



unl

Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de Energía, las Industrias y los Recursos

Naturales no Renovables

Carrera de Ingeniería en Sistemas

Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja.

Trabajo de Titulación previo
a la obtención del título de
Ingeniera en Sistemas.

AUTORA:

Dayanna Magdalla Alvarado Castillo

DIRECTOR:

Ing. Oscar Miguel Cumbicus Pineda, Mg.Sc.

LOJA - ECUADOR

2023

Certificación

Loja, 5 de mayo del 2023

Ing. Oscar Miguel Cumbicus Pineda Mg.Sc

DIRECTOR DE TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Titulación denominado: **Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja.**, previo a la obtención del título de **Ingeniera en Sistemas** de autoría del estudiante: **Dayanna Magdalla Alvarado Castillo**, con cédula de identidad Nro. **1150034401**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Oscar Miguel Cumbicus Pineda Mg.Sc

DIRECTOR DE TRABAJO DE TITULACIÓN

Autoría

Yo, **Dayanna Magdalla Alvarado Castillo**, declaro ser autora del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación del presente trabajo de titulación en el Repositorio Institucional-Biblioteca Virtual.

Firma:

Cédula de Identidad: 1150034401

Fecha: 10 de mayo de 2023

Correo electrónico: dayanna.alvarado@unl.edu.ec

Teléfono: 0995087934

Carta de autorización por parte de la autora, para la consulta, reproducción parcial y/o total, publicación electrónica de texto completo del Trabajo de Titulación

Yo **Dayanna Magdalla Alvarado Castillo**, declaro ser autora del Trabajo de Titulación denominado: **Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja.**, como requisito para optar el título de **Ingeniera en Sistemas**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los diez días del mes de mayo de los dos mil veintitrés.

Firma:

Autor: Dayanna Magdalla Alvarado Castillo

Cédula: 1150034401

Dirección: Argelia-Loja

Correo electrónico: dayanna.alvarado@unl.edu.ec

Teléfono: 0995087934

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Oscar Miguel Cumbicus Pineda Mg.Sc.

Dedicatoria

El presente Trabajo de Titulación lo dedico a Dios por darme la sabiduría, salud y fuerza ante los momentos más difíciles que se han presentado, él ha sabido encontrar la manera de seguir dándome razones por las que esforzarme día a día sin dejarme desmayar.

A mis padres, pilar fundamental para este logro.

Dayanna Magdalla Alvarado Castillo

Agradecimiento

Primeramente, a Dios, por la vida y por permitirme culminar esta meta académica con éxito.

Agradezco a mis padres quienes han sido mi apoyo incondicional guiándome con sus experiencias, consejos, valores y por motivarme a seguir en el camino del aprendizaje, infinitas gracias, por inculcarme la ideología y mentalidad de lograr este fin a pesar de las adversidades que la vida pone a prueba.

Finalmente, mi agradecimiento al Ingeniero Óscar Cumbicus, director del presente Trabajo de Titulación, por su paciencia, tiempo y orientación ideal para que este documento se finalice de manera exitosa.

Dayanna Magdalla Alvarado Castillo

Índice de Contenidos

Portada	i
Certificación.....	ii
Autoría	iii
Carta de autorización.....	iv
Dedicatoria	v
Agradecimiento	vi
Índice de Contenidos	vii
Índice de tablas.....	x
Índice de figuras	xii
Índice de anexos.....	xvii
1. Título.....	1
2. Resumen	2
2.1. Abstract	3
3. Introducción	4
4. Marco teórico.....	6
4.1. Conceptos Preliminares	6
4.1.1. Inteligencia Artificial.....	6
4.1.2. Visión por Computador.....	6
4.1.3. Reconocimiento Facial.....	7
4.1.4. Ataques de presentación.....	13
4.1.5. Detección de Vida	14
4.1.6. Red Neuronal convolucional	16
4.1.7. Odoo	20
4.1.8. OpenCV	22
4.1.9. Matriz de Confusión	22
4.2. Herramientas.....	23

4.2.1. PostgreSQL	23
4.2.2. Lucidchart	24
4.2.3. BPMN	24
4.3. Marcos de trabajo.....	26
4.3.1. Scrum	27
4.3.2. Design Thinking.....	27
4.4. Estudios Relacionados.....	29
5. Metodología.....	33
5.1. Área de estudio	33
5.2. Procedimiento	33
5.3. Recursos	35
5.3.1. Recursos Científicos	35
5.3.2. Recursos Técnicos.....	35
5.3.3. Recursos Tecnológicos.....	36
5.4. Participantes.....	36
6. Resultados.....	38
6.1. Objetivo 1:	38
6.1.1. Tarea 1: Analizar el proceso actual de registro de asistencia al laboratorio de Software mediante la aplicación de entrevistas y encuestas a docentes y estudiantes de CIS/C-UNL.....	39
6.1.2. Tarea 2: Diagramar el proceso de registro de asistencia en el laboratorio de Software haciendo uso de la notación BPMN con la herramienta LucidChart. ...	46
6.2. Objetivo 2:	50
6.2.1. Tarea 1: Analizar los requerimientos del software de reconocimiento facial con detección de vida para el registro de asistencias.....	54
6.2.2. Tarea 2: Modelar la arquitectura del software empleando la notación UML haciendo uso de la herramienta LucidChart.	56

6.2.3. Tarea 3: Desarrollar el prototipo de software para el registro de asistencias mediante reconocimiento facial con detección de vida usando el Framework Odoo ERP con el gestor de Base de Datos PostgreSQL.....	62
6.3. Objetivo 3:	89
6.3.1. Tarea 1: Evaluar mediante métricas de la Matriz de confusión la precisión del reconocimiento facial.....	89
6.3.2. Tarea 2: Evaluar mediante métricas de la Matriz de Confusión la precisión de la detección de vida.....	93
7. Discusión	99
7.1. Desarrollo de la propuesta.....	99
7.2. Valoración técnica económica ambiental	100
7.3. Limitaciones de la propuesta	102
8. Conclusiones.....	104
9. Recomendaciones	105
9.1. Trabajos Futuros	105
10. Referencias	106
11. Anexos.....	111

Índice de tablas:

Tabla 1. Trabajos relacionados	30
Tabla 2. Utilización de laboratorio de software.....	42
Tabla 3. Registro de asistencia en laboratorio.....	43
Tabla 4. Riesgo de contagio	44
Tabla 5. Método de registro de asistencia	45
Tabla 6. Descripción de las tareas del proceso actual de registro de asistencia.....	47
Tabla 7. Comparativa de las ideas para registro de asistencia.....	49
Tabla 8. Comparativa de las ideas para el reconocimiento facial	52
Tabla 9. Comparativa de las ideas para la detección de vida	54
Tabla 10. Requerimientos funcionales	54
Tabla 11. Requerimientos no funcionales	55
Tabla 12. Identificación de actores.....	56
Tabla 13. Comparación de la experimentación con modelos de reconocimiento facial	78
Tabla 14. Matriz de confusión de pruebas de reconocimiento facial.	92
Tabla 15. Matriz de confusión de pruebas de detección de vida.	95
Tabla 16. Cumplimiento de los requerimientos funcionales	97
Tabla 17. Cumplimiento de los requerimientos no funcionales	97
Tabla 18. Valoración económica - Talento humano	101
Tabla 19. Valoración económica – Servicios	101
Tabla 20. Valoración económica – Recursos hardware	102
Tabla 21. Valoración económica – Recursos Software	102
Tabla 22. Costo total del proyecto.....	102
Tabla 23. Manipulación de instrumentos de uso comunitario	148

Tabla 24. Tiempo estimado de registro de asistencia 149

Tabla 25. Satisfacción con el software propuesto 150

Índice de figuras:

Figura 1. Fases del Reconocimiento facial [8].....	7
Figura 2. Autenticación por reconocimiento facial en bienes inmuebles [1].	9
Figura 3. Reconocimiento facial con Eigenfaces [13].....	10
Figura 4. Reconocimiento Facial con Fisherfaces [13].....	11
Figura 5. Reconocimiento facial con LBP [9].....	12
Figura 6. Detección de rostros con Face-Recognition [15].	12
Figura 7. Ataques de presentación [18].....	14
Figura 8. Arquitectura de una Red Neuronal Convolutacional [22].....	17
Figura 9. Ejemplo de convolución en una Red Neuronal Convolutacional [22].....	18
Figura 10. Ejemplo de la capa de pooling de una Red Neuronal Convolutacional [23].	19
Figura 11. Red Neuronal Artificial [23].....	19
Figura 12. Kernel de una red Neuronal Convolutacional 3D [24].....	20
Figura 13. Convolución de una 3D CNN [24].	20
Figura 14. Organización de la matriz de confusión [27].....	22
Figura 15. Objetos de Flujo BPMN [31].....	25
Figura 16. Objetos Artefactos BPMN [31].....	25
Figura 17. Objetos de Conexión BPMN [31].....	26
Figura 18. Fases de Design Thinking [35].	28
Figura 19. Utilización de laboratorio de Software.....	42
Figura 20. Registro de Asistencia	43
Figura 21. Riesgo de Contagio	44
Figura 22. Método de registro de asistencia.....	45
Figura 23. Proceso actual de registro de asistencia.....	46

Figura 24. Lluvia de ideas desarrollo Reconocimiento Facial.....	51
Figura 25. Lluvia de ideas para el desarrollo de Detección de vida.	53
Figura 26. Estilo arquitectónico general de Odoo	56
Figura 27. Estilo arquitectónico de despliegue	57
Figura 28. Fases del reconocimiento facial	58
Figura 29. Vista de escenarios - Casos de uso.....	59
Figura 30. Modelo Relacional	60
Figura 31. Diagrama de flujo: Registro de asistencia.....	61
Figura 32. Arquitectura de 3D CNN para la detección de vida	62
Figura 33. Estructura del sistema de registro de asistencias.....	63
Figura 34. Modelo Estudiante.	65
Figura 35. Crear Estudiante	66
Figura 36. Editar Estudiante.....	66
Figura 37. Listar Estudiante	66
Figura 38. Eliminar Estudiante	67
Figura 39. Tecnologías aplicadas en las fases de reconocimiento facial con los modelos EigenFaces, FisherFaces y LBPHFaces.....	68
Figura 40. Recopilación de 300 fotogramas del rostro detectado.....	69
Figura 41. Captura de 300 fotogramas.....	70
Figura 42. Base de datos para el entrenamiento del modelo de Reconocimiento Facial.	71
Figura 43. Entrenamiento del modelo de reconocimiento LBPHFace.....	71
Figura 44. Reconocimiento Facial con el modelo EigenFaces	72
Figura 45 Reconocimiento Facial con FisherFaces	73
Figura 46. Reconocimiento Facial con LBPHFace.....	74

Figura 47. Tecnologías aplicadas en las fases de reconocimiento facial con el modelo Face-Recognition.....	75
Figura 48. Base de datos para Face-Recognition	75
Figura 49. Modelo Face-Recognition.....	76
Figura 50. Reconocimiento facial con Face-recognition	77
Figura 51. Puntos de referencia de los ojos [46].	80
Figura 52. Detección de vida mediante contador de parpadeos.....	81
Figura 53. Registro de asistencia con reconocedor facial y contador de parpadeos.	82
Figura 54. Predicción del modelo de detección de vida con CNN 3D.	83
Figura 55. Arquitectura de la red neuronal convolucional 3D para la detección de vida.....	84
Figura 56. Kernel de una Red neuronal convolucional de 3 dimensiones.....	84
Figura 57. Código de creación de Red Neuronal Convolutacional 3D	85
Figura 58. Detección de vida con CNN 3D (usuario en tiempo real)	87
Figura 59. detección de vida con CNN 3D (reproducción de video)	87
Figura 60. Listar registros de asistencia.	88
Figura 61. Fotografías de la BD de pruebas.....	90
Figura 62. Registro de los participantes en el sistema.....	90
Figura 63. Ejemplo de ejecución de la primera prueba Reconocimiento facial	91
Figura 64. Ejemplo de ejecución de pruebas de reconocimiento facial con rostros de personas no registradas en el sistema	91
Figura 65. Ejemplo de ejecución de la prueba de detección de vida con persona en tiempo real.....	94
Figura 66. Ejemplo de ejecución de la prueba de detección de vida con una fotografía.	94
Figura 67. Diagrama de flujo: Acceso al sistema	125
Figura 68. Diagrama de flujo: Crear Registro	126

Figura 69. Diagrama de flujo: Registro de Asistencia	127
Figura 70. Diagrama de flujo: Listar registros.....	128
Figura 71. Diagrama de flujo: Modificar registro.....	129
Figura 72. Diagrama de flujo: Eliminar registro	130
Figura 73. Diagrama de flujo: Generar reporte de asistencias	131
Figura 74. Crear Estudiante	132
Figura 75. Editar Estudiante.....	132
Figura 76. Listar Estudiante	132
Figura 77. Eliminar Estudiante	133
Figura 78. Crear Docente.....	134
Figura 79. Editar Docente	134
Figura 80. Eliminar Docente.....	134
Figura 81. Listar Docente	134
Figura 82. Crear Carrera	135
Figura 83. Editar Carrera.....	135
Figura 84. Eliminar Carrera	135
Figura 85. Listar Carrera	135
Figura 86. Crear Materia	136
Figura 87. Editar Materia.....	136
Figura 88. Eliminar Materia	136
Figura 89. Listar Materia	137
Figura 90. Crear Horario	137
Figura 91. Editar Horario	137
Figura 92. Eliminar Horario	138

Figura 93. Listar Horario.....	138
Figura 94. Listar registros.....	138
Figura 95. Manipulación de instrumentos de uso comunitario	148
Figura 96. Tiempo estimado de registro de asistencia.....	149
Figura 97. Satisfacción con el software propuesto.....	150

Índice de anexos:

Anexo 1. Entrevista a docente usuario del laboratorio de software	111
Anexo 2. Entrevista a técnico encargado del laboratorio de software	114
Anexo 3. Formato actual de registro de asistencia	118
Anexo 4. Encuesta a estudiantes	119
Anexo 5. Validación BPMN proceso actual	123
Anexo 6. Especificación de requerimientos de software	124
Anexo 7. Diagramas de flujo del prototipo	125
Anexo 8. Interfaces de módulo de reconocimiento facial con detección de vida para el registro de asistencia	132
Anexo 9. Encuesta de recopilación de datos de participantes para pruebas de los modelos	139
Anexo 10. Validación del cumplimiento de los requerimientos de software	143
Anexo 11. Encuesta de satisfacción a estudiantes del prototipo propuesto	144
Anexo 12. Código del prototipo	151
Anexo 13. Certificado de traducción de resumen al idioma inglés	152

1. Título

**Prototipo de reconocimiento facial con detección de vida para el registro de
asistencia al laboratorio de Software de la Carrera de Ingeniería en
Sistemas/Computación de la Universidad Nacional de Loja.**

2. Resumen

El presente trabajo de titulación tuvo como objetivo desarrollar un prototipo de reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja, como alternativa de registro de asistencia que no requiera la manipulación de instrumentos de uso comunitario, además acorta los tiempos de registro y facilita la creación de reportes del uso del laboratorio. Se guía según el marco de trabajo Design Thinking. La fase de empatía permitió conocer el proceso actual de registro de asistencia. En la segunda fase se definieron las revelaciones obtenidas en la etapa de empatía y se plantearon los retos para el desarrollo de la propuesta. En la tercera fase se recopilaron ideas como solución a los retos. En la cuarta fase se realizaron los prototipos de las ideas obtenidas. Finalmente, en la quinta fase se realizaron las pruebas del prototipo desarrollado. El sistema consta con un módulo web que permite la administración de los procesos de registro de asistencia y a través del cual se pone en funcionamiento el reconocimiento facial. El módulo de reconocimiento facial se desarrolló empleando el clasificador preentrenado Face-recognition el cual utiliza redes neuronales convolucionales, y esta sustenta con un modelo de detección de vida desarrollado con redes neuronales convolucionales de tres dimensiones. El prototipo desarrollado no requiere la manipulación física de instrumentos de uso comunitario y tiene una exactitud del 95% en las predicciones que realiza. Además, reduce el tiempo de registro de asistencia pasando de un mínimo de 2 minutos estimados que se empleaban de manera tradicional a 9,2 segundos promedio por persona con esta herramienta y facilita la creación de reportes del uso del laboratorio en un tiempo oportuno.

Palabras clave: *Reconocimiento facial, detección de vida, visión por computador, registro de asistencia.*

2.1. Abstract

The objective of this degree work was to develop a prototype of face recognition with liveness detection for attendance record at the Software laboratory of the Systems/Computer Engineering Career of the National University of Loja, as an alternative for the attendance record which does not require the manipulation of instruments of community use. In addition, it shortens registration times and facilitates the creation of reports on the use of the laboratory. It is guided according to the Design Thinking framework. The empathy phase allowed to know the current attendance record process. In the second phase, the revelations gathered in the empathy phase were defined and the challenges for the development of the proposal were raised. In the third phase, ideas were collected as a solution to the challenges. In the fourth phase, the prototypes of the ideas obtained were made. Finally, in the fifth phase, the developed prototype was tested. The system consists of a web module that allows the administration of attendance record processes and through which face recognition is put into operation. The face recognition module was developed using the Face-recognition pre-trained classifier, which uses convolutional neural networks, and is supported by a liveness detection model developed with three-dimensional convolutional neural networks. The prototype developed does not require the physical manipulation of instruments for community use and has an accuracy of 95% in the predictions it makes. Moreover, it reduces attendance record time from a minimum of 2 estimated minutes used in the traditional way to 9.2 seconds average per person with this tool, and facilitates the creation of reports on the use of the laboratory in a timely manner.

Keywords: *Face recognition, liveness detection, computer vision, attendance record.*

3. Introducción

La tecnología de reconocimiento facial actualmente se ha acoplado a múltiples procedimientos globales, tales como la autenticación en aplicaciones web/móvil, desbloqueo de dispositivos, el caso de registro de asistencia en instituciones públicas y privadas, entre otros [1], logrando reemplazar en muchos de los casos a los métodos tradicionales de autenticación con relleno de formularios, además, debido a la crisis sanitaria por la que atraviesa el mundo entero por la presencia del Covid-19, el reconocimiento facial al ser una tecnología de bajo costo, su alcance y adaptación a la vida cotidiana ha ido en constante crecimiento, sobre todo por caracterizarse en que su aplicación evita la manipulación de instrumentos de uso comunitario para realizar autenticaciones, lo que se traduce en que esta tecnología aporta positivamente en evitar el riesgo de contagio[2]. Sin embargo, en Ecuador son pocas las instituciones que han decidido aplicar la tecnología de reconocimiento facial, a pesar de que está en apogeo para el bien de la comunidad, esta tecnología sigue siendo un tema poco frecuente que no ha alcanzado un buen desarrollo, sin embargo, es importante, que los usuarios utilicen e implementen este tipo de tecnologías que permiten realizar tareas de forma ágil, rápida y sencilla de manera automatizada y que, a más de ser conveniente frente a la crisis sanitaria, brinda la oportunidad de ser competitivos con la constante evolución de las tecnologías.

Bajo este contexto, el presente trabajo de titulación tuvo como objetivo general desarrollar un prototipo de reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja, con el fin de brindar una propuesta de registro de asistencia que no requiera la manipulación física de instrumentos de uso comunitario, a más de reducir los tiempos de registro y facilitar al personal técnico la creación de reportes del uso del laboratorio del cual está encargado.

Este trabajo se desarrolló tomando como guía el marco de trabajo Design Thinking, la ejecución de cada una de sus fases permitió el cumplimiento de tres objetivos específicos planteados. A) Analizar el proceso actual de registro de asistencia en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja. B) Desarrollar el prototipo de reconocimiento facial con detección de vida aplicando clasificadores preentrenados con redes neuronales convolucionales. C) Evaluar mediante métricas de la Matriz de Confusión la precisión del prototipo de reconocimiento facial con detección de vida para el registro de asistencias en entornos simulados.

Cabe señalar que con el desarrollo de este prototipo se permite el registro de asistencias sin requerir la manipulación de instrumentos de uso comunitario, además se logra agilizar el proceso de registro de asistencia y la construcción de reportes.

Finaliza el trabajo con la discusión del análisis de los resultados obtenidos, conclusiones y recomendaciones.

4. Marco teórico

La información bibliográfica necesaria para el desarrollo del presente trabajo de titulación está organizada en los siguientes apartados.

4.1. Conceptos Preliminares

4.1.1. Inteligencia Artificial

La Inteligencia Artificial (IA) es una especialidad en el campo de las ciencias computacionales, la cual tiene la finalidad de estudiar modelos de cómputo que tienen la capacidad de hacer actividades innatas de los humanos, tomando como base principal a dos de sus propiedades primordiales: el razonamiento y la conducta.

Las primeras aplicaciones en esta área estuvieron enfocadas a desarrollar algoritmos para juegos, actualmente, la IA es una rama de la teoría de la computación que incluye áreas tales como el razonamiento automático, la demostración de teoremas, los sistemas expertos, el procesamiento de lenguaje natural, robótica, lenguajes y ambientes de IA, aprendizaje automático, redes neuronales, algoritmos genéticos y visión por computador [3].

4.1.2. Visión por Computador

La visión por computador, también conocida como visión artificial o interpretación de imágenes, es una disciplina que busca la deducción automática de la estructura y propiedad de un escenario o mundo en tercera dimensión, tiene como objetivo principal la interpretación de escenas a partir de las imágenes suministradas por cámaras, para ello se utiliza la potencia de procesamiento y las prestaciones de un computador digital [4].

Se considera como un conjunto de técnicas científicas que mediante el procesamiento computacional y aplicando algoritmos de identificación permite la adquisición, análisis y procesamiento de imágenes para obtener información específica, la cual es procesada por un ordenador, al igual que los seres humanos quienes adquieren información del entorno a través de los ojos, los sistemas de visión artificial intentan establecer un efecto similar obteniendo datos útiles de una o varias imágenes por medio de un dispositivo electrónico programable que realice un control automático [5]. Los sistemas de visión artificial completan tareas de inspección con un alto nivel de flexibilidad y repetitividad; nunca existe cansancio, aburrimiento ni distracción y pueden trabajar en ambientes donde las personas no podrían realizar una correcta inspección visual [6].

4.1.3. Reconocimiento Facial

El reconocimiento facial es una técnica en el campo de la biometría por características, la misma que se desarrolla a través de la utilización de algoritmos computacionales, haciendo uso de un procesamiento mediante ordenador, son capaces de identificar automáticamente rostros de personas a partir de imágenes digitales empleando un profundo análisis de las características faciales de la persona y comparándolas con un conjunto de imágenes de la misma almacenados en el sistema. La mayor parte de los sistemas de reconocimiento facial actualmente clasifican la apariencia, por medio de la identificación de ciertos puntos nodales en el rostro como lo son: la distancia entre ojos, nariz o entre los ojos y la boca [5].

En otras palabras, se puede definir el sistema de reconocimiento facial como una aplicación dirigida por ordenador que identifica de forma automática a un individuo en una imagen digital, esto es posible mediante un análisis de las características faciales del sujeto extraídas de la imagen o de un fotograma clave de una fuente de video [7].

4.1.3.1. Fases del reconocimiento facial.

Los sistemas de reconocimiento facial cumplen varias etapas, mismas que se presentan en la Figura 1 a continuación:

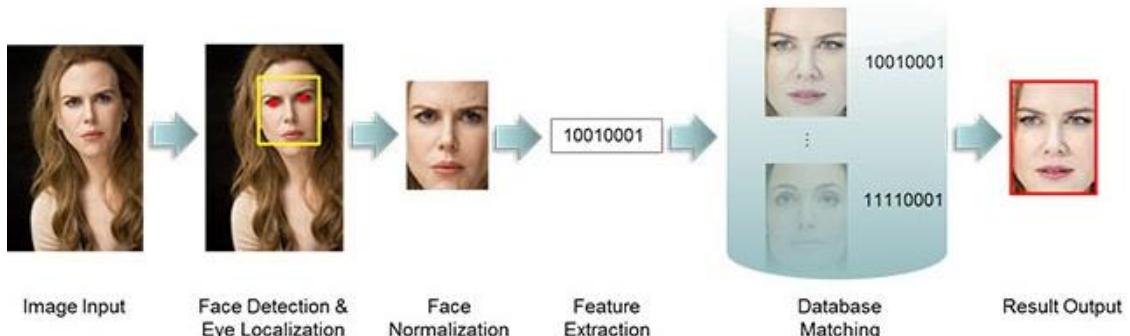


Figura 1. Fases del Reconocimiento facial [8].

- **Image input / Imagen de entrada:** No se considera como una fase del reconocimiento facial, sin embargo, corresponde al punto de partida para que el proceso se lleve a cabo.
- **Face detection & Eye Localization / Detección del rostro y localización de los ojos:** Fase inicial del reconocimiento facial, tiene como finalidad detectar rostros humanos en un segmento de video o imagen recibidos como entrada, esto se realiza

independientemente de la ubicación, orientación y escala, para lo cual se requiere una cámara y correcta iluminación [5].

- **Face Normalization / Normalización del Rostro:** La normalización del rostro prepara la imagen para su fácil manipulación, depende de un análisis previo haciendo uso de matemáticas para establecer una adecuada posición, iluminación requerida y el tamaño adecuado de imagen [5]. Dentro de esta fase se encuentra:
 - a) **Transformación a escala de grises:** realiza la tarea de transformar la imagen a escala de grises discriminando el color al ser poco relevante en la tarea de reconocimiento.
 - b) **Recorte:** secciona el rostro para eliminar las áreas que no aportan información.
 - c) **Escalado:** redimensiona la imagen del rostro a un tamaño estandarizado.
 - d) **Caracterización:** representa la imagen del rostro en un formato simplificado que permita facilitar la diferenciación entre expresiones [9].
- **Extracción de características:** La extracción de las características se da al aplicar el análisis de la información de los rostros los mismos que varían en características mediante cambios geométricos.
- **Reconocimiento:** El reconocimiento se da como objetivo final mediante la comparación de los patrones en medida de toda la información obtenida en la base de datos para establecer el reconocimiento. Da como resultado la clasificación dentro de un conjunto de posibles clases [5].

4.1.3.2. Autenticación por reconocimiento facial.

Se corresponde como un proceso cuya finalidad es verificar la identidad de una persona, para ello hace uso de las características innatas de cada individuo, se lo emplea para acceder a un servicio web, aplicativos, dispositivos electrónicos así como también actualmente está siendo utilizado para controlar el acceso a bienes inmuebles, tal como se ilustra en la Figura 2, debido a la medida de seguridad que otorga a los usuarios, además de proporcionar una experiencia novedosa e intuitiva para los mismos, con un índice de confiabilidad mucho mayor en comparación a las contraseñas convencionales [10].



Figura 2. Autenticación por reconocimiento facial en bienes inmuebles [1].

4.1.3.3. Métodos de Reconocimiento Facial.

Existen dos alternativas a la hora de realizar las comparaciones entre la imagen de entrada con una almacenada previamente en la base de datos, se puede utilizar el conjunto de todas las características del rostro, enfoque holístico, o utilizar diferentes partes del rostro de manera individual. En función del enfoque y de los algoritmos utilizados para reducir la dimensionalidad y simplificar los cálculos, en la web existen diferentes modelos preentrenados para el reconocimiento facial. En cualquier caso, en la mayoría de ellos es necesaria una fase de reentrenamiento previo con las imágenes de los usuarios ya registrados [11].

En los siguientes apartados se presentan algunos de los modelos más utilizados actualmente en el reconocimiento facial.

- **Eigenfaces.**

Este método se fundamenta en el análisis de componentes principales o PCA (Principal Component Analysis), su principal objetivo es reducir el número de dimensiones con pérdida no considerable de información tal y como se utiliza en la estadística para el análisis de datos multivariantes debido a que permite reducir el número de variables simplificando el sistema [12].

Este método fue empleado con éxito en el análisis de componentes principales para el reconocimiento facial, posterior a este suceso se hizo uso de una nueva técnica basado en PCA, denominada Eigenfaces (caras propias), para el reconocimiento facial.

Eigenfaces toma todas las caras de entrenamiento de todas las personas a la vez y las mira como un todo y luego guarda los componentes más importantes y descarta el resto. Estos importantes componentes se conocen como características principales.

Por desventaja, Eigenfaces considera la iluminación como un factor importante, por lo tanto, a medida que capta iluminaciones ligeras, lo considera como una característica que representa un rostro. Este algoritmo no presta atención a las características que diferencian a un individuo de otro, simplemente se concentra en las características que representan todos los rostros de todas las personas.

En la Figura 3 se ilustra un ejemplo del procesamiento de imagen con eigenfaces, se observa que con este método se recoge a la luminosidad como una característica del rostro.



Figura 3. Reconocimiento facial con Eigenfaces [13].

- **Fisherfaces.**

FisherFaces es una mejora sobre EigenFaces y utiliza el análisis de componentes principales (PCA) y el análisis discriminante lineal (LDA).

Eigenfaces, al considerar la iluminación como una característica importante, puede descartar las características de otras personas considerándolas menos útiles. Fisherfaces soluciona esto ajustando las caras propias de modo que extraiga características de todos los individuos por separado en lugar de mirarlos como un todo. Por lo tanto, ahora, incluso si los datos de la cara de una persona tienen grandes cambios de iluminación, no afectarán las características de otras personas.

El algoritmo de Fisherfaces extrae los componentes principales que separan a un individuo de otro. Entonces, ahora las características de un individuo no pueden dominar las características de otra persona. El reconocimiento de imágenes que utiliza este algoritmo se basa en la reducción de las cúpulas del espacio facial utilizando el método PCA y luego

aplicando el método LDA también conocido como método Fisher Linear Discriminant (FDL) para obtener los rasgos característicos de la imagen.[12]

Fisherfaces utiliza una matriz específica de transformación y no captura tan bien la iluminación como Eigenfaces. Por lo que la iluminación le afecta en menor medida.

La reconstrucción en Fisherfaces difiere mucho de la de Eigenfaces, como se puede apreciar en la Figura 4.



Figura 4. Reconocimiento Facial con Fisherfaces [13].

El proceso de reducción de dimensión de PCA puede causar cierta pérdida de información discriminante útil en el proceso LDA, lo cual es una desventaja de este proceso.

- **Local Binary Pattern (LBP).**

El patrón binario Local LBP es un método de reconocimiento facial que soluciona la desventaja de fisherfaces, ya que no reduce la dimensión y por tanto no pierde información útil para discriminar características propias de cada rostro.

Básicamente LBP tiene como propósito recorrer toda la imagen y etiquetar los píxeles de la misma usando un umbral, este umbral utilizado es la diferencia entre el píxel central y sus vecinos, el resultado de esta comparación con sus vecinos es un número binario. El operador básico del LBP utiliza una matriz de 3x3, es decir ocho píxeles de vecindad, en la cual se considera al pixel central como un umbral (ver Figura 5). Los píxeles vecinos con valores superiores o iguales al píxel central se les otorga el valor 1 caso contrario reciben el valor de 0. De esta manera se puede obtener el código binario de ocho bits, que describe a la región o vecindad a la que se encuentra atribuido el píxel central dado [9].

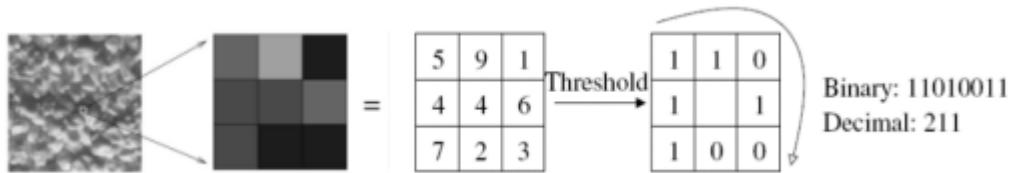


Figura 5. Reconocimiento facial con LBP [9].

- **Face-Recognition.**

Se trata de una librería de código abierto desarrollada en lenguaje Python que a su vez implementa dlib, la librería de reconocimiento facial más simple del mundo que incluye una red neuronal de tipo convolucional pre-entrenada que permite detectar caras y rasgos faciales con una efectividad del 99.38% [14].

El modelo Face-recognition es el resultado de la unión del reconocimiento facial con el aprendizaje métrico profundo, este último, genera un vector de características de valor real. En la red neuronal de reconocimiento facial dlib, el vector de características de salida es 128-d (es decir, una lista de 128 números con valores reales) que se utiliza para cuantificar el rostro.

Para llevar a cabo la fase de detección de rostro Face-Recognition utiliza el método llamado Histograma de gradientes orientados (HOG), luego se visualiza a cada píxel de la imagen uno a la vez. Para cada píxel, se observan los píxeles que lo rodean directamente, con el objetivo de averiguar qué tan oscuro es el píxel actual en comparación con los píxeles que lo rodean directamente. Luego se dibuja una flecha que muestre en qué dirección la imagen se oscurece, tal como se visualiza en la figura 6:

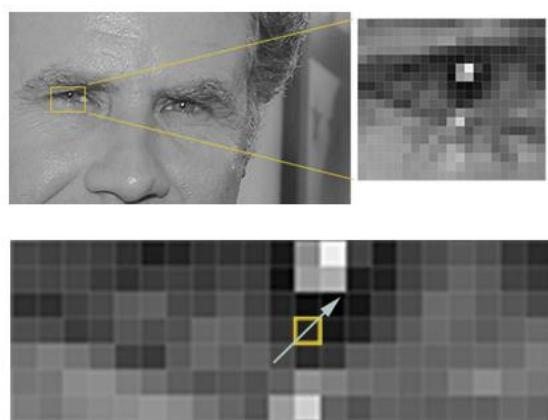


Figura 6. Detección de rostros con Face-Recognition [15].

Si repite ese proceso para cada píxel de la imagen, terminará reemplazando cada píxel por una flecha. Estas flechas se denominan gradientes y muestran el flujo de claro a oscuro en toda la imagen. Para hacer esto, se divide la imagen en pequeños cuadrados de 16x16 píxeles cada uno. En cada cuadrado, se cuentan cuántos gradientes apuntan en cada dirección principal (cuántos apuntan hacia arriba, hacia arriba a la derecha, hacia la derecha, etc.). Luego, se reemplaza ese cuadrado en la imagen con las direcciones de flecha que eran más fuertes.

El resultado final es que se convierte la imagen original en una representación muy simple que captura la estructura básica de una cara de una manera sencilla.

Para la medición del rostro, se ejecuta una red neuronal convolucional profunda para que genere 128 medidas para cada rostro y calcular un promedio de entre esas medidas que será único para cada rostro presentado en el entrenamiento. Para el proceso de predicción realiza la misma acción, ejecutando la red neuronal para encontrar las medidas del rostro de la imagen de entrada, calcular el promedio y comparar con los promedios antes recopilados, se predice como el mismo rostro a los promedios más cercanos [15].

4.1.4. Ataques de presentación.

Los avances del reconocimiento facial están haciendo que muchas áreas de la vida sean más óptimas. Las personas ahora pueden verificar su identidad sin siquiera una contraseña o huella digital. Pero como cualquier otra tecnología de privacidad, los estafadores están apuntando al reconocimiento facial a través de la suplantación de identidad.

Cuando se produce una suplantación facial, generalmente se hace bajo el disfraz de lo que se llama un ataque de presentación. El Instituto de Biometría en [16] lo describe como una suplantación de reconocimiento facial que se produce a través de datos biométricos obtenidos ilegalmente, ya sea de forma directa o encubierta de una persona en línea o mediante sistemas pirateados [17].

Generalmente, los sistemas de reconocimiento facial pueden falsificarse presentando a la cámara una fotografía, un video o una máscara 3D de una persona objetivo, usar maquillaje o cirugía plástica. No obstante, el uso de fotografías y videos es el tipo de ataques más común debido a la alta exposición del rostro y al bajo costo de las cámaras digitales de alta resolución. Seguidamente se presentan los ataques de presentación comúnmente utilizados.

- **Ataques fotográficos:** un ataque fotográfico no es más que presentar una fotografía de la identidad atacada al sensor del sistema de reconocimiento facial.
- **Ataques de video:** un atacante podría reproducir un video del usuario legítimo en cualquier dispositivo que reproduzca video y luego lo presente al sensor / cámara.
- **Ataques con máscara 3D:** en este tipo de ataque, el atacante realiza una reconstrucción 3D del rostro y la presenta al sensor / cámara.

Otros ataques: maquillaje, cirugía [18].

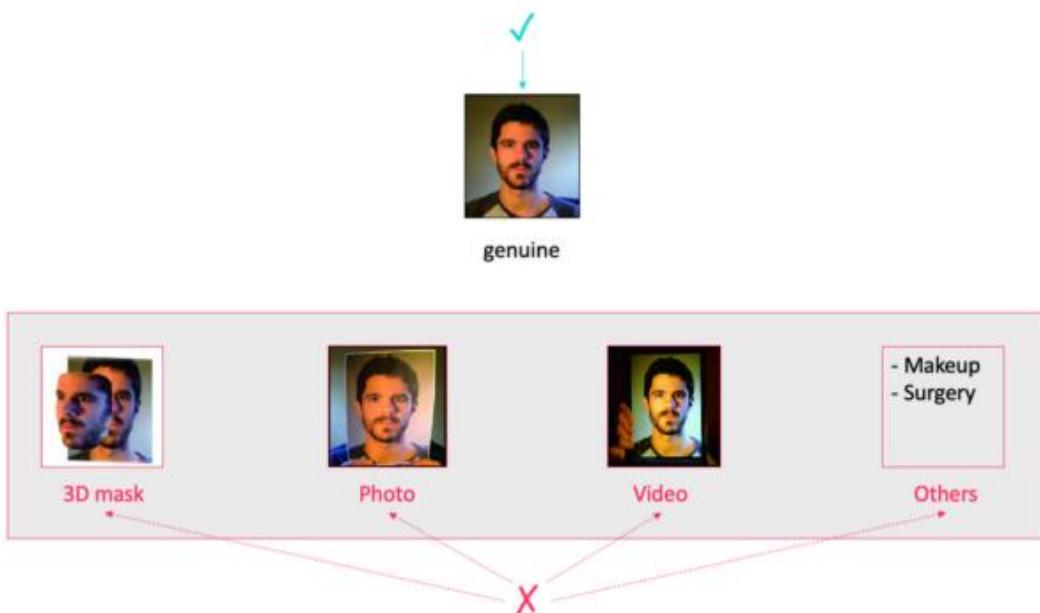


Figura 7. Ataques de presentación [18].

Como se describió en líneas anteriores y como se observa en la figura 7, son varios los ataques de presentación, es por ello que, para diseñar un sistema de reconocimiento facial seguro en un escenario real, las técnicas de detección de vida deben ser una prioridad desde la planificación inicial del sistema.

4.1.5. Detección de Vida

Para que la autenticación por reconocimiento facial pueda ser catalogada como un método de autenticación robusto, es indispensable distinguir entre un rostro vivo genuino y un intento de falsificar el sistema con un ataque de presentación. La detección automática de ataques de presentación, y en particular la detección de vida, se han vuelto un componente necesario de cualquier sistema de autenticación que se base en tecnología biométrica facial en la que el ser humano no supervise la autenticación [19].

La detección de vida en biometría es la capacidad de un sistema para detectar si una huella dactilar o un rostro (u otros datos biométricos) es real (de una persona viva presente en el punto de captura) o falsa (de un artefacto falso o una parte del cuerpo sin vida como fotografías o videos) [20].

4.1.5.1. Importancia de la detección de vida en la autenticación por reconocimiento facial.

El reconocimiento facial realiza una comparación entre las características mapeadas de un usuario inscrito previamente en la base de datos con una nueva entrada presentada ante el sensor de reconocimiento, tales características pueden ser la longitud de la línea de la mandíbula, la distancia entre sus ojos o con una plantilla biométrica para verificar su identidad, examina la nueva imagen que ve y realiza mediciones. Sin embargo, no verifica la presencia física de un individuo frente a una impresión de calidad o a una representación digital. Una fotografía o imagen en una pantalla funciona tan bien como la persona real [19].

Cuando se utiliza el reconocimiento facial para la autenticación, un atacante tiene la posibilidad de explorar la limitación del reconocimiento para engañar al sistema y hacerle creer que ve al usuario legítimo.

La detección de vida trabaja con un sistema biométrico para medir y analizar las características físicas con el propósito de verificar si se está capturando la muestra biométrica de un personaje vivo que está presente en el punto de captura, es decir que frente al sensor no se presente una imagen impresa, una reproducción de video o una máscara en 3D. Se debe tomar en cuenta que esta tecnología no realiza ninguna función de comparación, sino que detecta ataques de presentación.

4.1.5.2. Enfoques para la detección de vida.

Existen algunos enfoques para detectar la vida en un sistema biométrico facial. A un alto nivel, estas pueden clasificarse como activas o pasivas: La detección de vida activa demanda que los usuarios participen en una verificación respondiendo a un desafío que propone el sistema. Entre los ejemplos de sistemas implementados hoy en día se encuentran los siguientes [19]:

- Asentir o girar la cabeza de lado a lado
- Parpadear
- Seguir un punto en una pantalla

- Sonreír
- Hablar una serie de palabras o números
- Mover la cámara hacia la cara o inclinarse hacia la cámara
- Grabar un breve video

En el caso de la detección de vida pasiva, esta no requiere que el usuario genere ninguna acción. La detección de vida se produce cuando el usuario se toma una selfie. Existen algunas posibilidades en cuanto a técnicas para la vida pasiva, que van desde el análisis de una imagen de selfie hasta la captura de un video y la proyección de diferentes luces sobre el sujeto. Cada enfoque de la vida pasiva tiene un impacto diferente en la experiencia del usuario y en los requisitos de procesamiento [19].

4.1.6. Red Neuronal convolucional

Las redes neuronales convolucionales (CNN) son un tipo particular de red neuronal inspirada en el funcionamiento de la corteza visual del cerebro. Estas redes están diseñadas para resolver problemas de visión artificial tales como el reconocimiento de patrones, la clasificación de textos o el procesamiento de lenguaje natural. [21].

Al igual que las redes neuronales “convencionales”, reciben una entrada que transforman a través de una serie de capas de neuronas, pero en este caso la entrada es una imagen representada en forma de matriz tridimensional ancho, alto y profundidad de color que contiene los valores numéricos de los píxeles. Las capas de neuronas también se organizan de manera tridimensional, y el resultado de las distintas transformaciones llevadas a cabo en la red es la clase o categoría a la que pertenece dicha imagen. [21].

Las redes convolucionales se construyen utilizando cuatro tipos de capas principales: capa de entrada, capa convolucional, pooling y capa totalmente conectada.

Las CNN son en la actualidad el “estado del arte” en el campo de la visión artificial, donde demostraron una efectividad superior a otras técnicas, y esto es debido a tres ventajas clave de su arquitectura:

- Facilitan la detección de un atributo en cualquier posición de la imagen gracias al deslizamiento de la ventana.
- Tolerancia a leves variaciones de rotación, traslación o escala, gracias a las capas de pooling.

- Tiene la capacidad para detectar atributos o características complejas de alto nivel mediante la combinación de los filtros de las capas convolucionales [21].

4.1.6.1. Arquitectura de una Red Neuronal Convolutacional.

Cuando se habla de redes neuronales convolucionales se refiere, como su nombre lo indica, a un tipo de red neuronal que incluyen capas de convolución, esta convolución hace referencia al tratamiento de una imagen como una matriz que almacena los pixeles que conforman dicha imagen, a esta matriz se le aplican filtros (matrices pequeñas) que mediante operaciones matemáticas permiten la extracción de características de la imagen de entrada, además, este tipo de redes neuronales contienen capas de pooling necesarias para pasar, sin ninguna pérdida, la información que contiene la imagen a una matriz más pequeña que la inicial, estas redes contienen varias capas de convolución con la finalidad de extraer las características más complejas de una imagen, resultando una matriz cada vez más pequeña, adicional a ello, los valores finales de la matriz se traspasan a un vector, este último se conectará a la capa neuronal clasificadora que dará el resultado de la categoría a la que pertenece la imagen que se está examinando [22]. Para mejor entendimiento de la arquitectura de una red neuronal convolucional, a continuación, se ilustra en la figura 8.

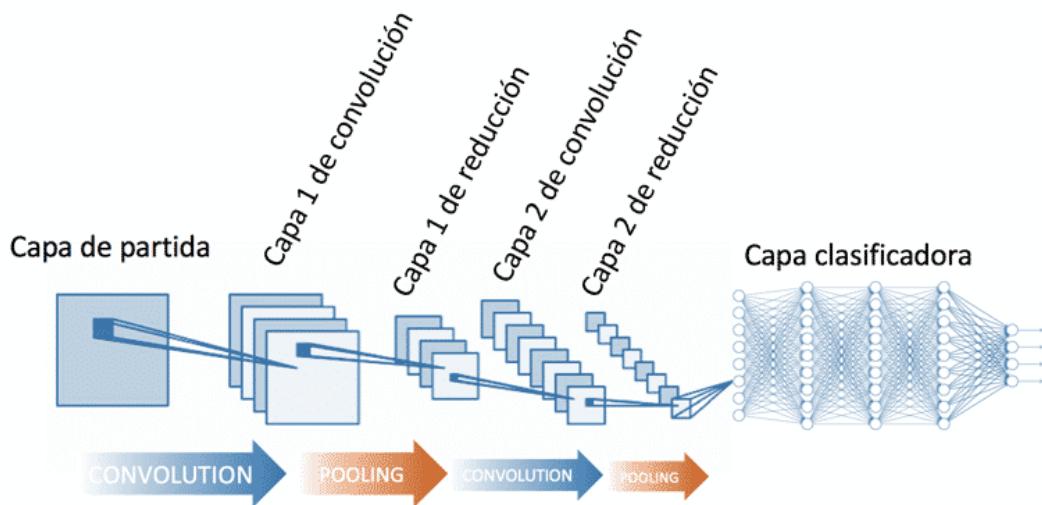


Figura 8. Arquitectura de una Red Neuronal Convolutacional [22].

Los pequeños cuadros presentes en las capas de convolución describen los filtros que se aplican a la imagen de entrada, para ejemplificar el proceso de la convolución, a continuación, la figura 9.

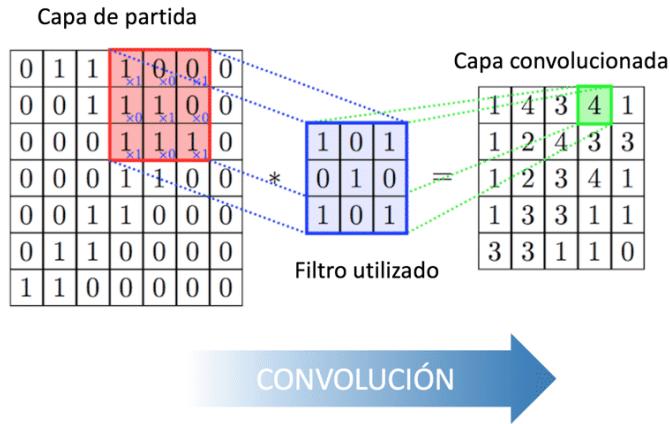


Figura 9. Ejemplo de convolución en una Red Neuronal Convolucional [22].

La figura 9 ejemplifica el proceso de convolución, se tiene la imagen de entrada representada en una matriz, a la cual se le aplican filtros, estos filtros son matrices más pequeñas que las de la capa de entrada, en este caso de dimensión 3x3 (largo y ancho), entre ambas se realizan operaciones matemáticas y se obtiene como resultado una capa convolucionada, siendo esta una nueva matriz con el resultado de la aplicación de los filtros [22].

Es muy común que en este tipo de neuronas se utilice una función de activación llamada ReLU [23], por las siglas en inglés de unidad lineal rectificada. Y su cometido es el de evitar números negativos en la salida de la neurona tomando en cuenta:

$$\begin{cases} 0 \text{ si } x < 0 \\ x \text{ si } x \geq 0 \end{cases} \quad (1)$$

En el caso de las capas de pooling, la razón de su existencia es la creencia de que la localización exacta de una característica es menos importante que su localización aproximada con respecto a otras características. De esta forma, la capa de reducción de muestreo (pooling) permite reducir paulatinamente el consumo espacial de la representación de la imagen, el número de parámetros y el costo computacional de la red. Además, permite reducir el sobreajuste y aumentar la invariancia frente a la traslación. Hay diversas funciones de reducción de muestreo no lineales entre las que elegir a la hora de implementar este tipo de capas en redes convolucionales, destacando por ser la más utilizada la técnica de max-pooling, la cual se basa en elegir, de entre la región seleccionada, la activación máxima. Otras formas, también usadas son average-pooling y L2-norm pooling.

Un ejemplo de la utilización de max-pooling se presenta a continuación.

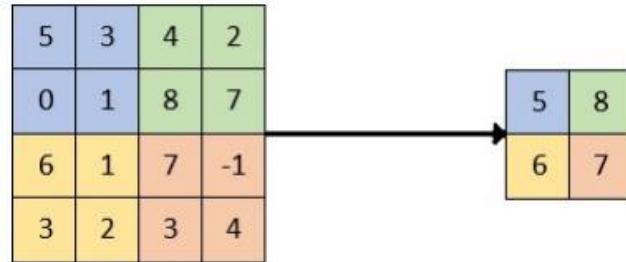


Figura 10. Ejemplo de la capa de pooling de una Red Neuronal Convolucional [23].

Como se observa en la figura 10, se realiza el proceso de reducción empleando la técnica de max-pooling, se aplica un filtro de 2×2 del cual se obtiene la mayor de las activaciones, el resultado es una matriz más pequeña que la capa convolucionada y servirá como entrada para la próxima convolución.

Finalmente, en las redes neuronales convoluciones están presentes las capas de clasificación, la cuales cuentan con neuronas conectadas como las de la figura 11, estas suelen recibir las características de las capas convolucionales y de pooling para decidir, en el caso de la clasificación de imágenes, qué objetos hay en una imagen [23].

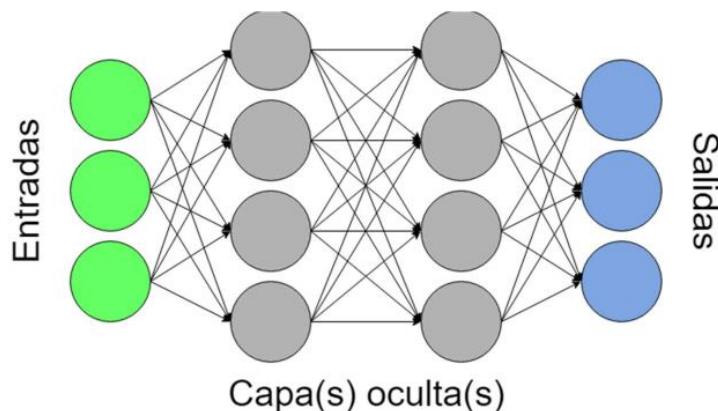


Figura 11. Red Neuronal Artificial [23].

Es importante mencionar la tendencia de “apagar” algunas de las neuronas de la red en el proceso de aprendizaje, esto con el fin de evitar el sobreajuste y lograr que no exista memorización durante el entrenamiento.

4.1.2.1. Red Neuronal Convolucional 3D.

Las redes neuronales convolucionales 3D, consideran, a diferencia de las bidimensionales, la profundidad, es decir la imagen de entrada es tratada como una matriz con dimensiones de alto, ancho y profundo, lo mismo ocurre con los filtros que se aplican, estos tendrán que tomar

en cuenta la profundidad, la capa convolucionada, así como las de pooling que también serán matrices de 3 dimensiones [39].

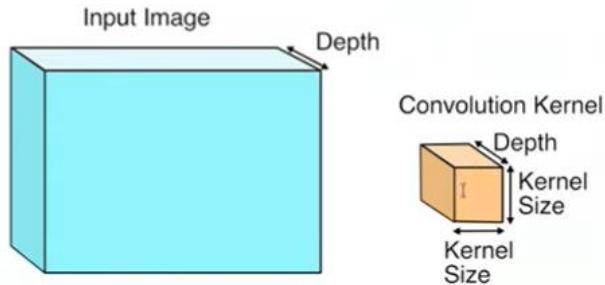


Figura 12. Kernel de una red Neuronal Convolutacional 3D [24].

En la figura 12 se puede apreciar, el tratamiento de la imagen de entrada tomando en cuenta la profundidad, además se presenta un filtro o kernel con 3 dimensiones.

El proceso de convolución se realiza sobreponiendo el filtro sobre la matriz de entrada y se realizan cálculos matemáticos entre esta y los valores del filtro correspondiente [39], el resultado será una nueva matriz, tal y como se observa en la figura 13.

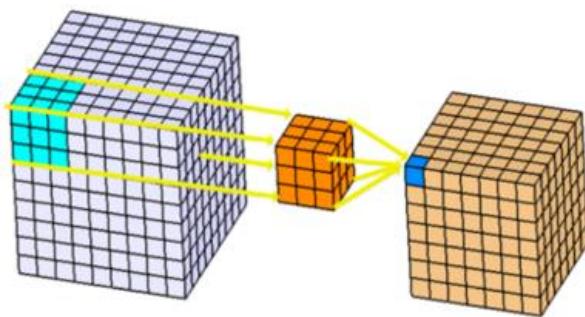


Figura 13. Convolución de una 3D CNN [24].

Por tanto, la arquitectura de una red neuronal convolucional 3D, contiene las mismas capas que una red neuronal convolucional tradicional, su diferenciación radica en la consideración de la profundidad de las matrices. Las matrices obtenidas por convolución y por pooling, también son matrices que consideran las dimensiones de alto, largo y ancho.

4.1.3. Odoo

Odoo es una suite de aplicaciones empresariales de código abierto que cubre todas las necesidades para la gestión empresarial, contiene un conjunto de aplicaciones como: Sitios Web, Ventas, Financieras, Operaciones, Fabricación, Contabilidad, RRHH, Proyectos, eCommerce, etc. Todas estas aplicaciones están integradas entre sí, lo que significa que

cualquier dato añadido en una aplicación, se sincroniza con las otras aplicaciones que también podrían necesitar esos datos [25].

Lo utilizan desde startups (1 usuario) hasta grandes empresas. Odoo se encuentra conformado por tres componentes principales, los cuales son:

- El servidor de base de datos PostgreSQL
- La aplicación del servidor de Odoo
- El servidor web

4.1.3.1. Composición de un módulo.

Un módulo de Odoo puede contener varios elementos.

- **Objetos de negocio:** Declaradas como clases de Python, Odoo conserva automáticamente estos recursos según su configuración.
- **Vistas de objetos:** Definición de visualización de la interfaz de usuario de los objetos comerciales
- **Archivos de información:** Archivos XML o CSV que declaran los metadatos del modelo:
 - Vistas o informes,
 - Datos de configuración (parametrización de módulos, reglas de seguridad),
 - Datos de demostración.
- **Controladores web:** Manejan solicitudes de navegadores web.
- **Datos web estáticos:** Imágenes, archivos CSS o javascript utilizados por la interfaz web o el sitio web [25].

4.1.3.2. Arquitectura de Odoo.

Un sistema de OdooERP está compuesto por tres elementos primordiales:

- El servidor de base de datos PostgreSQL, el cual almacena todas las bases de datos, cada una contiene todos los datos y la mayor parte de los elementos de configuración relacionados con el sistema OpenERP
- La aplicación del servidor de OpenERP el cual contiene toda la lógica empresarial y asegura que OpenERP corra eficazmente.

- Servidor web el cual es una aplicación separada llamada Open Object client-web, que le permite conectarse a OpenERP desde navegadores web estándar [26].

4.1.4. OpenCV

OpenCV es una librería de visión artificial creada por Intel y en la actualidad publicada bajo licencia BSD (*Berkeley Software Distribution*), contiene una variedad de algoritmos optimizados que permiten realizar las principales tareas de visión artificial, tales como la detección de características, reconocimiento de objetos o el procesamiento digital de imágenes.

La librería está escrita en C++, es multiplataforma y cuenta con interfaces para trabajar con lenguajes como Java o Python. OpenCV es una herramienta indispensable para desarrollar sistemas de visión artificial basados en software de código abierto debido a la gran cantidad de algoritmos disponibles, su velocidad de ejecución y la extensa comunidad de usuarios de que dispone [21].

4.1.5. Matriz de Confusión.

Una matriz de confusión se emplea en el campo de la inteligencia artificial y aprendizaje automático que en términos generales permite identificar cuántos aciertos y errores presenta un modelo realizado [27].

La matriz se organiza como se presenta en la figura 14.

		Valores de predicción	
		Verdaderos	Falsos
Valores reales	Positivos	Positivos	Positivos
	Falsos	Negativos	Verdaderos
		Negativos	Negativos

Figura 14. Organización de la matriz de confusión [27].

En Donde:

Verdadero positivo: El valor real es positivo y la predicción es positiva.

Verdadero negativo: El valor real es negativo y la predicción es negativa.

Falso negativo: El valor real es positivo y la predicción es negativa.

Falso positivo: El valor real es negativo y la predicción es positiva.

Ahora bien, para la métrica se utiliza una serie de valores como lo son precisión, recall, F1-score y accuracy definidos de la siguiente manera [27]:

Precisión: porcentaje de casos positivos detectados. Se calcula como:

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos positivos}} \quad (2)$$

Recall: valor que indica la capacidad del modelo para distinguir los casos positivos de los negativos. Se calcula como:

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos Negativos}} \quad (3)$$

F1-score: Este es un valor que resume la precisión y recall en una sola métrica. Se calcula como:

$$2 \frac{(\text{recall} * \text{precisión})}{\text{recall} + \text{precisión}} \quad (4)$$

Accuracy: Es un valor que mide el porcentaje de casos que el modelo de clasificación ha acertado. Se calcula como:

$$\frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Verdadero Positivos} + \text{Verdaderos Negativos} + \text{Falsos Positivos} + \text{Falsos Negativos}} \quad (5)$$

4.2. Herramientas

4.2.1. PostgreSQL

PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto que usa y extiende el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas.

PostgreSQL se ha ganado una sólida reputación por su arquitectura probada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para brindar soluciones innovadoras y de alto rendimiento de manera consistente.

PostgreSQL viene con muchas características destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y a crear entornos tolerantes a fallas, y a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos [28]. Algunas de sus características se citan a continuación.

- Diversos tipos de datos.
- Integridad de los datos.
- Simultaneidad, rendimiento.
- Fiabilidad, recuperación ante desastres.
- Seguridad.
- Extensibilidad.
- Internacionalización, búsqueda de texto.
- Código abierto.

4.2.2. Lucidchart

Lucidchart es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML, mapas mentales, prototipos de software y muchos otros tipos de diagrama. Construida con estándares web, como HTML5 y JavaScript, Lucidchart funciona en todos los navegadores web modernos, como Google Chrome, Firefox, Safari e Internet Explorer 8+, esta herramienta combina diagramas, visualización de datos y colaboración para acelerar la comprensión e impulsar la innovación [29].

4.2.3. BPMN

El estándar BPMN (Business Process Model and Notation) permite a las empresas visualizar sus procedimientos internos de negocio de forma gráfica y proporciona la notación estándar para la comunicación de procesos. Además, la notación gráfica facilitará la comprensión de las colaboraciones de desempeño y las transacciones comerciales entre las organizaciones [30]. La notación BPMN comprende los objetos que se detallan en los apartados a continuación.

4.2.3.1. Objetos de Flujo.

Los objetos de flujo que establece BPMN, son: rectángulos con las puntas redondeadas, dichos cuadrangulares significan actividades; círculos (más pequeños que las tareas) que

representan eventos y, rombos o diamantes (igual tamaño que los eventos) que representan una compuerta [31].

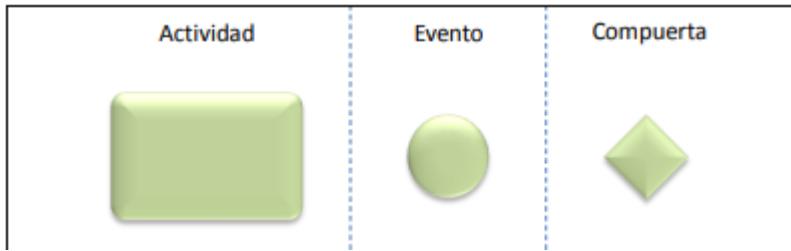


Figura 15. Objetos de Flujo BPMN [31].

4.2.3.2. Objetos Artefactos.

Los artefactos son elementos gráficos que se utilizan para proporcionar más información sobre el proceso de negocio, los artefactos no afectan los flujos del procedimiento, simplemente contribuyen a un mejor entendimiento del mismo; los objetos gráficos que integran esta categoría son: anotaciones, objetos de datos, grupos y por último pool, canales y fases [31].

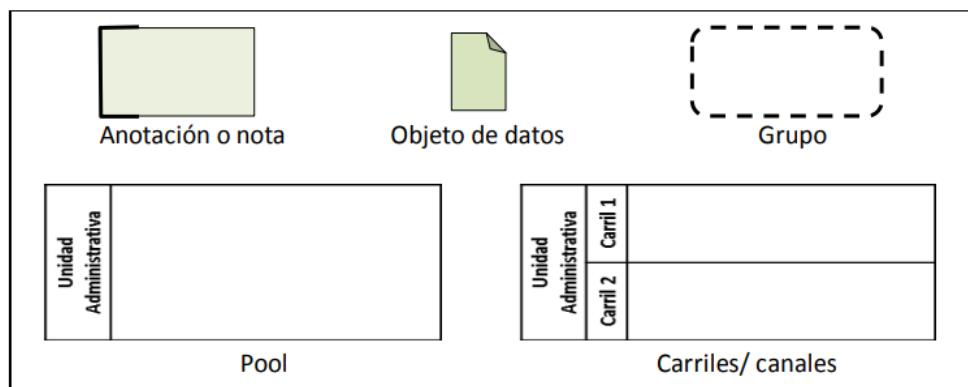


Figura 16. Objetos Artefactos BPMN [31].

- **Las anotaciones o notas:** son utilizadas para proporcionar información adicional y de utilidad al procedimiento, o, para hacer algún señalamiento importante sobre alguna actividad, subprocesso o proceso. Se representan mediante cuadros que se enlazan a una actividad.
- **El objeto de datos:** representa los insumos de documentos e información que es utilizada o modificada mediante la aplicación de un procedimiento, se representan mediante un recuadro con una esquina doblado, simulando una hoja de papel.

- **El grupo:** figura utilizada para agrupar un conjunto de actividades para efectos de análisis y comprensión. Se representa mediante rectángulo con puntas redondeadas y con línea intermitente o discontinua.
- **Los pool y canales/carriles:** Son contenedores gráficos que se utilizan para organizar las tareas, procesos y subprocesos; al contenedor mayor se le denomina pool y a las divisiones dentro de ellos, se nombran canales o carriles [31].

4.2.3.3. Objetos de Conexión.

Los conectores son elementos gráficos que se utilizan para enlazar los objetos del flujo; dicho enlace debe hacerse a través del enlace de los conectores con los límites. Se utilizan para representar la secuencia de las actividades, la interacción entre canales y, además, vinculan la información adicional que se deseé integrar al diagrama o modelado de procesos [31].

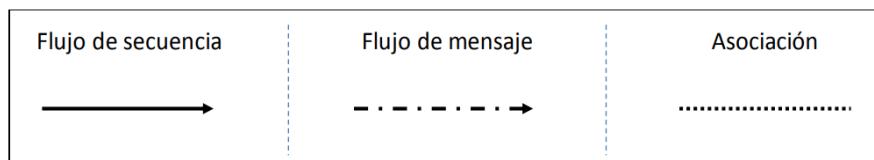


Figura 17. Objetos de Conexión BPMN [31].

- **Los flujos de secuencia:** se representan mediante línea con punta de flecha, son utilizados para representar la secuencia de las actividades y el control del flujo del procedimiento.
- **Los flujos de mensaje:** se representan mediante línea discontinua o intermitente con punta de flecha, son utilizados para representar interacción entre varios procesos o pool, son mensajes, no controlan el flujo, no tienen un orden en particular. Pueden unir actividades, inicios de otros procesos y directamente a la línea del pool.
- **Asociación:** se representa a través de una línea a base de puntos consecutivos, la cual se utiliza para vincular o conectar un artefacto (anotaciones, objeto de datos o grupos) o para asociar tareas de compensación [31].

4.3. Marcos de trabajo

Los marcos de trabajo son un procedimiento de gestión de proyectos que disminuye la complejidad en el desarrollo de productos para satisfacer las necesidades de un público

objetivo. Al seguir el procedimiento se aplican continuamente un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto.

Los marcos de trabajo son considerados como modelos de referencia que define un conjunto de prácticas y roles y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante la puesta en marcha de un proyecto [32].

A continuación, se citan los dos marcos de trabajo más utilizados, por un lado, Scrum y por otro, Design Thinking.

4.3.1. Scrum

Scrum es un marco de trabajo o framework, lo emplean los equipos que trabajan proyectos complejos. Se trata de un marco de trabajo ágil que tiene como objetivo primordial la entrega de valor en periodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación.

Al estar enmarcada dentro de las metodologías agile, Scrum se basa en aspectos como:

- La flexibilidad en la adaptación a cambios y nuevos requisitos durante un proyecto complejo.
- El recurso humano.
- La colaboración e interacción continua con el cliente.
- El desarrollo iterativo como garantía de buenos resultados.

El corazón de Scrum es el Sprint, el cual es un bloque de tiempo de generalmente dos semanas, durante el cual se crea un incremento de producto, utilizable y potencialmente entregable.

Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo [33].

4.3.2. Design Thinking

El design thinking es el marco de trabajo sobre la que se establecen los procesos de pensamiento que apuntan a la innovación y al desarrollo de nuevas ideas basadas en necesidades reales, siempre en un marco de actuación que persigue la satisfacción del cliente por la vía más directa, la entrega de valor [34].

Permite generar ideas innovadoras y centra su eficiencia en comprender y dar solución a las necesidades reales de los clientes. Se sustenta de la forma en la que trabajan los diseñadores de producto. Este marco de trabajo se empezó a desarrollar de forma teórica en la Universidad de Stanford en California, EEUU a partir de los años 70, y su primera aplicabilidad con fines lucrativos como “Design Thinking” la llevó a cabo la consultoría de diseño IDEO (Firma de diseño y consultoría), siendo hoy en día su principal precursora [35], [36].

4.3.2.1. ¿Quiénes lo utilizan?

Según Tim Brown, actual CEO de IDEO, el Design Thinking es una disciplina que usa la sensibilidad y métodos de los diseñadores para hacer coincidir las necesidades de las personas con lo que es tecnológicamente factible y con lo que una estrategia viable de negocios puede convertir en valor para el cliente, así como en una gran oportunidad para el mercado [36].

4.3.2.2. Fases de Design Thinking.

Como todo marco de trabajo, Design Thinking contiene una serie de fases que se deben realizar para poder aplicarlo con éxito, mismas que se presentan en la Figura 18.

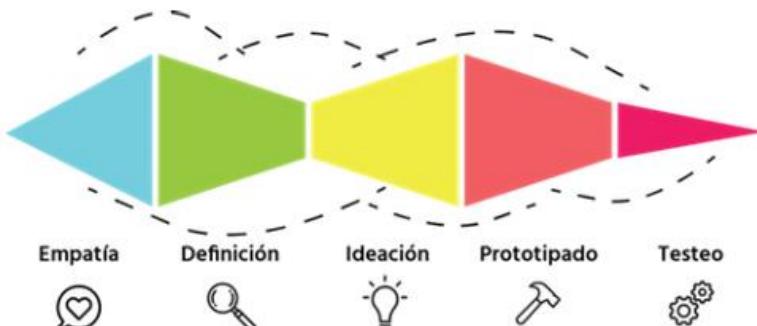


Figura 18. Fases de Design Thinking [35].

Fase de Empatía: El proceso de Design Thinking comienza con una profunda comprensión de los problemas, las necesidades y del entorno de los usuarios implicados en la solución que se está desarrollando. La atención debe centrarse sobre los involucrados y es necesario ponerse en el lugar de ellos para entenderlos y ser capaces de generar soluciones consecuentes con su realidad, en esta etapa se utilizan técnicas que permitan obtener información mediante una interacción directa, ya sea hablando con ellos, observándolos o conociendo su entorno [35].

Fase de Definición: Durante la fase de Definición, se clasifica la información recopilada durante la fase de Empatía y se reserva lo que realmente aporta valor. Se centra en identificar áreas de oportunidad para aportar soluciones innovadoras a los problemas y deseos de los usuarios, durante esta fase se encuentra el reto que se desea resolver mediante la aplicación de diversas técnicas, entre ellas la Identificación de Insights la cual busca patrones de comportamiento y extrae conclusiones, estas conclusiones son las denominadas revelaciones o “insights”; otra de las técnicas que se emplean en esa fase es la denominada ¿Cómo podríamos?, esta se centra en convertir las revelaciones en preguntas, se llaman así porque tratan de trasladar las revelaciones a preguntas que empiecen por: “¿Cómo podríamos...?” [35].

Fase de Ideación: La etapa o fase de Ideación tiene como objetivo generar el mayor número de soluciones posible a los problemas encontrados. No es recomendable quedarse con la primera idea. En esta fase, las actividades favorecen el pensamiento expansivo y se deben eliminar los juicios de valor y las ideas preconcebidas. A veces, las ideas más estrambóticas son las que generan soluciones visionarias [35].

Fase de Prototipado: En la etapa de Prototipado las ideas se vuelven realidad. Construir prototipos hace las ideas palpables. El propósito de esta fase es conseguir una maqueta lo más cercano a la realidad de la solución deseada, de tal forma que posteriormente se pueda validar, y se compruebe que resuelve las necesidades iniciales del público objetivo. No se lo considerará como la solución final del problema, sino como un avance en medio de un ciclo de mejora continua [35].

Fase de Testeo: Durante la fase de Testeo, se evalúa el prototipo con los usuarios involucrados en la solución que se está desarrollando. Esta fase es esencial, ayudará a identificar mejoras significativas, fallos a resolver, posibles carencias. Durante esta fase evoluciona la idea hasta convertirse en la solución que se está buscando.

Es importante tener en cuenta que las cinco fases no siempre son secuenciales. No tienen que seguir ningún orden específico, pueden ocurrir en paralelo y repetirse iterativamente [35].

4.4. Estudios Relacionados

El desarrollo del presente trabajo de titulación tuvo como referencia trabajos relacionados propuestos a nivel nacional, estos trabajos surgen como resultado de una serie de criterios de inclusión y exclusión.

Los criterios de inclusión efectuados fueron:

- Trabajos sobre reconocimiento facial para el control de acceso.
- Los trabajos deben ser a partir del año 2015.
- Trabajos en idioma español.
- Búsqueda de información en artículos, tesis, libros o trabajos relacionados (incluyendo patentes y citas).
- Los resultados son referentes al Ecuador.

Los criterios de exclusión aplicados fueron:

- Los resultados no se relacionan al tema.
- El resultado no cumple los criterios de inclusión.

Obteniendo como resultado 5 trabajos los cuales fueron aporte en el desarrollo del presente documento. Tales estudios se presentan en la tabla 1:

Tabla 1. Trabajos relacionados

#	Título	Palabras clave	Repositorio	Referencia
1	"Sistema biométrico de reconocimiento facial para el control de asistencia del personal docente y administrativo de la Uniandes Tulcán."	Automatización, biométrico, estándar, implantar.	Dspace.uce.edu.ec	[37]
2	"Sistema de control de acceso al personal de la lavadora de jeans fashion mediante reconocimiento facial"	Control de Acceso, Registro, Base de Datos, Reconocimiento Facial, PCA.	repositorio.uta.edu.ec	[38]
3	"Registro de asistencia de alumnos por medio de reconocimiento facial utilizando visión artificial"	Visión artificial, registro de asistencia, reconocimiento facial, verificación de imágenes, comparación de imágenes, procesamiento de imágenes, redes neuronales	repositorio.uta.edu.ec	[39]

	artificiales, red hog, red cnn, aprendizaje supervisado, aprendizaje de refuerzo.	
4	"Reconocimiento facial aplicado a la autenticación de usuarios en cursos online de la carrera de docencia en informática de la facultad de ciencias humanas y de la educación de la universidad técnica de Ambato"	
5	"Sistema informático de reconocimiento facial para el Registro y control de asistencia de los socios de la Cooperativa de taxis y camionetas Puyo".	repository.uta.edu.ec [40]

El primer trabajo [37], es una propuesta de un sistema informático con dispositivo biométrico para mejorar el control de asistencia del personal docente y administrativo de la Uniandes Tulcán, con la tecnología de Visual Basic y MySQL utilizando el modelo preentrenado EigenFaces. El segundo trabajo [38] se trata de una propuesta que plantea el método de reconocimiento facial para el control de acceso al personal de una empresa, está desarrollado con la tecnología Matlab, MySQL y la técnica LBPH (Local Binary Patterns). El tercer trabajo relacionado [39], corresponde a un trabajo de titulación para la obtención de grado académico de Magister en automatización de sistemas de control y plantea una propuesta basada en la visión artificial para el registro de asistencia de los alumnos de un centro educativo a fin de automatizar el proceso tradicional de llevar un registro de asistencia en documentación física, utilizó redes neuronales artificiales de Histograma de Gradientes orientados HOG y redes neuronales convolucionales con la librería OpenCV y Face-recognition (dlib). El cuarto trabajo [40], propone un recurso tecnológico para mejorar el Sistema de Reconocimiento facial para la autenticación de cursos online en la Carrera de Docencia en Informática de la Universidad Técnica de Ambato, utiliza Matlab y EigenFaces como recurso de reconocimiento facial para la autenticación de rostros de usuarios. El último trabajo [41], se trata de una propuesta de implementación de un sistema informático de reconocimiento facial para el registro y control de asistencia de los socios de la cooperativa de Taxis y Camionetas Puyo, una herramienta

útil para obtener datos confiables de los socios en las reuniones beneficiando y facilitando el trabajo manual al personal administrativo de la institución. En esta propuesta se utiliza OpenCV, FisherFaces para el reconocimiento facial, el servidor web Apache y la base de datos PostgreSQL.

En resumen, los trabajos anteriormente citados son propuestas de tesis con temas relacionados al reconocimiento facial para la autenticación de usuarios y registro de asistencia, sin embargo, ninguno de estos presenta una propuesta de detección de vida en el proceso de reconocimiento facial.

5. Metodología

Esta sección explica los materiales y métodos utilizados durante el desarrollo del presente Trabajo de Titulación. En la sección 1 se explica el área de estudio en donde se llevó a cabo el presente TT; en la sección 2 se indica el proceso para cumplir con cada objetivo y sus respectivas tareas; además, la sección 3 presenta los recursos utilizados; finalmente en la sección 4 se detallan los participantes del TT.

5.1. Área de estudio

El presente Trabajo de Titulación se realizó en el ámbito académico durante el período de octubre 2021 hasta diciembre 2022 en la Carrera de Ingeniería en Sistemas/Computación de la FEIRNNR de la Universidad Nacional de Loja, en colaboración con los docentes y estudiantes que hacen uso del laboratorio de Software de CIS-C/UNL.

5.2. Procedimiento

Para alcanzar el objetivo general del presente proyecto de investigación se consideró las fases del marco de trabajo Design Thinking y se utilizó el siguiente procedimiento por cada uno de los objetivos específicos.

Etapa 1: Empatía y Definición.

Objetivo 1. Analizar el proceso actual de registro de asistencia en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja.

Empatía

- Se analizó el proceso actual de registro de asistencia al laboratorio de Software mediante la aplicación de entrevistas y encuestas a docente, técnico del laboratorio y estudiantes de CIS/C-UNL (Véase sección Resultados [Objetivo 1 apartado 6.1.1](#))
- Se diagramó el proceso de registro de asistencia en el laboratorio de Software haciendo uso de la notación BPMN con la herramienta LucidChart (Véase sección Resultados [Objetivo 1 apartado 6.1.2](#)).

Definición.

- Se identificó las revelaciones según el análisis de las encuestas y entrevistas (Véase sección Resultados [Objetivo 1 apartado Revelación](#)).

- Se definió el reto a resolver aplicando la técnica “¿Como podríamos?” (Véase sección Resultados [Objetivo 1 apartado Planteamiento del reto](#)).

Etapa 2: Ideación y Prototipado.

Objetivo2. Desarrollar el prototipo de reconocimiento facial con detección de vida aplicando Clasificadores Preentrenados con Redes Neuronales Convolucionales.

Ideación

- Se ejecutó la fase de *Ideación* para ello se empleó la técnica de lluvia de ideas a fin de recopilar cada una de las posibles soluciones ante el reto propuesto, se analizó una a una cada idea y se eligió la que mejor aporta valor innovador y que cubra la necesidad real de las personas involucradas (Véase sección Resultados [Objetivo 2](#)).
- Se analizaron los requerimientos del software de reconocimiento facial con detección de vida para el registro de asistencias (Véase sección Resultados [Objetivo 2 apartado 6.2.1](#)).

Prototipado

- Se modeló la arquitectura del software empleando la notación UML haciendo uso de la herramienta LucidChart (Véase sección Resultados [Objetivo 2 apartado 6.2.2](#)).
- Se desarrolló el prototipo de software para el registro de asistencias mediante reconocimiento facial con detección de vida usando el Framework Odoo ERP con el gestor de Base de Datos PostgreSQL. (Véase sección Resultados [Objetivo 2 apartado 6.2.3](#)).

Etapa 3: Testeo

Objetivo 3. Evaluar mediante métricas de la Matriz de Confusión la precisión del prototipo de reconocimiento facial con detección de vida para el registro de asistencias en entornos simulados.

Testeo

- Se evaluó mediante métricas de la Matriz de confusión la precisión del reconocimiento facial, se empleó el método experimental (Véase sección Resultados [Objetivo 3 apartado 6.3.1](#)).
- Se evaluó mediante métricas de la Matriz de Confusión la precisión de la detección de vida, se empleó el método experimental (Véase sección Resultados [Objetivo 3 apartado 6.3.2](#)).

5.3. Recursos

Con la finalidad de lograr el cumplimiento de los objetivos planteados, se hizo uso de los siguientes recursos:

5.3.1. Recursos Científicos

- **Investigación Bibliográfica:** con esta técnica se fundamentó la revisión de literatura del presente TT, se consultó fuentes bibliográficas confiables, libros, revistas científicas, artículos, casos de éxito, documentos varios, entre otros (Véase sección [Marco teórico](#)).
- **Estudios de caso:** Esta técnica permitió identificar trabajos elaborados dentro de Ecuador y relacionados al reconocimiento facial para el registro de asistencia (Véase sección [Estudios Relacionados](#)).
- **Método Científico:** este método fue la base para el desarrollo del presente TT, desde que se planteó la Propuesta de Trabajo de Titulación (PTT), hasta la culminación del mismo. Esto se puede evidenciar a través del desarrollo de las actividades planteadas por cada uno de los objetivos y los resultados obtenidos en cada una de sus fases (véase sección [Resultados](#)), en conjunto con la respectiva discusión (véase sección [Discusión](#)) y sus conclusiones (véase sección [Conclusiones](#)).
- **Método experimental:** Este método se empleó en la fase de desarrollo y pruebas del presente prototipo, de tal manera que permitió realizar diversos experimentos del funcionamiento del prototipo de reconocimiento facial con prueba de vida, dichos experimentos se basaron en cambios de posición del usuario, cambios de expresiones faciales y cambios de iluminación, en caso de la prueba de vida, se experimentó con fotos impresas y videos de un dispositivo móvil (véase sección Resultados [Objetivo 3](#)).

5.3.2. Recursos Técnicos

- **Herramientas Colaborativas:** se utilizó Zoom como herramienta de software para la comunicación, permitió el trabajo colaborativo y la realización de entrevistas, además se utilizó Google Forms para la realización de encuestas (Véase [Anexo 1](#), [Anexo 2](#), [Anexo 4](#), [Anexo 9](#), [Anexo 11](#)).
- **Software para la diagramación:** se utilizó LucidChart para diagramar el proceso actual del registro de asistencias con la notación BPMN y UML (véase sección Resultados [Objetivo 1 apartado 6.1.2](#) y [Objetivo 2 apartado 6.2.2](#)).

- **Marco de Trabajo Design Thinking:** Se empleó durante todo el TT en cada uno de los objetivos se aplicaron determinadas fases para cumplir el objetivo general. En el Objetivo 1 se cumplió las fases de Empatía y Definición (véase Resultados sección [Etapa 1: Empatía y Definición](#)), en el Objetivo 2 se cumplió las fases de Ideación y Prototipado (véase Resultados sección [Etapa 2: Ideación y Prototipado](#)), en el Objetivo 3 se cumplió la Fase de Testeo (véase Resultados sección [Etapa 3: Testeo](#)).

5.3.3. Recursos Tecnológicos.

- **Pycharm:** Entorno de Desarrollo Integrado que permitió la escritura y depurado de código.
- **Odoo ERP:** Sistema ERP de código abierto en el cual se desarrolló y ejecutó el módulo de Software.
- **PostgreSQL:** Permitió gestionar la Base de Datos del módulo de reconocimiento facial dado que esta es la utilizada por el ORM (mapeo objeto-relacional) de Odoo.
- **OpenCV:** Biblioteca de código abierto especializada en el sistema de visión artificial y machine learning, permitió realizar el sistema de reconocimiento facial.
- **Face-recognition:** Librería de código abierto desarrollada en lenguaje Python que a su vez implementa dlib, la librería de reconocimiento facial más simple del mundo que incluye una red neuronal de tipo convolucional pre-entrenada que permite detectar caras y rasgos faciales con una efectividad del 99.38%
- **Keras:** Biblioteca de Redes Neuronales de código abierto escrita en Python.

5.4. Participantes

El presente TT enfocado en la línea de investigación de Sistemas Inteligentes fue desarrollado por los siguientes participantes:

- Dayanna Alvarado Castillo, como estudiante autora del presente TT. Sus actividades iniciaron desde el planteamiento del tema del PTT, hasta el desarrollo y finalización de los objetivos planteados en el presente TT.
- El Ing. Oscar Cumbicus Pineda como tutor académico y director del TT, quien supervisó los avances académicos y técnicos desarrollados por la tesista.
- El Ing. Luis Sinche, técnico encargado del laboratorio de Software de CIS-C/UNL, quien participó en la entrevista para conocer el proceso actual del registro de asistencia en el laboratorio de software de CIS-C/UNL, revisó y validó el documento de especificación de requerimientos de software.

- El Ing. Wilman Chamba, docente en planificación de retorno a las prácticas presenciales según el documento “*Plan general para el retorno seguro a las prácticas presenciales en los laboratorios de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables*” ([Véase el Documento](#)), quién participó en la entrevista para conocer el proceso actual del registro de asistencia en el laboratorio de software de CIS-C/UNL, revisó y validó el diagrama BPMN del estado actual de registro de asistencia y el documento de especificación de requerimientos de Software.
- Estudiantes de sexto a décimo ciclo de CIS-C/UNL, seleccionados debido a que en tiempos prepandemia hicieron uso del laboratorio durante al menos un ciclo académico, participaron en la encuesta para conocer el proceso actual del registro de asistencia en el laboratorio de software de CIS-C/UNL y durante la fase de pruebas del prototipo.

6. Resultados

En esta sección se presenta los resultados obtenidos durante el desarrollo del presente trabajo de Titulación. Se enumeran las fases del marco de trabajo Design Thinking ejecutadas para cumplir con las tareas y objetivos planteados.

Para alcanzar el primer objetivo, se inició con la ejecución del marco de trabajo Design Thinking, se desarrollaron las fases de Empatía y Definición, para la primera se aplicó las técnicas de encuesta y entrevista a fin de obtener información de los usuarios de tal manera que se conozca el estado actual del proceso de registro de asistencias en el laboratorio de software, seguidamente, la segunda fase permitió identificar y definir las revelaciones obtenidas de las encuestas y entrevistas y se prosiguió con el planteamiento del reto a resolver.

En el segundo objetivo se completaron las fases de Ideación y Prototipado, para la primera se aplicó la técnica de lluvia de ideas, a fin de recopilar todas las posibles soluciones que se le pudiesen dar al reto planteado, se analizaron los requerimientos funcionales y no funcionales que debe cumplir la alternativa de registro de asistencia y se modeló la arquitectura a desarrollar. Para la etapa de prototipado se desarrolló el Prototipo de Reconocimiento Facial con Prueba de Vida para el registro de asistencias. Como producto final del objetivo 2 se tiene documentación de la especificación de requerimientos y el prototipo funcional del sistema propuesto.

En el tercer objetivo se ejecutó la etapa de Pruebas, se realizaron pruebas en ambientes reales de los modelos que conforman el prototipo de reconocimiento facial con detección de vida y los resultados se evaluaron mediante métricas de la matriz de confusión.

Etapa 1: Empatía y Definición

6.1. Objetivo 1:

Analizar el proceso actual de registro de asistencia en el laboratorio de Software de la Facultad de Ingeniería en Sistemas/Computación del Área de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja.

Fase de Empatía

Con el fin de enfatizar y obtener información verídica de los usuarios y conocer el proceso actual que se lleva a cabo para el registro de asistencia en el laboratorio de Software de la Carrera de Ingeniería en Sistemas-Computación de la Universidad Nacional de Loja, se

realizaron un conjunto de actividades que involucra a estudiantes y docentes a través de métodos y técnicas de investigación que permitieron la recolección de información necesaria.

6.1.1. Tarea 1: Analizar el proceso actual de registro de asistencia al laboratorio de Software mediante la aplicación de entrevistas y encuestas a docentes y estudiantes de CIS/C-UNL

La aplicación de las técnicas de entrevista y encuesta permitió resolver la etapa de Empatía correspondiente a la primera fase del marco de trabajo Design Thinking, se pudo conocer las experiencias y opiniones de los usuarios en cuanto al proceso actual del registro de asistencia en el laboratorio de Software de CIS-C UNL.

6.1.1.1. Objetivo de las entrevistas y encuestas.

Se realizó la aplicación de encuestas y entrevistas con el objetivo de corroborar el uso del laboratorio de software de CIS/C UNL, además de captar información acerca del proceso actual que se lleva a cabo para el registro de asistencia de los estudiantes en mencionado laboratorio, así como también, tomando en cuenta la actual crisis sanitaria por Covid-19, obtener sugerencias, opiniones o deseos en cuanto al proceso de registro.

6.1.1.2. Selección de participantes.

Los usuarios seleccionados para este proceso se describen a continuación:

- **Estudiantes.** – Quienes han cumplido sus labores académicas haciendo uso del laboratorio de Software al menos un ciclo académico, por tanto, se seleccionó a estudiantes a partir de 6to ciclo, los mismos que, antes de la suspensión de las prácticas presenciales a causa de COVID-19, cursaban ciclos superiores al primero, los mencionados estudiantes son parte de los usuarios que utilizan con mayor concurrencia el laboratorio, por tanto, su participación forma parte de las bases para el presente trabajo de titulación.
- **Docentes.** - Al igual que los estudiantes, los docentes hacen uso frecuente del laboratorio y tienen experiencia adecuada para brindar información oportuna para el desarrollo del presente trabajo de titulación, en este caso se seleccionó al ingeniero Wilman Chamba, docente que consta en el “*Plan general para el retorno seguro a las prácticas presenciales en los laboratorios de la facultad de la Energía, las Industrias y los Recursos Naturales no Renovables*” ([Véase el Documento](#)).

- **Técnico encargado del Laboratorio de Software.** - Es la persona encargada netamente de la gestión del laboratorio, su labor es fundamental, controla que el laboratorio sea usado de forma correcta, ya sea por personal de la universidad o ajeno a ella, además de ser responsable del material tecnológico utilizado dentro de la instalación. Él es el conocedor de todos los procesos que se llevan a cabo en cada jornada académica y es el usuario con mayor información primordial a aportar al trabajo de titulación.

6.1.1.3. Análisis e Interpretación de las entrevistas.

- **Entrevista realizada a docente usuario del laboratorio de Software.**

En la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja (CIS/C-UNL), mediante la entrevista realizada al Ingeniero Wilman Chamba, docente en planificación del Retorno Seguro a Prácticas de CIS/C-UNL, se obtuvo que en tiempos prepandemia él hacía uso del laboratorio de software y los estudiantes registraban la asistencia colocando manualmente su nombre y firma en un formulario, dicho formulario se traspasaba de estudiante en estudiante, del mismo modo, el docente también firmaba ese documento. El docente da a conocer que se encuentra aprobado el Plan de retorno seguro a las prácticas presenciales en los laboratorios para el periodo académico octubre-abril 2021 y que continuar con el proceso de registro de asistencia de la misma forma que se hacía antes de la suspensión presencial, puede ser riesgoso en temas de salud. El entrevistado menciona que, “*no es recomendable que los estudiantes se traspasen documentos ya que puede ser un foco infeccioso de coronavirus, a pesar de que el plan de retorno progresivo a las prácticas presenciales manifiesta hacer uso constante de alcohol y mascarilla, hay que evitar en lo máximo compartir instrumentos con las demás personas, hay que estar alertas y no subestimar ninguna acción ante el hecho del contagio*”, por tanto considera que, “*sería factible cambiar la forma del registro de asistencia, pudiera ser por métodos biométricos de reconocimiento facial, con dispositivos colocados en la entrada o dentro de los laboratorios de prácticas, a fin de que detecte y reporte la asistencia de los estudiantes*” (Ver [Anexo 1](#)).

- **Entrevista realizada al técnico encargado del laboratorio de Software.**

Mediante la entrevista realizada al Ingeniero Luis Sinche, técnico encargado del laboratorio de Software de CIS-C /UNL, se obtuvo que, en tiempos prepandemia, el docente y los estudiantes ingresaban al laboratorio, se le entregaba al docente un formulario de registro, en

donde cada estudiante iba llenando su nombre, su número de cédula, el número de computador que utilizará y su firma, a fin de tener registro de los estudiantes que utilizaron el laboratorio en un determinado horario; cuando todos los estudiantes se registraban, el docente colocaba su firma al final de la hoja. El tiempo que conlleva el registro de asistencia, depende del número de estudiantes que asistan, variando de cinco a quince minutos. En caso de que se suscite algún daño en los equipos tecnológicos del laboratorio, es el docente quien informa al técnico encargado y se registra el daño en el apartado de observaciones presente en el formulario de registro, colocando el nombre del estudiante causante del daño y el equipo afectado. Por otro lado, la planificación para la asignación de horarios se realiza con los directores de cada carrera a fin de evitar cruces de horario, en esta planificación consta la materia y el horario en el que se va a ocupar el laboratorio, en cuanto a las temáticas que se van a impartir, es el docente quien lleva una planificación y antes de realizar la práctica da a conocer la temática para su respectivo registro. Además, el entrevistado menciona que no se ha experimentado que los estudiantes ingresen al laboratorio en un horario que no les corresponde, dado que es el docente quien ingresa primero y conoce la planificación donde se detalla el horario que se le ha establecido. Entre los inconvenientes que se ha tenido con los registros, se menciona que en ocasiones los estudiantes olvidan registrarse y realizan reclamos de que, si han asistido, así mismo, en ocasiones el docente olvida firmar el documento o mencionar el nombre de la práctica y se debe acudir a él a fin de obtener los datos faltantes, además, de que se corre el riesgo de que los archivos, al ser material físico, se puedan destruir.

También se obtuvo que, las hojas de registro se almacenan en archivos a fin de poder construir informes cuando el decanato lo requiera, se suelen solicitar informes sobre el docente o la materia que más utilizó el laboratorio, así como el máximo o mínimo número de estudiantes que acudieron, para construir estos informes es necesario ir revisando hoja por hoja para contabilizar y llenar los datos en una hoja de Excel. Finalmente, el técnico menciona que “*continuar con el mismo proceso de registro que se hacía antes de la pandemia, podría ser un foco infeccioso de coronavirus, el registro de asistencia debería hacerse de una forma en la que el estudiante no tenga contacto con el material, sería factible hacer el registro por medio de visión artificial que detecte el rostro. Además, este mecanismo disminuiría el tiempo de registro de asistencia y si se contara con un software de administración, se facilitaría en gran medida la construcción de los informes del uso del laboratorio*”. (Ver [Anexo 2](#)).

Además, se obtuvo el formato actual de registro de asistencia de los estudiantes, el cual contempla campos como: ciclo, paralelo, docente, fecha, asignatura, periodo académico, tema de la práctica, nombre del alumno, número de computador que utilizará, horario, número de cédula y firma del estudiante, observaciones y firma del docente (Ver [Anexo 3](#)).

6.1.1.4. Análisis e Interpretación de la encuesta.

De conformidad al Trabajo de Titulación se realizó una encuesta dirigida a los estudiantes en Sistemas/Computación de la Universidad Nacional de Loja, se seleccionó como participantes a los estudiantes que actualmente se encuentran cursando ciclos superiores al quinto, dado que ellos hacían uso del laboratorio de Software antes de la suspensión de las prácticas presenciales, se consiguió un total de 47 respuestas.

Pregunta N°1. ¿En tiempos prepandemia del COVID 19, realizaba usted prácticas en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación?

Tabla 2. Utilización de laboratorio de software.

Alternativas	Frecuencia	Porcentaje
Si	35	74,5%
No	12	25,5%
TOTAL	47	100%

Fuente: Encuesta a estudiantes

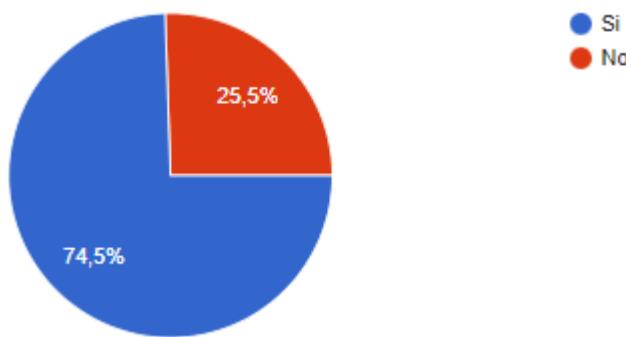


Figura 19. Utilización de laboratorio de Software

Obtenido de *Encuesta estructurada Google Forms*

Interpretación

Se puede evidenciar en la Figura 19 que el 74,5% de los estudiantes encuestados han hecho uso alguna vez del laboratorio de software de CIS/C, a diferencia del 25% que afirman no haberlo utilizado.

Pregunta N°2. ¿En tiempos prepandemia del COVID 19, cómo registraba usted su ingreso al laboratorio de Software de CIS/UNL?

Tabla 3. Registro de asistencia en laboratorio

Alternativas	Frecuencia	Porcentaje
El encargado del laboratorio se acercaba a correr lista.	1	2,1%
El encargado del laboratorio acercaba un formulario de registro donde usted y sus compañeros firmaban manualmente.	37	78,7%
No registraba asistencia al laboratorio de Software.	9	19,1%
TOTAL	47	100%

Fuente: Encuesta a estudiantes



Figura 20. Registro de Asistencia

Obtenido de Encuesta estructurada Google Forms

Interpretación

Respecto a la Pregunta 2, el 78,7% de los estudiantes encuestados aseguran que el registro de su asistencia se llevaba a cabo firmando manualmente sobre un formulario que el encargado del laboratorio les acercaba, mientras que el 19,1% responde que no registraba

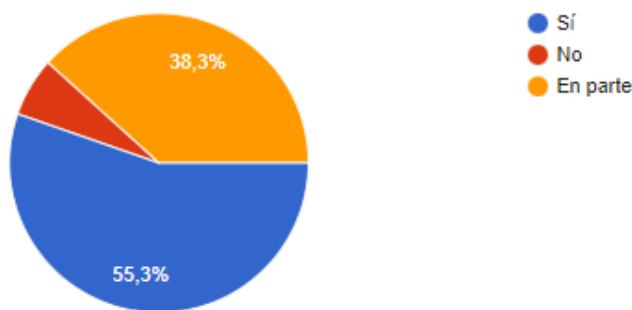
asistencia en el laboratorio y el 2,1% de los encuestados responden que el encargado del laboratorio se acercaba a correr lista. Por tanto, se puede constatar que el registro de asistencia en el laboratorio de software se realiza mediante el llenado manual de un formulario de registro, mencionado formulario es entregado por el encargado del laboratorio y traspasado a cada uno de los estudiantes de un determinado curso.

Pregunta N°3. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, considera usted que si se continúa llevando a cabo el proceso de registro de asistencia del mismo modo que se hacía antes de la pandemia, en el retorno a las prácticas presenciales pudiera ser un foco infeccioso en temas de Coronavirus?

Tabla 4. Riesgo de contagio

Alternativas	Frecuencia	Porcentaje
SI	26	55,3%
No	3	6,4%
En parte	18	38,3%
TOTAL	47	100%

Fuente: Encuesta a estudiantes



*Figura 21. Riesgo de Contagio
Obtenido de Encuesta estructurada Google Forms*

Interpretación

En cuanto a la pregunta 3, según se observa en la Figura 21, el 55,3% afirma que continuar con el mismo proceso de registro de asistencia al laboratorio podría ser un foco de contagio de coronavirus, mientras que el 38,3% respondieron que en parte y el 6,4% afirman que no

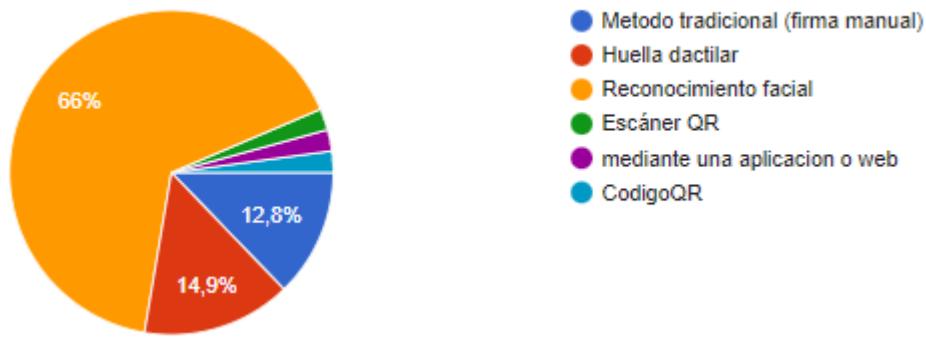
es un foco de contagio continuar con el mismo proceso de registro. Por tanto, se puede manifestar que los encuestados consideran que tomando en cuenta la crisis sanitaria por la que se atraviesa actualmente debido a la presencia de Covid-19, continuar con el mismo proceso de registro de asistencia que se seguía en tiempos prepandemia podría ser un foco infeccioso de coronavirus.

Pregunta N°4. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, con cuál de los siguientes métodos se sentiría usted más cómodo, al momento de registrar su asistencia en el laboratorio de Software de CIS-C/UNL.

Tabla 5. Método de registro de asistencia

Alternativas	Frecuencia	Porcentaje
Método tradicional (firma manual)	6	12,8%
Huella Dactilar	7	14,9%
Reconocimiento facial	31	66,0%
Escáner QR	2	4,2%
Mediante una aplicación o web	1	2,1%
TOTAL	47	100%

Fuente: Encuesta a estudiantes



*Figura 22. Método de registro de asistencia
Obtenido de Encuesta estructurada Google Forms*

Interpretación

Como resultado de la pregunta 4, se obtiene que el 66% de los encuestados se inclinan por el reconocimiento facial como método de registro de asistencia, mientras que el 14,9% prefieren el método de huella dactilar, el 4.2% prefieren el método de escáner de código QR

y el 2.1% se inclinan por la opción de llevar un registro de asistencia mediante una aplicación o servicio web.

6.1.2. Tarea 2: Diagramar el proceso de registro de asistencia en el laboratorio de Software haciendo uso de la notación BPMN con la herramienta LucidChart.

En base a la recopilación de la información en la tarea 1 del presente objetivo, se obtiene el proceso actual para llevar el registro de asistencia de los estudiantes en el laboratorio de software de la Carrera de Ingeniería en Sistemas-Computación, según la notación BPMN en la Figura 23 a continuación. El diagrama fue revisado y validado por el Ingeniero Wilman Chamba, docente de CIS/UNL y usuario del laboratorio de Software ([Ver Anexo 5](#)).

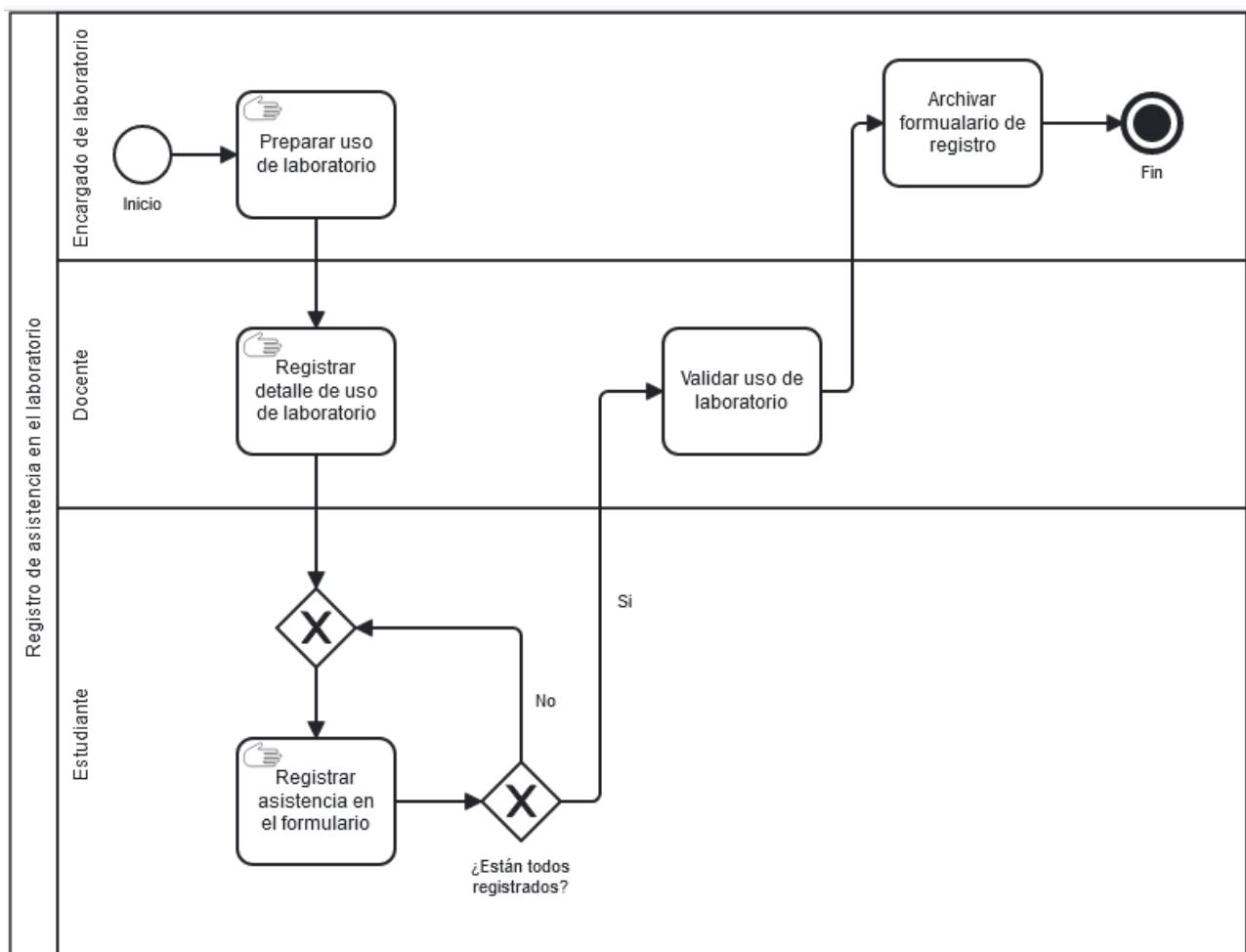


Figura 23. Proceso actual de registro de asistencia

Descripción del proceso

Tabla 6. Descripción de las tareas del proceso actual de registro de asistencia

Proceso actual de registro de asistencia en el laboratorio de CIS-C/UNL

Evento de inicio:	Calendarización de uso del laboratorio	
Rol	Tarea	Descripción
Encargado del laboratorio de Software.	Preparar el uso del laboratorio.	Abre el laboratorio, verifica el correcto estado de los equipos tecnológicos y entrega al docente un formulario de registro de asistencia.
	Archivar formulario de registro.	Recepta el formulario lleno y lo almacena.
Docente	Registrar detalle del uso del laboratorio	Detalla el ciclo y paralelo con el que se va a realizar la práctica, la asignatura y el tema de la práctica que se va a impartir.
	Validar uso del laboratorio	Verifica que todos los estudiantes hayan llenado el formulario correctamente, en caso de existir observaciones las detalla en el formulario, y hace válido el documento colocando su firma al final.
Estudiante	Registrar asistencia en el formulario.	Registran asistencia colocando datos tales como nombres, apellidos, hora de ingreso y salida del laboratorio, el número de computadora que se va a utilizar, número de cédula y firma. El documento se traspasa a todos los compañeros del curso.

Fase de Definición

Esta fase se centra en identificar las áreas de oportunidad para aportar un valor a los usuarios. Para ello se identificó revelaciones o también conocidas como ‘Insights’, estas revelaciones se las obtienen a partir de la identificación de patrones de las encuestas y entrevistas realizadas, una vez conocidas las revelaciones se procede a plantear un reto al que se va a dar solución en el presente TT.

a. Revelación

En base a las encuestas realizadas a los estudiantes (preguntas 3 y 4), a la entrevista realizada al docente (preguntas 4 y 5), y a la entrevista realizada al técnico encargado (preguntas 8 y 9), se obtienen las siguientes revelaciones:

- Se considera que si se continúa llevando a cabo el proceso de registro de asistencia del mismo modo que se hacía antes de la pandemia, en el retorno a las prácticas presenciales pudiera ser un foco infeccioso en temas de Coronavirus.
- Se desea una alternativa para el registro de asistencia que no requiera la manipulación de instrumentos de uso comunitario.
- Se desea una alternativa donde el registro de asistencia requiera un tiempo mínimo.
- Se desea una alternativa que facilite la elaboración de los reportes del uso del laboratorio.

b. Planteamiento del reto

Para el planteamiento del reto, se seleccionará en primer lugar la idea que será solución y cubra las necesidades identificadas en las revelaciones anteriormente descritas, para ello, se realiza una comparativa de las ideas tomando en cuenta la pregunta 4 de la encuesta a estudiantes, pregunta 4 de la entrevista al docente y pregunta 9 de la entrevista al técnico encargado del laboratorio. La idea que se tome como solución deberá contemplar los siguientes criterios.

1. La idea debe permitir el registro de asistencia.
2. Tomando en cuenta el marco de trabajo Design Thinking que se está siguiendo en el presente TT, la idea debe ser innovadora.
3. La idea debe prevenir ataques de presentación.
4. Debe ser fácil de usar.
5. La idea no debe requerir la manipulación de objetos para el registro de asistencia.
6. El registro de asistencia debe tomar un tiempo mínimo.
7. La idea debe tener aceptación de los usuarios finales.

A continuación, se presenta la tabla 7 como comparativa de las ideas para alternativas al registro de asistencia.

Tabla 7. Comparativa de las ideas para registro de asistencia

	Huella Dactilar	Tarjeta RFID	Reconocimiento de Iris	Escáner de código QR	Reconocimiento Facial	
¿Permite el registro de asistencia?	Alto	Alto	Alto	Alto	Alto	—
¿Es innovadora?	Medio Alto	Bajo	Alto	Medio	Alto	—
¿Probabilidad de ataques de presentación?	Media	Alta	Baja	Alta	Alta	
¿Facilidad de uso?	Alta	Alta	Alta	Alta	Alta	
¿Requiere la manipulación de instrumentos de uso comunitario?	Alto	Bajo	Bajo	Bajo	Bajo	
¿Tiempo que toma el registro de asistencia?	Bajo	Bajo	Bajo	Bajo	Bajo	
¿Aceptación de los usuarios finales?	Media Baja	Baja	Baja	Baja	Alta	

Se selecciona el reconocimiento facial como alternativa para el registro de asistencia en el laboratorio de Software de CIS/C UNL, dado que, considerando las respuestas obtenidas durante la aplicación de encuestas y entrevistas se tiene que el 66% de los estudiantes se apegan por el reconocimiento facial como alternativa para el registro de asistencia, y según la comparativa realizada en la Tabla 7, esta idea cumple con los criterios de aceptación establecidos, tiene un alto grado de innovación, es confiable, no requiere la manipulación de instrumentos de uso comunitario, el reconocimiento se realiza en segundos y es fácil de usar. Sin embargo, el reconocimiento facial por sí solo no brinda prevención contra ataques de

presentación, puesto que este tipo de biometría reconoce el rostro que se presenta, pero no detecta si se trata de una foto o un video en lugar de la persona en tiempo real, es por ello que se adiciona mecanismos de detección de vida para solventar este inconveniente. El reconocimiento facial con detección de vida se integra a un módulo web que facilite la administración y la generación de informes, a fin de dar solución a la necesidad planteada por el técnico encargado del laboratorio.

Una vez que se obtiene la solución a desarrollar y aplicando la técnica sugerida por el marco de trabajo que se está siguiendo en el presente TT, Design Thinking “¿Cómo podríamos?”, se obtiene los siguientes retos.

- ¿Cómo podríamos desarrollar una alternativa para el registro de asistencia en el laboratorio de software utilizando reconocimiento facial?
- ¿Cómo podríamos desarrollar detección de vida en el reconocimiento facial para el registro de asistencia en el laboratorio de software?

Etapa 2: Ideación y Prototipado

6.2. Objetivo 2:

Desarrollar el prototipo de reconocimiento facial con detección de vida aplicando Clasificadores Preentrenados con Redes Neuronales Convolucionales.

Para dar cumplimiento al presente objetivo se ejecutan las fases de Ideación y Prototipado del Marco de Trabajo Design Thinking.

Fase de Ideación

En esta fase se aplicó la técnica de lluvia de ideas, a fin de poder recopilar el mayor número de posibles soluciones a los retos planteados en la fase de Definición, esta lluvia de ideas es trabajada por la autora del presente TT y el Ingeniero Oscar Cumbicus como director de TT.

A continuación, se presenta la lluvia de ideas para dar respuesta a la pregunta ¿Cómo podríamos desarrollar una alternativa para el registro de asistencia en el laboratorio de software utilizando reconocimiento facial?

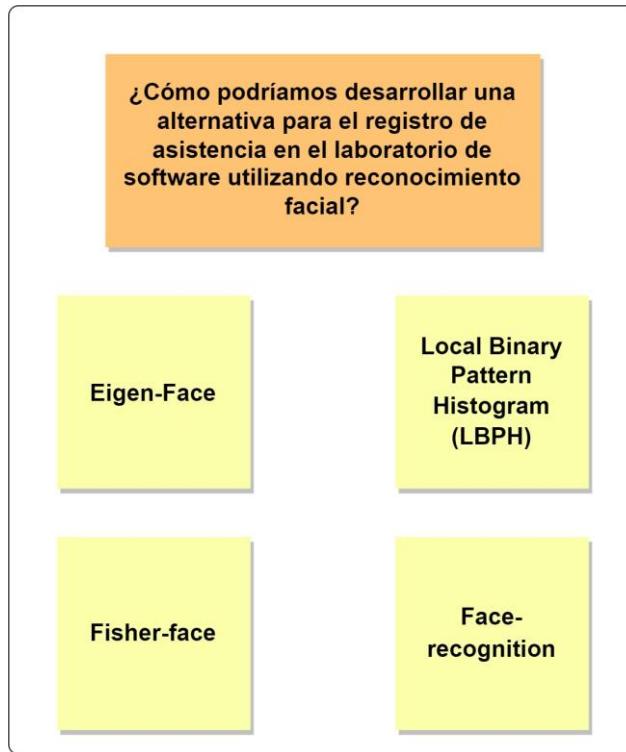


Figura 24. Lluvia de ideas desarrollo Reconocimiento Facial.

La figura 24 recopila cuatro ideas diferentes, por ello, para llevar a cabo la selección de la solución más adecuada para desarrollar el reconocimiento facial se prosigue a comparar cada una de las ideas propuestas considerando los siguientes criterios:

1. Tolerancia a cambios de luminosidad.
2. Tolerancia a cambios de expresiones faciales.
3. Fotografías por individuo en la base de datos.
4. Tiempo de entrenamiento.
5. Tiempo de respuesta de reconocimiento.
6. Fiabilidad de reconocimiento.

Para dar respuesta a cada uno de los criterios, se toma en cuenta los resultados obtenidos en estudios relacionados [42][43][44]. La idea que presente mejores resultados, se considerará como solución para desarrollar el reconocimiento facial para el registro de asistencia en el Laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación. La comparativa se presenta en la Tabla 8 a continuación.

Tabla 8. Comparativa de las ideas para el reconocimiento facial

	Eigen-Faces	Fisher Faces	Local Binary Patterns	Face-Recognition
Tolerancia a cambios de luminosidad.	Baja	Media	Media	Muy Alta
Tolerancia a cambios de expresiones faciales.	Baja	Media	Media	Muy Alta
Cantidad de fotografías por individuo en la base de datos.	Muy Alta	Muy Alta	Muy Alta	Muy Baja
Tiempo de entrenamiento.	Alto	Alto	Medio	Muy bajo
Tiempo de respuesta de reconocimiento.	Bajo	Bajo	Bajo	Bajo
Fiabilidad de reconocimiento.	Media	Media	Alta	Muy Alta

Según la comparación de cada una de las ideas, se obtiene que Face-Recognition es la opción más acertada para el reconocimiento facial, puesto que presenta muy alta tolerancia a cambios de luminosidad y expresiones faciales, además a diferencia de los otros métodos, Face-Recognition requiere solo una imagen de entrenamiento, mientras que los otros requieren aproximadamente 300 fotos por cada individuo, por tanto esta ventaja trae consigo un tiempo de entrenamiento mínimo, además este método tiene muy alta confiabilidad de reconocimiento, reconoce rostros con o sin accesorios y expresiones faciales a pesar de que las fotos de entrenamiento no los tengan.

A continuación, se presenta la lluvia de ideas para dar respuesta a la pregunta ¿Cómo podríamos desarrollar detección de vida en el reconocimiento facial para el registro de asistencia en el laboratorio de software?

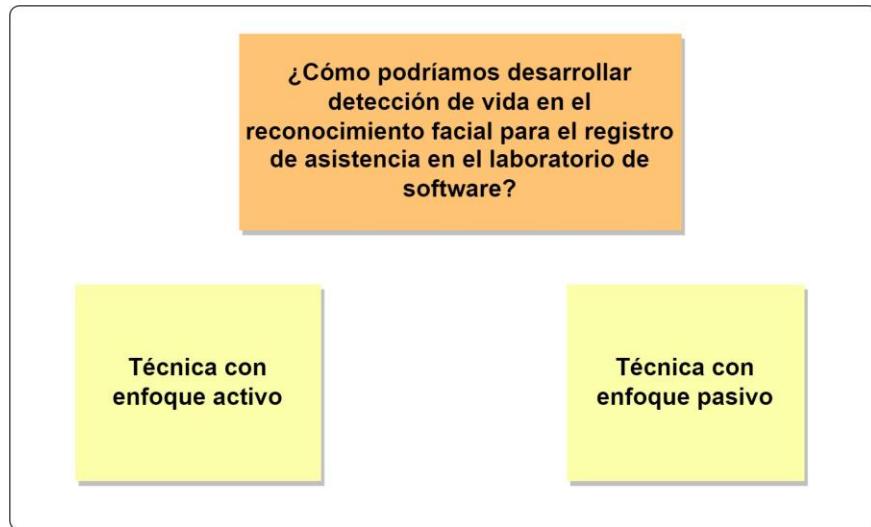


Figura 25. Lluvia de ideas para el desarrollo de Detección de vida.

En la figura 25 se recopilan dos ideas distintas para implementar la detección de vida en el reconocimiento facial, por un lado, el enfoque activo el cual requiere que los usuarios participen en una verificación respondiendo a un desafío, y el enfoque pasivo que no requiere ninguna acción por parte del usuario. Para llevar a cabo la selección de la idea más adecuada, se prosigue a compararlos considerando los siguientes criterios:

1. La idea debe tomar poco tiempo de desarrollo.
2. La dificultad de desarrollo debe ser baja.
3. La idea debe ajustarse a los requerimientos.
4. La idea debe presentar robustez ante la falsificación.

La idea que presente mejores resultados, se considerará como solución para desarrollar la detección de vida en el reconocimiento facial para el registro de asistencia del laboratorio de software de CIS/UNL. La comparativa se presenta en la Tabla 9 a continuación.

Según la tabla comparativa, ambos enfoques se ajustan a los requerimientos, pero, por un lado, las técnicas basadas en enfoque pasivo presentan mayor robustez ante la falsificación, dado que los intrusos no tienen pistas sobre cómo engañar la verificación de vida, sin embargo, la dificultad de desarrollo es alta y consecuentemente su desarrollo requiere un tiempo considerablemente alto; por otro lado, las técnicas basadas en el enfoque activo presentan un nivel medio de robustez ante falsificación y su tiempo y dificultad de desarrollo es medio. Con esta comparativa se evidencia que ambas ideas pueden ser ejecutadas en el presente proyecto, se realizará un prototipo con cada una de las ideas y se elegirá como

solución final la que mejor resuelva la detección de vida y en la experimentación presente mayor robustez ante la falsificación.

Tabla 9. Comparativa de las ideas para la detección de vida

	Técnica con enfoque activo	Técnica con enfoque pasivo
Tiempo de desarrollo.	Medio	Alto
Dificultad de desarrollo.	Media	Alto
Ajuste a los requerimientos	Muy Alto	Muy Alto
Robustez ante la falsificación	Medio	Alto

6.2.1. Tarea 1: Analizar los requerimientos del software de reconocimiento facial con detección de vida para el registro de asistencias.

En esta sección se presenta los requerimientos funcionales y no funcionales generados en la Especificación de Requerimientos de Software (ERS) basado en el estándar IEE830 (Ver [Anexo 6](#)), requerimientos que debe cumplir el prototipo a desarrollar, estos son producto de la reunión de socialización con el encargado del laboratorio y con el docente usuario, para ello se describe su código identificador, nombre y descripción.

6.2.1.1. Requerimientos Funcionales

La tabla de resumen de los requerimientos funcionales se presenta a continuación:

Tabla 10. Requerimientos funcionales

Código	Nombre	Descripción
RF001	Acceso al sistema	El sistema debe permitir al administrador acceder al sistema colocando sus credenciales de ingreso (usuario y contraseña)
RF002	Administrar docentes	El sistema debe permitir crear, modificar, listar o eliminar información de los docentes que utilizarán el laboratorio
RF003	Administrar estudiantes	El sistema debe permitir crear, modificar, listar o eliminar información de los estudiantes que utilizarán el laboratorio

RF004	Administrar carreras	El sistema debe permitir crear, modificar, listar o eliminar información de las carreras que utilizarán el laboratorio
RF005	Administrar materias	El sistema debe permitir crear, modificar, listar o eliminar información de las materias que se impartirán en el laboratorio
RF006	Administrar horarios	El sistema debe permitir crear, modificar, listar o eliminar información de los horarios en los que está disponible el laboratorio para cada grupo de usuarios.
RF007	Generar reportes	El sistema debe permitir generar reportes de los registros de asistencia según intervalos de fechas.
RF008	Registrar asistencia	El sistema debe permitir registrar asistencia a partir de la identificación del rostro mediante cámara web.

6.2.1.2. Requerimientos no Funcionales.

Los requerimientos no funcionales son aspectos que el prototipo no se ve obligado a contemplar pero que en sí influencian directamente en el producto.

Tabla 11. Requerimientos no funcionales

Código	Nombre	Descripción
RNF001	Interfaz gráfica de usuario	El sistema debe tener una IGU basada en plataforma web, visible a través de navegadores de internet.
RNF002	Tiempo de respuesta	El sistema debe dar respuesta rápida como máximo hasta 10 segundos.
RNF003	Usabilidad	Las interfaces deben ser intuitivas para el usuario y que permitan llevar a cabo los procesos fácilmente.
RNF004	Prevención de ataques de presentación	El sistema debe tener mecanismos de detección de vida que prevengan ataques de presentación en el reconocimiento facial.

6.2.1.3. Identificación de actores.

Una vez analizados los requerimientos se procedió a hacer la identificación de actores que participarán en los distintos casos de uso del sistema con su respectivo código del requerimiento funcional al cual tiene acceso.

Tabla 12. Identificación de actores

Actor	Código RF	Caso de Uso
Administrador	RF001, RF002, RF003, RF004, RF005, RF006, RF007.	Administración de sistema de registro de asistencia.
Docente	RF008.	Registro de asistencia.
Estudiantes	RF008.	Registro de asistencia.

Fase de Prototipado

6.2.2. Tarea 2: Modelar la arquitectura del software empleando la notación UML haciendo uso de la herramienta LucidChart.

6.2.2.1. Diseño arquitectónico de la solución.

Para el desarrollo del prototipo de reconocimiento facial con prueba de vida para el registro de asistencias, se utiliza OpenERP actualmente conocido como Odoo, el estilo arquitectónico para la organización estructural que presenta este ERP es el denominado “en capas”, en la cual sus capas principales son la capa de presentación, la lógica de negocios y acceso a base de datos, tal y como se presenta en la figura 26 a continuación.

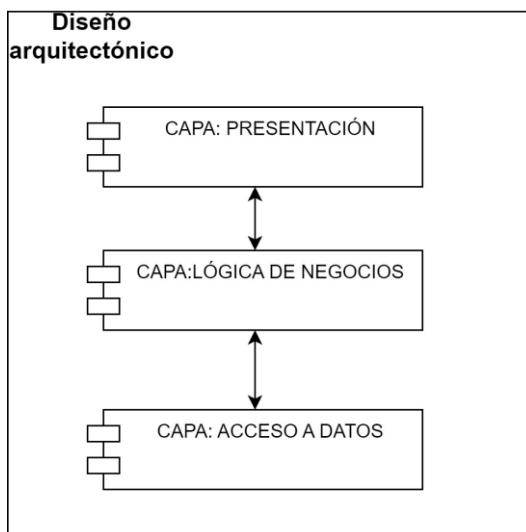


Figura 26. Estilo arquitectónico general de Odoo

En la capa de presentación se incluye la interfaz gráfica, conocida como frontEnd, siendo la parte visual que se le presenta al usuario por medio de un navegador web, en esta constan aspectos como: formularios, tablas, botones, así como aspectos de pre validación y validación

de datos a nivel de usuario, esta capa, al ser con la que interactúa el usuario, deberá ser lo más intuitiva posible. A su vez la capa de presentación se comunica únicamente con la capa de lógica de negocios y para su desarrollo se suele usar: XML, HTML, JavaScript y CSS.

En la capa de lógica de negocios, conocida como backEnd, está presente OpenERP Server, es aquí donde se implementa todo lo relacionado con el funcionamiento del sistema, en esta capa se define las reglas de negocio (comportamiento del sistema), manejo y procesado de datos, hace cálculos, validaciones y retorna resultados al cliente, a su vez se realiza post validaciones de información para lo cual interactúa con la capa de acceso a la base de datos.

La capa de acceso a datos, recibe la información en crudo que fue ingresada por el usuario y se encarga de procesarla y de entregar una respuesta para que sea interpretada en la capa de presentación. OpenERP maneja en esta capa la base de datos relacional PostgreSQL.

El diseño de la arquitectura de la solución de software se presenta en la figura 27.

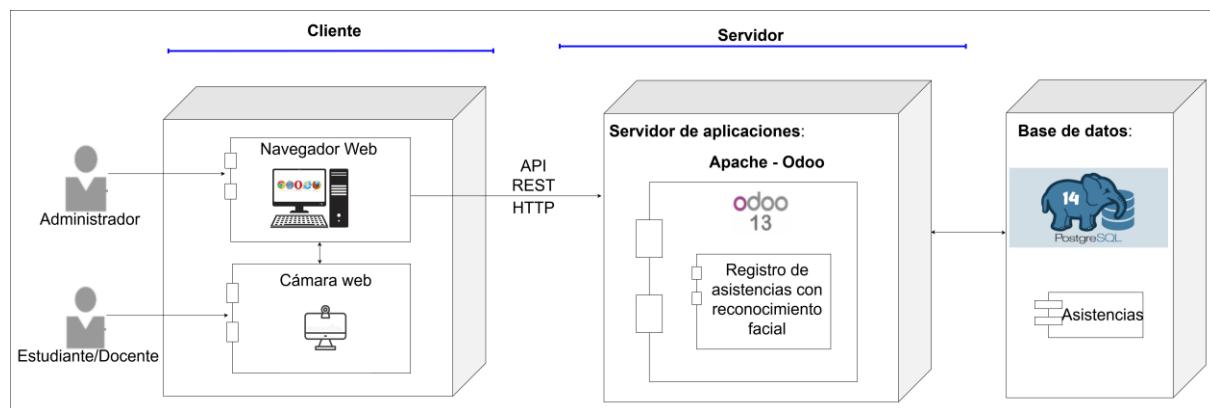


Figura 27. Estilo arquitectónico de despliegue

Dentro de la capa de presentación, se encuentra la interfaz de usuario a la cual se accede por medio de un navegador web (Chrome, Opera, Firefox, etc.) disponible en un computador, y a su vez se dispone de una cámara web que actuará como punto de captura para la detección de rostros, toda esta interfaz se comunica directamente con el servidor web que proporciona OpenERP, el mismo que interactúa con el framework de Odoo perteneciente a la capa de lógica de negocios, aquí se encuentra, el modelo, la vista, el controlador y el ORM que la lógica de negocios requiere, el modelo se encuentra en constante comunicación con el modelo de reconocimiento facial con detección de vida. El ORM (mapeo objeto-relacional) es quien se comunica directamente con la base de datos relacional PostgreSQL.

6.2.2.2. El modelo de Reconocimiento Facial

Como se indicó en la revisión de literatura, el reconocimiento facial requiere del cumplimiento de varias fases, en primera instancia, se dispone de una imagen de entrada e inicia la fase de detección de rostro, se procede a la normalización de la imagen del rostro detectado convirtiéndola a escala de grises y reduciendo su tamaño, se continúa con la extracción de características del rostro y se realiza la comparación con la base de datos. La culminación de las fases, dará como salida el resultado de la comparación, es decir, si se ha encontrado una coincidencia del rostro detectado en la imagen de entrada con algún rostro almacenado en la base de datos.

Las fases se figuran a continuación.

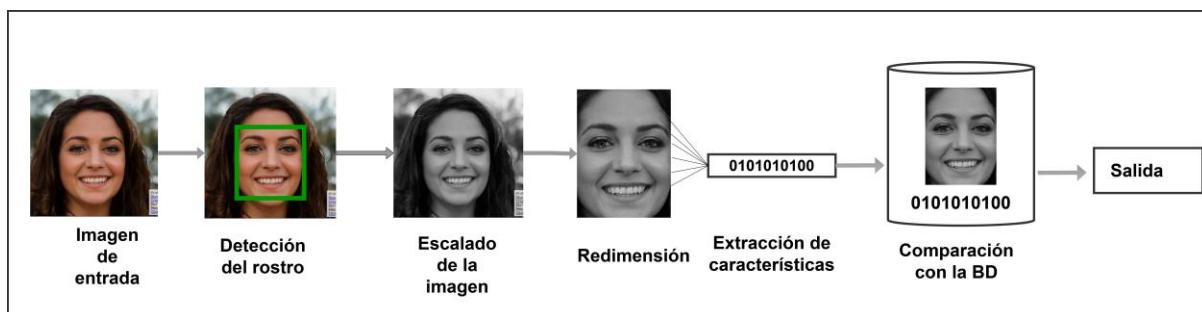


Figura 28. Fases del reconocimiento facial

6.2.2.3. Vista de escenarios.

Para representar la vista de posibles escenarios, se presenta el diagrama de casos de uso desarrollado en la especificación de requerimientos de software (Ver [Anexo 6](#)), en el cual se relaciona a los actores con las distintas funcionalidades o casos de uso a los que tienen acceso respectivamente dentro del sistema.

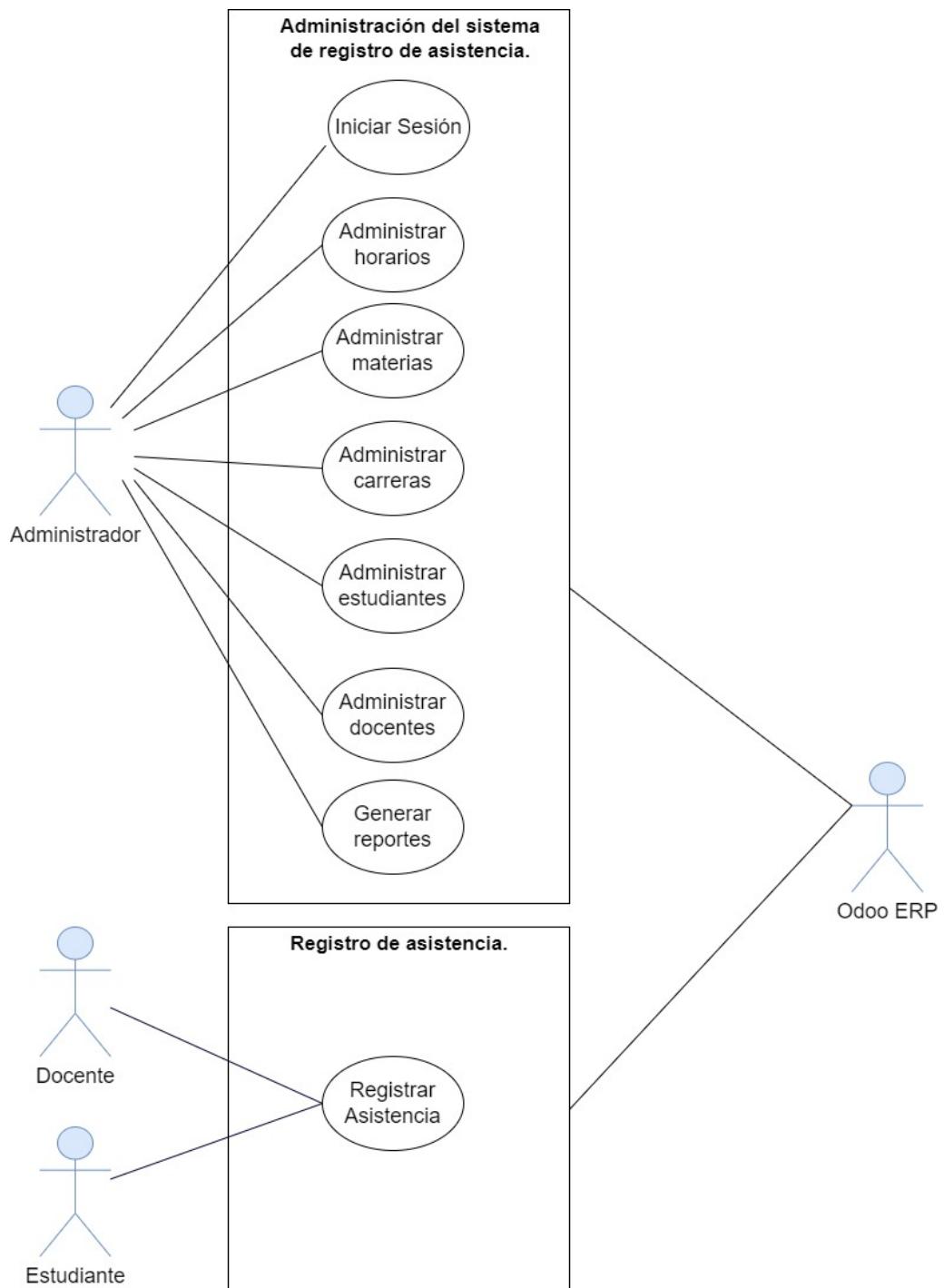


Figura 29. Vista de escenarios - Casos de uso

En el sistema de reconocimiento facial con detección de vida para el registro de asistencia, están presentes tres actores internos (representados a la izquierda) y un actor externo (representado a la derecha). Por un lado, el administrador, único usuario que tendrá acceso al sistema y podrá administrar a los estudiantes, docentes, horarios, carreras y materias que

hacen uso del Laboratorio de Software, así como también podrá generar reportes del registro de asistencia, adicionalmente se encuentra los actores Docente y Estudiante, los cuales podrán registrar su asistencia al mencionado laboratorio mediante reconocimiento facial con detección de vida. Por otro lado, está presente Odoo ERP, el cuál interactúa directamente con la administración del sistema, así como con el registro de asistencia.

6.2.2.4. Vista lógica.

En esta vista se puede observar la representación del modelo relacional del sistema, obteniendo un total de seis tablas, las cuales cuentan con campos específicos que permite el almacenamiento de información de forma adecuada.

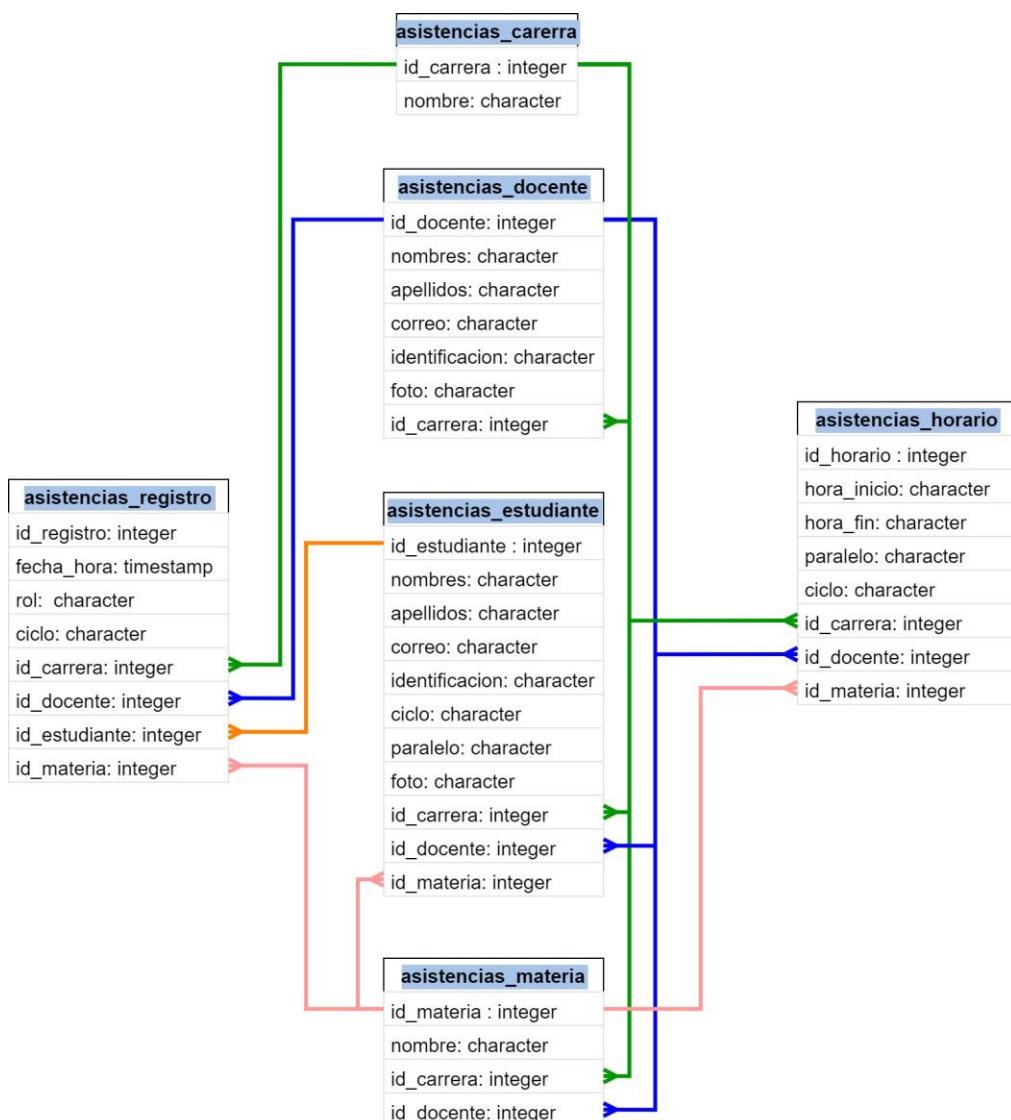


Figura 30. Modelo Relacional

6.2.2.5. Vista de procesos.

Esta vista se enfoca en describir el comportamiento que desarrolla el sistema en ejecución y sus procesos o actividades de una manera simple, para ello, se emplea diagramas de flujo que describen las actividades secuencialmente. A continuación, se presenta el diagrama de flujo del registro de asistencia, la totalidad de los diagramas se detallan en el [Anexo 7](#)

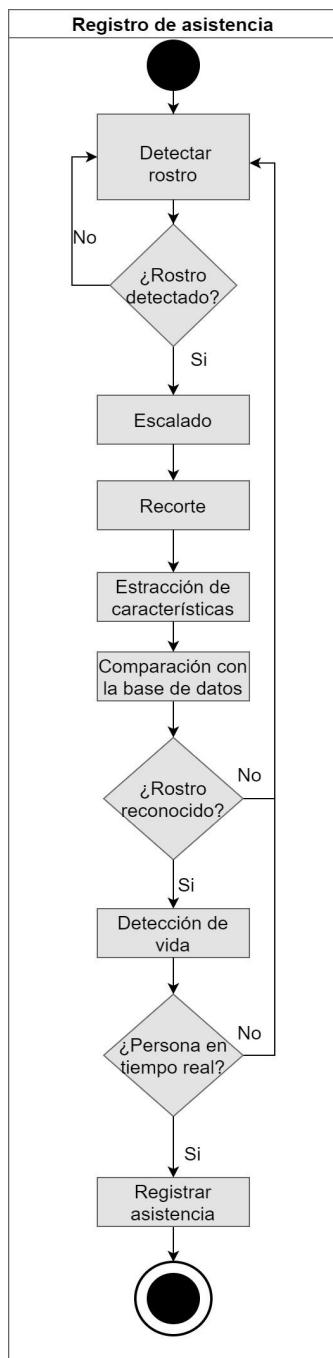


Figura 31. Diagrama de flujo: Registro de asistencia

El proceso de registro de asistencia con reconocimiento facial con detección de vida, inicializa al detectar un rostro presente en el punto de captura, si esto sucede, la imagen detectada es transformada a escala de grises, seguidamente se realiza el recorte de la imagen a tal punto que solo se obtenga el rostro presente en la imagen, excluyendo cualquier otro tipo de objeto, posterior a ello, se extrae las características de la imagen para compararlas con la base de datos, en caso de que exista alguna coincidencia, se prosigue con la detección de vida, si el modelo de detección de vida arroja un resultado correcto se realiza el respectivo registro de asistencia del usuario en cuestión. Si el sistema no ha reconocido o si no se ha detectado la vida del rostro, se volverá a esperar la detección de un nuevo rostro.

6.2.2.6. Arquitectura de la Red Neuronal Convolutacional.

La arquitectura de la red neuronal convolucional de tres dimensiones, empleada en el presente proyecto, necesaria para el desarrollo del modelo de detección de vida se ilustra a continuación en la figura 32.

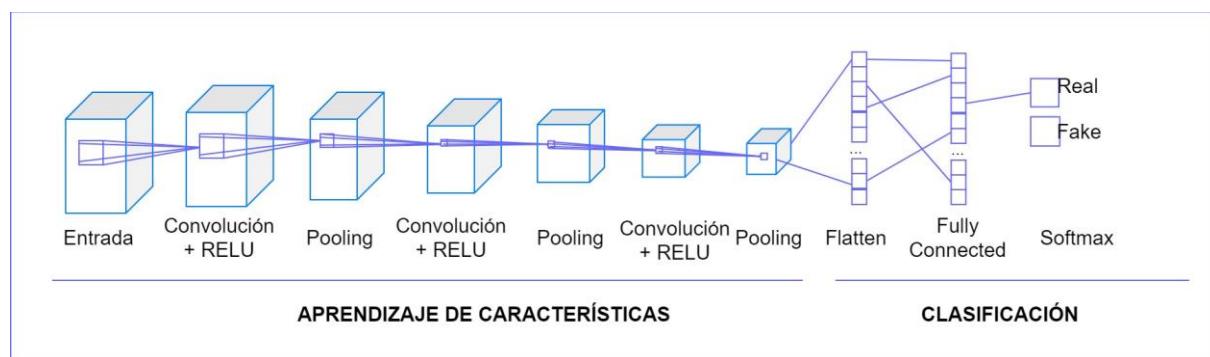


Figura 32. Arquitectura de 3D CNN para la detección de vida

La arquitectura de la 3D CNN cuenta con tres capas de convolución, tres capas de pooling, una capa de flatten y una capa totalmente conectada para la respectiva clasificación.

6.2.3. Tarea 3: Desarrollar el prototipo de software para el registro de asistencias mediante reconocimiento facial con detección de vida usando el Framework Odoo ERP con el gestor de Base de Datos PostgreSQL.

En el presente apartado se detalla la fase de desarrollo del sistema de registro de asistencias mediante reconocimiento facial con detección de vida, se desarrolló haciendo uso de Odoo ERP versión 13 y el lenguaje de programación Python versión 3.7, además, para la codificación del módulo se utilizó el IDE de programación PyCharm que permitió la edición,

depuración del código y la configuración de la ejecución del servicio de Odoo, ejecutado en un entorno virtual para controlar las versiones de librerías utilizadas, también se utilizó la plataforma GitHub para el alojamiento del código y con ello llevar el control de versiones.

6.2.3.1. Estructura del módulo de software.

Los módulos por defecto que forman parte del ERP se encuentran en la ruta /Odoo 13.0/server/odoo/adodons/, de igual forma, dentro de este directorio se encuentra almacenado el módulo para el registro de asistencia que se desarrolló para el presente TT, se lo ha denominado asistencias, su estructura se presenta en la Figura 33.

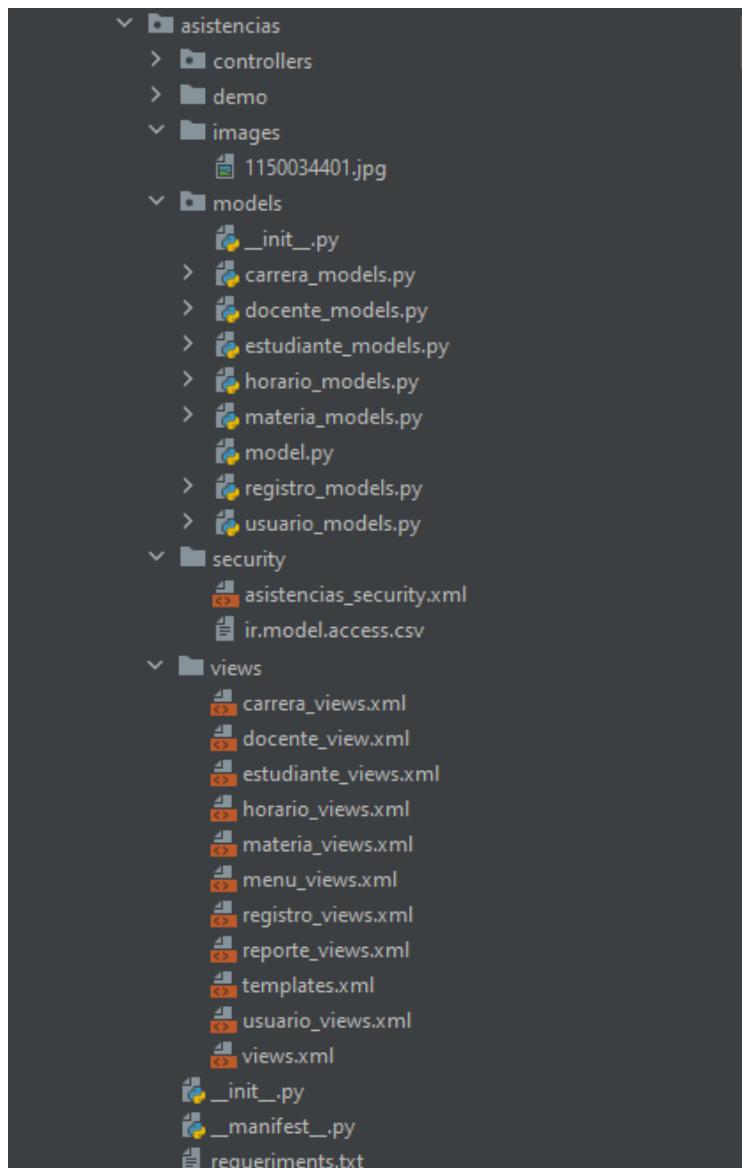


Figura 33. Estructura del sistema de registro de asistencias.

El sistema está estructurado de la siguiente manera:

- **asistencias:** Carpeta que contiene la estructura del módulo de software de reconocimiento facial con detección de vida para el registro de asistencias, el ERP reconoce este nombre como el identificador del módulo.
- **controllers:** El directorio almacena archivos de Python que establece cierta lógica de negocios, en este caso, no ha sido necesaria su utilización ya que las funciones han sido definidas en los archivos del directorio models.
- **images:** Se trata del directorio que almacena las fotografías que se descargan al momento de registrar un estudiante o docente en el sistema, se considera como la base de datos de fotografías, estas son necesarias para la comparación en el momento en que el reconocimiento facial se lleve a cabo.
- **models:** Carpeta que contiene los archivos escritos en Python, los cuales definen los objetos y son almacenados en la base de datos PostgreSQL. El mapeo de la base de datos es gestionado automáticamente por Odoo, y el mecanismo responsable de esto es el modelo objeto relacional (ORM). Dentro de estos archivos es posible definir funciones que serán invocados desde la interfaz gráfica de usuario.
- **security:** Carpeta que contiene archivos donde se declaran los grupos de usuarios, los permisos y las reglas de seguridad del módulo de software.
- **Views:** Carpeta que contiene los archivos .xml con la definición de las vistas, menuitems, actions, etc, es decir contiene la interfaz de usuario del módulo.
- **init.py:** Archivo que contiene la referencia de los modelos y controladores.
- **manifest.py:** Archivo de suma importancia ya que contiene las configuraciones necesarias para el módulo y decide que ficheros ejecutar y cuando.
- **requeriments.txt:** Archivo que almacena las librerías y dependencias necesarias para la compilación del módulo asistencias.

Desarrollo del módulo de software.

Dado que el marco de trabajo Design Thinking, el cual se está siguiendo en el presente TT, establece que los prototipos que se desarrollen durante esta fase, no se entenderán como productos finales, si no que serán una representación de las posibles soluciones definidas en la etapa anterior, además de no tener una limitante en la creación de prototipos, se realizó algunos experimentos con el modelo de reconocimiento facial y con el de detección de vida, antes de ello, se desarrolló el código necesario para la administración del módulo, con las funcionalidades de crear, modificar, listar y eliminar docentes, estudiantes, horarios, carreras

y materias, dando solución a los requerimientos RF002, RF003, RF004, RF005, RF006. Posteriormente se experimentó con algunas soluciones para el reconocimiento facial, una vez que se estableció el modelo más adecuado se prosiguió con la experimentación del modelo de detección de vida. El proceso se detalla a continuación.

6.2.3.2. Sistema de administración del registro de asistencias.

Cuando se crea un nuevo módulo en Odoo, existe la posibilidad de heredar una plantilla por defecto denominada “base”, en la cual se encuentran definidos los métodos de inicio de sesión, los métodos para crear, modificar, listar y borrar objetos, por lo que no fue necesaria la codificación de estas funcionalidades, se tuvo que crear los objetos del módulo con sus atributos y la codificación de las respectivas vistas.

A modo de ejemplo a continuación se presenta el código de creación del modelo del objeto Estudiante, así como sus funcionalidades en la interfaz gráfica.

```
class AsistenciasEstudiante(models.Model):
    _name = 'asistencias.estudiante'
    _rec_name = 'identificacion'
    nombres = fields.Char(string="Nombres", required=True)
    apellidos = fields.Char(string="Apellidos", required=True)
    identificacion = fields.Char(string="Identificación", required=True, size=10)
    _sql_constraints = [
        ('def_identification_unique', 'unique(identificacion)', 'Identificación ya registrada en el sistema!')]
    foto = fields.Image(required=True)
    correo = fields.Char(string="Dirección de Correo")
    ciclo = fields.Selection([
        ('1', 'Primerº'), ('2', 'Segundoº'), ('3', 'Terceroº'), ('4', 'Cuartoº'), ('5', 'Quintoº'), ('6', 'Sextoº'),
        ('7', 'Séptimoº'), ('8', 'Octavoº'), ('9', 'Novenoº'), ('10', 'Décimoº'), ('egre', 'Egresado')], string="Ciclo")
    carrera_name = fields.Many2one('asistencias.carrera', 'Carrera')
    materia_name = fields.Many2one('asistencias.materia', 'Materia')
    docente_name = fields.Many2one('asistencias.docente', 'Docente')
    registro_id = fields.One2many('asistencias.registro', 'estudiante_name')
```

Figura 34. Modelo Estudiante.

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Usuarios Administrator (tesis) ▾

Estudiante / Nuevo

Identificación	0000000	Ciclo	Seleccionar
Nombres	ej. Juan Luis	Paralelo	Seleccionar
Apellidos	ej. Perez	Carrera	Seleccionar
Dirección de Correo	Juan@tudominio.com	Materia	Seleccionar
Docente			Seleccionar
Foto			

Figura 35. Crear Estudiante

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Usuarios Administrator (tesis) ▾

Estudiante / 1150034401

39 / 39

Identificación	1150034401	Ciclo	10
Nombres	Dayanna	Paralelo	A
Apellidos	Alvarado	Carrera	Ingeniería en sistemas
Dirección de Correo	dayannaalvarado99@unl.edu.ec	Materia	Compiladores
Docente			Oscar Cumbicus
Foto			

Figura 36. Editar Estudiante

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Usuarios

Estudiante

<input type="checkbox"/>	Identific...	Nombres	Apellidos...	Ciclo	Paralel...	Dirección d...	Carrera	Materia	Docente
<input type="checkbox"/>	1900692...	Jennifer	Jazmín	9		jennifer.jara...	Ingeniería en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1106067...	Jhandry Ramiro	Lapo	9	A	jhandry.lapo...	Ingeniería en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1104414...	Luis Javier	Paz	9	A	luis.j.paz@u...	Ingeniería en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1105116...	Geovanny Samael	Poma	9	A	geovanny.po...	Ingeniería en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1150256...	Luis Homero	Bravo	9	A	luis.h.bravo...	Ingeniería en sistemas	Compiladores	Oscar Cumbicus

Figura 37. Listar Estudiante

The screenshot shows a web-based application for managing student records. At the top, there's a navigation bar with links for Estudiantes, Docentes, Carreras, Materias, Horarios, Registro de Asistencia, and Usuarios. There are also icons for notifications and user profile.

The main content area displays a student record for "Estudiante / 1150034401". The student's name is Dayanna Alvarado, and her email is dayanna.alvarado99@unl.edu.ec. The record includes fields for Identificación (1150034401), Nombres (Dayanna), Apellidos (Alvarado), Dirección de Correo (dayanna.alvarado99@unl.edu.ec), Paralelo (Paralelo), Carrera (Ingeniería en sistemas), Materia (Compiladores), and Docente (Oscar Cumbicus).

A "Foto" (Photo) field contains a small thumbnail of a smiling woman. To the right of the photo, a context menu is open with the option "Suprimir" (Delete) highlighted.

Figura 38. Eliminar Estudiante

Cada una de estas funcionalidades están presentes en Estudiantes, Docentes, Carreras, Materias y Horario. (Ver [anexo 8](#))

6.2.3.3. Modelo de reconocimiento facial.

Una vez que se desarrolló el módulo de administración, se prosigue con la construcción del modelo de reconocimiento facial.

Para resolver las fases que requiere el reconocimiento facial, en el presente prototipo, se experimentó, por un lado, con los métodos EigenFace, FisherFace, LBPHFace y por otro lado con el método Face-Recognition, el detalle de cada uno se presenta a continuación.

Reconocimiento facial con los métodos EigenFace, FisherFace y LBPHFace

En primera instancia se realizó el reconocimiento facial aplicando los métodos EigenFace, Fisherface, y LBPHFace, esto con la finalidad de experimentar con cada uno de ellos y si alguno se adaptara a los requerimientos, tomarlo como solución.

Las tecnologías empleadas en las fases de reconocimiento facial se ilustran a continuación.

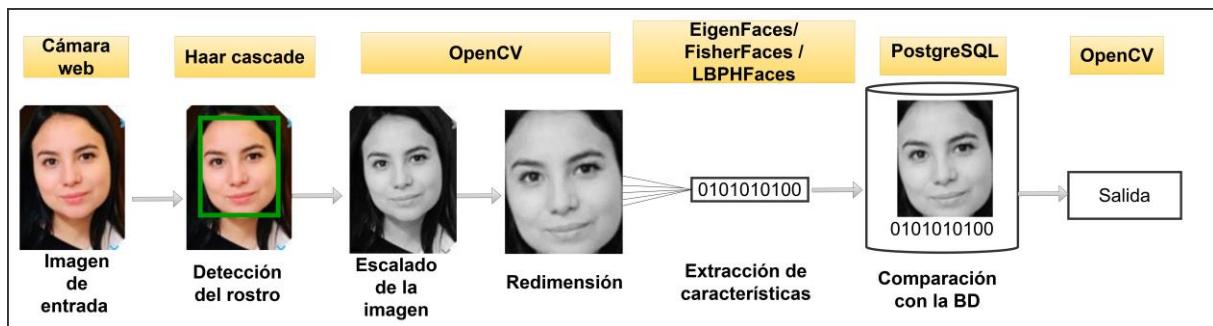


Figura 39. Tecnologías aplicadas en las fases de reconocimiento facial con los modelos EigenFaces, FisherFaces y LBPHFaces.

Para dar solución a la primera fase del reconocimiento facial la cual es detectar un rostro en una imagen proporcionada, se empleó los clasificadores pre-entrenados para cara, disponibles dentro de la librería OpenCV, se trata de clasificadores en cascada basados en funciones de Haar los cuales son 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 [45]”.

La segunda y tercera fase, son parte de la normalización de la imagen, para ello se redimensiona y se cambia a escala de grises la imagen de entrada.

En la fase de extracción de características se experimenta con EigenFace, FisherFace y LBPHFace, los cuales son métodos pre-entrenados con redes neuronales convolucionales para la extracción de características, estas características son representadas en forma de códigos únicos. Cada uno de estos métodos extraen las características de distinta forma, por lo que los resultados de la aplicación individual diferirán entre sí. Al poner en práctica estos métodos en el reconocimiento facial, permitirán obtener los códigos faciales de las fotografías proporcionadas de las personas que se intentará reconocer, para llevar a cabo el reconocimiento facial como tal, es necesario contar previamente con el conjunto de códigos faciales de las personas que se deseé reconocer, por tanto al momento de proporcionar una imagen como entrada, luego de las fases de detección del rostro y de normalización de la imagen, se extraerán las características y se comparará con los códigos faciales previamente obtenidos, en caso de encontrar una coincidencia se clasificará a la imagen de entrada como una persona reconocida.

Al aplicar estos métodos, para que el reconocimiento facial sea efectivo, es necesario obtener el conjunto de códigos faciales de una base de datos que contenga al menos 300 fotogramas de cada individuo que se deseé reconocer, estos fotogramas deben tener diferentes

expresiones faciales y si fuera posible cambios de luminosidad, los métodos extraerán las características de cada una de las fotografías proporcionadas, entre más se acerquen los valores de los códigos, se asumirá que se trata de la misma persona. El conjunto de códigos faciales se lo denominará modelo.

Para poner en práctica los métodos EigenFace, FisherFace y LBPHFace, en primer lugar, se escribió el código que permita capturar los 300 fotogramas necesarios de cada persona a reconocer, para ello, se realizó un algoritmo que permita inicializar la cámara web del computador y se aplicó clasificadores pre-entrenados para la detección de rostro, en caso de ser positiva la detección, se redimensionará la imagen y se cambiará a escala de grises, seguidamente se capturará 300 fotogramas del rostro detectado y se los almacenarán en un directorio del Sistema Operativo con el nombre proporcionado del individuo, es necesario que el usuario frente al punto de captura realice algunos movimientos y variados gestos para lograr tener una base de datos diversa.

El código se presenta a continuación:

```
1 import cv2
2 import os
3 import imutils
4
5 persoName = "DayannaAlvarado"
6 dataPath = "C:/Users/Usuario/Desktop/PRUEBA FACE/Data"
7 personPath = dataPath + "/" + persoName
8 print('Carpeta creada: ',personPath)
9 os.makedirs(personPath)
10
11 cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
12
13 faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
14 count = 0
15 while True:
16     ret, frame = cap.read()
17     if ret == False: break
18     frame = imutils.resize(frame, width=640)
19     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
20     auxFrame = frame.copy()
21     faces = faceClassif.detectMultiScale(gray,1.3,5)
22     for (x,y,w,h) in faces:
23         cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
24         rostro = auxFrame[y:y+h,x:x+w]
25         rostro = cv2.resize(rostro,(150,150),interpolation=cv2.INTER_CUBIC)
26         cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count),rostro)
27         count = count + 1
28     cv2.imshow('frame',frame)
29     k = cv2.waitKey(1)
30     if k == 27 or count >= 300:
31         break
32 cap.release()
33 cv2.destroyAllWindows()
```

Figura 40. Recopilación de 300 fotogramas del rostro detectado

Descripción del código:

personName: Es el nombre del individuo del cual se tomarán los 300 fotogramas para la base de datos, este nombre tendrá el directorio y es el que se presentará en el momento del reconocimiento facial.

dataPath: La ruta donde se crearán los directorios que contendrán los rostros de los individuos.

cap: Inicializa la cámara web del computador.

faceClassif: Variable que invoca al modelo preentrenado CascadeClassifier de la librería OpenCV, modelo necesario para la detección del rostro.

El bucle While: Verifica que se inicialice la cámara, si esto sucede se presentará un frame, se detectará el rostro del individuo y se lo encerrará en un rectángulo, se redimensiona la imagen y se cambia a escala de grises, leerá 300 fotogramas y los irá almacenando dentro de la ruta correspondiente. Una vez se culminen los 300 fotogramas, el frame se cerrará.

Al ejecutar el código se visualiza como en la figura 41 a continuación:

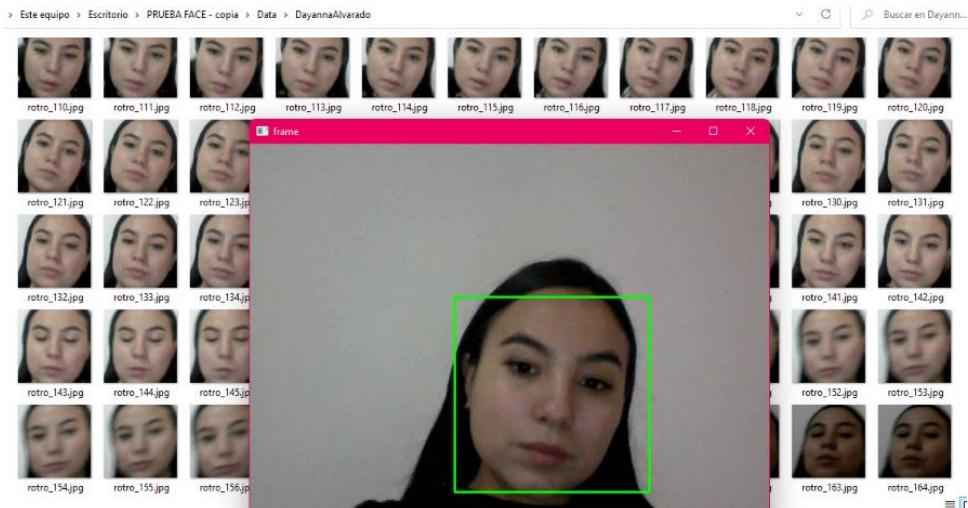


Figura 41. Captura de 300 fotogramas.

En la figura 41 se visualiza el frame con la detección del rostro presentado en el punto de captura y la recopilación de los 300 fotogramas, mismos que serán guardados en el directorio, para este ejemplo con el nombre “Dayanna Alvarado”. Cabe resaltar que este proceso se debe realizar por cada individuo que hará uso del sistema. En esta ocasión, a manera de experimentación, se tiene una base de datos de cinco personas diferentes, cada una con 300 fotogramas, los directorios creados se visualizan en la figura 42.

Este equipo > Escritorio > PRUEBA FACE - copia > Data >	
Nombre	Tipo
Carolina Martinez	Carpeta de archivos
Cinthya Poma	Carpeta de archivos
Dayanna Alvarado	Carpeta de archivos
Geovanny Alvarado	Carpeta de archivos
Itaty Poma	Carpeta de archivos

Figura 42. Base de datos para el entrenamiento del modelo de Reconocimiento Facial.

Una vez que se tiene lista la base de datos con los rostros de los individuos, se procede a entrenar el modelo, es decir, se van a extraer las características de cada fotograma de cada individuo y se armará un archivo (modelo) con las características faciales de cada persona. Esta acción se realiza individualmente para los métodos EigenFace, FisherFace y LBPHFace, los cuales están disponibles dentro de la librería OpenCV.

A manera de ejemplo, a continuación, se presenta el código para entrenar el modelo con el método LBPHFace.

```

1 import cv2
2 import os
3 import numpy as np
4 dataPath = "C:/Users/Usuario/Desktop/PRUEBA FACE/Data"
5 peopleList= os.listdir(dataPath)
6 print("Lista de personas: ", peopleList)
7 labels = []
8 facesData = []
9 label = 0
10
11 for nameDir in peopleList:
12     personPath = dataPath + '/' + nameDir
13     print('Leyendo las imágenes')
14     for fileName in os.listdir(personPath):
15         print('Rostros: ', nameDir + '/' + fileName)
16         labels.append(label)
17         facesData.append(cv2.imread(personPath+'/'+fileName,0))
18     label = label + 1
19
20 # Métodos para entrenar el reconocedor
21 #face_recognizer = cv2.face.EigenFaceRecognizer_create()
22 #face_recognizer = cv2.face.FisherFaceRecognizer_create()
23 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
24
25 # Entrenando el reconocedor de rostros
26 print("Entrenando...")
27 face_recognizer.train(facesData, np.array(labels))
28
29 # Almacenando el modelo obtenido
30 #face_recognizer.write('modeloEigenFace.xml')
31 #face_recognizer.write('modeloFisherFace.xml')
32 face_recognizer.write('modeloLBPHFace.xml')
33 print("Modelo almacenado...")

```

Figura 43. Entrenamiento del modelo de reconocimiento LBPHFace.

Las líneas de código de la figura 43, permiten leer cada fotograma de la base de datos construida anteriormente, extrae las características de los rostros y construye un archivo .xml, a este archivo se accederá durante la ejecución del reconocedor facial.

Finalmente se realiza el código para el reconocimiento facial, se incluyen comandos para inicializar la cámara web, detectar un rostro en el punto de captura y de así ser, invoca al modelo entrenado anteriormente y realiza la comparación de los códigos faciales, si existe una coincidencia, aparecerá el nombre de la persona reconocida, caso contrario aparecerá el mensaje “Desconocido”.

Cabe mencionar que cada modelo toma un umbral diferente de comparación, la diferenciación del código facial obtenido de la imagen de entrada con la coincidencia encontrada en el conjunto de códigos faciales debe ser menor al umbral establecido, de ser así el rostro se clasificará como reconocido. Es necesario ir experimento con el umbral según convenga. A continuación, se presenta el reconocimiento facial con los modelos entrenados.

- **Prototipo 1: Modelo EigenFace**

Al ejecutar el reconocedor facial que invoca al modelo EigenFace se obtiene el siguiente resultado.

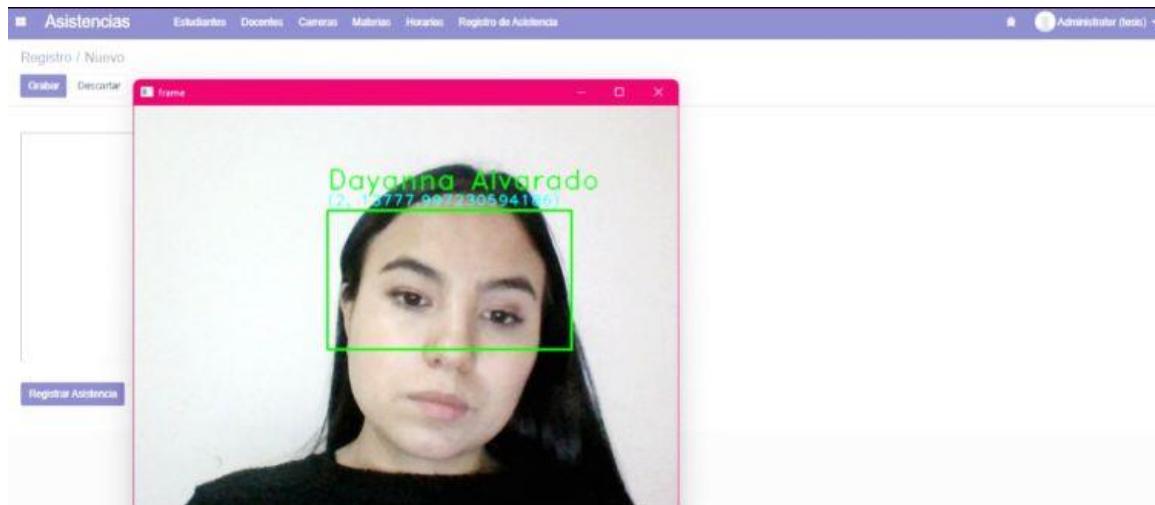


Figura 44. Reconocimiento Facial con el modelo EigenFaces

Como se observa en la figura 44, efectivamente se realizó el reconocimiento facial y el nombre presentado es el correcto, sin embargo, en ocasiones, no reconoce y presenta el mensaje “Desconocido”, para llegar a este resultado se estableció un umbral de 15000. El umbral es específico para cada persona que el modelo intenta reconocer, por tanto, si una persona

distinta a “Dayanna Alvarado” experimenta el reconocimiento facial con este modelo, es necesario modificar el umbral, es por ello que, para obtener un valor general, se debe calcular un umbral promedio de todos los usuarios del sistema.

Es importante destacar que en el entrenamiento (extracción de características de las fotografías de la base de datos) este modelo tardó alrededor de 7 minutos, tomando en cuenta que se tiene una base de datos de cinco personas con 300 fotogramas cada una.

- **Prototipo 2: Modelo FisherFace**

Al ejecutar el reconocedor facial que invoca al modelo FisherFace se obtiene el siguiente resultado.

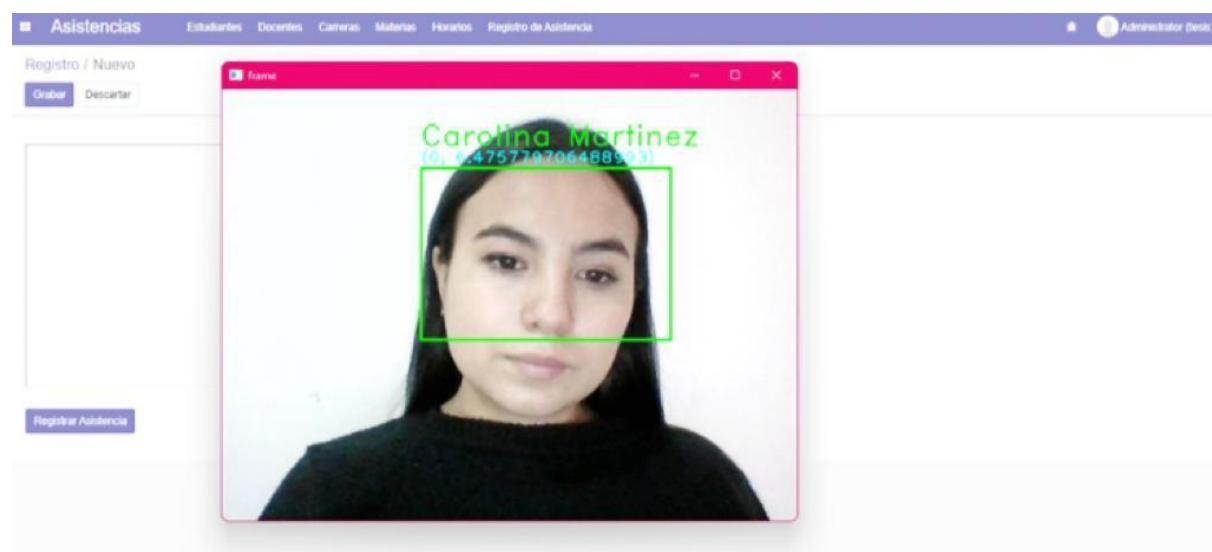


Figura 45 Reconocimiento Facial con FisherFaces

Como se observa en la figura 45, se realizó el reconocimiento facial y el nombre presentado es incorrecto, el rostro presentado en el punto de captura pertenece a Dayanna Alvarado, es decir, existe un error de reconocimiento, en este caso se estableció un umbral de 500. En la experimentación, si se establece un umbral mayor a 500, este modelo no reconoce el rostro y coloca el mensaje “Desconocido”, además, al experimentar con el rostro de otra persona guardada en la base de datos, el modelo se equivoca en el resultado intercambiando los nombres erróneamente.

Este modelo al igual que EigenFace, para tener un umbral general requiere el promedio de umbral de todos los usuarios del sistema, así mismo, tardó alrededor de 7 minutos en entrenarse teniendo la base de datos de cinco personas con 300 fotogramas cada uno.

- **Prototipo 3: Modelo LBPHFace**

Al ejecutar el reconocedor facial que invoca al modelo LBPHFace se obtiene el siguiente resultado.

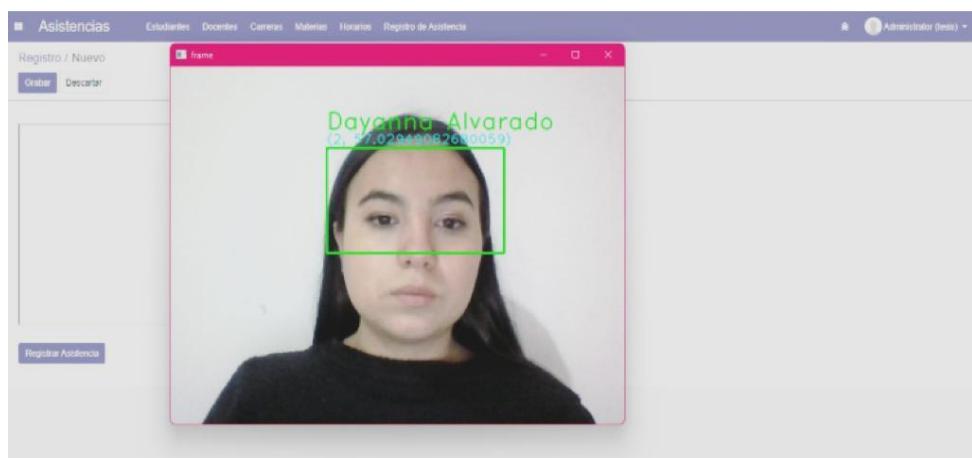


Figura 46. Reconocimiento Facial con LBPHFace

Como se observa en la figura 46, efectivamente se realizó el reconocimiento facial y el nombre presentado es el correcto, para llegar a este resultado se estableció un umbral de 94. Al igual que los modelos EigenFace y FisherFace, este modelo requiere establecer un umbral específico para cada persona y para obtener un valor general, se debe calcular un umbral promedio de todos los usuarios del sistema.

A diferencia de los modelos anteriores LBPHFace tiene una exactitud de reconocimiento mayor, además el tiempo de entrenamiento de este modelo es de alrededor de 2 minutos, tomando en cuenta que se mantiene la base de datos de cinco individuos con 300 fotogramas cada uno.

Conclusión de la experimentación con los modelos EigenFace, FisherFace y LBPHFace

Como conclusión de la experimentación con los modelos EigenFace, FisherFace y LBPHFace, se obtiene que, estos requieren un costo computacional de entrenamiento alto, tardaron de 2 a 7 minutos con una base de datos que almacena 1500 fotogramas de cinco personas distintas, además, requieren alta experimentación para establecer un umbral general que se adapte a todos los usuarios del sistema. Por tanto, tomando en cuenta que el modelo de reconocimiento facial se lo requiere para el registro de asistencia de los estudiantes de CIS/C UNL, carrera que contiene 10 paralelos, cada uno con más de 15 estudiantes, la recopilación de los fotogramas, seguido del tardado entrenamiento, adicional a la baja

exactitud de reconocimiento y finalmente el exhaustivo cálculo del promedio del umbral, son aspectos que hacen que ninguno de estos modelos sea elegible.

- **Prototipo 4: Reconocimiento facial con el modelo Face-Recognition.**

A consecuencia de la no elección de ninguno de los modelos anteriores, se procede a experimentar con Face-Recognition, se trata de una librería de código abierto desarrollada en lenguaje Python que a su vez implementa dlib, la librería de reconocimiento facial más simple del mundo que incluye una red neuronal de tipo convolucional pre-entrenada que permite detectar caras y rasgos faciales con una efectividad del 99.38% [14].

Las tecnologías utilizadas en cada una de las fases del reconocimiento facial se ilustran a continuación.

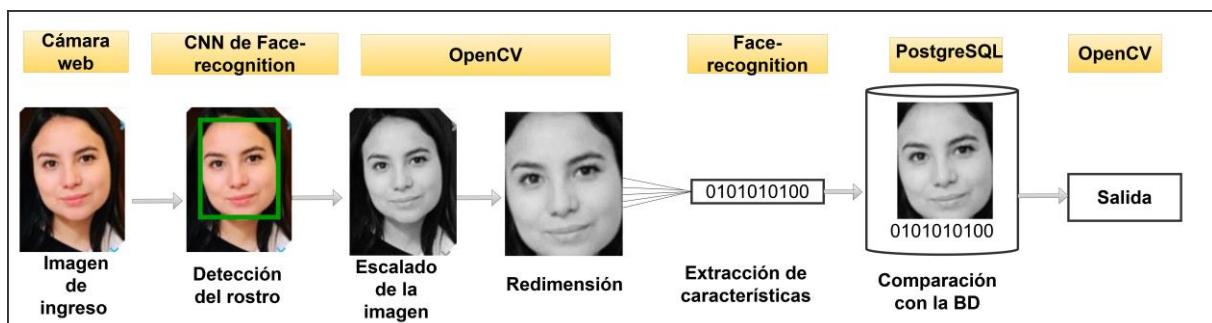


Figura 47. Tecnologías aplicadas en las fases de reconocimiento facial con el modelo Face-Recognition

Para emplear el modelo de reconocimiento facial con Face-Recognition, es necesario tener una base de datos con los rostros de las personas que se desea reconocer, con una fotografía de cada individuo es suficiente. Para la presente experimentación se tiene una base de datos con cinco fotografías almacenadas dentro de un directorio, cada una con el nombre de la persona correspondiente, tal como se presenta en la figura 48.

Este equipo > Escritorio > PRUEBA FACE - copia (2) > Data > faces



Figura 48. Base de datos para Face-Recognition

A continuación, se presenta el código desarrollado.

```
1 import face_recognition
2 import cv2
3 import numpy as np
4 import os
5 import glob
6
7 faces_encodings = []
8 faces_names = []
9 cur_direc = os.getcwd()
10 path = os.path.join(cur_direc, 'Data/faces/')
11 list_of_files = [f for f in glob.glob(path+'*.jpg')]
12 number_files = len(list_of_files)
13 names = list_of_files.copy()
14 for i in range(number_files):
15     globals()['image_{}'.format(i)] = face_recognition.load_image_file(list_of_files[i])
16     globals()['image_encoding_{}'.format(i)] = face_recognition.face_encodings(globals()['image_{}'.format(i)])[0]
17     faces_encodings.append(globals()['image_encoding_{}'.format(i)])
18 # Create array of known names
19 names[i] = names[i].replace(cur_direc, "")
20 faces_names.append(names[i])
21 #print(faces_names)
22 #Reconocimiento
23 face_locations = []
24 face_encodings = []
25 face_names = []
26 process_this_frame = True
27 input_vid = []
28 video_capture = cv2.VideoCapture(0)
29 while True:
30     ret, frame = video_capture.read()
31     small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
32     rgb_small_frame = small_frame[:, :, ::-1]
33     if process_this_frame:
34         face_locations = face_recognition.face_locations(rgb_small_frame)
35         face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
36         face_names = []
37         for face_encoding in face_encodings:
38             matches = face_recognition.compare_faces(faces_encodings, face_encoding)
39             name = "Unknown"
40             face_distances = face_recognition.face_distance(faces_encodings, face_encoding)
41             best_match_index = np.argmin(face_distances)
42             #print(best_match_index)
43             if matches[best_match_index] == True:
44                 name = faces_names[best_match_index]
45             face_names.append(name)
46     process_this_frame = not process_this_frame
47 # Display the results
48 for (top, right, bottom, left), name in zip(face_locations, face_names):
49     top *= 4
50     right *= 4
51     bottom *= 4
52     left *= 4
53     # Draw a rectangle around the face
54     cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
55     # Input text label with a name below the face
56     cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
57     font = cv2.FONT_HERSHEY_DUPLEX
58     cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
59 # Display the resulting image
60 cv2.imshow('Video', frame)
61 # Hit 'q' on the keyboard to quit!
62 if cv2.waitKey(1) & 0xFF == ord('q'):
63     break
64 video_capture.release()
65 cv2.destroyAllWindows()
```

Figura 49. Modelo Face-Recognition

Descripción de código.

Inicialmente se importan los complementos necesarios para el modelo, a continuación, se recorre mediante una sentencia repetitiva las imágenes guardadas en la base de datos, de estas se detectan los rostros y se extraen los códigos faciales, estos códigos, así como los nombres de cada rostro se almacenan en variables de tipo array. A continuación se inicializa la cámara web del computador, se crea un frame que leerá el video en tiempo real de la persona de la cual se intenta hacer el reconocimiento facial, el frame se redimensiona y se lo cambia a escala de grises, seguidamente se detecta los rostros en el punto de captura y se almacena las coordenadas de cada rostro en una variable de tipo arreglo, además, por cada rostro detectado en el punto de captura se extrae la codificación facial, luego se compara esa codificación facial con las codificaciones faciales obtenidas de las fotos almacenadas en la base datos, en caso de ser iguales, arrojará el nombre de la persona reconocida, en caso de no coincidir, aparecerá el mensaje “Desconocido”.

La ejecución se presenta a continuación.

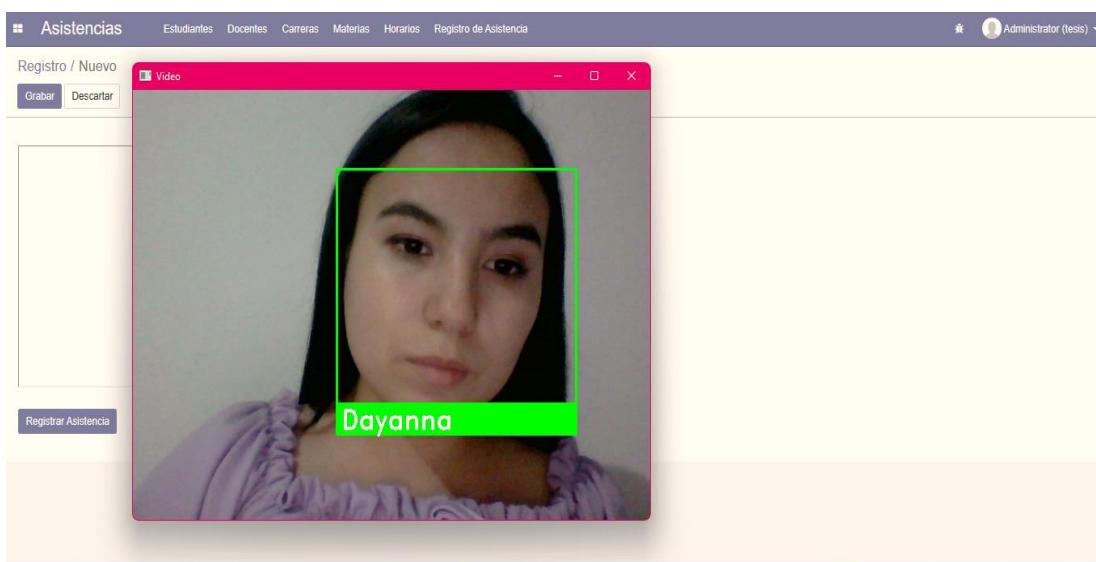


Figura 50. Reconocimiento facial con Face-recognition

Como se observa en la figura 50, se realizó el reconocimiento facial haciendo uso del modelo face-recognition, los resultados obtenidos son correctos, el rostro se corresponde con el nombre que se presenta. Se realizaron experimentaciones con las demás personas almacenadas en la base de datos y se corroboró el buen funcionamiento de este modelo. Cabe resaltar que face-recognition es un modelo pre-entrenado y no requiere un entrenamiento previo de las fotografías de la base de datos, si no que en cada ejecución calcula las codificaciones faciales tanto de las imágenes guardadas como de las de entrada,

realiza una comparación de la codificación facial de la imagen de entrada con cada una de las codificaciones de las imágenes guardadas, en caso de coincidir con alguna, presenta el nombre correspondiente, caso contrario presenta “Desconocido”. Este modelo ya incluye un detector facial, por tanto, solo ha sido necesaria su invocación.

Conclusión del modelo de reconocimiento facial con el modelo Face-Recognition.

A pesar de que las fotos almacenadas en la base de datos fueron capturadas en un escenario diferente al que se realizó el reconocimiento facial y que contenían diferente luminosidad y gestos, Face-recognition arrojó buenos resultados con un total de 95% de aciertos, además de tener la ventaja que solo requiere una imagen por usuario en la base de datos, su coste computacional es bajo, inmediatamente que detecta el rostro presenta si se trata de una persona reconocida o no.

Tabla de resumen de experimentación con los modelos de reconocimiento facial.

A continuación, en la tabla 13 se presenta la comparación del funcionamiento de los modelos de reconocimiento facial con los que se experimentó.

Tabla 13. Comparación de la experimentación con modelos de reconocimiento facial

	Eigen-Face	Fisher-Face	LBPH-Face	Face-Recognition
Número de fotografías por individuo para la base de datos.	300	300	300	1
Tiempo de entrenamiento.	7 minutos	7 minutos	2 minutos	0 minutos
Requiere establecer umbral de comparación.	Si requiere	Si requiere	Si requiere	No requiere
Incluye modelo de detección facial	No incluye	No incluye	No incluye	Si incluye
Precisión de reconocimiento.	Media	Baja	Media	Alta

Según la experimentación con los modelos Eigen-Face, Fisher-Face, LBPH-Face y Face-recognición, se obtiene que con los tres primeros modelos es necesario contar mínimo con 300 fotografías por individuo para la construcción de la base de datos, además, dado que es un gran número de fotografías, el entrenamiento toma tiempo considerable que va desde 2 a

7 minutos con un total de 1500 fotos; de igual manera, estos modelos emplean un umbral de comparación que varía según las características faciales del individuo, por tanto, para obtener un umbral general de toda la base de datos se requiere de mucha experimentación y alto coste computacional; en cuanto a la precisión de reconocimiento, el modelo Fisher-face se equivocó en su predicción, mientras que los demás modelos acertaron correctamente. Por otro lado, para el funcionamiento del modelo denominado face-recognition, es necesario únicamente una fotografía por individuo para la construcción de la base de datos, además, no requiere un nuevo entrenamiento previo de las fotografías almacenadas, si no que en cada ejecución calcula las codificaciones faciales tanto de las imágenes guardadas como de las de entrada, por tanto, su coste computacional es bajo; otra característica es que este modelo ya incluye un detector facial y solo ha sido necesaria su invocación, finalmente, su precisión de reconocimiento es alta y no requiere un umbral de comparación.

Tomando en cuenta la comparación anterior, se considera que el modelo preentrenado Face-recognition, tiene importantes ventajas con respecto a los demás modelos, es por ello que se elige este modelo como solución para el reconocimiento facial para el registro de asistencia en el laboratorio de software de CIS/C UNL.

6.2.3.4. Modelo de detección de vida.

Según la literatura, los modelos de reconocimiento facial, detectan y clasifican las imágenes de entrada como persona reconocida o no reconocida, en caso de que se presente una foto en el punto de captura en vez de un usuario en tiempo real, el reconocimiento facial igual cumplirá con su función de determinar si el rostro es reconocido o no. Por tanto, en la práctica, si se deja únicamente un registro de asistencia con reconocimiento facial, el sistema es vulnerable a que los usuarios puedan, mediante ataques de presentación, registrar asistencia de personas que no se encuentran en tiempo real en las instalaciones, es decir, un usuario pudiese registrar asistencia de otro usuario colocando una foto, un video o una máscara en el punto de captura. Es por ello, que es necesario elaborar un mecanismo capaz de detectar que la persona que se está registrando está presente en tiempo real frente al punto de captura.

Para realizar el modelo de detección de vida, se toma como base el modelo de reconocimiento facial elegido en la sección anterior, únicamente si el rostro es reconocido, comenzará el proceso de detección de vida.

En esta sección se experimenta con algunas ideas de detección de vida y se escogerá para el prototipo del presente TT la que mejor se adapte a los requerimientos establecidos. A continuación, se presenta el detalle de experimentación.

- **Prototipo 1: Detección de vida mediante contador de parpadeos.**

Inicialmente se realizó un modelo de detección de vida con el cual el usuario reconocido en el punto de captura del reconocimiento facial tiene que parpadear frente a la cámara las veces que se solicite en la pantalla, se desarrolló esta idea debido a que si se intenta vulnerar el sistema de reconocimiento facial con una fotografía de un usuario, el detector de vida actuará y no permitirá el registro de asistencia dado que no detectará el parpadeo en una fotografía, en caso de que se intente vulnerar el sistema con una reproducción de video, el detector de vida comprobará que se parpadeen únicamente las veces que se solicita por pantalla, en caso de no coincidir, no se registrará asistencia, el número de parpadeos que solicita el modelo detector de vida es aleatorio, por tanto, no se tendrá un video previsto para vulnerar el sistema.

Para el desarrollo del algoritmo se ha tomado como referencia al estudio *Detección de parpadeo en tiempo real usando puntos de referencia faciales* [46]. El documento propone un método para poder reconocer cuando los ojos han pasado de estado abierto, cerrado y nuevamente abierto, es decir, reconocer cuando ha habido un parpadeo, el método se basa en el uso de puntos de referencia faciales, específicamente los de los ojos.

Según la visión por computador, cada parte del rostro tienen puntos referenciales, conociendo estos puntos tanto del ojo izquierdo como el derecho, es posible calcular la distancia entre estos para identificar un parpadeo.

Para el desarrollo del algoritmo se toma en cuenta 6 puntos referenciales del ojo, que van en orden de las manecillas del reloj, tal como se muestra en la figura 51.

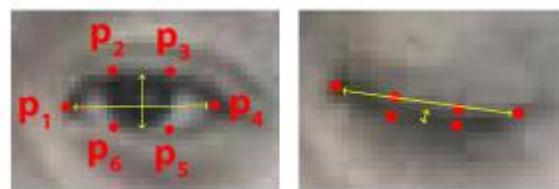


Figura 51. Puntos de referencia de los ojos [46].

El estudio propone la fórmula matemática descrita a continuación, para determinar la relación de aspecto del ojo.

$$\frac{(P2 - P6) + (P3 - P5)}{2} * (P1 - P4) \quad (6)$$

El cálculo se realiza tanto con las coordenadas del ojo izquierdo como del derecho, el promedio de la relación de aspecto del ojo izquierdo con el derecho se determinará como un umbral, además, es necesario definir un número de frames específicos que deben transcurrir mientras el ojo está cerrado, entonces, cuando se esté obteniendo por medio de cámara un video en tiempo real, se realizará constantemente estos cálculos, si el promedio de relación de aspecto de los ojos se encuentra por debajo del umbral, y si el número de frames que han transcurrido es mayor o igual al establecido, se contabilizará como un parpadeo.

Según los requerimientos del prototipo en desarrollo, se debe comprobar que el número de parpadeos sea igual al que se solicita en pantalla, si es así, se registra la asistencia.

La ejecución del código desarrollado se presenta a continuación en la figura 52.



Figura 52. Detección de vida mediante contador de parpadeos

Como se visualiza en la figura 52, primero se realiza el reconocimiento facial, seguidamente se detecta los puntos referenciales de los ojos y se solicita que se parpadee 6 veces, a manera de ejemplo se han parpadeado únicamente 5 veces, dado que aún no se ha alcanzado el valor solicitado, no se registra asistencia, en caso de cumplir con los parpadeos solicitados, aparecerá un mensaje de “Registrado” y desaparecerá el contador de parpadeos, tal y como se presenta en la figura 53.

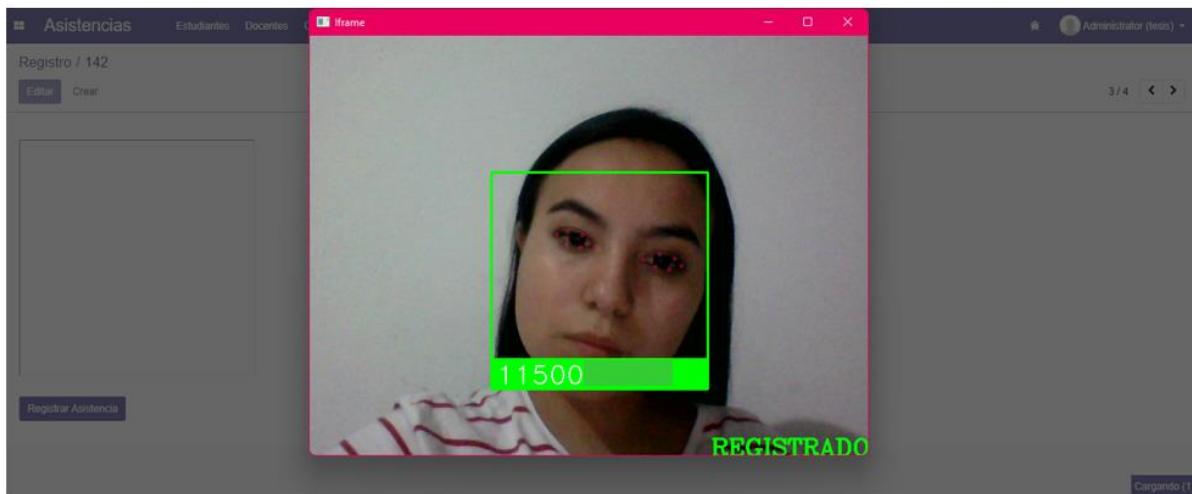


Figura 53. Registro de asistencia con reconocedor facial y contador de parpadeos.

En este caso se ha completado el número de parpadeos solicitados, por tanto, se ha llevado a cabo el registro de asistencia y se presenta un mensaje informativo anunciando que la asistencia del usuario ya ha sido registrada.

Para observar los experimentos realizados tanto con una foto como con una reproducción de video dirigirse a [Enlace de Github](#)

Conclusión:

Según la experimentación de este modelo, cuando se presenta una fotografía en el punto de captura, si existe movimiento continuo y cambios de luminosidad, el modelo en ocasiones equívocamente lo toma como un parpadeo, pero no se detecta la vida mientras no se haya complementado el número total de parpadeos solicitado, y como esto no sucede con una fotografía, no se registra la asistencia, sin embargo, cuando se presenta una reproducción de video frente al punto de captura, el modelo contabiliza los parpadeos realizados por el usuario presente en la grabación y cuando se complete el número de parpadeos solicitado, el modelo detecta a ese video como un usuario en tiempo real, a pesar de que el número de parpadeos que se solicita es randómico, este no sobrepasa el valor de 6, ya que, para el presente TT se requiere que el sistema de registro de asistencia sea rápido, por tanto, si se tiene una grabación de un usuario que realice 6 parpadeos y se lo presenta en el punto de captura y el sistema en el mejor de los casos solicite 6 parpadeos, este video será detectado como un usuario en tiempo real frente al punto de captura al trascurrir los 6 parpadeos solicitados, por consiguiente, se realiza el registro de asistencia.

Dados los resultados expuestos en líneas anteriores, el modelo contabilizador de parpadeos para detectar la vida en el reconocimiento facial, con la reproducción de un video, no es eficiente, el resultado arroja un falso positivo, es por ello, que no se lo toma como solución y se experimentará con otras técnicas.

- **Prototipo 2: Detección de vida aplicando redes neuronales convolucionales 3D.**

En este experimento, se utilizó una red convolucional 3D para el desarrollo del modelo de detección de vida. Este tipo de redes convolucionales a diferencia de las 2D, miden la profundidad de una entrada, es por ello que es útil este mecanismo de tal manera que se detecte la profundidad del rostro presentado en el punto de captura, dado que una fotografía es una imagen coplanar de 2 dimensiones, esta no presenta profundidad, así mismo, de una presentación de una reproducción de video, tampoco es posible medir su profundidad ya que la profundidad sería del dispositivo que reproduce el video y no del rostro presentado en él.

Lo que se intentó con este experimento es que, en caso de que una persona en tiempo real se presente frente al punto de captura, se podrá medir la profundidad del rostro y se aprobará la detección de vida.

El modelo de detección de vida, deberá predecir si se trata de una persona en tiempo real, o un ataque de presentación ya sea mediante la presentación de fotografías, reproducciones de video o máscaras frente al punto de captura, tal y como se ilustra a continuación.



Figura 54. Predicción del modelo de detección de vida con CNN 3D.

La arquitectura de la red neuronal convolucional empleada en el presente experimento se ilustra a continuación.

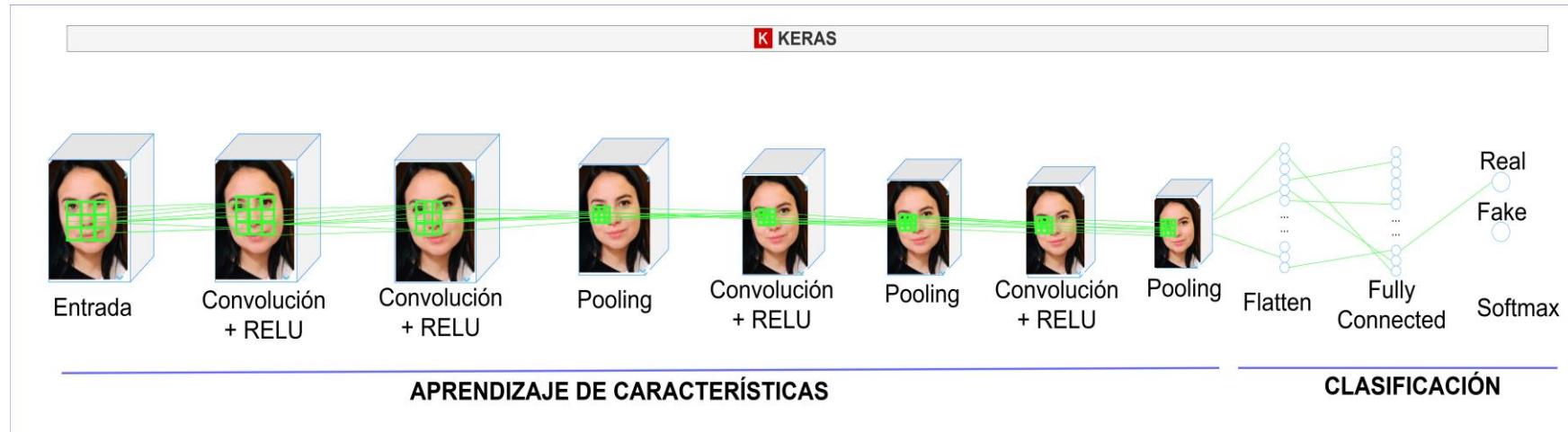


Figura 55. Arquitectura de la red neuronal convolucional 3D para la detección de vida

El kernel empleado se figura a continuación.

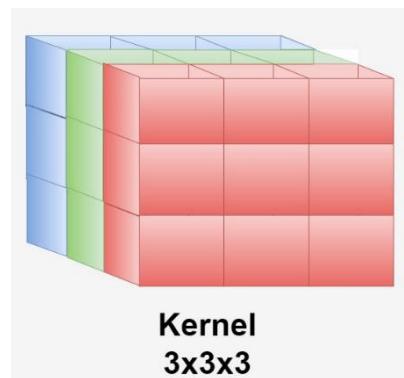


Figura 56. Kernel de una Red neuronal convolucional de 3 dimensiones.

El código desarrollado para la creación de la red neuronal convolucional de 3 dimensiones se presenta a continuación.

```
8  font = cv2.FONT_HERSHEY_DUPLEX
9  import keras
10 from keras.models import model_from_json
11 from keras.models import Sequential
12 from keras.layers import Dense, Dropout, Flatten
13 from keras.layers import Conv3D, MaxPooling3D
14 from keras import backend as K
15
16
17 model = Sequential()
18 model.add(Conv3D(32, kernel_size=(3, 3, 3), activation='relu', input_shape=(24, 100, 100, 1)))
19 model.add(Conv3D(64, (3, 3, 3), activation='relu'))
20 model.add(MaxPooling3D(pool_size=(2, 2, 2)))
21 model.add(Conv3D(64, (3, 3, 3), activation='relu'))
22 model.add(MaxPooling3D(pool_size=(2, 2, 2)))
23 model.add(Conv3D(64, (3, 3, 3), activation='relu'))
24 model.add(MaxPooling3D(pool_size=(2, 2, 2)))
25 model.add(Dropout(0.25))
26 model.add(Flatten())
27 model.add(Dense(128, activation='relu'))
28 model.add(Dropout(0.5))
29 model.add(Dense(2, activation='softmax'))
30
31
32 #model = get_liveness_model()
33 model.load_weights('model/model.h5')
34 print("Loaded model from disk")
35
```

Figura 57. Código de creación de Red Neuronal Convolucional 3D

Descripción del código e hiperparámetros:

En primera instancia se realiza la importación de Keras, una librería de código abierto que contiene herramientas que sirven para crear redes neuronales, a su vez se importan algunos de sus complementos.

Lo que procede es instanciar el sistema con la llamada a Sequential(), que crea una red neuronal vacía a la que habrá que agregar las capas correspondientes con la instrucción model.add(), este modelo de tipo secuencial permite que las capas de la red se agreguen en secuencias y es un modelo apropiado para una simple pila de capas donde cada una tiene exactamente un tensor de entrada y un tensor de salida.

A continuación, se agrega la primera capa convolucional de tres dimensiones y un tipo de activación ReLu, este tipo de activación considera únicamente valores positivos, en donde:

$$\begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (7)$$

Luego se agregan tres capas convolucionales también con función de activación ReLU y max-pooling (el mayor de los valores), siempre asignando al tamaño del kernel o filtro la dimisión 3*3*3 (alto, ancho, profundidad).

Adicional se agrega una capa Flatten la cual permite aplanar la salida de la capa convolucional, convirtiendo la matriz multidimensional en un vector de una sola dimensión, este último será la entrada de red neuronal.

Luego se agrega la capa Dense, la cual es la red neuronal que está completamente conectada y tiene una dimensionalidad de salida de 128 y una función de activación ReLu.

Se adiciona la capa Dropout, que permite que las neuronas seleccionadas al azar se ignoren durante el entrenamiento, al ignorarse, desempeñan un papel del 0% en la propagación hacia adelante y hacia atrás y, por lo tanto, el modelo tiene que usar las otras neuronas para continuar con el aprendizaje, se consideran como capas de regularización para que no exista memorización de entrenamiento.

Luego se añade una segunda capa Dense esta vez con activación softmax, este tipo de activación devuelve una predicción en forma de probabilidad (el sumatorio de todas las probabilidades de las salidas de 1) cuya fórmula matemática es:

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (8)$$

Finalmente se guarda los pesos del entrenamiento en un archivo, el cual va a ser el modelo de detección de vida y será llamado posterior al proceso de reconocimiento facial.

En resumen, lo que hacen la red neuronal convolucional de 3 dimensiones en este experimento, es aprender a reconocer el alto, ancho y la profundidad de una imagen. Como resultado se tiene un documento con los valores aprendidos de dimensionalidad, el cual es llamado después de hacer el reconocimiento facial.

Una vez que el rostro ha sido reconocido, empieza el detector de vida, para ello primero se toma como entrada la secuencia de fotogramas presente en el frame, se lo redimensiona y se lo cambia a escala de grises, seguidamente se detecta el alto, el ancho y la profundidad de la imagen entrante. Estos cálculos se los compara con los valores obtenidos en el modelo de detección de vida con redes neuronales convolucionales 3D, y se realiza una predicción, si la

predicción es mayor a 85%, se define a la entrada como una imagen de un usuario en tiempo real.

Con la experimentación se verificó que el sistema no sea vulnerable a ataques de presentación tanto de fotos como de videos, los resultados se presentan a continuación.

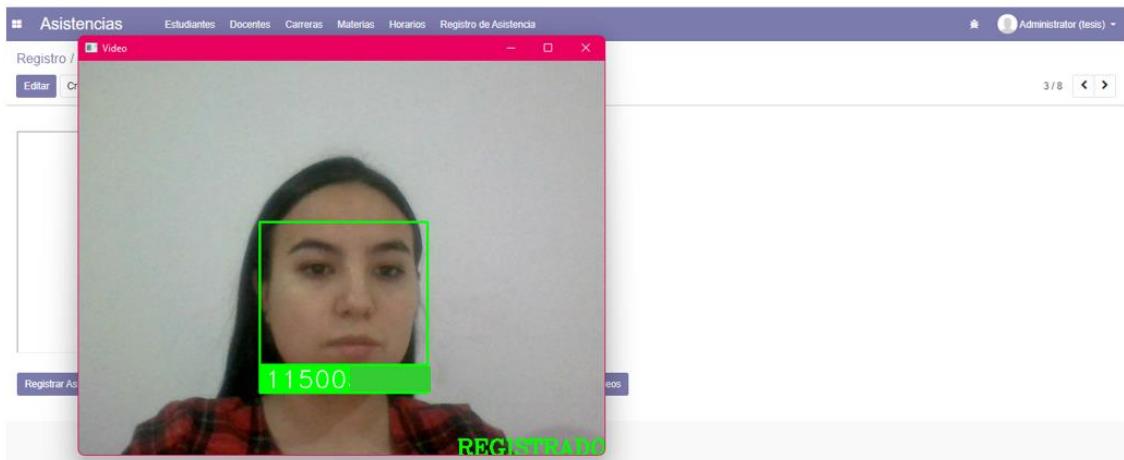


Figura 58. Detección de vida con CNN 3D (usuario en tiempo real)

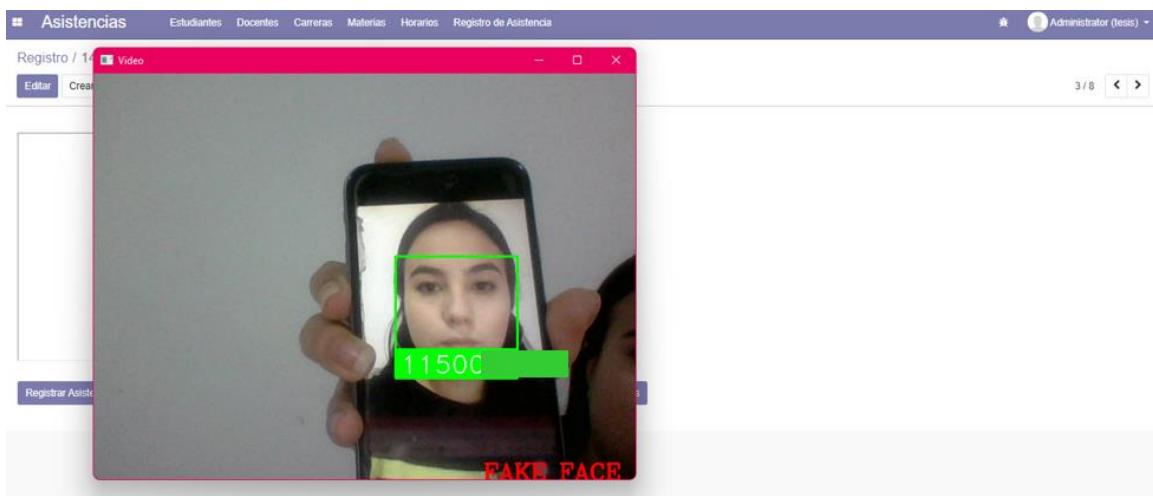


Figura 59. detección de vida con CNN 3D (reproducción de video)

En el caso de la figura 58, se presentó frente al punto de captura el rostro de un usuario en tiempo real, y se observa que el detector de vida funcionó correctamente, el resultado es el esperado, dado que se comprobó la vida de la imagen de entrada se registró la asistencia y se emitió el enunciado “Registrado”. Para el experimento de la figura 59, el modelo también funciona correctamente, detectó que no se trata del rostro de una persona en tiempo real, por

tanto, aunque se reconoció el rostro, no se registró la asistencia y se emitió el mensaje “FAKE FACE”, haciendo referencia a que se trata de un ataque de presentación.

Para observar los experimentos realizados tanto con una foto como con una reproducción de video dirigirse a [Enlace de Github](#)

Conclusión

A consecuencia de que el modelo de detección de vida con redes neuronales convolucionales de 3 dimensiones, arrojó los resultados que se esperaban, se lo selecciona como modelo para detección de vida en el reconocimiento facial para el registro de asistencia en el laboratorio de CIS/C UNL.

Hasta este punto del presente TT, ya se tiene desarrollado el modelo de reconocimiento facial con detector de vida para el registro de asistencias. A continuación, se presenta registros de asistencia dentro del sistema de administración.

Asistencias								
	Estudiantes	Docentes	Carreras	Materias	Horarios	Registro de Asistencia	Usuarios	
Registro								Búsqueda.
<input type="checkbox"/>	Laboratorio	Fecha	Nombres Y Apellidos	Identifi...	Rol	Ciclo	Paralelo	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:11:56	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:18:44	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:45:48	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:51:22	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:53:11	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:58:22	Carmen Azucena False	110608...	Estudiante			
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:58:24	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 14:11:57	Dayanna Alvarado	115003...	Estudiante	10	A	
<input type="checkbox"/>								

Figura 60. Listar registros de asistencia.

Ya obtenidos el listado de registros, se puede exportar todos estos datos en un documento de Excel, presionando en el botón de exportación, además, es posible realizar filtros para exportar únicamente los registros que se requieran.

Etapa 3: Testeo

6.3. Objetivo 3:

Evaluar mediante métricas de la Matriz de Confusión la precisión del prototipo de reconocimiento facial con detección de vida para el registro de asistencias en entornos simulados.

En este apartado se presenta las pruebas realizadas para determinar la precisión del prototipo realizado de reconocimiento facial con detección de vida, se realizó una medición implementando la matriz de confusión, la cual generalmente se utiliza para sistemas de predicción, es decir sistemas que implementan inteligencia artificial.

Cabe mencionar que las pruebas del sistema fueron desarrolladas con estudiantes de CIS/C que acuden al laboratorio de software en el periodo académico abril-septiembre 2022, obtenido un total de 40 estudiantes voluntarios pertenecientes a séptimo y décimo ciclo.

En primera instancia, se realizaron las pruebas respectivas para evaluar el reconocimiento facial, seguidamente se procedió a evaluar el modelo de detección de vida, finalmente se cuantifica el resultado de las pruebas realizadas en la matriz de confusión correspondiente.

6.3.1. Tarea 1: Evaluar mediante métricas de la Matriz de confusión la precisión del reconocimiento facial.

A fin de evaluar el modelo de reconocimiento facial realizado, fue necesario construir la base de datos con las fotos de los rostros de los participantes, para ello, se realizó una encuesta que se compartió con los participantes a fin de que estos pudiesen aportar una foto que cumpla con los siguientes requerimientos:

- La fotografía debió contener el rostro del participante de manera frontal.
- Antigüedad de fotografía máxima de un año.
- El rostro en la fotografía no debió contener accesorios (lentes, mascarillas).
- La fotografía debió tener buena iluminación.
- La fotografía no debió contener filtros embellecedores.

Se obtuvo un total de 40 estudiantes participantes, véase la encuesta realizada para la recopilación de datos en [Anexo 9](#).

Cabe mencionar, que a los participantes se los registró en el sistema como estudiantes, se llenó el formulario correspondiente, agregando las fotos y asignándoles un id (número de

cédula) de usuario con el que fueron identificados durante la prueba, estas imágenes se almacenaron en el equipo de cómputo configurado como servidor donde se encuentra la base de datos. Una muestra de la recopilación de fotografías se presenta en la figura 61, seguida de la figura 62 donde se presenta el registro en el sistema de los participantes.

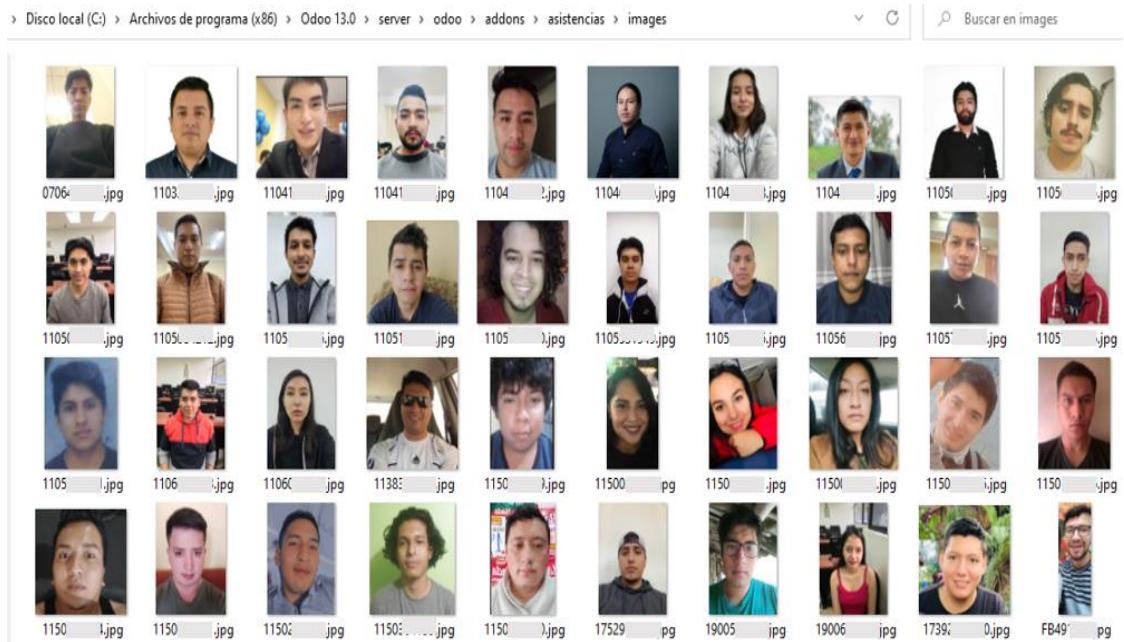


Figura 61. Fotografías de la BD de pruebas.

Asistencias									
Estudiante									
	Identificación	Nombres	Apellidos...	Ciclo	Paralelo...	Dirección de Correo	Carrera	Materia	Docente
<input type="checkbox"/>	1150	Dayanna	Alvarado	10	A	dayan...	Ingenieria en sistem...	Compiladores	Oscar Cumbic...
<input type="checkbox"/>	0706	Lenin Fabricio	Yanagomez	10	A	lenin.y...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1105	Jimmy	Jaramillo	10	A	jimmy.j...	Ingenieria en sistem...	Compiladores	Oscar Cumbic...
<input type="checkbox"/>	1104	Verónica Estefanía	Placencia	7	B	veronic...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1150	Jhandy Marcel	Santorum	7	A	jhandy...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1104	Danny Michael	Jaramillo	7	A	danny.j...	Ingenieria en sistem...	Compiladores	José Guamán
<input type="checkbox"/>	1150	Vladimir Abelardo	Celi	7	A	vladimi...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1105	Xavier	Gutierrez	7	A	xavier.g...	Ingenieria en sistem...	Compiladores	Wilman Cham...
<input type="checkbox"/>	1150	Gilson Orlando	Quezada	7	A	gilson.q...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1104	Yohanna Elizabeth	Ureña	7	A	yureen...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1150	Edinson Roberto	Quizhpe	7	A	edinson...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1105	José Andrés	Rios	7	A	jose.ri...	Ingenieria en sistem...	Base de dat...	Wilman Cham...
<input type="checkbox"/>	1105	Alex Santiago	Malo	0	A	alex.santi...	Ingenieria en sistem...	Compiladores	José Guamán

Figura 62. Registro de los participantes en el sistema.

Para llevar a cabo la respectiva evaluación del modelo de reconocimiento, se solicitó a los participantes colocarse frente al punto de captura. Un ejemplo se presenta a continuación.

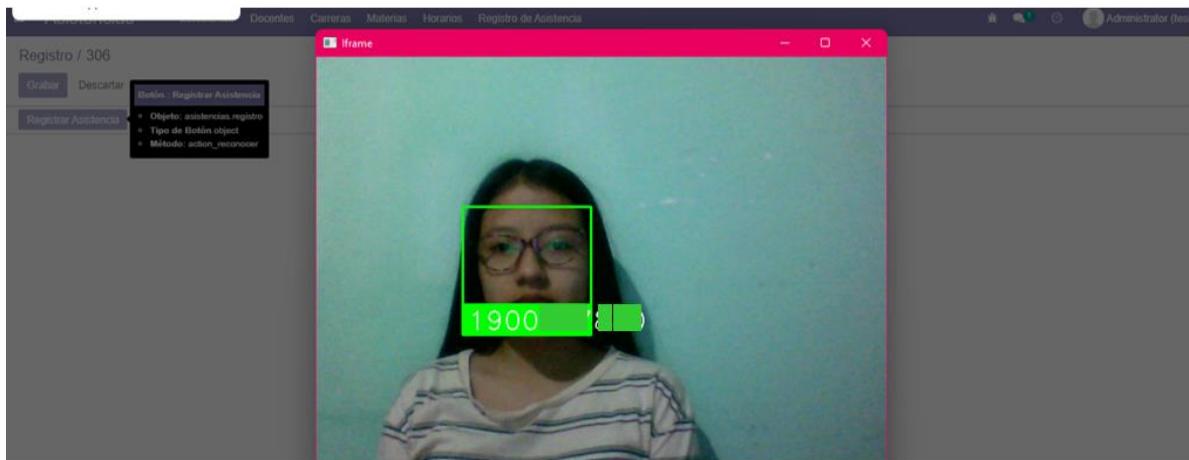


Figura 63. Ejemplo de ejecución de la primera prueba Reconocimiento facial

Para evaluar el modelo de reconocimiento facial, el proceso se centró en comparar si el número de cédula que el sistema presenta, dentro del rectángulo verde, es igual al número de cédula que el participante proporcionó, en caso de ser iguales, se considera a la predicción del modelo como un verdadero positivo.

Además, en las pruebas realizadas para el modelo de reconocimiento facial, no se tomó en cuenta el modelo de detección de vida, es por ello que también se emplearon fotografías de personas no registradas en el sistema para evaluar el modelo de reconocimiento.

Un ejemplo se presenta a continuación.

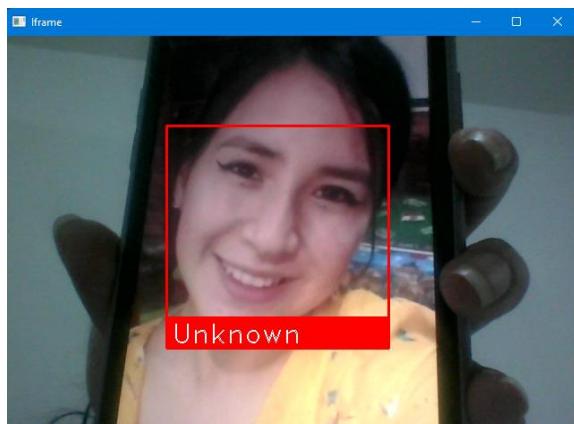


Figura 64. Ejemplo de ejecución de pruebas de reconocimiento facial con rostros de personas no registradas en el sistema

Véase evidencia del total de pruebas realizadas del modelo de reconocimiento facial en este [enlace](#).

En la presente evaluación, se realizaron 80 experimentos, entre los cuales 40 corresponden a personas registradas en el sistema y los restantes con personas no registradas.

A continuación, se presenta la matriz de confusión resultante.

Tabla 14. Matriz de confusión de pruebas de reconocimiento facial.

		Predicción	
		VP	FN
Real	40	0	
	FP	VN	
	4	36	

En donde:

VP: Corresponden a los verdaderos positivos, es decir el número de veces que el sistema reconoció al usuario y le asignó correctamente su identificador.

FP: Corresponden a los falsos positivos, es decir el número de veces que el sistema se equivocó y trató como conocidos a usuarios no registrados en el sistema. Adicional al número de veces que el sistema reconoció equívocamente a un usuario y le asignó erróneamente el identificador.

FN: Corresponden a los falsos negativos, el número de veces que se presentó a usuarios registrados y el sistema no los reconoció.

VN: Corresponden a los verdaderos negativos, el número de veces que se presentó a usuarios no registrados y el sistema no los reconoció.

Según las pruebas realizadas, se obtuvo que el modelo reconoció perfectamente a 40 personas y colocó el identificador correspondiente, en ninguna ocasión trató como desconocidas a personas registradas; por otro lado, cuando se presentó frente al punto de captura a rostros de usuarios no registrados en el sistema los trató correctamente como desconocidos en 36 ocasiones, mientras que se equivocó en 4 ocasiones asignándoles un identificador de personas registradas en el sistema.

A continuación, se evalúa el modelo según las métricas de la matriz.

Precisión: porcentaje de casos positivos detectados.

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos positivos}} = \frac{40}{44} = 0.90 \quad (9)$$

Sensibilidad: valor que indica la capacidad del modelo para distinguir los casos positivos de los negativos.

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos Negativos}} = \frac{40}{40} = 1.00 \quad (10)$$

Media armónica: Este es un valor que resume la precisión y recall en una sola métrica.

$$2 \frac{(\text{sensibilidad} * \text{precisión})}{\text{sensibilidad} + \text{precisión}} = 2 \frac{0.90 * 1}{0.90 + 1} = 2 * \frac{0.90}{1.90} = 0.94 \quad (11)$$

Exactitud: Es un valor que mide el porcentaje de casos que el modelo de clasificación ha acertado.

$$\frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Verdadero Positivos} + \text{Verdaderos Negativos} + \text{Falsos Positivos} + \text{Falsos Negativos}} = \frac{76}{80} = 0.95 \quad (12)$$

Para un total de 80 experimentos realizados, en los que se incluyen personas registradas y no registradas en el sistema, se logró construir la matriz de confusión, obteniendo como resultado una precisión del modelo con respecto a los valores positivos de 0.90 lo cual corresponde al 90%, con respecto a la sensibilidad de reconocimiento en cuanto a los valores negativos se obtuvo 1.00 lo que corresponde al 100%, en total la media armónica entre los valores positivos y negativos es de 0,94 lo que corresponde al 94%, finalmente la exactitud del modelo en general es de 95%.

6.3.2. Tarea 2: Evaluar mediante métricas de la Matriz de Confusión la precisión de la detección de vida.

Para evaluar la precisión del modelo de detección de vida del prototipo se experimentó presentando frente al punto de captura a personas en tiempo real, fotografías y videos de las personas almacenadas en la base de datos construida para la evaluación del modelo de reconocimiento facial.

- Un ejemplo de persona en tiempo real frente al punto de captura se presenta a continuación.

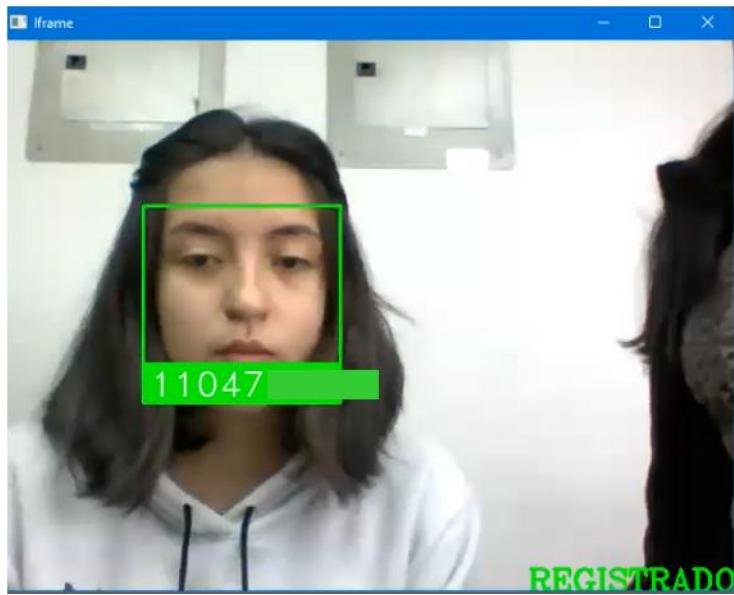


Figura 65. Ejemplo de ejecución de la prueba de detección de vida con persona en tiempo real.

La figura 65, corresponde a una prueba del modelo de detección de vida, se observa en la parte inferior la palabra “REGISTRADO”, escrito en letras verdes, lo cual hace referencia que el rostro presentado en el punto de captura corresponde a una persona en tiempo real.

Un ejemplo de fotografía frente al punto de captura se presenta a continuación.

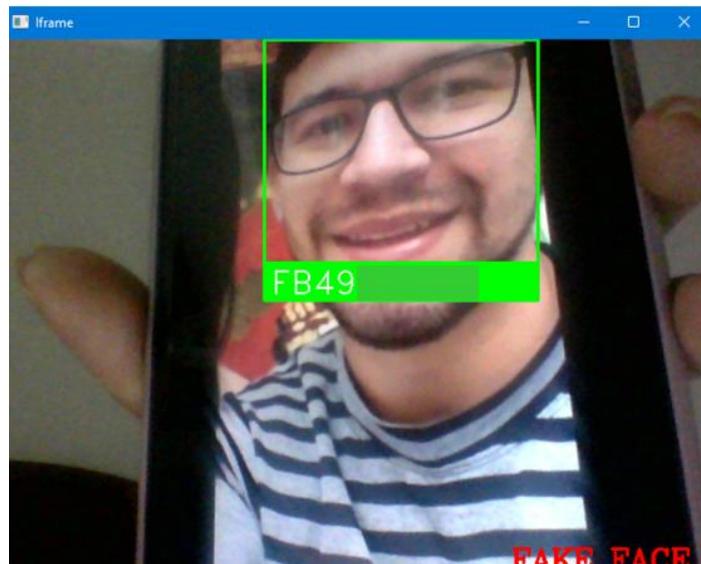


Figura 66. Ejemplo de ejecución de la prueba de detección de vida con una fotografía.

La figura 66, corresponde a una prueba del modelo de detección de vida, se observa en la parte inferior la palabra “FAKE FACE”, escrito en letras de color rojo, lo cual hace referencia

que el rostro presentado en el punto de captura no corresponde a una persona en tiempo real, es decir pudiese ser una fotografía o una reproducción de video.

Véase evidencia del total de pruebas realizadas del modelo de detección de vida en este [enlace](#).

A continuación, se presenta la matriz de confusión resultante.

Tabla 15. Matriz de confusión de pruebas de detección de vida.

		Predicción	
		VP	FN
Real	37	3	
	FP	VN	
	1	39	

En donde:

VP: Corresponden a los verdaderos positivos, es decir el número de veces que el sistema detectó correctamente a una persona en tiempo real.

FP: Corresponden a los falsos positivos, es decir el número de veces que el sistema se equivocó y detectó a una fotografía como una persona en tiempo real.

FN: Corresponden a los falsos negativos, el número de veces que se presentó personas en tiempo real y el sistema las detectó como fotografías.

VN: Corresponden a los verdaderos negativos, el número de veces que se presentó fotografías frente al punto de captura y el sistema la detectó correctamente.

Para la evaluación del presente modelo, se realizaron 80 experimentos, 40 de ellos fueron con personas en tiempo real y los restantes con fotografías. Según las pruebas realizadas, se obtuvo que el modelo detectó 37 veces correctamente la vivacidad del personaje frente al punto de captura, y se equivocó en 3 ocasiones al detectar como fotografía a una persona en tiempo real, además, cuando se presentó en el punto de captura a una fotografía o reproducción de video, el sistema acertó 39 veces al no detectar la vivacidad, y se equivocó en 1 ocasión ya que detectó como persona en tiempo real a una fotografía.

A continuación, se evalúa el modelo según las métricas de la matriz.

Precisión: porcentaje de casos positivos detectados.

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos positivos}} = \frac{37}{37 + 1} = 0.97 \quad (13)$$

Sensibilidad: valor que indica la capacidad del modelo para distinguir los casos positivos de los negativos.

$$\frac{\text{verdadero positivo}}{\text{verdadero positivos} + \text{Falsos Negativos}} = \frac{37}{37 + 3} = 0.93 \quad (14)$$

Media armónica: Este es un valor que resume la precisión y sensibilidad en una sola métrica.

$$2 \frac{(\text{sensibilidad} * \text{precisión})}{\text{sensibilidad} + \text{precisión}} = 2 \frac{0.93 * 0.97}{0.93 + 0.97} = 2 * \frac{0.90}{1.90} = 0.95 \quad (15)$$

Exactitud: Es un valor que mide el porcentaje de casos que el modelo de clasificación ha acertado.

$$\frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Verdadero Positivos} + \text{Verdaderos Negativos} + \text{Falsos Positivos} + \text{Falsos Negativos}} = \frac{37 + 39}{80} = 0.95 \quad (16)$$

Para evaluar el modelo de detección de vida, se realizaron un total de 80 experimentos, entre los cuales se presentó frente al punto de captura 40 veces a personas en tiempo real y 40 veces a fotografías o reproducciones de video, con ello se logró construir la matriz de confusión, obteniendo 97% como resultado de la precisión del modelo en cuanto a los valores positivos, además según los valores negativos se obtuvo una sensibilidad del modelo de 93%, el total de la media armónica entre los valores positivos con respecto a los negativos es de 95%, finalmente la exactitud global del modelo construido es de 0.95 lo que corresponde al 95%.

6.3.3. Verificación de cumplimiento de los requerimientos de software.

Una vez finalizado el prototipo se procedió a la respectiva verificación del cumplimiento de los requerimientos de software especificados en el [Anexo 6](#), para ello se solicitó al técnico encargado del laboratorio de software de CIS/C UNL que realice pruebas reales con el software propuesto. A continuación, se presenta las tablas de resumen del cumplimiento de los requerimientos funcionales y no funcionales.

Tabla 16. Cumplimiento de los requerimientos funcionales

Código	Nombre	Descripción	Cumple
RF001	Acceso al sistema	El sistema debe permitir al administrador acceder al sistema colocando sus credenciales de ingreso (usuario y contraseña)	Si
F002	Administrar docentes	El sistema debe permitir crear, modificar, listar o eliminar información de los docentes que utilizarán el laboratorio	Si
RF003	Administrar estudiantes	El sistema debe permitir crear, modificar, listar o eliminar información de los estudiantes que utilizarán el laboratorio	Si
RF004	Administrar carreras	El sistema debe permitir crear, modificar, listar o eliminar información de las carreras que utilizarán el laboratorio	Si
RF005	Administrar materias	El sistema debe permitir crear, modificar, listar o eliminar información de las materias que se impartirán en el laboratorio	Si
RF006	Administrar horarios	El sistema debe permitir crear, modificar, listar o eliminar información de los horarios en los que está disponible el laboratorio para cada grupo de usuarios.	Si
RF007	Generar reportes	El sistema debe permitir generar reportes de los registros de asistencia según intervalos de fechas.	Si
RF008	Registrar asistencia	El sistema debe permitir registrar asistencia a partir de la identificación del rostro mediante cámara web.	Si

Tabla 17. Cumplimiento de los requerimientos no funcionales

Código	Nombre	Descripción	Cumple
RNF001	Interfaz gráfica de usuario	El sistema debe tener una IGU basada en plataforma web, visible a través de navegadores de internet.	Si
RNF002	Tiempo de respuesta	El sistema debe dar respuesta rápida como máximo hasta 10 segundos.	Si
RNF003	Usabilidad	Las interfaces deben ser intuitivas para el usuario y que permitan llevar a cabo los procesos fácilmente.	Si

RNF004	Prevención de ataques de presentación	El sistema debe tener mecanismos de detección de vida que prevengan ataques de presentación en el reconocimiento facial.	Si
--------	---------------------------------------	--	----

Como se observa en la Tabla 16 y Tabla 17, el prototipo propuesto da cumplimiento a todos los requerimientos planteados.

La validación del cumplimiento de los requerimientos de software por parte del Ingeniero Luis Sinche, técnico encargado del laboratorio de software se evidencia en el [Anexo 10](#).

6.3.4. Encuesta de satisfacción del prototipo propuesto.

Luego de realizar las pruebas del prototipo en un ambiente real con estudiantes usuarios del laboratorio de software, se encontró conveniente realizar una encuesta de satisfacción de la alternativa propuesta para el registro de asistencia, para ello se pidió a los participantes de las pruebas del prototipo responder un formulario ([véase Anexo 11](#)), con lo cual se obtuvo lo siguiente.

- El 100% de los participantes no necesitaron manipular instrumentos de uso comunitario para registrar asistencia con el software propuesto.
- El 67.5% de los estudiantes encuestados aseguran que el registro de su asistencia con el software propuesto se lleva a cabo en un rango entre 5 y 10 segundos, mientras que el 25% afirman que el tiempo que conlleva es de entre 11 a 20 segundos, además el 7.5% asegura que el software requiere un tiempo de entre 31 segundos a un minuto. Por tanto, al calcular el promedio entre los valores obtenidos, se tiene como resultado que el tiempo estimado para el registro de asistencia es de 9,2 segundos, perteneciente a la primera alternativa correspondiente a entre 5 y 10 segundos.
- El 100% de los estudiantes encuestados afirman que sí les gustaría contar con el método de reconocimiento facial propuesto para el registro de asistencias en el laboratorio de software.

7. Discusión

El propósito de este apartado es analizar los resultados obtenidos durante el desarrollo del presente Trabajo de Titulación.

7.1. Desarrollo de la propuesta

Para el desarrollo de la presente propuesta se experimentó con cada uno de los modelos preentrenados empleados en los trabajos relacionados citados en los apartados preliminares de este documento, esto con finalidad de llevar a cabo el proceso de reconocimiento facial. El primer trabajo relacionado [37], es una propuesta que emplea la tecnología de Visual Basic y MySQL utilizando el modelo preentrenado EigenFaces. El segundo trabajo [38], está desarrollado con la tecnología Matlab, MySQL y el modelo preentrenado LBPH (Local Binary Patterns). El tercer trabajo relacionado [39], plantea una propuesta basada en la visión artificial utilizando redes neuronales artificiales de Histograma de Gradientes orientados HOG y redes neuronales convolucionales con la librería OpenCV y dlib. El cuarto trabajo [40], propone un recurso tecnológico utilizando Matlab y EigenFaces como recurso de reconocimiento facial para la autenticación de rostros de usuarios. En el último trabajo [41], se utiliza OpenCV, FisherFaces para el reconocimiento facial, el servidor web Apache y la base de datos PostgreSQL.

En la experimentación realizada en la presente propuesta se obtuvo que los modelos EigenFaces, FisherFaces y LBPHFaces, no cuentan con un modelo detector de rostro, por tanto es necesario agregar al algoritmo uno externo, además, requieren un número mínimo de 300 fotogramas por cada individuo que se va a registrar en la base de datos, el proceso de entrenamiento tardó entre 2 a 7 minutos con el registro de cinco personas en el sistema, requieren un umbral de comparación el cual se define mediante una extensa experimentación y calibre, cada vez que se incrementa un nuevo rostro a la base de datos, además estos modelos tienen un alto índice de equivocación en las predicciones que realiza. Por otro lado, dlib, una librería de código abierto útil para la detección y reconocimiento de rostros, el cual fue combinado en esta propuesta con el modelo preentrenado Face-recognition, requiere únicamente una fotografía por cada persona que se registra en la base de datos, consecuentemente el tiempo de reentrenamiento es corto a diferencia de los otros 3 modelos con los que se experimentó, además este modelo no requiere definir manualmente un umbral de comparación y sus predicciones tienen una exactitud del 95%.

Por otro lado, ninguno de los trabajos relacionados implementa mecanismos de seguridad contra ataques de presentación, lo cual es indispensable en los procesos donde no se tiene control humano para llevar a cabo procesos de autenticación de usuarios, sin un control contra estos ataques el sistema es propenso a ser vulnerado por usuarios maliciosos, quienes intenten realizar un registro de asistencia presentando frente al punto de captura una fotografía en lugar del rostro de una persona en tiempo real.

Por ello, en la propuesta se desarrolló un modelo de detección de vida, que permita diferenciar cuando se trate de un ataque de presentación y una presentación auténtica, en primera instancia se experimentó con un enfoque activo mediante la contabilización de parpadeos y se determinó a este como un mecanismo de baja robustez ante ataques de presentación, consecuentemente, se desarrolló un modelo siguiendo un enfoque pasivo empleando redes neuronales convolucionales de 3 dimensiones que permitieron medir la profundidad de la imagen de entrada, se corroboró el buen funcionamiento ante ataques de presentación y se la definió como solución.

Durante la fase de pruebas se identificó que la solución propuesta tiene una exactitud global del 95%, considerando que el modelo de reconocimiento facial funciona correctamente cuando se presentan rostros con accesorios, maquillaje, mascarillas, lentes, siempre y cuando se deje al descubierto alguna porción del rostro, por otro lado, para el buen funcionamiento de modelo de detección de vida, se requiere ubicar la cámara web en un lugar con luminosidad intermedia, en un fondo llano, color claro y sin irregularidades en la superficie de fondo.

En respuesta de la pregunta de investigación, la cual interroga ¿Un prototipo de reconocimiento facial evitará la manipulación de los registros físicos de asistencia en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja?, se menciona que con las pruebas ejecutadas se corroboró que el prototipo de reconocimiento facial con detección de vida para el registro de asistencias no requiere la manipulación de instrumentos de uso comunitario, únicamente se requiere la presentación del rostro en el punto de captura, además, el prototipo toma alrededor de 9.2 segundos para realizar el registro y su módulo web facilita la creación de reportes de uso del laboratorio.

7.2. Valoración técnica económica ambiental

El desarrollo del presente Trabajo de Titulación se realizó a través de recursos técnicos, económicos y ambientales.

Los recursos técnicos permitieron el desarrollo del sistema de reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de CIS/C UNL, a través de diversas herramientas de software, con los cuales se pudo dar cumplimiento a los requisitos del sistema para llevar a cabo el registro de asistencia con mecanismos que no requieran la manipulación de instrumentos de uso comunitario y que faciliten la creación de reportes de uso del laboratorio en un tiempo mínimo.

Los recursos económicos que se requirieron para llevar a cabo el presente TT, fueron mínimos ya que se utilizaron herramientas de código abierto para el desarrollo del módulo web.

En cuanto al aspecto ambiental se contribuyó de forma positiva al medio ambiente ya que el sistema para el registro de asistencias mediante reconocimiento facial evita el uso de papel para el registro diario de asistencias.

A continuación, en la siguiente tabla se detalla el valor económico perteneciente al talento humano que intervino en la realización del presente trabajo.

Tabla 18. Valoración económica - Talento humano

Nombre	Cargo	# de Horas	Precio / Hora	Valor total
Dayanna Magdalla Alvarado Castillo	Desarrollador del Trabajo de Titulación	440	\$10.00	\$4400.00
Oscar Miguel Cumbicus Pineda	Director del Trabajo de Titulación	40	\$10.28	\$411.2
Tota				\$4811,20

A continuación, en la siguiente tabla se detalla el valor económico perteneciente a los servicios utilizados en la realización del presente trabajo.

Tabla 19. Valoración económica – Servicios

SERVICIOS			
Nombre	Descripción	Costo/Mes	Valor total
Internet	Desarrollador del trabajo de titulación	\$25.00	\$125.00
Total			\$125.00

Seguidamente, se detalla el valor económico perteneciente a los recursos Software y Hardware utilizados en la realización del presente trabajo.

Tabla 20. Valoración económica – Recursos hardware

HARDWARE			
Nombre	Cantidad	Precio Unitario	Valor total
Portátil Lenovo	1	\$500.00	\$500.00
Subtotal			\$500.00

Tabla 21. Valoración económica – Recursos Software

SOFTWARE		
Nombre	Descripción	Valor total
Framework	Odoo ERP	\$00.00
Editor de Código	Pycharm	\$00.00
Gestor de BD	PostgreSQL	\$00.00
Lenguaje de Programación	Python-JavaScript	\$00.00
Subtotal		\$00.00

Finalmente, la siguiente tabla presenta el costo total de la realización del Trabajo de Titulación:

Tabla 22. Costo total del proyecto

Recursos	Subtotal
Talento Humano	\$4811,20
Servicio	\$125.00
Recursos Hardware	\$500.00
Recursos Software	\$00.00
Subtotal	\$5436.20
Imprevistos 10%	\$543.62
Total:	\$5979.82

7.3. Limitaciones de la propuesta

Las limitaciones del sistema están relacionados a las dificultades que se detallan a continuación:

- Para el proceso de realización del presente TT se contó con un computador portátil con tarjeta gráfica Radeon, sin embargo, las recomendaciones para la utilización de ciertos modelos preentrenados para el reconocimiento facial, sugieren contar con GPU Nvidia, por tanto, el bajo rendimiento del computador utilizado presentó complicaciones durante el proceso de desarrollo.
- Odoo ERP no presenta tanta documentación para enlazar los lenguajes de programación Python y Javascript, por tanto, la realización de esta actividad tomó más tiempo de lo esperado.
- El prototipo propuesto, específicamente el modelo de detección de vida, para su buen funcionamiento requiere ser ejecutado en ambientes con alta iluminación y preferiblemente donde el fondo donde apunte la cámara web sea liso, no contenga adornos y en colores de pintura llanos.

8. Conclusiones

Una vez finalizado el presente Trabajo de Titulación denominado: Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja., se ha determinado las siguientes conclusiones:

- En el presente Trabajo de Titulación se desarrolló un prototipo de reconocimiento facial con detección de vida, para el registro de asistencias de estudiantes y docentes en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Facultad de la Energía, las industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja, el prototipo desarrollado no requiere la manipulación de instrumentos de uso comunitario, el registro toma alrededor de 9.2 segundos por usuario y su módulo web permite la creación de reportes de uso del laboratorio en un tiempo oportuno.
- El diagrama diseñado siguiendo la notación BPMN 2.0. permitió ilustrar el proceso actual de registro de asistencia en el laboratorio de software de CIS/C UNL, facilitó el entendimiento del procedimiento de registro y sirvió como base fundamental para la elaboración del presente prototipo.
- El clasificador pre-entrenado denominado Face-Recognition permitió la detección del rostro y su respectivo reconocimiento reentrenando el modelo con una base de datos considerablemente pequeña. Las redes neuronales convolucionales de 3 dimensiones permitieron el desarrollo del modelo de detección de vida con un enfoque pasivo, dado que determinan la profundidad de la imagen.
- Las pruebas realizadas a los modelos del prototipo permitieron la construcción de matrices de confusión para determinar la precisión de los mismos, obteniendo para cada uno de estos una exactitud global del 95%, considerándose al prototipo general como prototipo funcional.

9. Recomendaciones

- Cuando se requiere desarrollar sistemas basados en inteligencia artificial es recomendable contar con el hardware adecuado, dado que el desarrollo en esta área presenta alto costo computacional, se requiere buen rendimiento en el hardware que se emplea.
- El marco de trabajo Design Thinking se enfoca en comprender las necesidades de las personas implicadas en problemas reales, con el objetivo de entenderlos para poder desarrollar mejores soluciones, ayuda al investigador a crear prototipos y modelos que le permitan explorar diferentes soluciones y enfoques, lo cual es recomendable especialmente cuando se investigan temas que requieren una solución práctica o tangible.
- Para el desarrollo de un TT es recomendable seguir modelos de inteligencia artificial pre-entrenados disponibles en la web y adaptarlos a cada una de las necesidades, dado que la construcción de uno propio conlleva un coste computacional relativamente alto y el tiempo excede los límites previstos en el desarrollo de un trabajo de titulación.

9.1. Trabajos Futuros

- El prototipo se pudiera implementar como un módulo móvil.
- El prototipo requiere el registro manual de los estudiantes y docentes en el módulo web, como trabajo futuro se podría consumir las páginas oficiales universitarias donde ya están presentes los listados respectivos.

10. Referencias

- [1] Hik Vision, “Reconocimiento facial para control de acceso. - Telefonia total,” Apr. 24, 2020. <https://telefoniatotal.com/seguridad/reconocimiento-facial-rapido-y-preciso-para-control-de-acceso/> (accessed Nov. 25, 2021).
- [2] R. Cárcamo Catalán, “Pandemia incrementa uso de aplicaciones de identificación sin contacto - Pontificia Universidad Católica de Chile,” *Pontificia Universidad Católica de Chile*, 2021. <https://www.uc.cl/noticias/pandemia-incrementa-uso-de-aplicaciones-de-identificacion-sin-contacto/> (accessed Jun. 24, 2022).
- [3] B. Takeyas López, “Introducción a la inteligencia artificial,” Nuevo Laredo, México, 2007. [Online]. Available: <http://www.itnuevolaredo.edu.mx/takeyas>.
- [4] J. A. Cancelas, R. C. González, I. Álvarez, and J. M. Enguita, *Procesamiento Morfológico, Visión 3D: Estereoscopía, Álgebra lineal básica para visión por computador, Geometría Proyectiva para Visión 3D*, vol. 1. 2016.
- [5] A. F. Garcés Núñez, “Sistema De Reconocimiento Facial Con Visión Artificial Para Apoyar Al Ecu-911 Con La Identificación De Personas En La Lista De Los Más Buscados,” Universidad Técnica de Ambato, 2017.
- [6] A. David, C. Alonso, and A. Mart, “Visión por computadora : identificación , clasificación y seguimiento de objetos .,” p. 12.
- [7] D. L. CASTAÑO SAAVEDRA and J. D. ALONSO SIERRA, “SISTEMA DE RECONOCIMIENTO FACIAL PARA CONTROL DE ACCESO A VIVIENDAS,” Universidad Católica de Colombia, 2019.
- [8] Ignacio G. R. Gavilán, “Las cuatro fases del reconocimiento facial | Ignacio G.R. Gavilán,” Oct. 21, 2020. <https://ignaciogavilan.com/las-cuatro-fases-del-reconocimiento-facial/> (accessed Nov. 25, 2021).
- [9] Y. A. Cama Castillo, “Prototipo computacional para la detección y clasificación de expresiones faciales mediante la extracción de patrones binarios locales,” Pontifica Universidad Católica del Perú, 2015.
- [10] Electronic IDentification, “Casos de uso 2021 del servicio de autenticación biométrica facial SmileID,” May 10, 2021. <https://www.electronicid.eu/es/blog/post/autenticacion-biometrica-facial-smileid/es> (accessed Nov. 25, 2021).

- [11] I. Granja, D. Moreno, F. Cabrera, and P. Valle, "Procesamiento De Imágenes Para La Identificación De Personas Como Sistema De Seguridad En Zonas Domiciliarias/Image Processing for identification of people as a security system in domiciliary zones," *KnE Eng.*, vol. 2020, pp. 164–186, 2020, doi: 10.18502/keg.v5i2.6233.
- [12] N. Ríos del Moral, "CONTROL DE ACCESO CON IDENTIFICACIÓN MEDIANTE IMÁGENES," Universidad de Jaén, 2016.
- [13] OpenCV, "OpenCV: Face Recognition with OpenCV," Dec. 31, 2019. https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html (accessed Nov. 25, 2021).
- [14] D. King, "Librería dlib C ++," 2018. <http://blog.dlib.net/> (accessed Feb. 06, 2022).
- [15] A. Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning," *Research medium*, Jul. 24, 2016.
- [16] Biometrics Institute, "What does presentation attack detection and liveness actually mean?," Australia. Accessed: Nov. 25, 2021. [Online]. Available: <https://www.biometricsinstitute.org/wp-content/uploads/1809-liveness-questions-final.pdf>.
- [17] M. Faresse, "The Most Common Facial Recognition Spoofing Methods," May 21, 2020. <https://blog.dormakaba.com/the-most-common-facial-recognition-spoofing-methods-and-how-to-prevent-them/> (accessed Nov. 26, 2021).
- [18] Anh T. Dang, "Facial Recognition: Types of Attacks and Anti-Spoofing Techniques | by Anh T. Dang | Towards Data Science," Oct. 10, 2020. <https://towardsdatascience.com/facial-recognition-types-of-attacks-and-anti-spoofing-techniques-9d732080f91e> (accessed Nov. 25, 2021).
- [19] IDR&D, "El importante papel de la detección de vida en la autenticación biométrica facial," p. 14, 2020, [Online]. Available: <https://www.idrnd.ai/wp-content/uploads/2020/09/IDRD-Facial-Liveness-Whitepaper-Spanish.pdf>.
- [20] Thales, "Liveness in biometrics: spoofing attacks and detection," Dec. 04, 2020. <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/liveness-detection> (accessed Nov. 25, 2021).
- [21] S. K. Izquierdo and C. Ventura Royo, "Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional," *Univ.*

Oberta Catalunya, p. 53, 2016, [Online]. Available: <http://hdl.handle.net/10609/52222>.

- [22] D. Calvo, “Red Neuronal Convolucional CNN,” *Redes Neuronales*, Jul. 20, 2017. <https://www.diegocalvo.es/red-neuronal-convolucional/> (accessed Feb. 09, 2022).
- [23] P. Pastor, “Usando Redes Neuronales Convolucionales Para Convertir Características Visuales en Estímulos Sonoros,” Universidad de La Laguna, 2018.
- [24] S. BANSAL, “3D Convolutions : Understanding + Use Case | Kaggle,” 2019. <https://www.kaggle.com/shivamb/3d-convolutions-understanding-use-case> (accessed Feb. 15, 2022).
- [25] Odoo company, “Survival guide for appaying at Odoo,” 2012. [Online]. Available: <https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:32016R0679&from=PT%0Ahttp://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52012PC0011:pt:NOT>.
- [26] O. ERP, “Odoo Documentation.” https://doc.odoo.com/6.0/es/book/1/1_1_Inst_Config/1_1_Inst_Config_architecture/.
- [27] J. I. Barrios Arce, “La matriz de confusión y sus métricas – Inteligencia Artificial –,” 2019. <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/> (accessed Mar. 02, 2022).
- [28] PostgreSQL, “PostgreSQL: Acerca de.” <https://www.postgresql.org/about/> (accessed Dec. 02, 2021).
- [29] Lucidchart, “Software de diagramación en línea y solución visual | Lucidchart.” https://www.lucidchart.com/pages/es/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_es_tier2_mixed_search_brand_bmm_&km_CPC_CampaignID=1501207844&km_CPC_AdGroupID=63362147892&km_CPC_Keyword=%2Blucidcharts&km_CPC_MatchType=b&km_CPC_ExtensionID (accessed Dec. 02, 2021).
- [30] W. Rogowski and W. Swoboda, “Business Process Model and Notation,” *Management im Gesundheitswesen*, 2020. <https://www.bpmn.org/> (accessed Dec. 07, 2021).
- [31] R. Contreras, L. Meraz, and J. Gonzalez, “Guía para modelar procesos y procedimientos bajo el estándar BPMN 2.0,” p. 66, 2014, [Online]. Available: https://info.jalisco.gob.mx/sites/default/files/leyes/dom-p286-d2_001_guias_para_modelar_procesos_y_procedimientos_001.pdf.

- [32] A. Djandrw, “Qué es Scrum?,” *medium.com*, 2019. <https://medium.com/@andrewdjandrw/qu%C3%A9-es-scrum-674c6b791af4> (accessed Nov. 02, 2021).
- [33] ESI School of managment, “Scrum Un Marco De Trabajo Ágil,” 2021. <https://esieduc.org/nuestros-blogs/scrum-un-marco-de-trabajo-agil-en-los-negocios/> (accessed Nov. 02, 2021).
- [34] J. Irigaray, “Design thinking: qué es, características y fases | EAE,” Nov. 05, 2021. <https://retos-directivos.eae.es/design-thinking-que-es-caracteristicas-y-fases/> (accessed Nov. 25, 2021).
- [35] dinngo, “Design Thinking en Español.” <https://www.designthinking.es/inicio/index.php> (accessed Nov. 25, 2021).
- [36] ITMadrid, “Qué es y para qué sirve Design Thinking | ITMadrid,” Feb. 06, 2020. <https://www.itmadrid.com/que-es-y-para-que-sirve-design-thinking/#> (accessed Nov. 25, 2021).
- [37] M. S. AYALA GOYES, “SISTEMA BIOMÉTRICO DE RECONOCIMIENTO FACIAL PARA EL CONTROL DE ASISTENCIA DEL PERSONAL DOCENTE Y ADMINISTRATIVO DE LA UNIANDES TULCÁN.,” Universidad Regional Autónoma de los Andes “Uniandes,” 2018.
- [38] R.Castro, “Sistema De Control De Acceso Al Personal De La Lavadora De Jeans Fashion Mediante Reconocimiento Facial,” UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD, 2016.
- [39] J. R. Bastidas Gavilanes, “Registro De Asistencia De Alumnos Por Medio De Reconocimiento Facial Utilizando Visión Artificial,” Universidad Técnica de Ambato, 2019.
- [40] R. J. Naranjo Quispe, “RECONOCIMIENTO FACIAL APLICADO A LA AUTENTIFICACION DE USUARIOS EN CURSOS ONLINE DE LA CARRERA DE DOCENCIA EN INFORMATICA DE LA FACULTAD DE CIENCIAS HUMANAS Y DE LA EDUCACION DE LA UNIVERSIDAD TECNICA DE AMBATO,” Universidad Tecnica de Ambato, 2016.
- [41] C. C. Fernando, “SISTEMA INFORMÁTICO DE RECONOCIMIENTO FACIAL PARA EL REGISTRO Y CONTROL DE ASISTENCIA DE LOS SOCIOS DE LA

COOPERATIVA DE TAXIS Y CAMIONETAS PUYO.,” UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES “UNIANDES,” 2019.

- [42] C. E. Franco, C. T. Ospina, E. S. Cuevas, and D. V. Capacho, “RECONOCIMIENTO FACIAL BASADO EN EIGENFACES, LBHP Y FISHERFACES EN LA BEAGLEBOARD-xM,” *Rev. Colomb. Tecnol. Av.*, vol. 2, no. 26, p. 8, 2017, doi: 10.24054/16927257.v26.n26.2015.2387.
- [43] G. Ottado, “Reconocimiento de caras : Eigenfaces y Fisherfaces,” *Univ. la República Uruguay*, pp. 1–15, 2010, [Online]. Available: https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe_final_ottado.pdf.
- [44] D. Verdeguer Valderrama and N. Campos Vasquez, “Diseño E Implementación De Un Sistema De Identificación De Personas Para La Seguridad De Los Accesos A Condominios, Basado En El Algoritmo De Reconocimiento Facial LBPH Faces,” pp. 1–9, 2021, doi: 10.18687/laccei2021.1.1.213.
- [45] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, p. 9, 2021.
- [46] T. Soukupova and Cech Jan, “Real-Time Eye Blink Detection using Facial Landmarks,” Rimske Toplice, Slovenia, Feb. 2016. Accessed: Feb. 07, 2022. [Online]. Available: <http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>.

11. Anexos

Anexo 1. Entrevista a docente usuario del laboratorio de software.



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD LA ENERGÍA LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES



Carrera de Ingeniería en
Sistemas / Computación

CARRERA DE INGENIERIA EN SISTEMAS

Nombre:	Ing. Wilman Chamba Docente de CIS/UNL
Fecha de entrevista:	27 agosto 2021

Objetivo de la entrevista.

La presente entrevista se realiza con finalidad de corroborar el uso del laboratorio de software de CIS/C UNL, además de captar información acerca del proceso actual que se lleva a cabo para el registro de asistencia de los estudiantes en mencionado laboratorio, así como también, tomando en cuenta la actual crisis sanitaria por Covid-19, obtener sugerencias, opiniones o deseos en cuanto al proceso de registro.

Cuestionario

1. ¿En tiempos prepandemia del COVID 19, realizaba usted prácticas con los estudiantes en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación?

Sí, antes de la suspensión de las prácticas presenciales, hacía uso del laboratorio.

2. ¿En tiempos prepandemia del COVID 19, como era el proceso de registro de asistencia de los estudiantes en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación?

El encargado del laboratorio entregaba un formulario de registro en el cual cada uno de los estudiantes anotaban su nombre, la hora de entrada, hora de salida y su firma, de igual forma como docente colocaba los datos personales y el tema que se iba a impartir.

3. ¿Planea usted el próximo periodo académico retornar a las prácticas presenciales en el laboratorio de Software?

Sí, se encuentra establecido en un Plan de Retorno a las prácticas presenciales en los laboratorios.

4. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, considera usted que si se continúa llevando a cabo el proceso de registro de asistencia del mismo modo que se hacía antes de la pandemia, pudiera ser un foco infeccioso en temas de Coronavirus?

Sí, podría ser un foco infeccioso, sabemos que al mínimo descuido este virus se puede propagar entre las personas por más seguridad que se tenga, a pesar de que el Plan de Retorno a las prácticas presenciales, recomienda el distanciamiento social y el uso del alcohol, en ocasiones se nos pasa por alto y descuidamos algunas fuentes de contagio.

5. ¿Tiene usted alguna sugerencia o recomendación a fin de evitar el contacto físico de los estudiantes con los instrumentos de registro de asistencia en el laboratorio de Software?

Sería bueno que exista una alternativa de registro, pudiera ser un sistema de reconocimiento facial colocado en la entrada o dentro del laboratorio que además de reconocer las personas tenga un sensor de temperatura.

CONSTANCIA DE ENTREVISTA



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES
CARRERA DE INGENIERIA EN SISTEMAS



Carrera de Ingeniería en
Sistemas / Computación

Constancia de Entrevista

Fecha de Entrevista:	27 de agosto 2021
Hora de Entrevista:	16h00
Entrevistador:	Dayanna Alvarado Castillo
Canal de Comunicación:	Reunión virtual (meet.jit.si)

Entrevistado:

Nombre	Cargo	Firma
Wilman Chamba Zargocín	Docente	 Firmado electrónicamente por: WILMAN PATRICIO CHAMBA ZARAGOCIN

Ver documento original en el enlace. [Constancia de entrevista](#)

Anexo 2. Entrevista a técnico encargado del laboratorio de software.



UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES



Carrera de Ingeniería en
Sistemas / Computación

CARRERA DE INGENIERIA EN SISTEMAS

Nombre:	Ing. Luis Darío Sinche Técnico encargado del laboratorio de Software.
Fecha de entrevista:	21 enero 2022

Objetivo de la entrevista.

La presente entrevista se realiza con finalidad de corroborar el uso del laboratorio de software de CIS/C UNL, además de captar información acerca del proceso actual que se lleva a cabo para el registro de asistencia de los estudiantes en mencionado laboratorio, así como también, tomando en cuenta la actual crisis sanitaria por Covid-19, obtener sugerencias, opiniones o deseos en cuanto al proceso de registro.

Cuestionario

1. ¿En tiempos prepandemia del COVID 19, como era el proceso de registro de asistencia de los estudiantes en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación?

El docente y los estudiantes ingresaban al laboratorio, se le entregaba al docente la hoja de registro de estudiantes, en donde cada estudiante iba llenando su nombre, su número de cedula, el número de computador que utilizará y su firma, a fin de tener registro de los estudiantes que utilizaron el laboratorio en un determinado horario. Cuando ya todos los estudiantes se registraban, el docente colocaba su firma al final de la hoja.

2. ¿Qué lapso de tiempo considera usted que se ocupa en el registro de asistencia de los estudiantes?

Depende el número de estudiantes, si son pocos, alrededor de unos cinco minutos, si son bastantes se toman de quince a veinte minutos.

3. ¿Cómo se registra algún daño provocado por los estudiantes en los equipos del laboratorio?

El docente reporta el daño y se lo describe en la casilla de observaciones presente en el formulario de registro de asistencia de los estudiantes, en tal caso, se coloca el nombre del estudiante causante del daño y el equipo afectado.

4. ¿Tiene usted una planificación de asignación de horarios? ¿Esta planificación contiene el nombre del docente, la materia y la práctica que se va a impartir?

Se realiza la planificación con los directores de cada carrera a fin de que no existan cruces de horario, en la planificación constan las materias y el horario en el que va a ocupar. En cuanto a las temáticas que se van a impartir, es el docente quien lleva una planificación y antes de realizar la práctica o asistir a la clase, da a conocer la temática para su respectivo registro.

5. ¿Ha tenido algún tipo de inconveniente en el almacenaje de los registros?

El problema es que algunos estudiantes se olvidan de firmar y hay reclamos de que, si han asistido, además, en ocasiones el docente se olvida de firmar o de colocar el nombre de la práctica y toca estarlo buscando a fin de recopilar esos datos. Por otro lado, se corre el riesgo de que los archivos, al ser material físico, se puedan destruir.

6. ¿Entrega usted a alguien más los registros recopilados?

Los registros se almacenan en archivos a fin de poder hacer un informe cuando el decanato lo requiera, se suelen pedir informes de cuantas prácticas se hicieron en el ciclo académico, cual fue el docente o que materia fue la que más utilizó el laboratorio, el máximo y mínimo de estudiantes que acudieron, esto con finalidad de evaluar cómo se está trabajando en los laboratorios, para construir estos informes se requiere ir revisando y contabilizando manualmente hoja por hoja e ir colocando los datos en una hoja de Excel.

7. ¿Alguna vez ha experimentado que los alumnos ingresen al laboratorio en horarios que no les corresponden?

No se ha tenido ese inconveniente hasta el momento, dado que es el docente quien ingresa primero y posteriormente el grupo de alumnos a los que les corresponde el uso del laboratorio.

8. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, considera usted que si se continúa llevando a cabo el proceso de registro de asistencia del mismo modo que se hacía antes de la pandemia, pudiera ser un foco infeccioso en temas de Coronavirus?

Sí, si todos los usuarios manipulan el esfero o la misma hoja, se puede llegar a algún tipo de contagio, lo más factible es evitar cualquier tipo de contacto con los materiales de registro. Es por ello que, a los estudiantes que realizan las prácticas durante la pandemia no se les está registrando la asistencia a fin de que no tengan contacto con los materiales.

9. ¿Tiene usted alguna sugerencia o recomendación a fin de evitar el contacto físico de los estudiantes con los instrumentos de registro de asistencia en el laboratorio de Software?

El registro de asistencia debería hacerse de una forma en la que el estudiante no tenga contacto con el material, sería factible hacer el registro por medio de visión artificial que detecte el rostro, en algunas empresas de Loja están llevando este mecanismo de registro, sería una forma óptima a fin de evitar que se pueda propagar el virus dentro de la universidad. Además, este mecanismo disminuiría el tiempo de registro de asistencia y si se contara con un software de administración, se facilitaría en gran medida la construcción de los informes del uso del laboratorio.

CONSTANCIA DE ENTREVISTA



UNIVERSIDAD
NACIONAL
DE LOJA



Centro de Investigación en
Sistemas / Comunicación

CIS/C-UNL

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Ingeniería en Sistemas

Constancia de Entrevista

<i>Fecha de Entrevista:</i>	21/01/2022
<i>Hora de Entrevista:</i>	15:30
<i>Entrevistador:</i>	<i>Dayanna Alvarado Castillo</i>
<i>Canal de Comunicación:</i>	<i>Reunión virtual</i>

Entrevistado:

Nombre	Cargo	Firma
Ing. Luis Dario Sinche	Técnico Encargado del laboratorio de Software de CIS-C/UNL	 Scanned and verified by LUIS DARIO SINCHE

Ver documento original en el enlace. [Constancia de Entrevista.](#)

Anexo 3. Formato actual de registro de asistencia

UNIVERSIDAD NACIONAL DE LOJA FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES CARRERA: INGENIERIA EN SISTEMAS REGISTRO DE ESTUDIANTES QUE REALIZAN PRÁCTICAS DE LABORATORIO <i>LABORATORIO DE DESARROLLO DE SOFTWARE</i>							
Módulo/Ciclo		Paralelo:		Docente:			
Fecha:		Asignatura:				PERIODO ACADÉMICO: OCTUBRE 2019 - MARZO 2020	
TEMA DE LA PRÁCTICA:							Nro. PRACTICA
N°	ALUMNO	PREPARATORIO		Nº PC	HORARIO	Nro. CÉDULA	FIRMA
		SI	NO				
1							
2							
3							
4							
5							
6							
FIRMA DOCENTE							
OBSERVACIONES							

Anexo 4. Encuesta a estudiantes

Encuesta realizada mediante la herramienta Google Forms, las preguntas se detallan a continuación:

Encuesta CIS-C/UNL

Estimado estudiante sírvase de la manera más comedida responder la siguiente encuesta, el objetivo de la presente, es corroborar el uso del laboratorio de software de CIS/C UNL, además de captar información acerca del proceso actual que se lleva a cabo para el registro de asistencia en mencionado laboratorio, así como también, tomando en cuenta la actual crisis sanitaria por Covid-19, obtener sugerencias, opiniones o deseos en cuanto al proceso de registro. Sus respuestas serán tomadas en cuenta en la investigación acerca del registro de asistencia al laboratorio de Software de CIS/C UNL.

Indicaciones: Lea detenidamente cada ítem y conteste con toda seriedad.

La contestación no le tomará más de 3 minutos.

dayanna.alvarado@unl.edu.ec [Cambiar de cuenta](#)



Tu correo se registrará cuando envíes este formulario

*Obligatorio

1. ¿En tiempos prepandemia del COVID 19, realizaba usted prácticas en el laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación? *

- Si
- No

2. ¿En tiempos prepandemia del COVID 19, cómo registraba usted su ingreso al laboratorio de Software de CIS/UNL? *

- El encargado del laboratorio se acercaba a correr lista.
- El encargado del laboratorio acercaba un formulario de registro donde usted y sus compañeros firmaban manualmente.
- No registraba asistencia al laboratorio de Software.

3. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, considera usted que si se continúa llevando a cabo el proceso de registro de asistencia del mismo modo que se hacía antes de la pandemia, en el retorno a las prácticas presenciales pudiera ser un foco infeccioso en temas de Coronavirus? *

- Sí
- No
- En parte

4. ¿Tomando en cuenta la crisis sanitaria por la que atravesamos, con cuál de los siguientes métodos se sentiría ud más cómodo, al momento de registrar su asistencia en el laboratorio de Software de CIS-C/UNL? *

- Método tradicional (firma manual)
- Huella dactilar
- Reconocimiento facial
- Otro: _____

Enviar

Borrar formulario

Fuente: Encuesta estructurada

Nota: Obtenido de Google Forms

Estudiantes participantes de la encuesta

De los participantes en la encuesta se obtuvieron los correos institucionales, mismos que se detallan a continuación:

Preguntas Respuestas **47** Configuración

47 respuestas

No se aceptan más respuestas

Mensaje para los encuestados

Ya no se aceptan respuestas en este formulario

Resumen Pregunta Individual

Usuarios que han respondido

Correo electrónico

- .montano@unl.edu.ec
- artinez@unl.edu.ec
- ivera@unl.edu.ec
- atino@unl.edu.ec
- acas@unl.edu.ec
- menez@unl.edu.ec
- radeneira@unl.edu.ec
- hora@unl.edu.ec
- encalada@unl.edu.ec
- .guzman@unl.edu.ec
- i.guazha@unl.edu.ec
- iontano@unl.edu.ec
- ilva@unl.edu.ec

jaramillo@unl.edu.ec
rdo.v.paredes@unl.edu.ec
or.rojas@unl.edu.ec
artuchem@unl.edu.ec
l.pasaca@unl.edu.ec
naira.perez@unl.edu.ec
jr.sanmartin@unl.edu.ec
yalang@unl.edu.ec
tian.e.medina@unl.edu.ec
ardo.leon@unl.edu.ec
ls.canar@unl.edu.ec
vasquez@unl.edu.ec
son.quizhpe@unl.edu.ec
ny.torres@unl.edu.ec
la.castro@unl.edu.ec
a.conde@unl.edu.ec
sco.c.agreda@unl.edu.ec
ole@unl.edu.ec
salazar@unl.edu.ec
reuenes@unl.edu.ec
egron@unl.edu.ec
l.campoverde@unl.edu.ec
ortega@unl.edu.ec
is.a.cuenca.c@unl.edu.ec
c.paredes@unl.edu.ec
carrión@unl.edu.ec
rtandazo@unl.edu.ec
a.carrión.o@unl.edu.ec
an.correa@unl.edu.ec
bosc@unl.edu.ec
jndo.j.pezantes@unl.edu.ec
henao@unl.edu.ec

Fuente: Encuesta estructurada

Anexo 5. Validación BPMN proceso actual



**UNIVERSIDAD
NACIONAL
DE LOJA**



Carrera de Ingeniería en Sistemas / Computación

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Validación de Diagrama de Proceso

Validación del diagrama BPMN del proceso actual del registro de asistencia en el laboratorio de Software de la Carrera de Ingeniería en Sistemas-Computación de la Universidad Nacional de Loja.

Fecha de entrega para validación:	<i>14 de diciembre 2021</i>
Aspirante:	<i>Dayanna Alvarado Castillo</i>
Tema de Tesis:	Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja

Validador:

Nombre	Cargo	Firma
Wilman Chamba Zargocin	Docente CIS-C UNL.	 <small>Documento digitalmente firmado WILMAN PATRICIO CHAMBA ZARGOCIN</small>

Ver documento original en el enlace. [Validación de diagrama de procesos](#)

Anexo 6. Especificación de requerimientos de software

Véase [Documento de especificación de requerimientos de software](#)

Anexo 7. Diagramas de flujo del prototipo

En esta sección, se presenta los diagramas de flujo que detallan la secuencia que realiza cada función del prototipo.

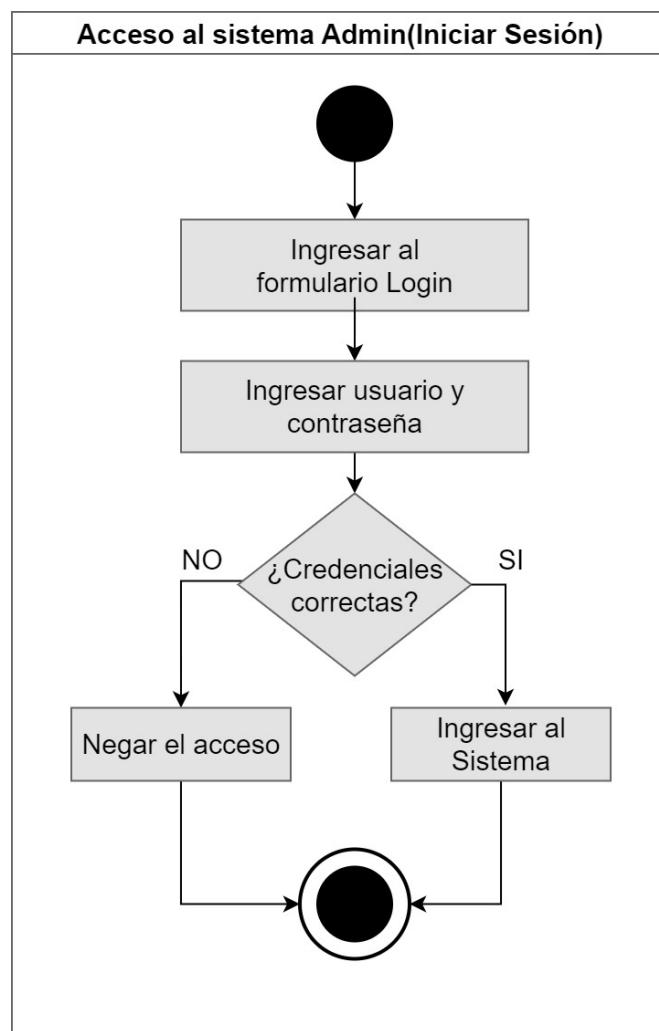


Figura 67. Diagrama de flujo: Acceso al sistema

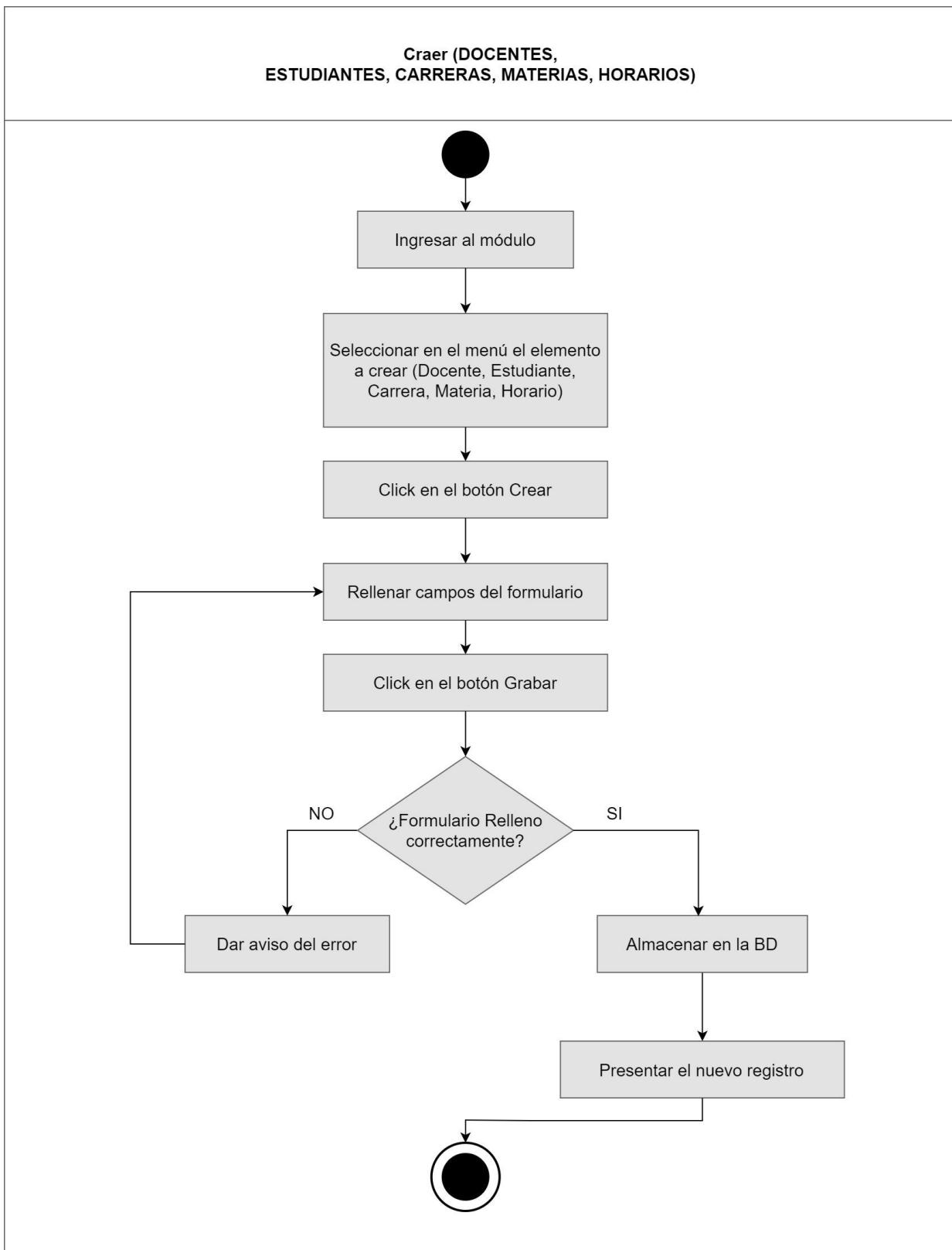


Figura 68. Diagrama de flujo: Crear Registro

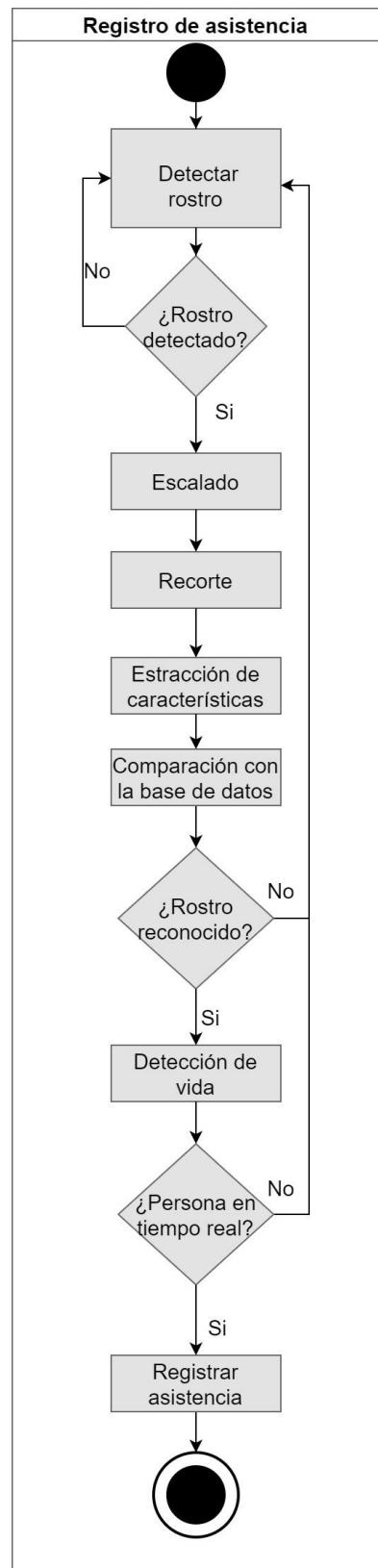


Figura 69. Diagrama de flujo: Registro de Asistencia

**Listar (DOCENTES,
ESTUDIANTES, CARRERAS, MATERIAS, HORARIOS)**

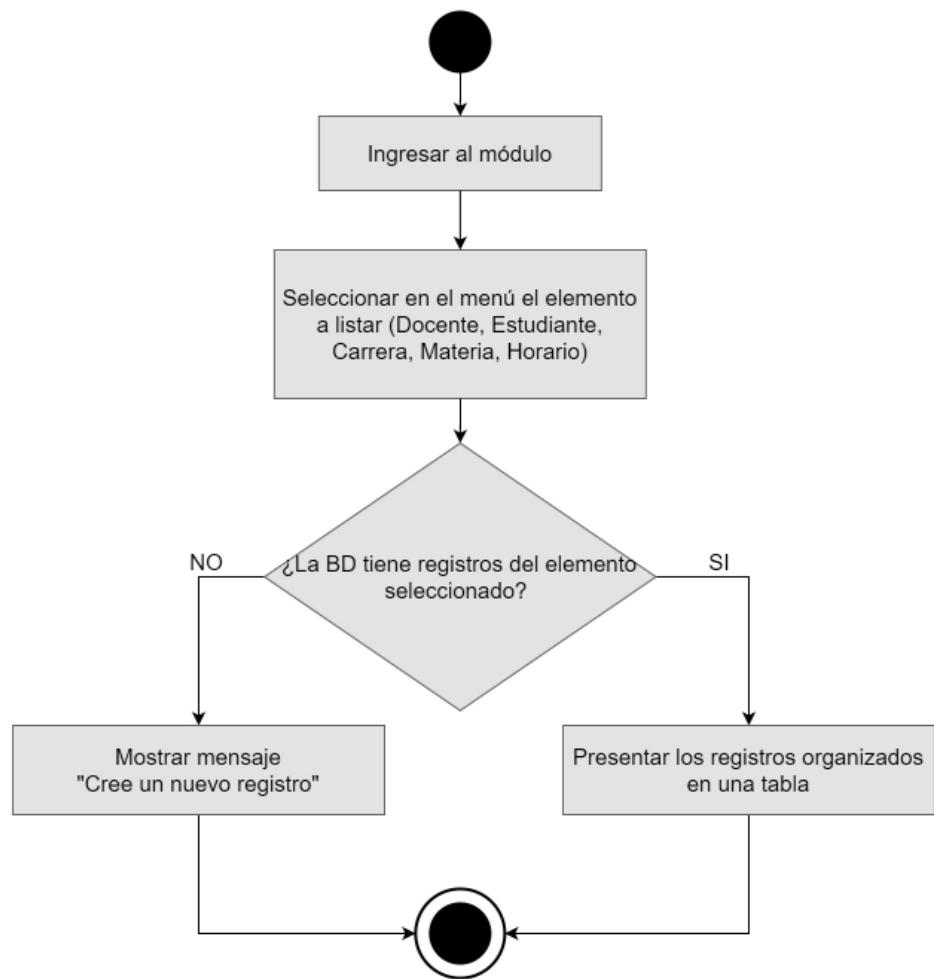


Figura 70. Diagrama de flujo: Listar registros

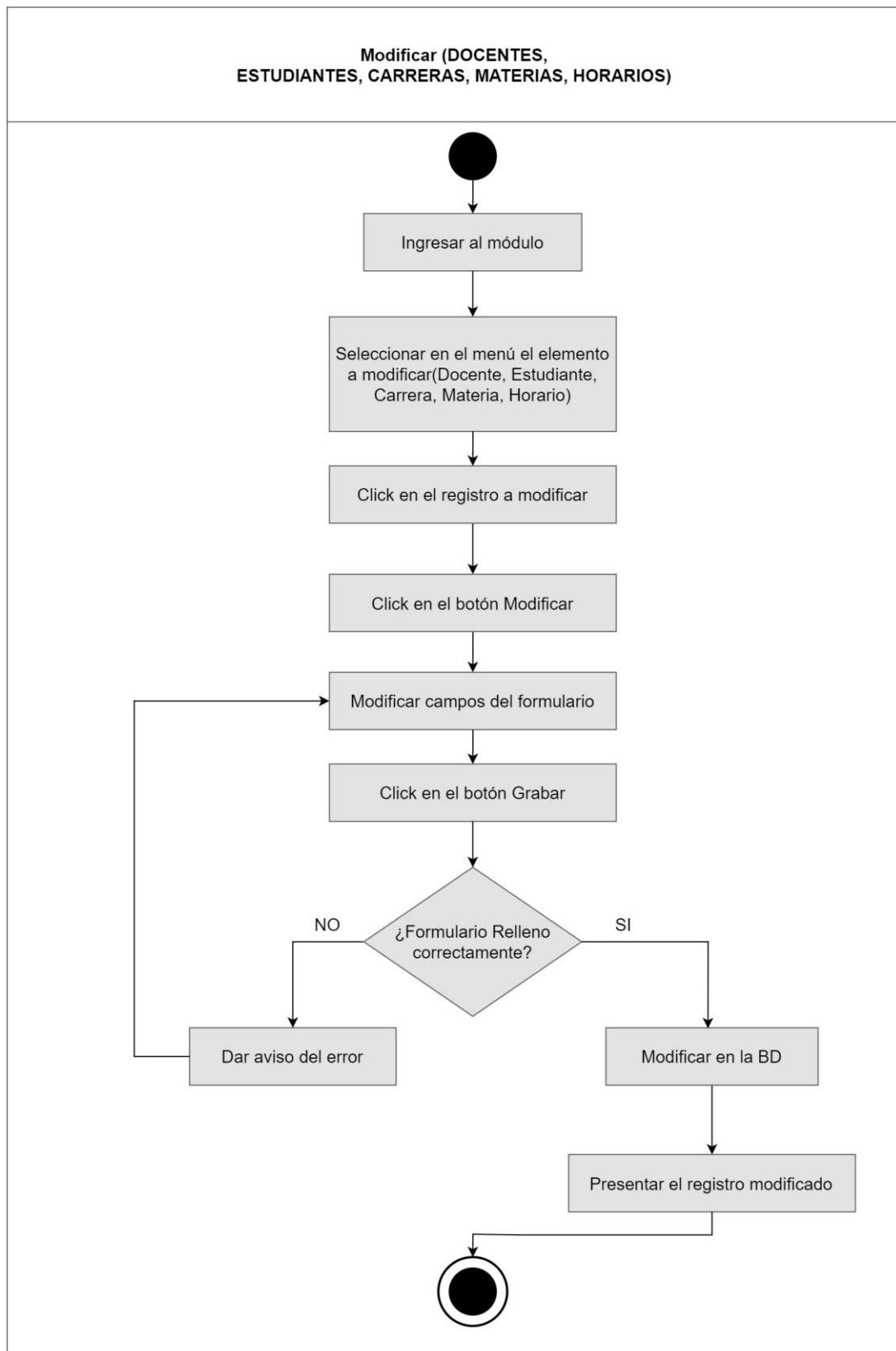


Figura 71. Diagrama de flujo: Modificar registro

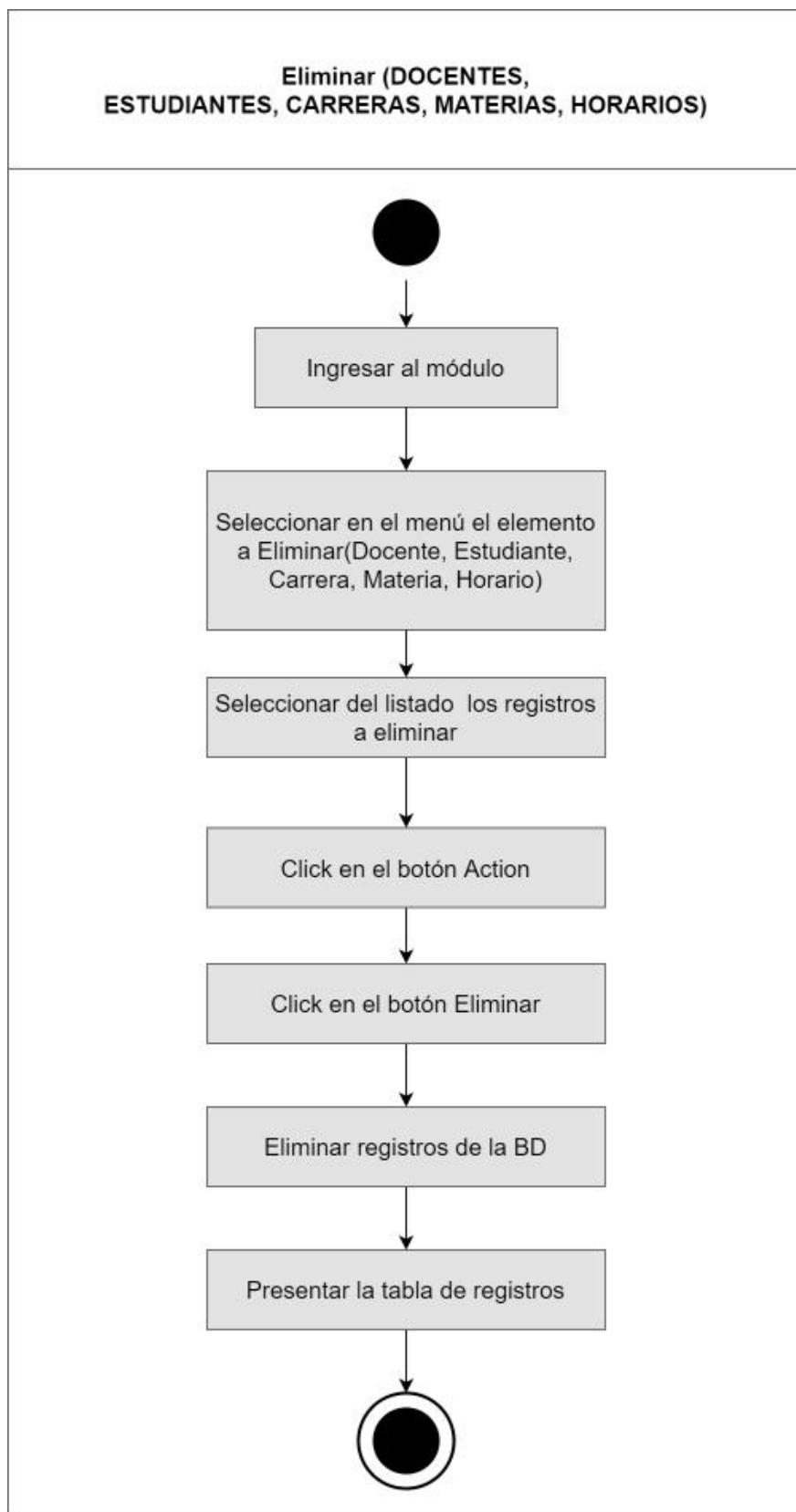


Figura 72. Diagrama de flujo: Eliminar registro

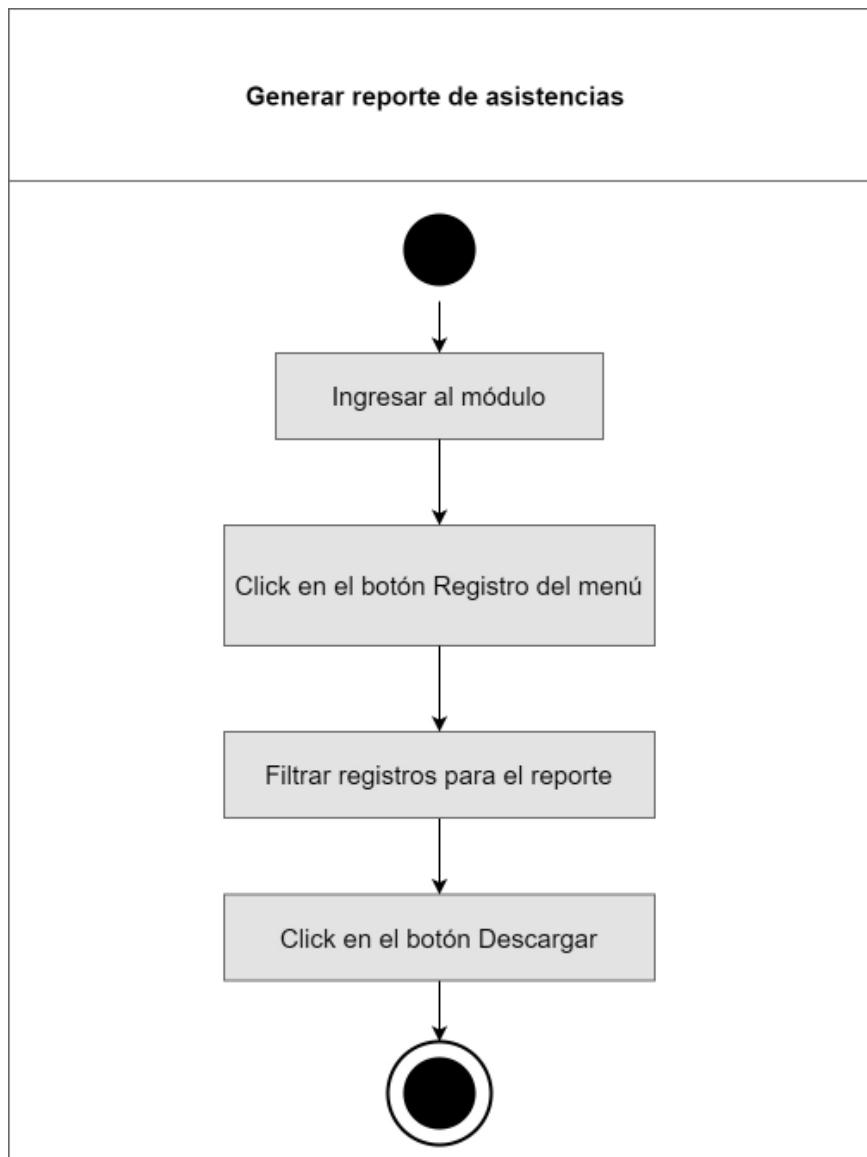


Figura 73. Diagrama de flujo: Generar reporte de asistencias

Anexo 8. Interfaces de módulo de reconocimiento facial con detección de vida para el registro de asistencia.

ESTUDIANTE

Figura 74. Crear Estudiante

Figura 75. Editar Estudiante

<input type="checkbox"/>	Identific...	Nombres	Apellidos...	Ciclo	Paralel...	Dirección d...	Carrera	Materia	Docente
<input type="checkbox"/>	1900692...	Jennifer	Jazmín	9		jennifer.jara...	Ingenieria en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1106067...	Jhandry Ramiro	Lapo	9	A	jhandry.lapo...	Ingenieria en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1104414...	Luis Javier	Paz	9	A	luis.j.paz@u...	Ingenieria en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1105116...	Geovanny Samael	Poma	9	A	geovanny.po...	Ingenieria en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	1150256...	Luis Homero	Bravo	9	A	luis.h.bravo...	Ingenieria en sistemas	Compiladores	Oscar Cumbicus

Figura 76. Listar Estudiante

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Usuarios

Estudiante / 1150034401

	Editar	Crear	Acción ▾
Identificación	1150034401		
Nombres	Dayanna	Paralelo	10
Apellidos	Alvarado	Carrera	A
Dirección de Correo	dayannaalvarado99@uni.edu.ec	Materia	Ingenieria en sistemas
		Docente	Compiladores
			Oscar Cumbicus
Foto			

Figura 77. Eliminar Estudiante

DOCENTE

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Usuarios

Docentes / Nuevo

	Guardar	Descartar
Identificación	0000	Foto
Nombres	ej. Juan Luis	
Dirección de Correo		
Carrera		
Estudiantes	<input type="button" value="Id..."/> <input type="button" value="N..."/> <input type="button" value="A..."/> <input type="button" value="Ci..."/> <input type="button" value="P..."/> <input type="button" value="Di..."/> <input type="button" value="C..."/> <input type="button" value="M..."/> <input type="button" value="D..."/> Agregar línea	

Figura 78. Crear Docente

Identificación	Nombres	Apellidos...	Ciclo	Paralelo...	Dirección de Correo	Carrera	Materia...	Docente
1150304135	Vladimir Abelardo				vladimir.celi@unl.edu.ec	Ingeniería en sistem...	Wilman Cham...	
1104750508	Verónica Estefanía				veronica.placencia@unl.edu...	Ingeniería en sistem...	Wilman Cham...	
1150006979	Gilson Orlando				gilson.quezada@unl.edu.ec	Ingeniería en sistem...	Wilman Cham...	

Figura 79. Editar Docente

Figura 80. Eliminar Docente

<input type="checkbox"/>	Identificaci...	Nombres y Apellidos	Dirección de Correo	Rol	Carrera	Estudiantes
<input type="checkbox"/>	000000001	Oscar Cumbicus	oscar.cumbicus@unl.edu.ec	Docente	Ingeniería en sistemas	13 registros
<input type="checkbox"/>	000000002	Wilman Chamba	wilman.chamba@unl.edu.ec	Docente	Ingeniería en sistemas	12 registros
<input type="checkbox"/>	000000003	José Guamán	jose.guaman@unl.edu.ec	Docente	Ingeniería en sistemas	14 registros

Figura 81. Listar Docente

CARRERA



Figura 82. Crear Carrera

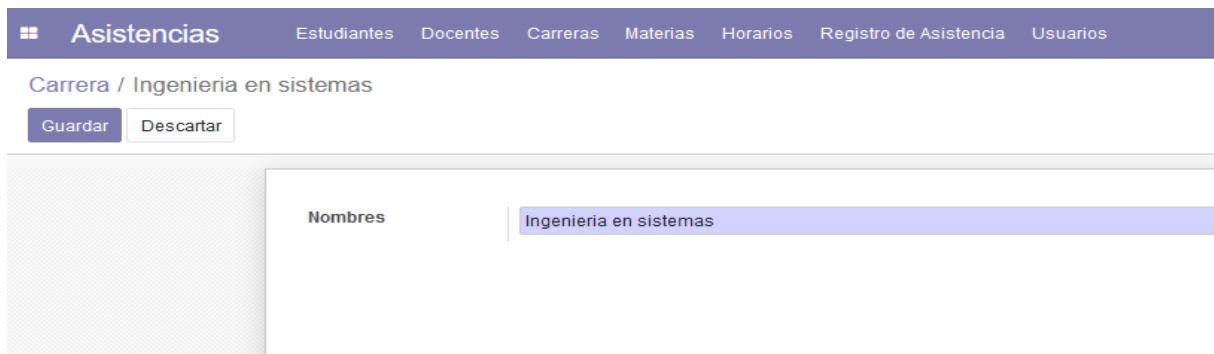


Figura 83. Editar Carrera



Figura 84. Eliminar Carrera

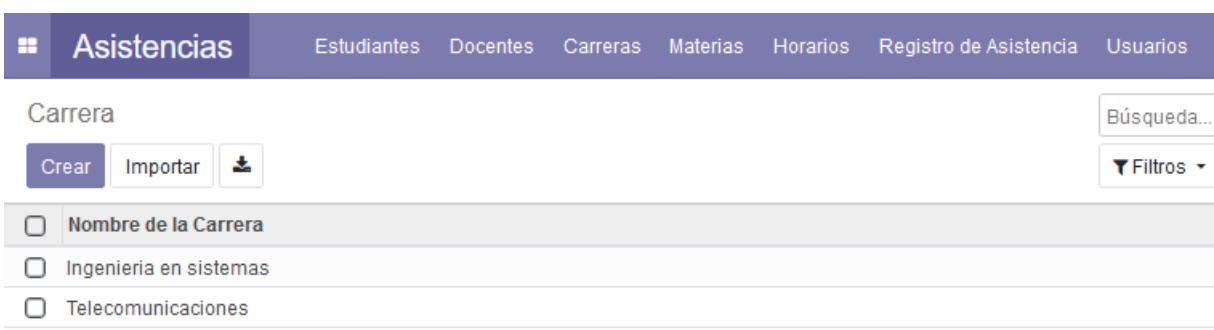


Figura 85. Listar Carrera

MATERIA

This screenshot shows the 'Crear Materia' (Create Subject) page. At the top, there is a navigation bar with links for Asistencias, Estudiantes, Docentes, Carreras, Materias, Horarios, Registro de Asistencia, and Usuarios. On the far right, there is a user icon labeled 'Administrator (tesis)'. Below the navigation bar, the page title 'Materia / Nuevo' is displayed. There are two buttons at the top left: 'Guardar' (Save) and 'Descartar' (Discard). The main form has two fields: 'Nombre de la Materia' (Subject Name) and 'Carrera a la que pertenece' (Belongs to the Career). The 'Nombre de la Materia' field contains 'Programación 2'. The 'Carrera a la que pertenece' dropdown menu is open, showing 'Ingeniería en sistemas' as the selected option, with other options like 'Telecomunicaciones' and a 'Crear y Editar...' link.

Figura 86. Crear Materia

This screenshot shows the 'Editar Materia' (Edit Subject) page. The top navigation bar is identical to Figure 86. The page title is 'Materia / Programación 2'. The main form displays the subject details: 'Nombre de la Materia' is 'Programación 2' and 'Carrera a la que pertenece' is 'Ingeniería en Sistemas'. On the right side of the page, there is a small preview window showing the same information. At the bottom right, there are navigation icons for '1/1' and arrows.

Figura 87. Editar Materia

This screenshot shows the 'Eliminar Materia' (Delete Subject) page. The top navigation bar is identical to the previous figures. The page title is 'Materia / Compiladores'. The main form displays the subject details: 'Nombre de la Materia' is 'Compiladores' and 'Carrera a la que pertenece' is 'Ingeniería en sistemas'. To the right, there is a 'Acción' (Action) button with a dropdown menu. The dropdown menu is open, showing 'Suprimir' (Delete) and 'Duplicar' (Duplicate) as options.

Figura 88. Eliminar Materia

Asistencias		Estudiantes	Docentes	Carreras	Materias	Horarios	Registro de Asistencia	Usuarios
Materia								
Crear	Importar							
<input type="checkbox"/>	Nombre de la Materia	Carrera a la que pertenece						
<input type="checkbox"/>	Compiladores	Ingenieria en sistemas						
<input type="checkbox"/>	Base de datos	Ingenieria en sistemas						

Figura 89. Listar Materia

HORARIOS

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Administrador

Horario / Nuevo

[Grabar](#) [Descartar](#)

Hora de inicio	<input type="text"/>
Hora de Fin	<input type="text"/>
Día al que pertenece	<input type="text"/>
Ciclo	<input type="text"/>
Carrera	<input type="text"/>
Materia	<input type="text"/>
Docente	<input type="text"/>

Figura 90. Crear Horario

Asistencias Estudiantes Docentes Carreras Materias Horarios Registro de Asistencia Administrador

Horario / asistencias.horario,1

[Grabar](#) [Descartar](#)

1 / 1 < >

Hora de inicio	<input type="text"/> 10:30
Hora de Fin	<input type="text"/> 13:30
Día al que pertenece	<input type="text"/> Martes
Ciclo	<input type="text"/> Cuarto
Carrera	<input type="text"/> Ingenieria en Sistemas
Materia	<input type="text"/> Programación 2
Docente	<input type="text"/> Manuela

Figura 91. Editar Horario

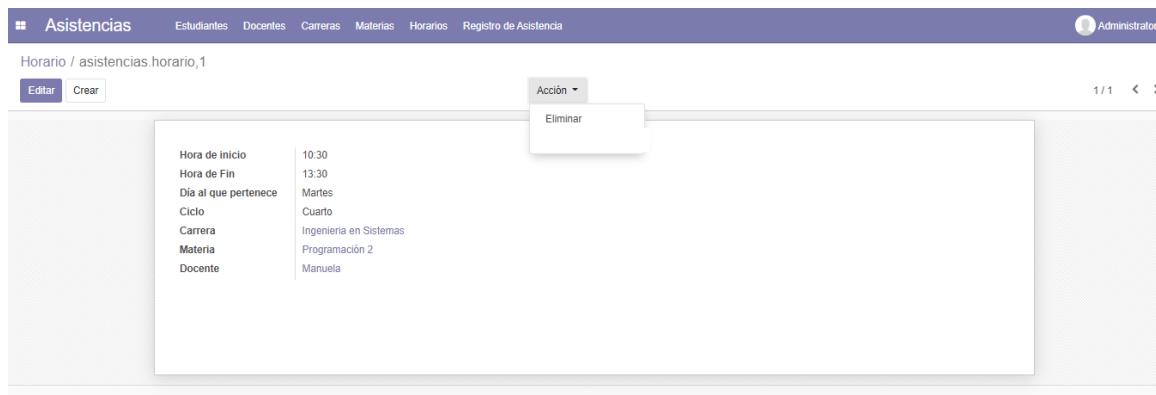


Figura 92. Eliminar Horario

Horario							
	Hora de inicio	Hora de Fin	Día al que pertenece	Ciclo	Carrera	Materia	Docente
<input type="checkbox"/>	7:30	9:30	Miercoles	Séptimo	Ingenieria en sistemas	Compiladores	Oscar Cumbicus
<input type="checkbox"/>	9:30	11:30	Lunes	Cuarto	Ingenieria en sistemas	Base de datos	Wilman Chamba

Figura 93. Listar Horario

REGISTROS

Registro							
	Laboratorio	Fecha	Nombres Y Apellidos	Identific...	Rol	Ciclo	Paralelo
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:11:56	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:18:44	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:45:48	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:51:22	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:53:11	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:58:22	Carmen Azucena False	110608...	Estudiante		
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 13:58:24	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>	Laboratorio de Software	01/05/2023 14:11:57	Dayanna Alvarado	115003...	Estudiante	10	A
<input type="checkbox"/>							

Figura 94. Listar registros.

Anexo 9. Encuesta de recopilación de datos de participantes para pruebas de los modelos

Encuesta realizada mediante la herramienta Google Forms, las preguntas se detallan a continuación:

Prototipo de reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de software de CIS/C UNL.

Estimado estudiante.

La presente es una encuesta necesaria para la recopilación de datos de los estudiantes que acuden a las prácticas presenciales en el laboratorio de software de CIS/C UNL. Los datos que aquí se proporcionen, serán utilizados con el fin de realizar pruebas al modelo de reconocimiento facial con detección de vida para el registro de asistencias que se desarrolló como proyecto de titulación. Los datos obtenidos serán utilizados únicamente con fines académicos y los resultados de las pruebas a realizarse serán divulgados en la documentación del trabajo de titulación.

Por favor, apelando a su espíritu de colaboración, solicito se rellene el siguiente formulario.

Correo *

Correo válido

Este formulario registra los correos. [Cambiar configuración](#)

Nombres y Apellidos completos

*

Texto de respuesta corta

Cédula de Identidad o Pasaporte *

Texto de respuesta corta

Correo Institucional *

Texto de respuesta corta

Ciclo al que pertenece *

- 6to ciclo
- 7mo ciclo
- 8vo ciclo
- 9no ciclo
- 10mo ciclo

Paralelo al que pertenece *

- A
- B

Fotografia de su rostro.

La *

fotografia debe contener buena iluminación, el rostro de la fotografía no debe contener accesorios (lentes, gafas, mascarilla). El rostro debe tener proyección hacia el frente (no de perfil). Evitar la utilización de filtros. La fotografía debe tener como máximo de antiguedad 1 año.

 Añadir archivo

 Ver carpeta

¿Está usted de acuerdo con que los datos que proporcionó sean utilizados con fines *
académicos?

- Si
- No

Fuente: Encuesta estructurada

Nota: Obtenido de Google Forms

Estudiantes participantes de la encuesta

De los participantes en la encuesta se obtuvieron los correos institucionales, mismos que se detallan a continuación:

Correo electrónico

iny.poma@unl.edu.ec

:a.placencia@unl.edu.ec

al.rios@unl.edu.ec

paredes@unl.edu.ec

ortega@unl.edu.ec

jo.tuqueres@unl.edu.ec

ile@unl.edu.ec

iyora@unl.edu.ec

l.rojas@unl.edu.ec

ir.celi@unl.edu.ec

n.quizhpe@unl.edu.ec

y.santorum@unl.edu.ec

ps@unl.edu.ec

m.jaramillo@unl.edu.ec

az@unl.edu.ec

oravo@unl.edu.ec

.....salazar@unl.edu.ec

jam@unl.edu.ec

.mendoza@unl.edu.ec

nacas@unl.edu.ec

l.a.torres@unl.edu.ec

y.lapo@unl.edu.ec

r.jaramillo@unl.edu.ec

:alva@unl.edu.ec

j@unl.edu.ec

i.correa@unl.edu.ec
.quezada@unl.edu.ec
iah@unl.edu.ec
ianangomez@unl.edu.ec
gutierrez@unl.edu.ec
n.toledo@unl.edu.ec
a.jimenez@unl.edu.ec
zalesch@unl.edu.ec
r.castillo@unl.edu.ec
ny.torres@unl.edu.ec
campoverde@unl.edu.ec
requenes@unl.edu.ec
na.alvarado@unl.edu.ec
ny.villalta@unl.edu.ec
es@unl.edu.ec

Fuente: Encuesta estructurada

Nota: Obtenido de Google Forms

Anexo 10. Validación del cumplimiento de los requerimientos de software.

Constancia de Validación

Fecha de validación: 2/08/2022

Nombre	Cargo	Firma
Luis Darío Sinche Cueva	Técnico encargado del laboratorio de software de CIS/C UNL.	 Firmado electrónicamente por: LUIS DARIO SINCHE

Véase documento original en el enlace. [Validación del cumplimiento de Requisitos de Software](#)

Anexo 11. Encuesta de satisfacción a estudiantes del prototipo propuesto

Encuesta realizada mediante la herramienta Google Forms, las preguntas se detallan a continuación:

Preguntas Respuestas 40 Configuración

Encuesta de aceptación del prototipo de reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de software de CIS/C UNL.

Estimado estudiante, la presente tiene la finalidad de captar su opinión acerca del prototipo de reconocimiento facial con detección de vida para el registro de asistencia en el cual usted participó en la fase de pruebas.

Por favor, responda con toda seriedad.

Este formulario registra automáticamente los correos de los usuarios de Universidad Nacional de Loja. [Cambiar configuración](#)

Correo electrónico institucional *

Texto de respuesta corta

Ciclo al que pertenece *

6to
 7mo
 8vo
 9no
 10mo

Paralelo al que pertenece *

- A
- B

¿Necesitó usted manipular instrumentos de uso comunitario para registrar asistencia con el * prototipo de reconocimiento facial en el cual participó?

- Si
- En parte
- No

¿Cuánto tiempo estima usted que le tomó registrar asistencia con el prototipo de * reconocimiento facial?

- De 5 a 10 segundos
- De 11 a 20 segundos
- De 21 a 30 segundos
- De 31 a 40 segundos
- De 41 a 50 segundos
- De 51 a 60 segundos
- De 1 a 2 minutos
- Más de 2 minutos

¿Le gustaría a usted contar con el método de reconocimiento facial para registrar asistencia en el laboratorio de Software? *

- Si
- En parte
- No

Fuente: Encuesta estructurada

Nota: Obtenido de Google Forms

Estudiantes participantes de la encuesta

De los participantes en la encuesta se obtuvieron los correos institucionales, mismos que se detallan a continuación:

Correo electrónico

iny.poma@unl.edu.ec

:a.placencia@unl.edu.ec

al.rios@unl.edu.ec

paredes@unl.edu.ec

ortega@unl.edu.ec

jo.tuquieres@unl.edu.ec

le@unl.edu.ec

iyora@unl.edu.ec

ir.celi@unl.edu.ec

n.quizhpe@unl.edu.ec

y.santorum@unl.edu.ec

os@unl.edu.ec

m.jaramillo@unl.edu.ec

az@unl.edu.ec

oravo@unl.edu.ec

.....salazar@unl.edu.ec

jam@unl.edu.ec
.mendoza@unl.edu.ec
nacas@unl.edu.ec
l.a.torres@unl.edu.ec
y.lapo@unl.edu.ec
r.jaramillo@unl.edu.ec
:alva@unl.edu.ec
j@unl.edu.ec
i.correa@unl.edu.ec
.quezada@unl.edu.ec
iah@unl.edu.ec
ianangomez@unl.edu.ec
gutierrez@unl.edu.ec
n.toledo@unl.edu.ec
a.jimenez@unl.edu.ec
zalesch@unl.edu.ec
r.castillo@unl.edu.ec
ny.torres@unl.edu.ec
campoverde@unl.edu.ec
requenes@unl.edu.ec
na.alvarado@unl.edu.ec
ny.villalta@unl.edu.ec
es@unl.edu.ec

Fuente: Encuesta estructurada

Nota: Obtenido de Google Forms

Análisis e Interpretación de la encuesta.

De conformidad al Trabajo de Titulación se realizó una encuesta dirigida a los estudiantes en Sistemas/Computación de la Universidad Nacional de Loja quienes participaron en el proceso de pruebas del prototipo planteado, con finalidad de captar su opinión en cuanto al software en cuestión, se obtuvieron 40 respuestas. A continuación, se presenta la tabulación de las preguntas necesarias para rescatar opinión de satisfacción, omitiendo aquellas preguntas sobre datos personales.

Pregunta 1. ¿Necesitó usted manipular instrumentos de uso comunitario para registrar asistencia con el prototipo de reconocimiento facial en el cual participó?

Tabla 23. Manipulación de instrumentos de uso comunitario

Alternativas	Frecuencia	Porcentaje
SI	40	100%
No	0	0%
En parte	0	0%
TOTAL	40	100%

Fuente: Encuesta a estudiantes

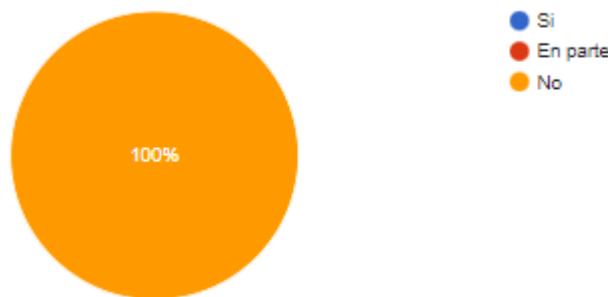


Figura 95. Manipulación de instrumentos de uso comunitario

Obtenido de *Encuesta estructurada Google Forms*

Interpretación

Se puede evidenciar en el diagrama que el 100% de los estudiantes no necesitaron manipular instrumentos de uso comunitario para registrar asistencia con el software propuesto.

Pregunta N°2. ¿Cuánto tiempo estima usted que le tomó registrar asistencia con el prototipo de reconocimiento facial?

Tabla 24. Tiempo estimado de registro de asistencia

Alternativas	Frecuencia	Porcentaje
De 5 a 10 segundos	27	67.5%
De 11 a 20 segundos	10	25%
De 21 a 30 segundos	0	0%
De 31 a 40 segundos	1	2.5%
De 41 a 50 segundos	1	2.5%
De 51 a 60 segundos	1	2.5%
De 1 a 2 minutos	0	0%
Más de 2 minutos	0	0%
TOTAL	40	100%

Fuente: Encuesta a estudiantes

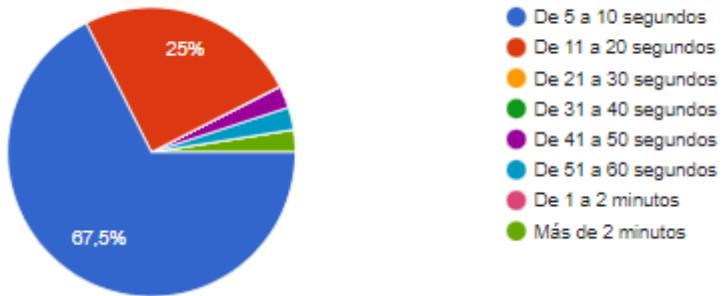


Figura 96. Tiempo estimado de registro de asistencia

Obtenido de Encuesta estructurada Google Forms

Interpretación

Respecto a la Pregunta 2, el 67.5% de los estudiantes encuestados aseguran que el registro de su asistencia con el software propuesto se lleva a cabo en un rango entre 5 y 10 segundos, mientras que el 25% afirman que el tiempo que conlleva es de entre 11 a 20 segundos, además el 7.5% asegura que el software requiere un tiempo de entre 31 segundos a un

minuto. Por tanto, al calcular el promedio entre los valores obtenidos, se tiene como resultado que el tiempo estimado para el registro de asistencia es de 9,2 segundos, perteneciente a la primera alternativa correspondiente a entre 5 y 10 segundos.

Pregunta 3. ¿Le gustaría a usted contar con el método de reconocimiento facial para registrar asistencia en el laboratorio de Software?

Tabla 25. Satisfacción con el software propuesto

Alternativas	Frecuencia	Porcentaje
SI	40	100%
En parte	0	0%
No	0	0%
TOTAL	40	100%

Fuente: Encuesta a estudiantes

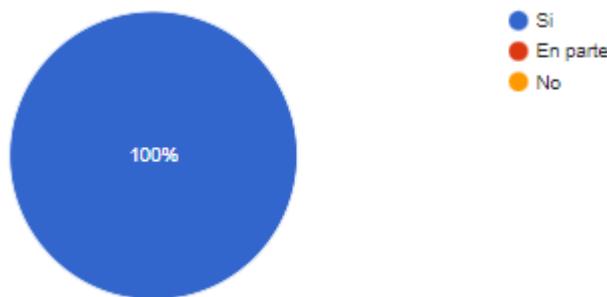


Figura 97. Satisfacción con el software propuesto

Obtenido de Encuesta estructurada Google Forms

Interpretación

Según se ilustra en el diagrama, el 100% de los estudiantes encuestados afirman que si les gustaría contar con el método de reconocimiento facial propuesto para el registro de asistencias en el laboratorio de software.

Anexo 12. Código del prototipo.

Ver código del prototipo propuesto en [Enlace de Github](#)

Anexo 13. Certificado de traducción de resumen al idioma inglés.

CERTIFICACIÓN

Loja, 8 de Mayo del 2023

Yo Karina Magdalena Beltrán Valverde, licenciada en Ciencias de la Educación en la especialidad de Idioma Inglés con registro Nro. 058492, y con título conferido por la Universidad Nacional de Loja, certifico que:

El resumen de la tesis titulada "Prototipo de Reconocimiento facial con detección de vida para el registro de asistencia al laboratorio de Software de la Carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja", de la autoría de Dayanna Magdalla Alvarado Castillo (Cl: 1150034401), estudiante de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, es fiel traducción al idioma inglés por mi persona.

Lo certifico en honor a la verdad y autorizo a la interesada a hacer uso del presente documento como estime conveniente.



Karina Magdalena Beltrán Valverde

LICENCIADA EN IDIOMA INGLÉS

CI.1900724590