



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



**MODELO PARA LA DETECCIÓN DE ANOMALÍAS EN
SECUENCIAS DE VIDEOS DE EXÁMENES EN LÍNEA
MEDIANTE INTELIGENCIA ARTIFICIAL CASO DE ESTUDIO:
UNIVERSIDAD NACIONAL DEL ALTIPLANO**

TESIS

PRESENTADA POR:

MILWART ROSSINI CALIZAYA BOBADILLA

FREDY CALSIN CARI

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO - PERÚ

2023



Reporte de similitud

NOMBRE DEL TRABAJO

MODELO PARA LA DETECCIÓN DE ANOMALÍAS EN SECUENCIAS DE VIDEOS DE EXÁMENES EN LÍNEA MEDIANTE INTELIGENCIA ARTIFICIAL CASO DE ESTUDIO: UNIVERSIDAD NACIONAL DEL ALTIPLANO

AUTOR

MILWART ROSSINI CALIZAYA BOBADILLA, FREDY CALSIN CARI

RECuento DE PALABRAS

15748 Words

RECuento DE CARACTERES

92196 Characters

RECuento DE PÁGINAS

121 Pages

TAMAÑO DEL ARCHIVO

4.6MB

FECHA DE ENTREGA

Nov 17, 2023 8:14 AM GMT-5

FECHA DEL INFORME

Nov 17, 2023 8:16 AM GMT-5


● 12% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos

- 10% Base de datos de Internet
- Base de datos de Crossref
- 9% Base de datos de trabajos entregados
- 3% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Coincidencia baja (menos de 10 palabras)
- Material citado


Dr. Miguel Romitio Acuña Pozo.
ASESOR


Dra. Guina G. Sotomayor Alzamora
INGENIERO DE SISTEMAS
CIP. 91234

Resumen



DEDICATORIA

A Dios, por despertar en nosotros conciencia y regalarnos el libre albedrío en este universo tal vez determinista; a los hombros de gigantes, quienes nos soportan, nos inspiran y son faros en la oscuridad; a aquellos que, con esperanza y pasión, intentan hacer de este un mundo mejor; a quienes que, aun sintiéndose excluidos, desafiando limitaciones y superando carencias, emprenden la travesía de renacer y cambiar su realidad. Profundo respeto y sincera admiración hacia ustedes.

A mis padres, por la esperanza depositada y el apoyo brindado.

Milwart Rossini Calizaya Bobadilla



DEDICATORIA

Este trabajo está dedicado a mi familia por la motivación y apoyo brindado, especialmente a mi hija Arleth Aysel y a mi esposa Erika. También quiero expresar mi agradecimiento a todas las personas cercanas que me apoyaron durante todo el proceso de la tesis.

Fredy Calsin Cari



AGRADECIMIENTOS

En primer lugar, agradecer a Dios por habernos guiado a lo largo de estos años de estudio, agradecemos a la universidad Nacional del altiplano por haber aportado en nuestro desarrollo.

A las autoridades, docentes y compañeros que fueron parte de la E.P. Ingeniería de sistemas de la Universidad Nacional del Altiplano por habernos encaminado en la búsqueda de construir una mejor sociedad.

También agradecemos al Centro Preuniversitario de la Universidad Nacional del Altiplano por habernos permitido realizar nuestra investigación en el simulacro de admisión. Así mismo agradecemos a nuestro asesor el Ing. Miguel Romilio Aceituno Rojo por habernos guiado y orientado en el proceso de investigación.

Milwart Rossini Calizaya Bobadilla

Fredy Calsin Cari



ÍNDICE GENERAL

Pág.

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

ACRÓNIMOS

RESUMEN 17

ABSTRACT..... 18

CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA..... 19

1.2. FORMULACIÓN DEL PROBLEMA 19

1.2.1. Problema general..... 19

1.3. JUSTIFICACIÓN DEL PROBLEMA 20

1.4. OBJETIVO DE LA INVESTIGACIÓN 21

1.4.1. Objetivo general..... 21

1.4.2. Objetivo específico 21

1.5. HIPÓTESIS DE LA INVESTIGACIÓN 21

1.5.1. Hipótesis general..... 21

CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES 22



2.2. MARCO TEÓRICO	27
2.2.1. Computación en la nube.....	27
2.2.1.1. Amazon Web Services (AWS).....	28
2.2.1.2. Amazon Elastic Compute Cloud (Amazon EC2).....	28
2.2.1.3. Amazon Simple Storage Service (Amazon S3)	29
2.2.1.4. Amazon Timestream	29
2.2.1.5. AWS Lambda.....	30
2.2.1.6. API Gateway	30
2.2.2. Inteligencia artificial	31
2.2.2.1 Autoencoders.....	32
2.2.2.2. Aprendizaje automático.....	33
2.2.2.3. Detección de rostros	33
2.2.2.4. Reconocimiento facial.....	34
2.2.2.5. MediaPipe.....	35
2.2.2.6. Anomalía	36
2.2.2.7. Redes LSTM	36
2.2.3. Procesamiento multimedia.....	37
2.2.3.1. Procesamiento de imágenes	37
2.2.3.2. Sustracción de fondo	38
2.2.3.3. Biblioteca Opencv	39
2.2.4. Herramientas de visualización	39
2.2.4.1. Grafana	39
2.2.5. Métricas de error	40
2.2.5.1. MAE.....	40
2.2.5.2. MSE.....	41



2.2.6. Definición de términos básicos	41
2.2.6.1. Python.....	41
2.2.6.2. JavaScript	42
2.2.6.3. Qt Quick	42
2.2.6.4. Paquete Pandas.....	42
2.2.6.5. Librería numpy.....	43
2.2.6.6. Sprint	43
2.2.6.7. Interfaz de programación de aplicaciones (API).....	43
2.2.6.8. Dataframe	44
2.2.6.9. Base de datos relacionales (SQLITE)	44
2.2.6.11. Modelo Vista Controlador (MVC).....	45
2.2.6.12. Ángulos de Euler.....	46
2.2.6.13. Protocolo http.....	47
2.2.6.14. Scrum	48

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. TIPO DE INVESTIGACIÓN	50
3.2. DISEÑO DE INVESTIGACIÓN.....	50
3.3. MÉTRICAS PARA LA EVALUACIÓN DEL MODELO.....	50
3.4. POBLACIÓN Y MUESTRA.....	51
3.4.1. Población.....	51
3.4.2. Muestra.....	51

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS.....	53
-----------------------------	-----------



4.1.1. Construir un dataset de secuencias de video de exámenes en línea.....	53
4.1.1.1. Desarrollo de la aplicación con modelos pre-entrenados y servicios en la nube para la adquisición de datos de exámenes en línea.....	53
4.1.1.2. Características de la aplicación con modelos preentrenados y servicios en la nube para la adquisición de datos de exámenes en línea.	55
4.1.1.3. Módulos de la aplicación.....	58
4.1.1.3.1. Inicio de sesión.....	58
4.1.1.3.2. Datos del estudiante	59
4.1.1.3.3. Control biométrico	60
4.1.1.3.4. Control de identidad.....	61
4.1.1.3.5. Reconocimiento Facial.....	62
4.1.1.3.6. Durante el examen.....	65
4.1.1.3.7. Envío y visualización	68
4.1.1.3.8. Análisis y reportes.....	69
4.1.1.3.9. Algoritmo para el cálculo de notas.....	70
4.1.1.3.10. Resultados del simulacro de examen de Admisión.....	71
4.1.1.4. Dataset de eventos de exámenes en línea.....	72
4.1.1.4.1. Dataset de anomalías en video	74
4.1.1.4.1.1. Dataset de movimiento excesivo.....	74
4.1.1.4.1.2. Dataset de postura de la cabeza.....	75
4.1.1.4.1.3. Dataset de eventos anómalos.....	76
4.1.1.4.1.4. Dataset de eventos normales	78
4.1.1.4.1.5. Construcción del Dataframe.....	79
4.1.2. Entrenar los modelos en la detección de anomalías en secuencias de video mediante inteligencia artificial	80



4.1.2.1. Análisis exploratorio de datos	80
4.1.2.2. Movimientos de la cabeza en los distintos ejes.....	81
4.1.2.3. Preprocesamiento y normalización de datos	84
4.1.2.4. Extracción de características de movimiento de la postura de la cabeza.....	84
4.1.2.5. Modelo	86
4.1.2.6. Entrenamiento de modelos.....	88
4.1.2.6.1. Entrenamiento con LSTM-AUTOENCODER.....	88
4.1.2.6.1. Entrenamiento con AUTOENCODER.....	90
4.1.3. Evaluar y comparar los modelos en la detección de anomalías en secuencias de video en exámenes en línea	91
4.2. DISCUSIÓN	94
V. CONCLUSIONES.....	96
VI. RECOMENDACIONES	98
VII. REFERENCIAS BIBLIOGRÁFICAS.....	99
ANEXOS.....	106

ÁREA: Inteligencia artificial

TEMA: Detección de anomalías

FECHA DE SUSTENTACIÓN: 21 de noviembre del 2023



ÍNDICE DE FIGURAS

	Pág.
Figura 1 Esquema de evaluación virtual.....	24
Figura 2 Modelos de Servicio de la Computación en la Nube y Tipo de Usuarios Asociados	28
Figura 3 Subida y descarga de objetos a S3 via HTTPS.....	29
Figura 4 Servicio AWS Lambda.....	30
Figura 5 Servicio API Gateway	31
Figura 6 Autoencoder	32
Figura 7 Proceso para codificar, comprimir y decodificar.	33
Figura 8 Características Haar.....	34
Figura 9 Proceso para el reconocimiento facial.....	35
Figura 10 Mediapipe hand landmarks.....	36
Figura 11 Estructura de la red LSTM	37
Figura 12 Imagen en escala de grises.	38
Figura 13 Etapas de sustracción de fondo.....	39
Figura 14 Plugin Amazon Timestream en Grafana.	40
Figura 15 Comunicación de la API.....	44
Figura 16 Tres procesos embebidos.....	45
Figura 17 Esquema Modelo Vista Controlador.	46
Figura 18 La notación de ángulo de Tait-Bryan de los ángulos de Euler.....	47
Figura 19 Matriz de rotación para detección de anomalías (Riba Pi, E, 2015).	47
Figura 20 Esquema del protocolo Http	48
Figura 21 Ilustración de un Sprint dentro de un proceso SCRUM con algunas adaptaciones a la metodología.....	49
Figura 22 Matriz de Confusión para detección de anomalías.....	51
Figura 23 Tareas registradas y realizadas en Jira.....	54
Figura 24 Fases contempladas durante el examen de admisión.	54
Figura 25 Lista de aplicaciones restringidas	56
Figura 26 Formato del archivo JSON	57
Figura 27 Interfaz de inicio de sesión en la aplicación.	59
Figura 28 Interfaz de verificación de datos del estudiante.	60
Figura 29 Interfaz de control de identidad en la aplicación.....	61



Figura 30	Interfaz de control de identidad en la aplicación.....	62
Figura 31	Puntos característicos de un rostro para el reconocimiento facial.....	63
Figura 32	Puntos característicos de un rostro que permiten el reconocimiento facial.	64
Figura 33	Reconocimiento facial mediante puntos característicos.....	64
Figura 34	Estructura del archivo JSON para una pregunta del examen	66
Figura 35	Formato JSON de las alternativas válidas.....	67
Figura 36	Preguntas del examen.....	68
Figura 37	Envío y visualización de usuarios en Grafana.....	68
Figura 38	Dashboard con estadísticas propias del examen.....	69
Figura 39	Dashboard con estadísticas del examen.	69
Figura 40	Resultado del examen de estudiantes que asisten en de forma presencial en Juliaca y Puno y de forma virtual en diferentes sedes.....	70
Figura 41	Resultados del examen de simulacro por escuela profesional.....	71
Figura 42	Resultados del examen de simulacro.....	72
Figura 43	Cantidad de datos recolectados durante el simulacro.....	73
Figura 44	Sustracción de fondo en secuencias de video.....	74
Figura 45	Algoritmo para la construcción y almacenamiento de un dataset de señales de video.	75
Figura 46	Esquema del problema PnP, dado un conjunto de puntos 3D y sus proyecciones 2D sobre la imagen, se busca recuperar la pose (R y T).	76
Figura 47	Orientación de la cabeza con los ángulos yaw, pitch y roll.....	77
Figura 48	Imágenes de movimientos hacia arriba, abajo, derecha e izquierda, son considerados anómalos.....	77
Figura 49	Dataset de videos de anomalías en exámenes en línea.....	78
Figura 50	Movimiento de frente considerado normal.....	78
Figura 51	Dataset de eventos normales.	79
Figura 52	Dataframe construido del movimiento de la postura de la cabeza	80
Figura 53	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje X de muestras anómalas.	81
Figura 54	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje X de muestras normales.....	82
Figura 55	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras anómalas.	82



Figura 56	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras normales.....	83
Figura 57	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Z de muestras anómalas.....	83
Figura 58	Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras normales.....	84
Figura 59	Histograma orientado de flujo óptico con 7 bins que describen 360°	85
Figura 60	Dataframe que contiene el nombre, histograma de movimiento y el tipo..	86
Figura 61	AUTOENCODER	87
Figura 62	LSTM-AUTOENCODER	87
Figura 63	Arquitectura del modelo 1.....	89
Figura 64	Entrenamiento del modelo 1.....	89
Figura 65	Arquitectura del modelo	90
Figura 66	Resultados del entrenamiento del Autoencoder.	91
Figura 67	Señal original (color azul) y señal reconstruida (color naranja).....	92
Figura 68	Distribución del Error medio absoluto.	93
Figura 69	Modelo para la detección de anomalías en secuencias de video de exámenes en línea.....	94



ÍNDICE DE TABLAS

	Pág.
Tabla 1 Diseño preexperimental.	50
Tabla 2 Algoritmo para el cálculo de notas.	70
Tabla 3 Cantidad de videos categorizados.	73
Tabla 4 Cantidad total de audios y videos.	73
Tabla 5 Tabla de resultados.....	93



ÍNDICE DE ANEXOS

	Pág.
ANEXO 1 BackLog de las tareas en Jira	106
ANEXO 2 Datos adquiridos por estudiante	106
ANEXO 3 Algoritmo para guardar rostros	107
ANEXO 4 Algoritmo para Monitoreo de la cámara y audio en el examen	107
ANEXO 5 Imágenes de la extracción de características del movimiento de rostro.....	111



ACRÓNIMOS

AWS:	Amazon Web Services
Amazon S3:	Amazon Simple Storage Service
API:	Interfaz de programación de aplicaciones
REST:	La transferencia de estado representacional
HTTP:	HyperText Markup Language
SQL:	Structured Query Language
QML:	Qt Modeling Language
Qt Quick:	Qt User Interface Creation Kit
OpenCV:	Open Computer Vision
MAE:	Mask Autoencoder
MSE:	Mean-squared error
MVC:	Modelo Vista Controlador



RESUMEN

La educación virtual se ha visto impulsada por la pandemia de COVID-19 y, aunque ya era una opción de aprendizaje, su uso se ha vuelto más frecuente. Las universidades no siempre cuentan con la infraestructura necesaria para dar cabida a grandes cantidades de estudiantes que eligen esta modalidad de educación, y la educación virtual puede ayudar a incrementar la cantidad de vacantes ofertadas. Sin embargo, es necesario establecer mecanismos que garanticen una educación de calidad, justa y equitativa. En la actualidad, se utiliza la supervisión humana, que requiere la presencia de un examinador para monitorear visual y acústicamente a los estudiantes durante los exámenes, lo cual resulta costoso y requiere un gran esfuerzo cuando se tienen que evaluar a muchas personas. En este estudio, se propone un modelo de detección de anomalías en los exámenes en línea. Se desarrolló una aplicación de escritorio para recolectar información de la cámara y el micrófono durante el examen simulacro de admisión del centro pre-universitario de la Universidad Nacional del Altiplano. Se recolectaron 180024 clips de video y 115292 clips de audio. La información se procesó y se extrajeron características de movimiento para construir un vector de características. Se desarrollaron, compararon y evaluaron tres modelos, basados en los algoritmos ISOLATIONFOREST, LSTM-AUTOENCODER y AUTOENCODERS y este último tuvo los mejores resultados, ya que se obtuvo un ACCURACY de 80.08 % y una PRECISIÓN 98.00 %. El modelo propuesto puede ayudar a reducir la posibilidad de que los estudiantes hagan trampa y mejorar la calidad y equidad de la educación virtual en línea.

Palabras clave: Aprendizaje profundo, Computación en la nube, Detección de anomalías, Inteligencia artificial, Monitoreo de exámenes en línea.



ABSTRACT

Virtual education has been boosted by the COVID-19 pandemic and, although it was already a learning option, its use has become more frequent. Universities do not always have the necessary infrastructure to accommodate large numbers of students who choose this type of education, and virtual education can help increase the number of vacancies offered. However, it is necessary to establish mechanisms that guarantee quality, fair and equitable education. Currently, human supervision is used, which requires the presence of an examiner to visually and acoustically monitor students during exams, which is expensive and requires a lot of effort when many people have to be evaluated. In this study, an anomaly detection model for online examinations is proposed. A desktop application was developed to collect information from the camera and microphone during the mock admission exam of the pre-university center of the National University of the Altiplano. 180024 video clips and 115292 audio clips were collected. The information was processed and motion features were extracted to construct a feature vector. Three models were developed, compared and evaluated, based on the ISOLATIONFOREST, LSTM-AUTOENCODER and AUTOENCODERS algorithms and the latter had the best results, since an ACCURACY of 80.08% and a PRECISION of 98.00% were obtained. The proposed model can help reduce the possibility of student cheating and improve the quality and equity of online virtual education.

Keywords: Deep Learning, Cloud Computing, Anomaly Detection, Artificial Intelligence, Online Exam Monitoring



CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En marzo del 2020 la Organización Mundial de la Salud (OMS), elevaba la situación de emergencia de salud pública por la pandemia internacional COVID-19. Esta llegó a afectar muchas situaciones, entre ellas el universitario y preuniversitario. Por lo que las universidades fueron forzadas a usar algún software que les permita brindar la enseñanza virtual. Sin embargo, los exámenes virtuales abrieron un nuevo enfoque en cuanto al docente y al estudiante y la principal preocupación fue la fiabilidad de los resultados de estos. Siendo un caso que llevó a muchas controversias por parte de alumnos y docentes (Montejo Bernardo, 2020). Alrededor de 138 países y 1.37 billones de estudiantes alrededor del mundo estaban estudiando en línea en casa (UNESCO, 2020). Con el aumento repentino del aprendizaje a distancia en línea y la necesidad de garantizar la integridad académica, las universidades han adoptado diferentes tecnologías de supervisión electrónica para monitorear exámenes en línea, estas tecnologías validan las identidades de los estudiantes y señalan las actividades sospechosas durante el examen para desalentar el engaño (Kharbat et al., 2021).

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema general

¿Cuál es el modelo más eficiente en la detección de anomalías en secuencias de videos de exámenes en línea utilizando inteligencia artificial?



1.3. JUSTIFICACIÓN DEL PROBLEMA

Según las estadísticas de World Wide Web alrededor de 6 millones de estudiantes optan por algún curso en línea dentro de un año, los cuales se desempeñan mejor que aquellos que asisten presencialmente. No obstante, el monitoreo manual conlleva costo y está propensa a errores por lo que una alternativa es el uso de algún software que permita el monitoreo y supervisión (Garg et al., 2020).

Ya existen herramientas de supervisión electrónica tales como ManageExam, ProctorExam, smowl, entre otros. Estas herramientas tienen diferentes enfoques y pueden presentar algunas limitaciones en el reconocimiento de rostros, análisis de la red, análisis y captura de ruidos, etc.

Sin embargo, durante la pandemia de COVID-19, surgió una necesidad sin precedentes de utilizar exámenes supervisados electrónicamente. Esta situación representó un desafío para la comunidad educativa, ya que las instituciones educativas se vieron obligadas a encontrar un equilibrio entre la calidad de la enseñanza y el mantenimiento de los procesos educativos necesarios (Kharbat et al., 2021). Plantear un modelo para la detección de anomalías en secuencias de video que supervise y analice alguna actividad sospechosa en un examen se ve como una necesidad muy importante. Esto se debe a la creciente demanda de exámenes en línea y la necesidad de garantizar la integridad de los procesos de evaluación.



1.4. OBJETIVO DE LA INVESTIGACIÓN

1.4.1. Objetivo general

Determinar el modelo más eficiente en la detección de anomalías en secuencias de video de exámenes en línea mediante inteligencia artificial.

1.4.2. Objetivo específico

- Construir un dataset de secuencias de video de exámenes en línea.
- Entrenar los modelos en la detección de anomalías en secuencias de video mediante inteligencia artificial.
- Evaluar y comparar los modelos en la detección de anomalías en secuencias de video en exámenes en línea.

1.5. HIPÓTESIS DE LA INVESTIGACIÓN

1.5.1. Hipótesis general

El modelo basado en AUTOENCODERS obtiene mejores resultados en la detección de anomalías en secuencias de video de exámenes en línea mediante inteligencia artificial.



CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES

En la investigación del autor Martínez Donoso, (2020) titulado: “Análisis de patrones de comportamiento en cursos Online para la detección de fraude” se observa un análisis de patrones y malas prácticas por parte de los usuarios como la sincronización de respuestas de dos o más estudiantes de la misma red. Se ha propuesto el “proceso Spark”, que fue desarrollado para leer un archivo que contiene los LOGs de la plataforma edX. No obstante, para este análisis se ha empleado el lenguaje de programación Python con la tecnología BIG DATA e inteligencia artificial para la predicción de resultados anómalos, obteniendo un 0.99 a 1.0 de precisión.

En el trabajo de Garg et al. (2020) titulado: “Controlador de examen virtual basado en red neuronal convolucional”. La investigación indica que el examen online se ha vuelto muy popular y sustituyó a los exámenes offline. Pero todavía hay una gran pregunta acerca de la seguridad del examen en línea. Por lo tanto, el autor presenta un sistema para monitorear el comportamiento de la persona que realiza la prueba y detectar las malas prácticas mediante los algoritmos *Haarcascade Classifier* y *deep learning*. Obteniendo una tasa de reconocimiento de 93% lo que hace que el sistema sea más preciso que los sistemas en línea actuales.

En el proyecto realizado por Ozgen et al.(2021) titulado: “Un sistema anti-trampas para entrevistas y exámenes en línea”. En este trabajo indica que el examen remoto ha ganado popularidad y se ha vuelto indispensable debido tanto a la pandemia como a la ventaja de las circunstancias de trabajo remoto. La mayoría de las empresas y



organizaciones educativas utilizan estas plataformas para los exámenes en línea. Sin embargo, uno de los problemas críticos de los sistemas de examen remoto es la confiabilidad. Por lo tanto, el objetivo principal del sistema es proporcionar un entorno de examen confiable y seguro para entrevistas y exámenes en línea. Para las pruebas se ha usado videos que fueron obtenidos de una cámara integrada, se enfoca en el reconocimiento de rostro, reconocimiento de rostros, seguimientos de rostros y detección de objetos.

Saba, et al. (2021), en la investigación titulada: “Automated Exam Proctoring Using Proposed Deep L2-GraftNet CNN Network and ASO Based Feature Selection Approach”, los investigadores indican de que supervisar las actividades durante un examen es muy tedioso para los evaluadores ellos proponen una arquitectura de redes neuronales convolucionales con 46 capas de profundidad la misma que está basada en arquitecturas como Alexnet y SqueezeNet para la extracción de características que posteriormente pasaron hacer clasificadas por algunas variantes de SVM, KNN obteniendo los mejores resultados con Fine KNN, obteniendo como mejores resultados de 93.88% de precisión, concluyendo que las pruebas validan la robustez de la propuesta.

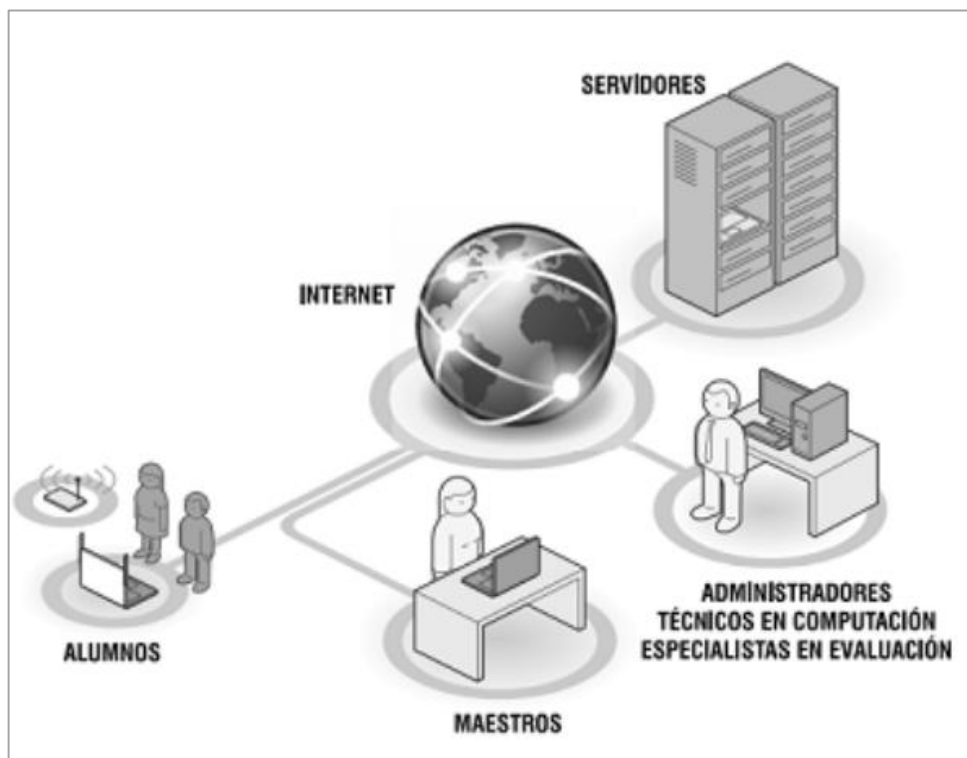
En la investigación de Beraún Barrantes (2021), titulada: “Sistema de reconocimiento facial en línea para prevenir la suplantación y el plagio en el examen de admisión virtual en la Universidad de Huanuco 2020”, tiene como finalidad plantear un método para el reconocimiento del rostro en exámenes virtuales de un gran número de estudiantes y prevenir alguna suplantación y plagio de un total de 1250 postulantes se reconoció 1132, el cual corresponde a 90.42%, 120 postulantes no fueron reconocidos, que corresponde al 9.58%. Sin embargo, en este sistema no se ha contemplado el audio, monitoreo del rostro en el examen, aplicaciones abiertas, doble pantalla, etc. Por lo que

es necesario investigar en estos temas para resolver el problema del plagio.

En el trabajo de Moyano et al., (2021) consiste en “Evaluar en Línea a través de un Entorno Virtual (Moodle). Experiencias en la UNTDF”. En esta investigación se describe y comparte la experiencia con respecto a la evaluación virtual en el marco de la pandemia COVID-19. Para este fin se ha usado la plataforma Moodle, tomando casos entre 100 alumnos y 15 alumnos respectivamente. Evaluar en línea es sumamente complejo por los diferentes valores atípicos, suplantación y plagio que se podría encontrar, también es una tarea compleja conectar a los objetivos de aprendizaje, ya que influyen varios factores como se muestra en la Figura 1.

Figura 1

Esquema de evaluación virtual.



Nota: Imagen extraída de Moyano et al., 2021, p. 2.

El trabajo realizado por los autores Concha Llorente & Vega Africano (2022) se



enfoca en el “uso de Inteligencia Artificial (IA) para mejorar la cultura de la originalidad de las evaluaciones en línea en la UNAD”. el sistema detecta el plagio en el sistema educativo de manera virtual, mediante reconocimiento facial, Block Watermark y algoritmos genéticos, siendo el propósito de mejorar el sistema educativo. No obstante, el sistema dificulta el plagio, retroalimenta el sistema educativo, también el sistema ayuda a mejorar el proceso al resolver las preguntas y genera una mayor preparación en los estudiantes, ya que estos se ven reflejados en el tiempo de respuesta de los cuestionarios.

En la investigación del autor Paramita et al., (2022) titulada “View of Cloud based LMS Security& Exam Proctoring Solution” indica que la educación está evolucionando rápidamente hacia un modelo basado en la nube, especialmente durante la pandemia de Covid-19. Aprender desde casa y tomar pruebas en línea es una práctica común. Sin embargo, la supervisión de exámenes en línea es un problema crítico que limita al docente obtener calificaciones correctas, no obstante, es un desafío importante en el sistema educativo. La supervisión humana es costosa y consume tiempo, por lo que se necesitan técnicas de evaluación supervisadas por Internet. Este estudio tiene como objetivo crear un sistema de inteligencia multimodal que también pueda proporcionar un sistema de supervisión continua en línea (OPS) para los exámenes y examinar los comportamientos de trampa. La primera fase en el presente trabajo se incluye la autenticación facial y de contraseña, y durante el examen se ha capturado señales audiovisuales del entorno, empleando funciones factorías y de covarianza.

Rabiha et al., (2022) desarrollaron una investigación titulada “Survey of online exam proctoring model to detect cheating behavior based on face recognition”, con el objetivo de profundizar en el monitoreo de exámenes en línea, el cual está centrado en el reconocimiento facial para detectar anomalías y/o trampas. Este estudio les permitió



realizar un análisis adicional con respecto al Método de detección de rostros, el Método de reconocimiento de rostros, la Característica inicial, el Análisis de comportamiento y las Métricas de evaluación; donde el método de detección de rostros más empleado fue Viola-Jones con una presentación del 20 %, posteriormente CNN y MTCNN con una presentación total del 21 %. Finalmente se concluyó que las características que también son indispensables fueron la detección de la boca, el vector facial, la ubicación de puntos de referencia, los gestos y la postura.

El trabajo de investigación de Kaddoura et al., (2022), denominado “A systematic review on machine learning models for online learning and examination systems”, donde el objetivo principal fue destacar la importancia de los exámenes y evaluaciones para los estudiantes y cómo la pandemia del COVID-19 ha afectado el aprendizaje y la evaluación en todo el mundo. Así mismo revisaron 135 estudios en los últimos cinco años para explorar el papel del aprendizaje automático en la gestión de exámenes en línea durante la pandemia. Los algoritmos de aprendizaje supervisado y no supervisado se han utilizado para la autenticación, programación, supervisión y detección de trampas durante el proceso de examen. Finalmente concluyen que la discusión de los problemas y desafíos que presenta el aprendizaje automático en el sistema de examen, junto con posibles soluciones.

Rodríguez Morales et.al., (2020) llevaron a cabo una investigación sobre “Cómo asegurar evaluaciones válidas y detectar falseamiento en pruebas a distancia síncronas”, donde su objetivo principal fue introducir la idea del falseamiento como un aporte al análisis de la validez psicométrica de las pruebas, mediante una revisión bibliográfica. Esta investigación les permitió determinar los principales métodos para detectar el falseamiento, que se pueden utilizar para asegurar la validez de los resultados en pruebas



sincrónicas, como las de opción múltiple, así mismo lograron describir los usos, limitaciones y potencialidades de los métodos mencionados.

2.2. MARCO TEÓRICO

2.2.1. Computación en la nube

La computación en la nube es una tecnología que permite acceder desde cualquier lugar geográfico en cualquier momento a recursos computacionales a través de internet, sin necesidad de contar con un dispositivo físico.

La computación en la nube es la entrega bajo demanda de recursos y aplicaciones de TI a través de Internet con precios de pago por uso. Ya sea que ejecute aplicaciones que comparten fotos con millones de usuarios móviles o que respalde las operaciones críticas de su empresa, la nube brinda acceso rápido a recursos de TI flexibles y de bajo costo. Con la computación en la nube, no necesita realizar grandes inversiones iniciales en hardware ni dedicar mucho tiempo a administrar ese hardware. En su lugar, puede aprovisionar exactamente el tipo y el tamaño correctos de los recursos informáticos que necesita para impulsar su nueva idea brillante u operar su departamento de TI. Con la computación en la nube, puede acceder a tantos recursos como necesite, casi al instante, y pagar solo por lo que usa. (Amazon web services, 2014), como se muestra en la Figura 2.

Figura 2

Modelos de Servicio de la Computación en la Nube y Tipo de Usuarios Asociados.



Nota: Adaptado de Fuentes & Hernandez, 2019.

2.2.1.1. Amazon Web Services (AWS)

Amazon Web Services es una plataforma que ofrece soluciones de computación, almacenamiento y redes. En AWS puede ejecutar todo tipo de aplicaciones, los cuales pueden ser uno o la combinación de varios servicios, como EC2, S3, Lambda, API Gateway, entre otros (Wittig & Wittig, 2018).

2.2.1.2. Amazon Elastic Compute Cloud (Amazon EC2)

Es una parte central de la computación en la nube de la empresa Amazon, ofrece a los usuarios la posibilidad de crear entornos virtuales en la nube, en donde los usuarios pueden ejecutar sus propias aplicaciones.

La mayor ventaja de EC2 es la auto escalabilidad. La máquina virtual EC2 se encuentra en la central de datos de Amazon y por lo tanto tiene un gran ancho de banda disponible. Puede crear tantas máquinas

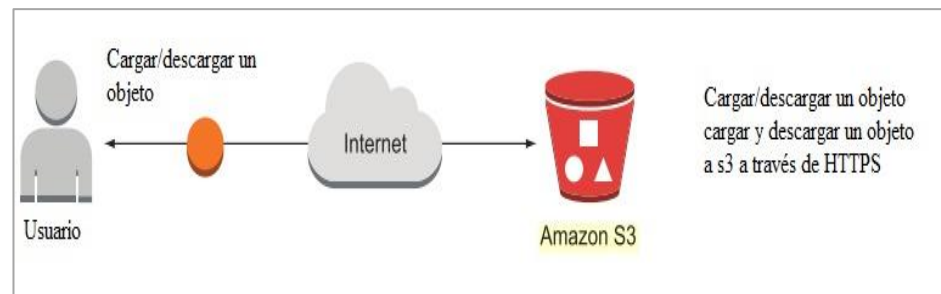
como desee e iniciarlas o detenerlas a pedido. Solo se factura por el tiempo de cómputo y el ancho de banda que se utiliza. (Diginmotion, 2010)

2.2.1.3. Amazon Simple Storage Service (Amazon S3)

Amazon S3 es un almacén de datos distribuidos totalmente con espacio de almacenamiento casi ilimitado, puede almacenar datos como imágenes, archivos binarios, documentos. Es un servicio web típico que ofrece escalabilidad, seguridad, disponibilidad de datos (Amazon.com, 2022), permite almacenar y recuperar datos organizados como objetos a través de una API accesible a través de HTTPS (Wittig & Wittig, 2018), como se muestra en la Figura 3.

Figura 3

Subida y descarga de objetos a S3 via HTTPS.



Nota: Adaptado de Wittig & Wittig, 2018, p. 237.

2.2.1.4. Amazon Timestream

En la página de Amazon Web Services (2022), indica que “Timestream es una base de datos de series temporales rápida, escalable, totalmente administrada y especialmente diseñada que facilita el almacenamiento y el análisis de billones de puntos de datos de series temporales al día”. A Timestream se puede enviar datos a través de AWS

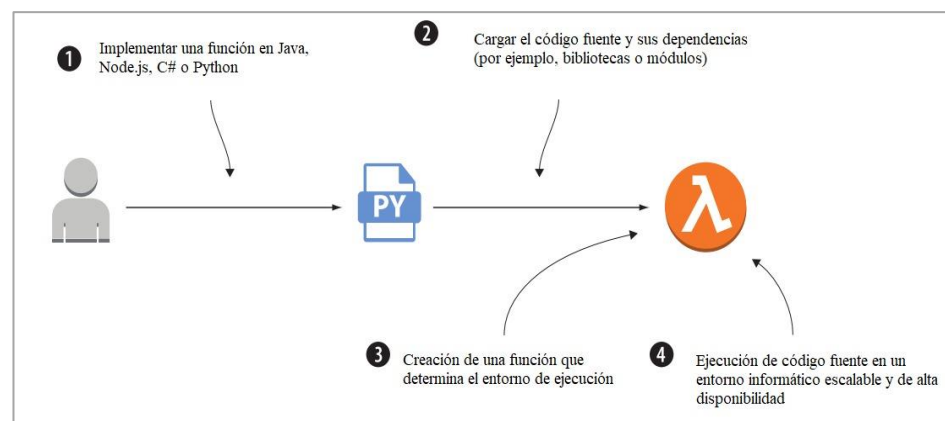
IoT Core, Amazon Kinesis, Amazon MSK y Telegraf. Estos datos se pueden visualizar mediante herramientas como JDBC, Grafana, D3, etc.

2.2.1.5. AWS Lambda

AWS Lambda es un servicio sin servidor (serverless), está basado en eventos y se puede activar más de 200 servicios de AWS y servicios (SaaS) (Patterson, 2019) (Whitepaper, 2022). Ofrece entornos para ejecutar código Java, C#, Python, Node.js, y Go. Para ejecutar estos lenguajes backend solo basta con crear una función e implementar código para comunicarse con su endpoint (Wittig & Wittig, 2018), como se muestra en la Figura 4.

Figura 4

Servicio AWS Lambda.



Nota: Imagen adaptada de Wittig & Wittig, 2018, p. 201.

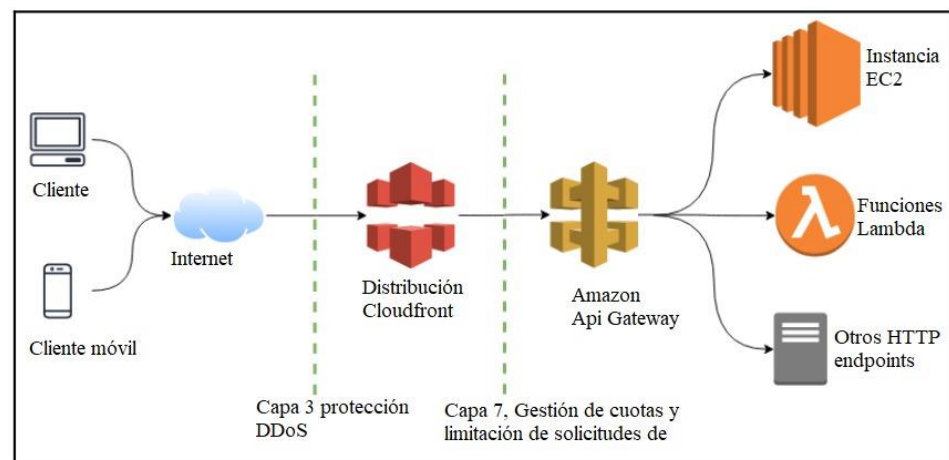
2.2.1.6. API Gateway

API Gateway es un servicio de AWS, que permite gestionar el tráfico web, ya que actúa como un servicio proxy y aplica sus propias definiciones de API RESTful y los controles de seguridad. Esta función

crea un punto de enlace HTTP para exponer un servicio de AWS como backend. Entonces se puede decir que la combinación de API Gateway y otros componentes de AWS es un mecanismo de seguridad para publicación de APIs (Patterson, 2019). En la Figura 5 se puede observar las capas de protección.

Figura 5

Servicio API Gateway.



Nota: Imagen adaptada de Patterson, 2019, p. 85.

2.2.2. Inteligencia artificial

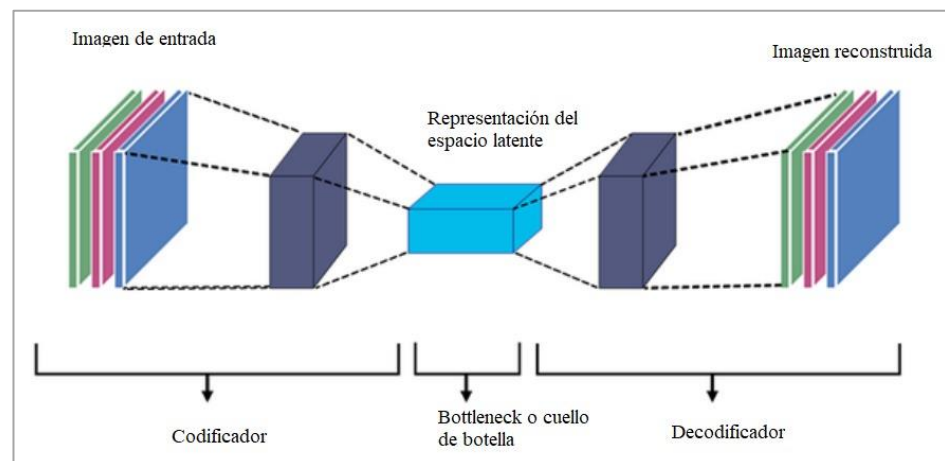
En el libro de Caballero & Martion (2021), la inteligencia artificial es definido como "el campo de la informática que busca replicar tareas complejas que habitualmente han venido realizando humanos, incluyendo reconocimiento de imágenes, escritura o voz, o incluso toma de decisiones en tiempo real, como las que deben tomar los vehículos autónomos". Entre otras aplicaciones se puede mencionar la robótica, como también en las importantes decisiones de negocio.

2.2.2.1 Autoencoders

El Autoencoder se define como un tipo de red neuronal artificial entrenada para reconstruir su entrada (Bank et al., 2020), siendo el objetivo principal aprender de una representación de un grupo de datos mediante el aprendizaje no supervisado. Los Autoencoders presentan una arquitectura simple de una red neuronal no recurrente, el cual tiene el mismo número de nodos en la capa de entrada y en la capa de salida. Hoy en día los Autoencoders (Figura 6) se utilizan en diferentes problemas, una de ellas es la detección de anomalías (Laguna, 2021), los Autoencoders también se usan para reconstruir con mejor calidad las imágenes como se observa en la Figura 7.

Figura 6

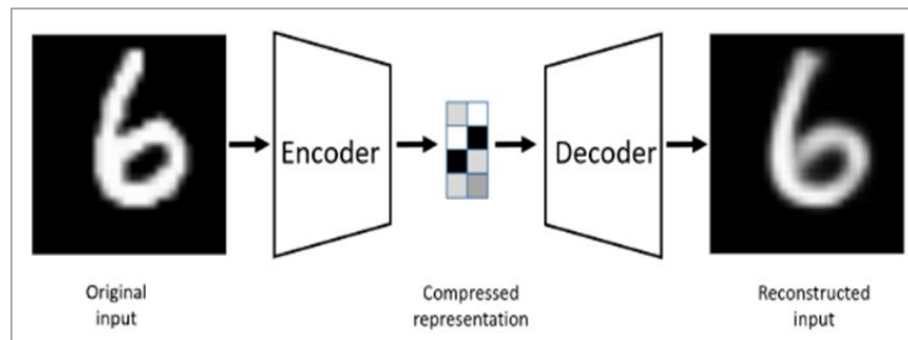
Autoencoder



Nota: Imagen adaptada de Laguna, 2021, p. 12

Figura 7

Proceso para codificar, comprimir y decodificar.



Nota: Imagen adaptada de Bank et al., 2020, p. 2

2.2.2.2. Aprendizaje automático

El aprendizaje automático, también llamado aprendizaje artificial [PLAZ92], es un área de interés muy desarrollada en la IA.

2.2.2.3. Detección de rostros

Una de las técnicas más usadas para la detección de rostros es Haar Cascade. Este clasificador tiene una primera versión llamada *Haar wavelet* para la detección de rostros y peatones, sin embargo no fue muy significativo hasta hace un tiempo, en donde Fang & Aarabi (2004), adoptaron la idea y la denominaron Características Heer (*Haar-Like Features*), este se basa en los niveles de intensidad que presentan los píxeles en una imagen. Una ventaja muy importante de "Características Haar" es detectar estructuras de objetos uniformes o no. En resumen, Haar se puede definir como una ventana de píxeles dividida en regiones rectangulares con tamaño y orientación variable (Barba Guamán, 2015), como se muestra en la Figura 8.

Figura 8

Características Haar.

+	-
+	-

Nota: Adaptado de Barba Guamán, 2015, p. 19.

Esta ventana de píxeles recorre toda la imagen, esto se puede representar ecuación 1 :

$$H(x,y) = \sum_p I(x,y) - \sum_n I(x,y) \quad (1)$$

Donde:

- $I(x,y)$: Imagen del rostro.
- p y n : son las regiones positivas y negativas
- $H(x,y)$: Característica Haar en el punto x,y .

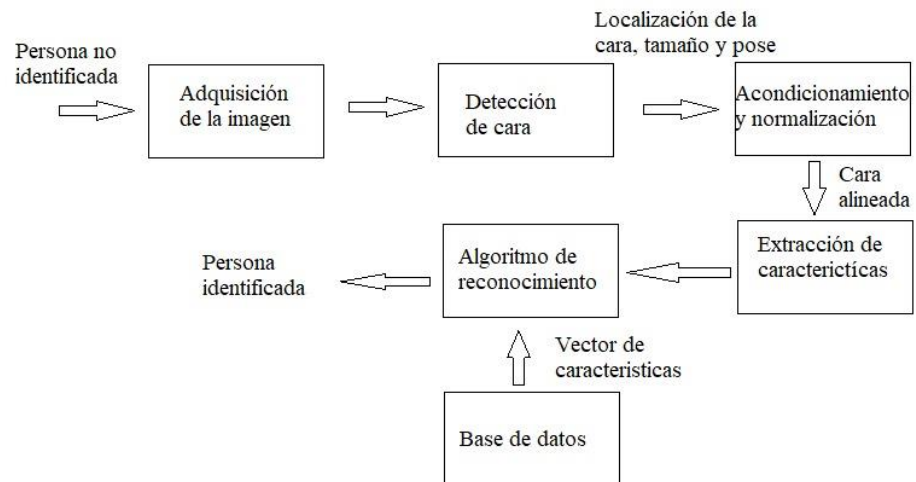
2.2.2.4. Reconocimiento facial

El reconocimiento facial es la toma de decisiones e identificación mediante el uso de características personales (fotos y videos) capturadas de cada persona, estas pueden ser reconocidas o detectadas de forma automática, para realizar este proceso se utilizan algoritmos para extracción de características como PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), etc. Entre las etapas más importantes para el reconocimiento facial se tiene la detección de rostros, acondicionamiento, normalización, extracción de características y

reconocimiento (Cadena Moreano et al., 2017). En la Figura 9 se muestra el proceso para el reconocimiento.

Figura 9

Proceso para el reconocimiento facial



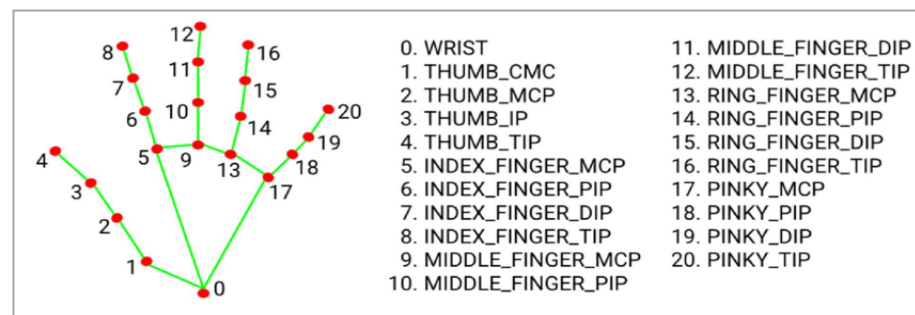
Nota: Adaptado de Cadena Moreano et al., 2017.

2.2.2.5. MediaPipe

MediaPipe es un framework desarrollado por Google, dicha biblioteca es multimodal y multiplataforma es decir puede ser desarrollada en web, escritorio y aplicaciones móvil, esta herramienta es ideal para desarrollar proyectos de visión artificial en tiempo real. En la Figura 10 se muestra la base para la comprensión del lenguaje de señas y el control de gestos con las manos.

Figura 10

Mediapipe hand landmarks.



Nota: Imagen extraída de Grif, M & Kondratenko, Y, 2021, p. 4.

2.2.2.6. Anomalía

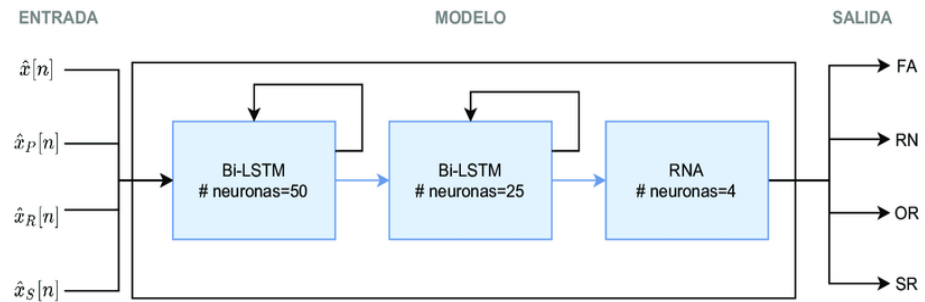
En exámenes en línea es un desafío detectar anomalías de manera automática mediante un algoritmo y de gestionar grandes cantidades de datos. En este último, el análisis de un conjunto de datos grandes podría ser de mucha importancia, ya que se puede detectar valores atípicos, una desviación, identificación de algo irregular o un evento que no es normal. Por ejemplo verificación y detección de rostros en exámenes virtuales (Servicenow,S/f).

2.2.2.7. Redes LSTM

Las LSTM son un tipo de redes recurrentes o Recurrent Neural Networks(RNN) en inglés, son una clase de redes para analizar datos de series temporales, una de sus principales particularidades es que la información es persistente al introducir bucles en el diagrama de red por lo que se puede inferir que estas redes pueden recordar estados previos y utilizar información para inferir cuál será el siguiente estado, como se ilustra en la Figura 11.

Figura 11

Estructura de la red LSTM.



Nota: En la imagen se puede observar datos de entrada ($\hat{x}[n]$, \hat{x}_P , \hat{x}_R y \hat{x}_S son señales normalizadas), el modelo y la salida (FA: fibrilacion auricular, RN: ritmo normal, OR: otros ritmos, SR: señal ruidosa). Hernandez, Fabio, 2020, p. 77.

2.2.3. Procesamiento multimedia

Procesamiento multimedia se refiere al proceso de tratar diferentes formatos como audio, video, texto, imágenes; mediante algún método o herramienta con el cual se pueda extraer características necesarias para poder analizarlos.

2.2.3.1. Procesamiento de imágenes

La imagen está constituida por una función bidimensional $f(x,y)$, donde x , y son coordenadas y cantidades finitas, discretas (Roncagliolo, 2008). Al transformar una imagen a escala de grises, en donde cada coordenada representa un pixel y este recibe un valor entre 0 y 255 (Taquiá, 2017) como se muestra en la Figura 12.

Figura 12

Imagen en escala de grises.



230	226	217	183	123	170	135	91	154	230	218	222	222	226	229	229	228	218	227	227	228
226	223	227	183	161	118	109	87	202	226	226	227	224	227	229	227	227	226	227	229	229
219	210	206	179	150	92	143	74	196	226	231	226	226	227	230	230	226	228	227	230	228
174	181	203	169	117	176	120	82	119	224	224	226	228	223	230	225	233	228	226	231	229
164	189	183	168	168	171	158	95	193	233	231	226	225	216	223	238	226	222	239	225	176
173	191	212	199	141	142	111	211	225	228	232	223	208	149	192	225	226	231	222	160	139
155	167	224	176	153	99	85	218	230	227	224	211	124	117	214	235	227	224	141	150	206
205	212	227	218	104	192	114	206	208	197	233	215	153	160	216	214	219	165	164	106	181
218	232	236	219	141	175	179	167	130	128	181	217	172	201	222	177	162	128	154	111	126
229	231	227	197	197	176	237	215	164	126	107	135	180	208	212	167	131	136	214	189	123
232	223	231	223	168	219	205	203	224	164	117	117	140	186	176	188	159	186	200	185	139
184	207	222	220	206	217	208	151	135	134	155	149	164	119	199	195	200	134	163	146	131
108	214	214	169	211	190	185	152	105	110	109	117	97	99	183	180	112	104	118	134	135
120	135	175	148	157	131	170	202	134	165	187	172	163	167	162	150	125	127	87	51	116
127	140	128	140	153	142	99	114	100	165	126	146	124	113	144	177	99	98	111	70	89
135	116	129	121	104	154	167	140	111	114	110	146	107	70	99	93	88	101	63	78	112
107	156	105	107	98	124	99	89	120	119	113	120	105	125	163	138	107	85	88	91	145
127	93	96	123	103	93	88	104	112	135	154	121	81	139	88	167	91	150	90	64	138
92	47	46	41	87	84	69	70	70	119	164	171	182	78	61	73	102	71	84	71	71

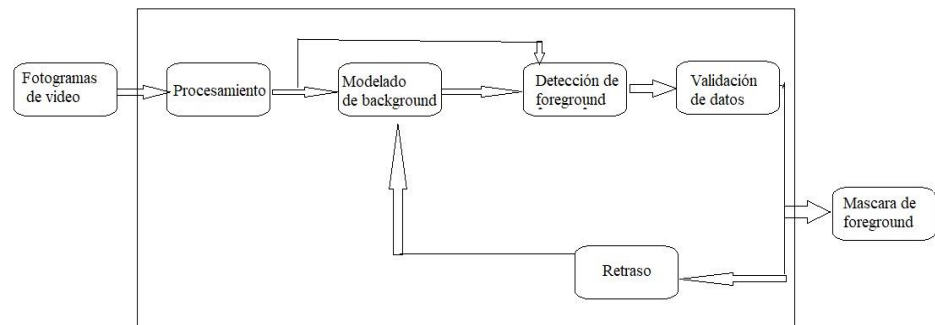
Nota: Adaptado de Taquíá, 2017, p. 17.

2.2.3.2. Sustracción de fondo

La sustracción de fondo es el proceso en el cual son comparados píxeles de una matriz con el modelo de fondo, con el fin de determinar los píxeles son parte del fondo o del frente, también es conocido como detección en primer plano. En el artículo de Fernando & Tafurt (2016), define que "En una sustracción los píxeles que no cambian son considerados como "background" y los píxeles que cambian son considerados como "foreground" u objetos en movimiento", como se muestra en la Figura 13.

Figura 13

Etapas de sustracción de fondo.



Nota: Adaptado de Fernando & Tafurt, 2016, p. 18

2.2.3.3. Biblioteca Opencv

Opencv es un biblioteca de visión computacional, que tiene como objetivo proporcionar una infraestructura de visión artificial fácil de usar, escrita en c y c++ y se ejecuta en diferentes sistemas operativos como Linux, Windows, Mac OS x. Opencv está desarrollado para aplicaciones en tiempo real, abarca muchas áreas como imágenes médicas, seguridad, calibración de cámaras, visión estéreo, robótica, etc, también incluye una biblioteca de aprendizaje automático, reconocimiento de patrones y clustering (Bradski & Kaebler, 2016).

2.2.4. Herramientas de visualización

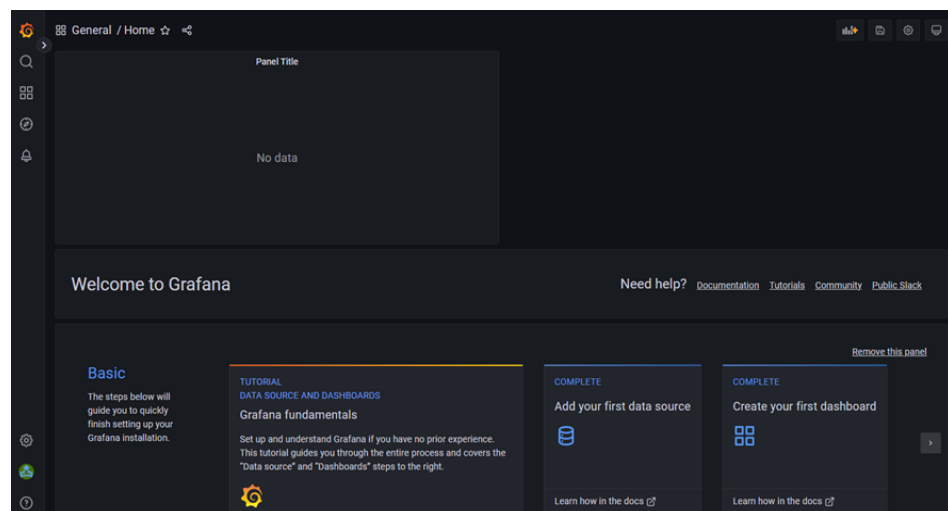
2.2.4.1. Grafana

Grafana es una herramienta para consulta y visualización de datos de series temporales como Prometheus, influxDB, timestream, OpenTSDB, etc. También se puede conectar a base de datos relacionales como Mysql, Microsoft SQL Server (MS SQL), etc. Grafana tiene diversos modos de

visualización de datos tales como en forma de gráficos, mapas de calor o histogramas, todo esto se muestra en un dashboard (tablero), el cual es un conjunto de paneles para mostrar información de un sistema o servicio en particular o en su totalidad. Una característica muy importante es que puede alertar sobre situaciones críticas que fueron definidas con umbrales (Mainak & Pratap, 2021). En la Figura 14 se muestra el dashboard HOME.

Figura 14

Plugin Amazon Timestream en Grafana.



Nota: Captura de la interfaz de Grafana.

2.2.5. Métricas de error

2.2.5.1. MAE

Es una función que se usa para pronosticar el error medio absoluto y los posibles resultados. Esta medida se calcula de la diferencia entre dos variables continuas, donde se consideran los datos calculados y los datos observados relativos ambos al mismo fenómeno, la finalidad es cuantificar

la precisión de una técnica de predicción. La ecuación 2 usada para pronosticarla es la siguiente:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x| \quad (2)$$

2.2.5.2. MSE

El error cuadrático medio se usa como métrica estadística estándar para medir el rendimiento del modelo en estudios meteorológicas, calidad del aire e investigación climática. La ecuación 3 para hallar dicho error es la siguiente:

$$MS3 = \frac{1}{n} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3)$$

2.2.6. Definición de términos básicos

2.2.6.1. Python

Python es un lenguaje de programación de más alto nivel, puesto que las instrucciones en el código son más entendibles por un humano (Delgado Quintero, 2022).

Python permite reflejar de forma simple y elegante, las ideas en forma algorítmica, tiene una sintaxis sencilla y clara que permite escribir programas en menos tiempo, además Python tiene un modo interpretado lo que permiten, que ofrece retroalimentación inmediata, lo que permite probar nuevas ideas casi al instante (Michael Dawson, 2003).



2.2.6.2. JavaScript

JavaScript es un lenguaje de programación interpretado muy usado para crear páginas web dinámicas y no requiere ser compilado para poder ejecutarlo en cualquier navegador (Eguíluz Pérez, 2018).

Aunque JavaScript no puede competir con los lenguajes compilados fuertemente tipados como C++, Java y C# en velocidad, proporciona suficiente rendimiento para muchos tipos de simulaciones y supera a sus competidores en facilidad de uso de productividad del desarrollador, especialmente en simulación basada en web (IEEE, 2016).

2.2.6.3. Qt Quick

El módulo Qt Quick es una biblioteca, el cual permite escribir aplicaciones en QML. En QML se puede integrar Javascript y C++, todo en el marco QT. En la página oficial de Qt-QML (Doc.Qt.Io, 2022) describe lo siguiente: "QML es un lenguaje declarativo que permite describir las interfaces de usuario en términos de sus componentes visuales y cómo interactúan y se relacionan entre sí. Es un lenguaje altamente legible que fue diseñado para permitir que los componentes se interconectan de manera dinámica y permite que los componentes se reutilicen y personalizan fácilmente dentro de una interfaz de usuario".

2.2.6.4. Paquete Pandas

En el libro de Delgado Quintero (2022) define: "pandas es un paquete open-source que nos proporciona una forma sencilla y potente de



trabajar con estructuras de datos a través de múltiples herramientas para su análisis”.

2.2.6.5. Librería numpy

En el libro de Delgado Quintero (2022) define: “NumPy es el paquete fundamental para computación científica en Python y manejo de arrays numéricos multidimensionales”.

2.2.6.6. Sprint

Un sprint es un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida.

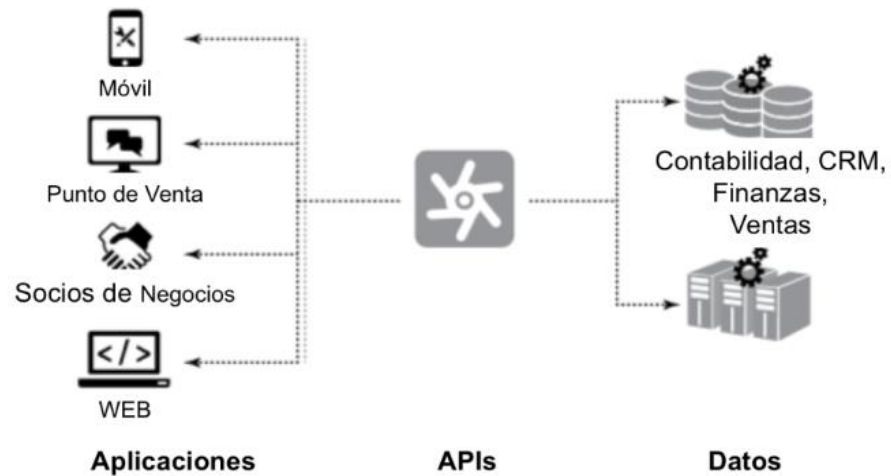
2.2.6.7. Interfaz de programación de aplicaciones (API)

El autor Pérez Porto, J.(2015), define API como “un código que indica a las aplicaciones cómo pueden mantener una comunicación entre sí. Estas reglas permiten que los distintos programas mantengan interacciones”.

Las interfaces de programación de aplicaciones son una de esas innovaciones en el dominio del desarrollo de software. Las API forman un componente integral del ecosistema de software (Manikas, 2016). Estos ecosistemas de software se han convertido en una forma ideal de construir grandes soluciones de software sobre una plataforma tecnológica común. (Ofoeda, Boateng, & Effah, 2019). En la Figura 15 se muestra el rol que juegan las API, haciendo posible conectar diferentes fuentes de datos y adecuar la cadena de valor digital.

Figura 15

Comunicación de la API.



Nota: Adaptado de Saffiro, 2018.

2.2.6.8. Dataframe

En la página Datascientists (S/f) se define Dataframe como: “Un DataFrame es una serie de Series Pandas indexadas por un valor. En este artículo, presentaremos la estructura de los DataFrames y luego veremos sus diferentes atributos y métodos básicos, y explicaremos su utilidad y funcionamiento”.

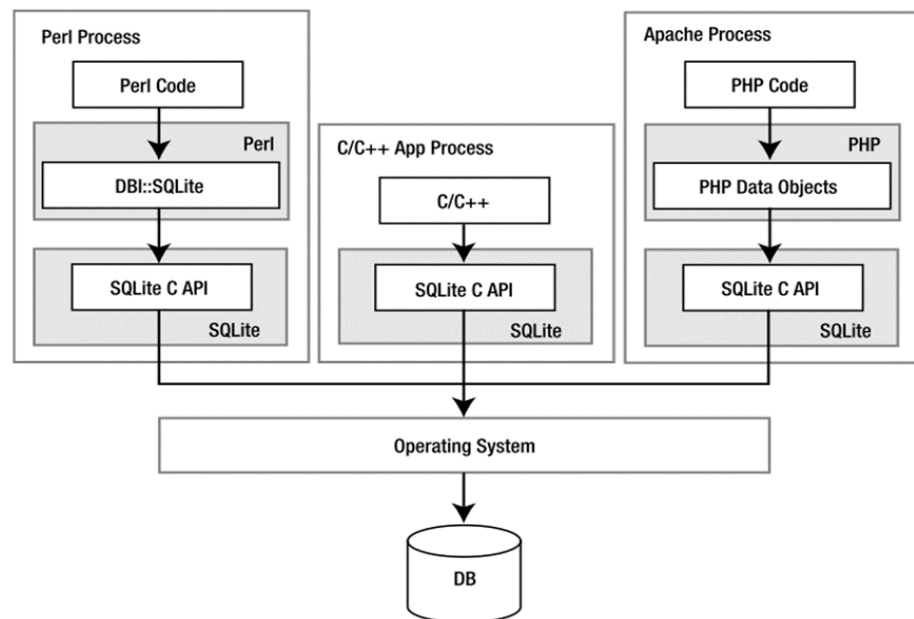
2.2.6.9. Base de datos relacionales (SQLITE)

SQLITE es una base de datos relacional de fácil uso, compacto, eficiente, confiable y altamente portátil, coexisten dentro de otra aplicación a la que sirve. Una de las ventajas es que no requiere configuración ni administración, cabe mencionar que el cliente y el servidor se ejecutan en el mismo proceso (Owens & Hipp, 2019). En la

Figura 16 se muestran tres procesos que interactúan como servidores independientes en sí mismos.

Figura 16

Tres procesos embebidos.



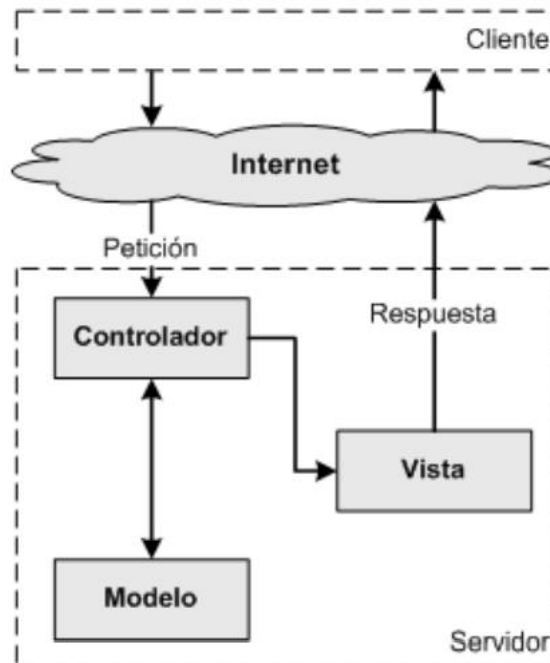
Nota: Imagen extraída de Owens & Hipp, 2019, p. 2.

2.2.6.11. Modelo Vista Controlador (MVC)

Esta arquitectura permite la separación de modelo, vista, controlador con el fin de facilitar el mantenimiento posterior y la reusabilidad del código. El modelo es la parte que realiza operaciones para obtener los resultados de la base de datos, las vistas son las encargadas de materializar las interfaces de la aplicación del usuario, se sabe que el modelo hace referencia al “backend” y la vista el “frondend”. No obstante, un controlador es enlace o comunicación entre las vistas y los modelos, es decir que se establecen reglas para la comunicación (Aranda, 2019), como se muestra en la Figura 17.

Figura 17

Esquema Modelo Vista Controlador.



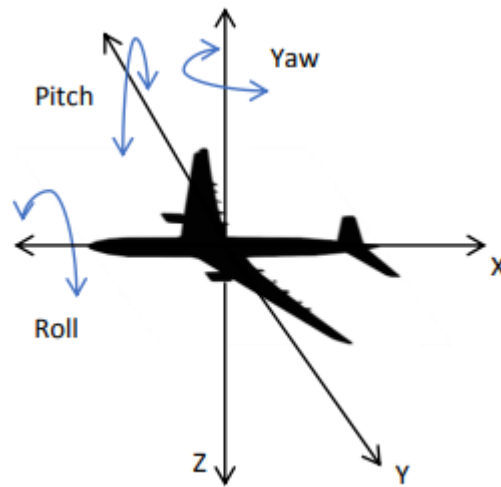
Nota: Adaptado de Aranda, 2019, p. 7.

2.2.6.12. Ángulos de Euler

Los ángulos de Euler constituyen en un cuerpo rígido tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, usando los ángulos α , β y γ es posible definir la orientación en ángulos de Euler alrededor de los ejes Z, Y, y X. Somma, F. J. (2018). Una representación particular de los ángulos Euler son los ángulos de Tait-Bryan como se muestra en la Figura 18.

Figura 18

La notación de ángulo de Tait-Bryan de los ángulos de Euler.



Nota: Imagen extraída de Somma, F. J., 2018) p. 59.

Por otro lado, también se puede representar una rotación mediante una matriz de rotación (Figura 19) mediante el producto de tres matrices de los ejes X, Y y Z y teniendo α , β y γ como ángulos de rotación alrededor de los ejes Z, Y y X (Riba Pi, E, 2015).

Figura 19

Matriz de rotación para detección de anomalías (Riba Pi, E, 2015).

$$\mathbf{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$

Nota: extraída de Riba Pi, E, 2015, p. 11.

2.2.6.13. Protocolo http

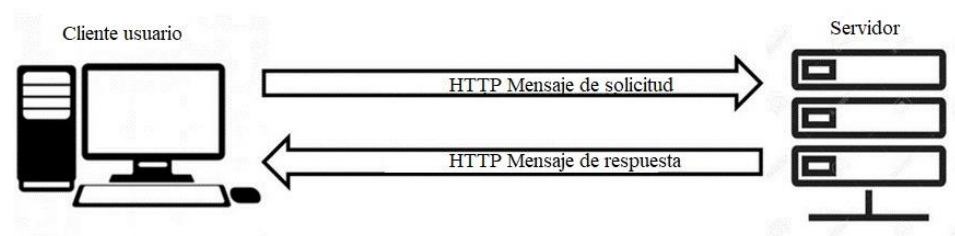
El Http (del inglés HyperText Transfer Protocol o Protocolo de Transferencia de Hipertextos) es el protocolo de transmisión de información de la World Wide Web este es un protocolo sin estado que

trabaja a nivel de la capa de aplicación (Enciso et al., 2018). Con el http se establecen criterios de significado y forma o también llamado sintaxis y semántica informática.

Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos), como se muestra en la Figura 20.

Figura 20

Esquema del protocolo Http.



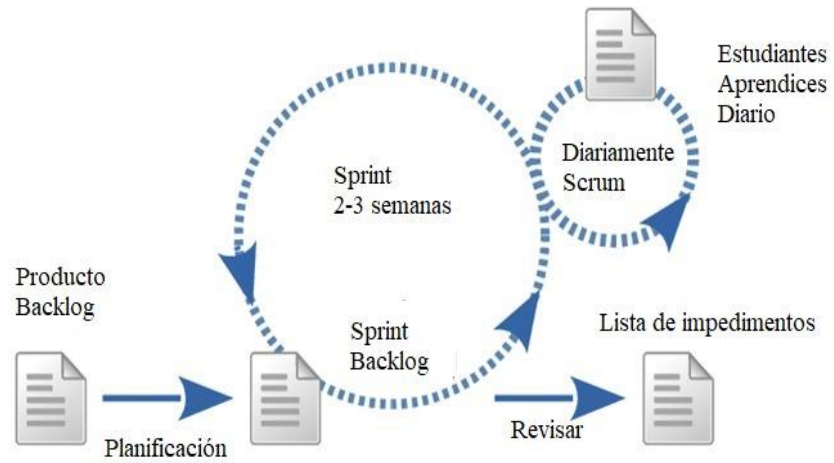
Nota: Khandkar, Vinod & Hanawal, Manjesh, 2021.

2.2.6.14. Scrum

Rodríguez, César & Dorado, Ruben. (2015) Scrum es un marco de trabajo usado para el desarrollo de aplicaciones y proyectos, esta metodología ágil es una de las más conocidas por ser iterativa e incremental. En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite, como se muestra en la Figura 21.

Figura 21

Ilustración de un Sprint dentro de un proceso SCRUM con algunas adaptaciones a la metodología.



Nota: Adaptado de Rodríguez, César & Dorado, Ruben, 2015.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. TIPO DE INVESTIGACIÓN

La investigación en cuestión es de naturaleza comparativa ya que su objetivo es confrontar el comportamiento de los modelos de detección de anomalías (Bermejo, 2021).

3.2. DISEÑO DE INVESTIGACIÓN

El diseño de este trabajo de investigación es experimental, el cual consiste si existe alguna diferencia en los tratamientos de los experimentos realizados, ya sea positiva o negativa.

El tipo de diseño es pre experimental, puesto que se realiza experimentos y luego se mide el experimento en dos diferentes tiempos (Arias & Covinos, 2021, p. 68). Este diseño se presenta en la Tabla 1.

Tabla 1

Diseño preexperimental.

Aplicación del tratamiento Grupo experimental	Medición de la variable independiente
X	O1
Fecha 1	Fecha 2

Nota: Cuadro extraído de Arias & Covinos, 2021, p. 74.

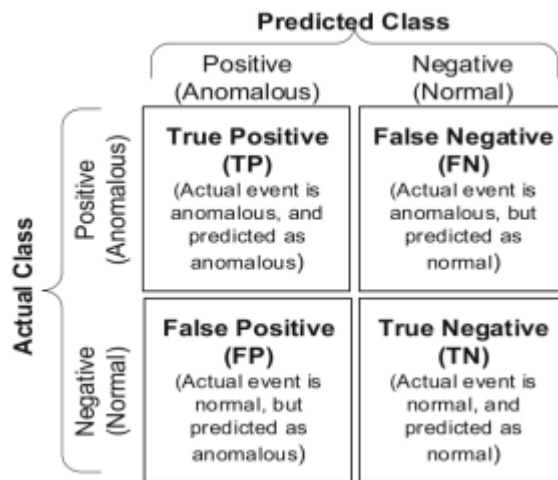
3.3. MÉTRICAS PARA LA EVALUACIÓN DEL MODELO

Matriz de confusión es una herramienta que permite la visualización del desempeño de un modelo que se emplea en aprendizaje supervisado. Cada columna de la

matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En la Figura 22 se muestra un ejemplo de matriz de confusión:

Figura 22

Matriz de Confusión para detección de anomalías.



Nota: Adaptado de Karishma et al., 2019.

3.4. POBLACIÓN Y MUESTRA

3.4.1. Población

La población está conformada por 180024 clips de video del examen simulacro virtual del centro preuniversitario de la Universidad Nacional del Altiplano.

3.4.2. Muestra

Se ha empleado como muestra el dataset validado por un ser humano el cual cuenta con 248 videos registrados del examen virtual del centro preuniversitario de la Universidad Nacional del Altiplano-Puno. Se está empleando el muestreo no probabilístico e intencional, puesto que depende de las



causas relacionadas con las características de la investigación y se establece la selección de los elementos de manera arbitraria (Hugo Sanchez Carlessi, 2019, p. 166).



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

Luego de realizar la extracción de videos de exámenes en línea se procedió al etiquetado manual en el que un experto determinó si el comportamiento de los estudiantes era o no una anomalía, se construyó un dataframe que almacenaba características del movimiento de la postura de la cabeza y a partir de estos se construyeron histogramas de movimiento de cada eje que posteriormente se utilizó para el entrenamiento de los modelos. En los experimentos se han probado distintas arquitecturas de cómo la ISOLATION FOREST, LSTM-AUTOENCODER y la AUTOENCODER presentado resultado en la Tabla 3 y obteniendo mejores resultados con el AUTOENCODER con 60% de MAPE.

4.1.1. Construir un dataset de secuencias de video de exámenes en línea

Para la construcción de un dataset se vio por conveniente el desarrollo de una aplicación de escritorio que mediante modelos pre entrenado de inteligencia artificial y servicios en la nube que permita la adquisición de datos de exámenes en línea.

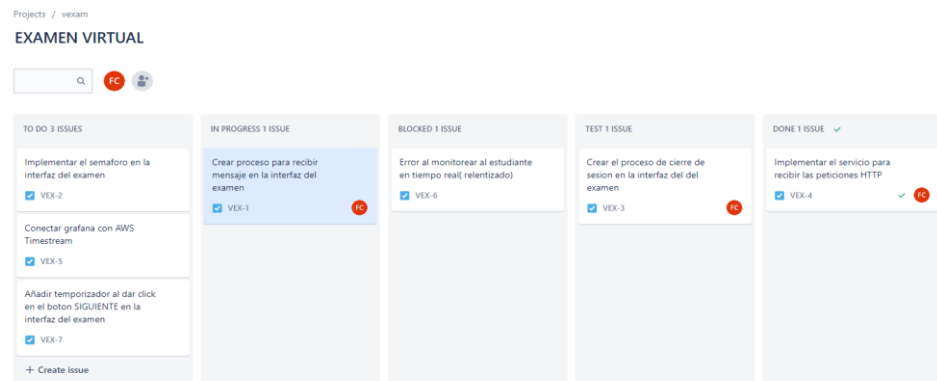
4.1.1.1. Desarrollo de la aplicación con modelos pre-entrenados y servicios en la nube para la adquisición de datos de exámenes en línea

La aplicación es implementada con la metodología ágil SCRUM y la herramienta Jira, dividida en cuatro Sprints con dos semanas de duración

cada una, adicionalmente los requerimientos se han colocado en el backlog propio del uso de esta metodología (ANEXO 1). Se han creado cinco columnas ISSUES, IN PROGRESS, BLOCKED, TEST y DONE, como se muestra en la Figura 23.

Figura 23

Tareas registradas y realizadas en Jira.

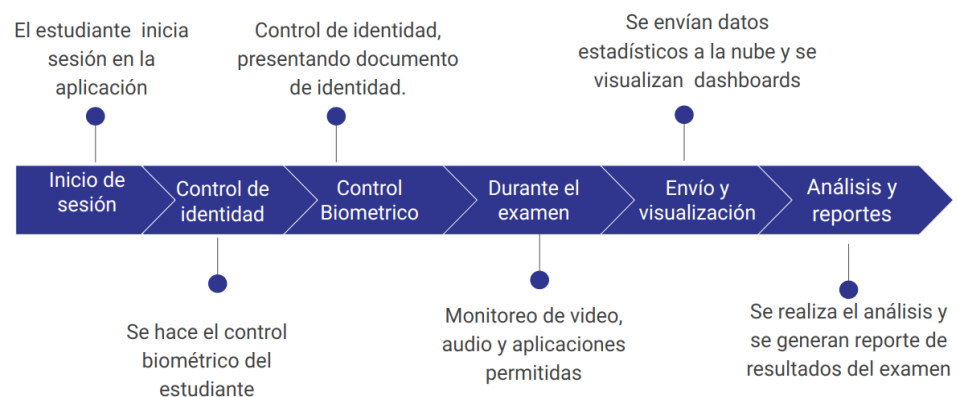


Elaborado por el equipo de trabajo.

Para el desarrollo del examen de admisión se contemplaron distintas fases, como se puede apreciar en la Figura 24.

Figura 24

Fases contempladas durante el examen de admisión.



Elaborado por el equipo de trabajo.



4.1.1.2. Características de la aplicación con modelos preentrenados y servicios en la nube para la adquisición de datos de exámenes en línea

Para el desarrollo de la aplicación se consideraron ciertas características producto de entrevistas las mismas que se describen en el backlog (ANEXO 1) y a continuación:

- **Condiciones generales para el examen:**
 - Tiempo para preguntas.
 - Tiempo de examen.
 - Cantidad de preguntas.
 - No se puede navegar entre preguntas.
- **Condiciones generales para el estudiante durante el examen:**
 - No realizar movimiento excesivo.
 - No girar la cabeza (no mirar abajo, arriba a la derecha, izquierda).
 - No mover la boca.
 - No hace ruido (no pronunciar palabras).
- **Restricciones del software**
 - No se podrá iniciar si es que se detecta dos cámaras web.
 - No se podrá iniciar si es que no se detecta micrófono.
 - No se podrá iniciar si es que se tiene varias pantallas conectadas.
 - No se podrá ingresar a la aplicación si se ejecuta en una máquina virtual.
 - El software verifica una lista (Black List) de aplicaciones restringidas, en caso de que el estudiante intente abrirlas. Un ejemplo se muestra en la Figura 25:

Figura 25

Lista de aplicaciones restringidas.

```
{psutil.Process(pid=9760, name='svchost.exe',  
status='running', started='14:49:36'),  
psutil.Process(pid=4892, name='WerFault.exe',  
status='running', started='14:50:46'),  
psutil.Process(pid=8592, name='obs64.exe',  
status='running', started='14:49:12'),  
psutil.Process(pid=10868, name='svchost.exe',  
status='running', started='14:50:46')}
```

Elaborado por el equipo de trabajo.

- Al ingresar al examen se tiene un temporizador de 10 segundos como mínimo para contestar y pasar a la siguiente pregunta.
- Hay un proceso que envía peticiones HTTP-GET para comprobar el estado de cada etapa en el examen.
- Se genera un archivo JSON de los datos del estudiante de todos los estados como la autenticación, datos del usuario, captura de foto, captura de DNI e inicio del examen. El formato del archivo JSON se muestra en la Figura 26.

Figura 26

Formato del archivo JSON.

```
{  
  
  "dni": "45454545",  
  
  "hora_login": "Jul. 23, 2022 | 09:58:23",  
  
  "escuela_profesional": "Derecho",  
  
  "hora_form_user": "Jul. 23, 2022 | 10:00:11 ",  
  
  "estado_foto": "foto_sin_parecido",  
  
  "hora_img_foto": "Jul. 23, 2022 | 10:02:08",  
  
  "hora_img_dni": "Jul. 23, 2022 | 10:03:11",  
  
  "hora_ini_examen": "Jul. 23, 2022 | 10:05:37"  
  
}
```

Elaborado por el equipo de trabajo.

Donde:

“dni”: es el DNI del estudiante.

“hora_login”: Hora de la autenticación del estudiante (estado uno).

"escuela_profesional": Escuela profesional que le corresponde al estudiante.

"hora_form_user": Hora en que el estudiante ha seleccionado su escuela profesional y ha corroborado sus datos (estado dos).

"estado_foto": Es el estado de la foto, el cual se obtiene al comparar una foto previa con la actual (adquirida al momento de captura de foto).

"hora_img_foto": Hora de la captura de la foto del estudiante.



"hora_img_dni": Hora de la captura del DNI del estudiante.

"hora_ini_examen": Hora del inicio del examen.

- Se genera un archivo JSON al contestar las preguntas del examen y esta se va actualizando conforme se van contestando las preguntas.
- Existe un servicio que verifica las carpetas (“exam”, ”stage”, ”process”, ”events”, ”mic” y “sshot”) donde fueron guardadas la foto, dni, los datos del examen, los archivos JSON, los videos y audios. Si existen cambios o archivos nuevos se procede a enviar a Amazon S3.
- Existe un proceso que verifica la conexión a internet en cada momento.
- La aplicación no se puede cerrar una vez que se ingrese.
- Existe un proceso que verifica y comprueba si el examen ha culminado y también recibe información (mensaje a los estudiantes).
- Existe un proceso que captura video en segundo plano
- Existe un proceso que captura el audio en segundo plano.
- Existe un proceso que controla cuando la aplicación se cierra, ya sea esto porque se culminó el examen o cuando recibe una petición HTTP GET desde el API Gateway.

4.1.1.3. Módulos de la aplicación

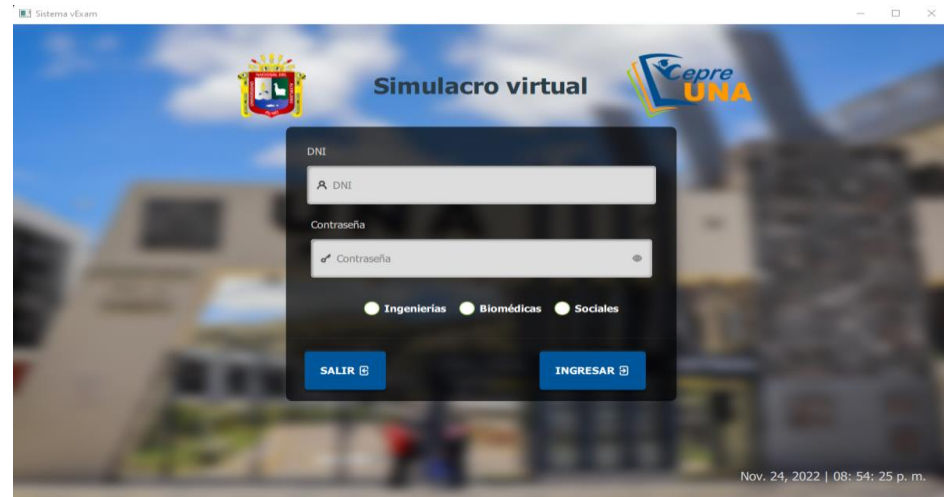
4.1.1.3.1. Inicio de sesión

En esta fase el usuario debe de colocar el nombre de usuario y la contraseña que previamente se les ha compartido, también se debe seleccionar el área del estudiante, si no tuvieran dichos accesos también se tiene la opción de SALIR. No obstante, en esta fase la interfaz de autenticación maximiza en toda la pantalla y se verifican las restricciones

de audio, cámara, etc. La interfaz del inicio de sesión se puede ver en la Figura 27.

Figura 27

Interfaz de inicio de sesión en la aplicación.



Elaborado por el equipo de trabajo.

4.1.1.3.2. Datos del estudiante

Después de que el estudiante se autentifique, este debe corroborar sus datos y selecciona la escuela profesional al que pertenece, finalizado este se activa el botón SIGUIENTE (Figura 28). Cabe señalar que en esta etapa se ejecutan las restricciones de aplicación y se abren los procesos para el control de los estudiantes como la peticiones HTTP GET, generar el archivo JSON para guardar datos. En esta fase se espera un acceso (activa el botón SIGUIENTE) que es obtenido de las peticiones realizadas al API Gateway, este es configurado para enviar estados (activo, desactivo y mensaje) y dar permiso al estudiante seguir la siguiente fase.

Figura 28

Interfaz de verificación de datos del estudiante.

Sistema vExam

Est: Biomédicas

CEPRE UNAP Datos del Estudiante

Nov. 24, 2022 | 08: 35: 45 p. m.

Cepre UNA

Datos del estudiante

Nombres: Calsin Cari Fredy

Escuela Profesional: Seleccione

DNI: 12345678

Área: Ingenierias

Sede: Juliaca

Colegio: Nacional

Ingrese su escuela profesional

SIGUIENTE

Elaborado por el equipo de trabajo.

4.1.1.3.3. Control biométrico

Luego de corroborar los datos del estudiante se pasa al control biométrico en el que la cámara se enciende y la aplicación detecta de manera automática el rostro del postulante esto se consigue gracias a un modelo pre entrenado basado en Haarcascade que permite la detección de rostros en tiempo real, el postulante puede capturar la imagen (Figura 29), también se puede reemplazar la imagen en caso de que la imagen no se tome adecuadamente, finalizado este proceso el botón SIGUIENTE se activa y se puede pasar al control de identidad. Cabe resaltar que en esta fase se espera un acceso (activa el botón SIGUIENTE) que es obtenido de las peticiones realizadas al API Gateway, este es configurado para enviar estados (activo, desactivo y mensaje) y dar permiso al estudiante seguir la siguiente fase.

Figura 29

Interfaz de control de identidad en la aplicación.



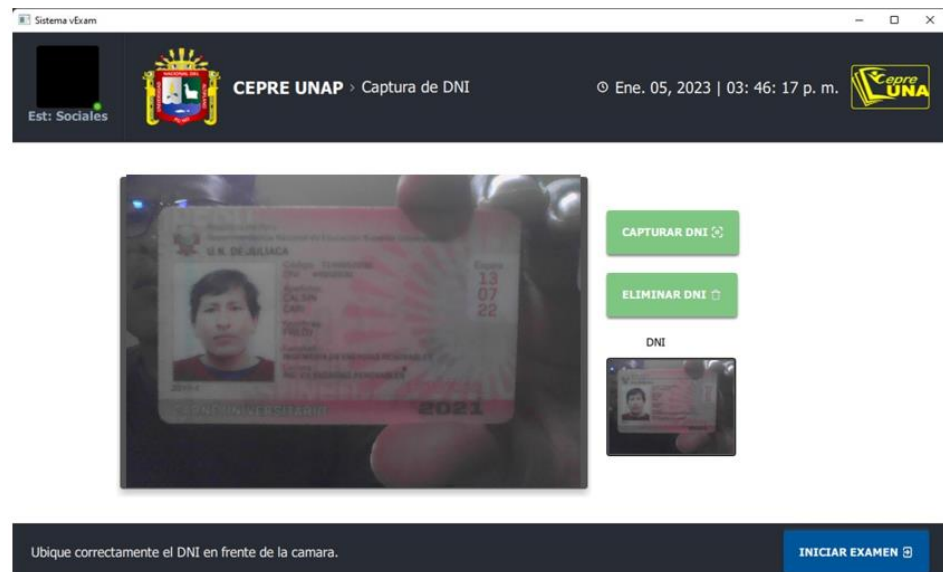
Elaborado por el equipo de trabajo.

4.1.1.3.4. Control de identidad

En esta fase se activa la cámara del equipo y el usuario debe de colocar su documento de identidad en el recuadro precisamente configurado y hacer la captura del documento (Figura 30), en el caso de que la imagen no se tome de manera adecuada la aplicación puede volver a intentarlo y eliminar la imagen, finalizado este proceso el botón SIGUIENTE se activará y ya se tiene listo para empezar el examen. Cabe resaltar que en esta fase se espera un acceso (activa el botón SIGUIENTE) que es obtenido de las peticiones realizadas al API Gateway, este es configurado para enviar estados (activo, desactivo y mensaje) y dar permiso al estudiante seguir la siguiente fase.

Figura 30

Interfaz de control de identidad en la aplicación.



Elaborado por el equipo de trabajo.

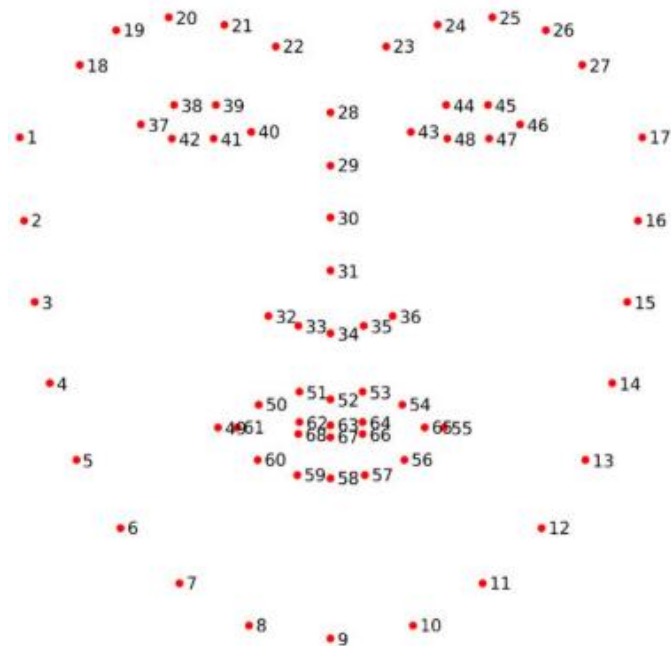
4.1.1.3.5. Reconocimiento Facial

El reconocimiento de puntos faciales es un problema que se aborda en imágenes y en video, las aplicaciones son muchas como el análisis de comportamiento facial, reconstrucción de rostros, edición de rostros y reconocimiento facial (Deng, J. et al., 2019)

En la Figura 31 se muestra un ejemplo de los puntos característicos de un rostro en entornos no controlados.

Figura 31

Puntos característicos de un rostro para el reconocimiento facial.

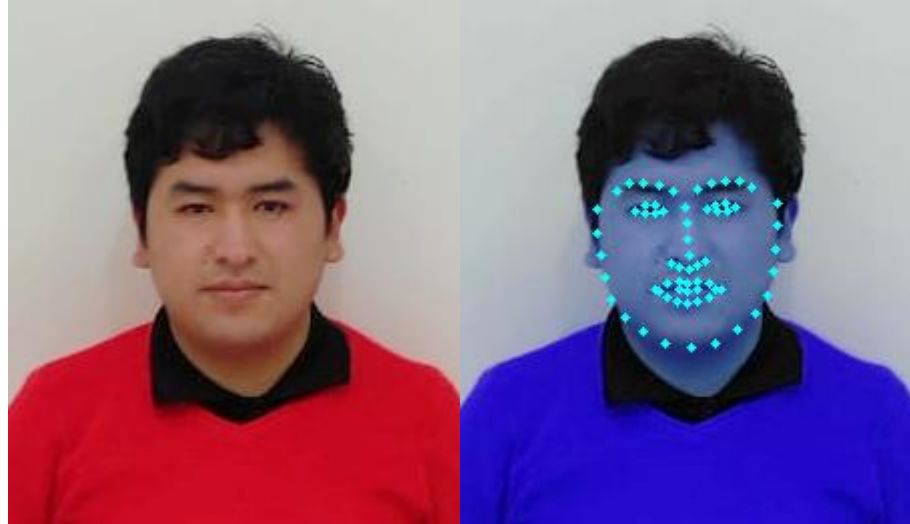


Nota: Adaptado de Deng, J. et al., 2019.

En el presente trabajo también se está utilizando reconocimiento facial para evitar la suplantación de postulantes para ello las personas o postulantes previamente han tenido que acudir a la toma de imagen en la que el personal de la Universidad Nacional del Altiplano valida la identidad del postulante. Para el reconocimiento facial se utiliza un modelo pre-entrenado el cual calcula un vector de características de la imagen que se obtuvo en la toma de imagen y mediante una métrica de distancia (distancia euclidiana) y un umbral previamente definido se valida la identidad del postulante, para este procedimiento se usan los puntos característicos (Figura 31) que permitirá el reconocimiento facial como se muestra en la Figura 32.

Figura 32

Puntos característicos de un rostro que permiten el reconocimiento facial.



Elaborado por el equipo de trabajo.

Figura 33

Reconocimiento facial mediante puntos característicos.



Elaborado por el equipo de trabajo.

4.1.1.3.6. Durante el examen

Una vez pasados los controles de identidad y control biométrico se pasan a esta fase en la que el estudiante puede rendir la evaluación, en la interfaz, en la parte superior izquierda se puede visualizar la cámara en tiempo real y en la parte superior derecha se puede monitorear en tiempo real la señal de cantidad de movimiento en un gráfico que corresponde a una serie de tiempo. Adicionalmente se ha colocado un semáforo el cual de acuerdo a un umbral previamente establecido se marca en colores amarillo rojo y verde cuando se registra cierta cantidad de movimiento y sonido, movimiento/sonido en condiciones normales, movimiento/sonido moderado y movimiento/sonido excesivo.

Ambas señales están sujetas a un umbral previamente definido, cuando se supera este umbral se almacena un archivo y con un servicio previamente definido se envían estos a AWS S3, que es un servicio de almacenamiento de objetos, los mismos que sirven para la construcción del dataset.

En la parte inferior se muestra un botón SIGUIENTE, este es activado si ha transcurrido el tiempo mínimo (diez segundos) y la conexión a internet es estable. Al costado izquierdo del botón SIGUIENTE se mostrará un mensaje si es conveniente, el mensaje es configurado en la API Gateway.

En la parte central de la Figura 36 se muestra la pregunta con sus respectivas alternativas, el estudiante puede seleccionar o dejar en blanco,

al hacer clic en SIGUIENTE se guardará la pregunta con su respectiva respuesta en un archivo JSON y esta será enviada a Amazon S3, este proceso se repite hasta que muestre la última pregunta, en donde cambia el botón SIGUIENTE por TERMINAR Y SALIR. Cabe resaltar que el botón TERMINAR y SALIR puede aparecer por dos razones, cuando el estudiante termina el examen o cuando el servicio que obtiene estados del API Gateway de por terminado el examen. La estructura del formato JSON para una pregunta del examen virtual se muestra en la Figura 34.

Figura 34

Estructura del archivo JSON para una pregunta del examen.

```
[
  {
    "curso": "Razonamiento-matematico_1",
    "alternativas": [
      "0",
      "0",
      "0",
      "0",
      "0"
    ],
    "responseValue": "",
    "path": "Razonamiento-matematico_1.jpg",
    "datetime": "2022-7-24 15:15:37"
  }
]
```

Elaborado por el equipo de trabajo.

Existe un formato JSON (Figura 35) de las alternativas válidas (correctas) con sus respectivas preguntas, este servirá para comprobar y evaluar el examen del estudiante.

Figura 35

Formato JSON de las alternativas válidas.

```
[  
  {"curso":"Razonamiento-matematico_1","peso":"7","alternativas":  
  ["0","0","0","0","1"],"path":"Razonamiento-matematico_1.jpg"},  
]
```

Elaborado por el equipo de trabajo.

Donde:

“curso”: representa al curso con su respectiva numeración.

“peso”: representa el peso de la pregunta.

“alternativas”: guarda la alternativa marcada por el usuario, donde el orden de estas es la siguiente: a, b, c, d, e.

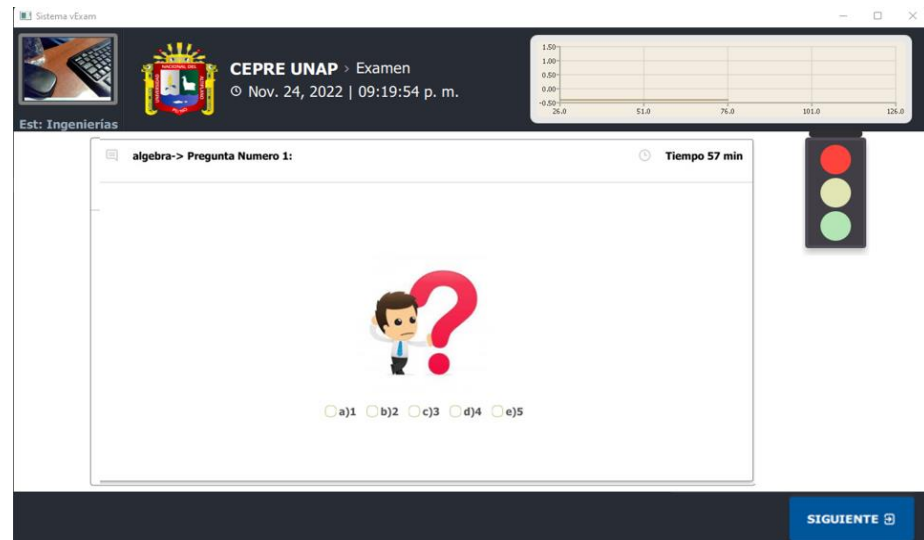
"responseValue": es la alternativa marcada por el estudiante.

“path”: es la dirección de una imagen recortada que representa la pregunta.

"datetime": la fecha en que fue registrada o marcada la alternativa.

Figura 36

Preguntas del examen.



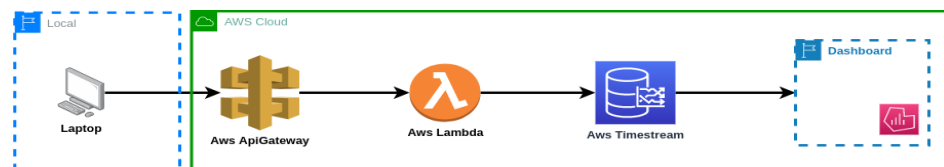
Elaborado por el equipo de trabajo.

4.1.1.3.7. Envío y visualización

En este componente se puede visualizar en tiempo real la cantidad de postulantes y en qué etapa se encuentran para esto se está recibiendo en la nube peticiones desde que son canalizadas a través de los servicios de Amazon como se puede visualizar en la Figura 37.

Figura 37

Envío y visualización de usuarios en Grafana.



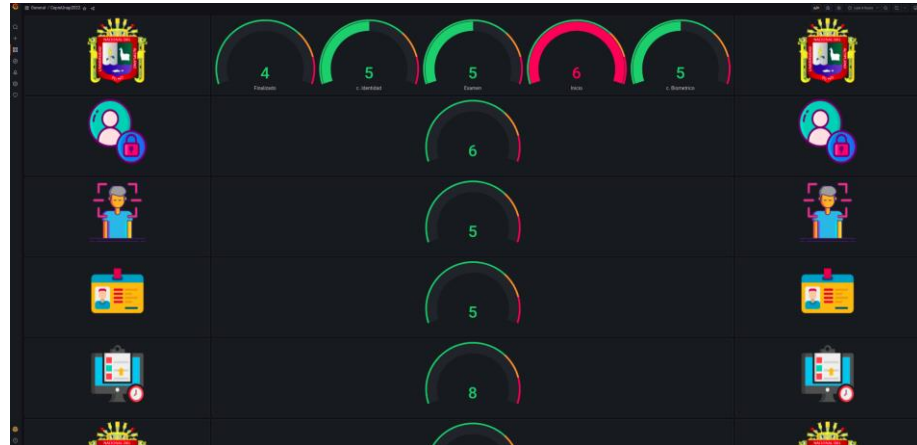
Elaborado por el equipo de trabajo.

Los datos que son almacenados en Aws Timestream son consultados desde Grafana para posteriormente ser visualizados en un

Dashboard (Figura 38) estos datos puedes ser visualizados de distintas formas (Figura 39)

Figura 38

Dashboard con estadísticas propias del examen.



Elaborado por el equipo de trabajo.

Figura 39

Dashboard con estadísticas del examen.



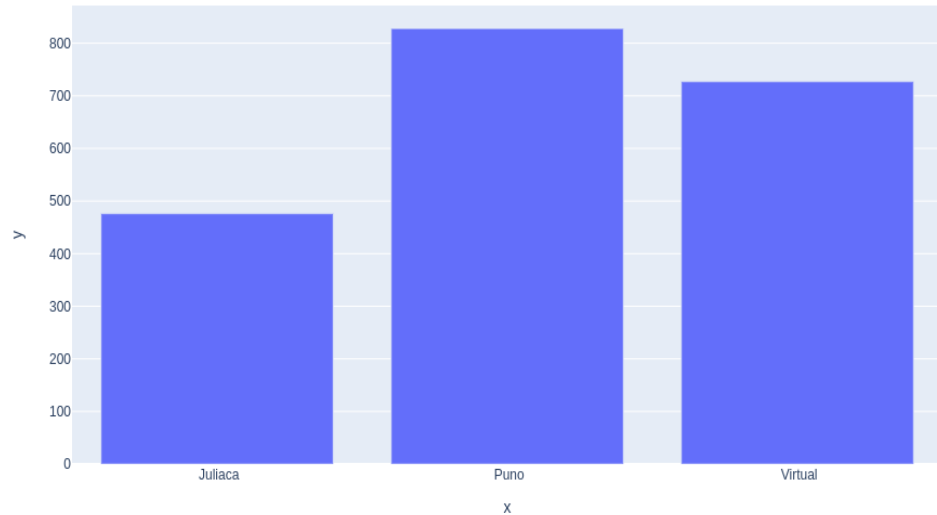
Elaborado por el equipo de trabajo.

4.1.1.3.8. Análisis y reportes

En este componente se contempló realizar estadísticas del uso de la aplicación y resultados del examen de simulacro, por ejemplo, se hizo una comparación del resultado de los estudiantes que asistente de manera presencial en las sedes Juliaca y Puno y de forma virtual (Figura 40).

Figura 40

Resultado del examen de estudiantes que asisten en de forma presencial en Juliaca y Puno y de forma virtual en diferentes sedes.



Elaborado por el equipo de trabajo.

4.1.1.3.9. Algoritmo para el cálculo de notas

Tabla 2

Algoritmo para el cálculo de notas.

Algorithm 1

Require: Dni de los estudiantes

Ensure: Nota de cada estudiante

```
1: import library
2: DNI<-getAllDniDatabase()
3: m<-0
3: for record in DNI
4:   try:
5:     data_json<-open('data.json')
6:   except:
7:     data_json<-""
8:   estudiante<-open('exam.json')
10:  data<-json.load(estudiante)
11:  areas<-["Ingenierias","Biomedicas","Sociales"]
12:  data_student<-[]
13:  for area in areas:
```

```
14:   correctas=open(area+“.json”)
15:   i<-0
15:   while i<cantidad_preguntas:
16:     j<-0
17:     k<-0
18:     if data_correctas[i]==data[i]
19:       similitud<-similitud+1
20:     while j<5:
21:       if data_correctas[correctas][i]==data[examen][i]:
22:         k<-k+1
23:         j<-j+1
24:       nota<-nota+k*10*peso
25:     guardarNota(nota)
```

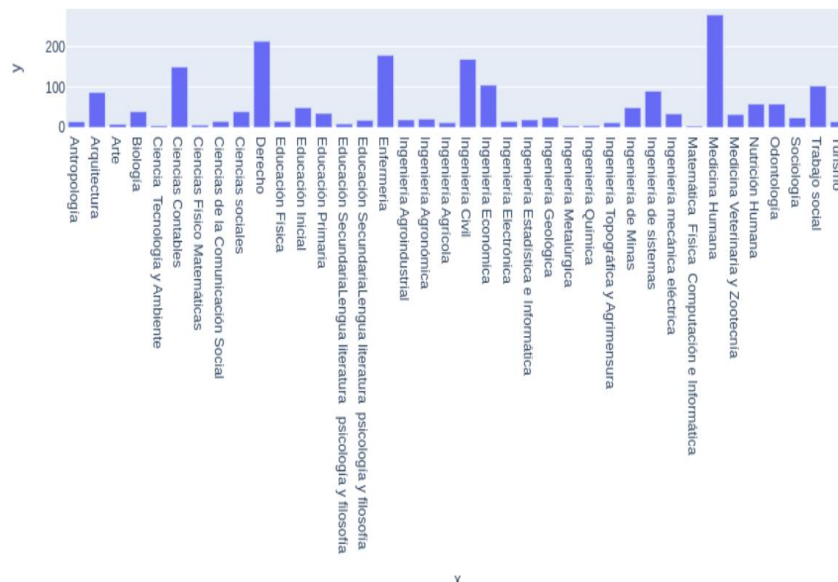
Elaborado por el equipo de trabajo.

4.1.1.3.10. Resultados del simulacro de examen de Admisión

En la Figura 41 y Figura 42 podemos visualizar los resultados del examen de admisión por escuela profesional de forma gráfica y numérica.

Figura 41

Resultados del examen de simulacro por escuela profesional



Elaborado por el equipo de trabajo.

Figura 42

Resultados del examen de simulacro.

Antropología	14
Arquitectura	87
Arte	8
Biología	39
Ciencia Tecnología y Ambiente	4
Ciencias Contables	150
Ciencias Físico Matemáticas	6
Ciencias de la Comunicación Social\n	15
Ciencias sociales	39
Derecho	214
Educación Física	15
Educación Inicial	49
Educación Primaria	35
Educación SecundariaLengua literatura psicología y filosofía	9
Educación SecundariaLengua literatura psicología y filosofía	18
Enfermería	179
Ingeniería Agroindustrial	19
Ingeniería Agronómica	21
Ingeniería Agrícola	12
Ingeniería Civil	169
Ingeniería Económica	105
Ingeniería Electrónica	15
Ingeniería Estadística e Informática	19
Ingeniería Geológica	25
Ingeniería Metalúrgica\n	4
Ingeniería Química	5
Ingeniería Topográfica y Agrimensura	12
Ingeniería de Minas	49
Ingeniería de sistemas	90
Ingeniería mecánica eléctrica	34
Matemática Física Computación e Informática	3
Medicina Humana	279
Medicina Veterinaria y Zootecnia	32
Nutrición Humana	58
Odontología	58
Sociología	24
Trabajo social	103
Turismo	14

Elaborado por el equipo de trabajo.

4.1.1.4. Dataset de eventos de exámenes en línea.

Para la elaboración del presente trabajo se construyó un dataset de secuencias de vídeo y secuencias de audio, los mismos que se extrajeron del simulacro de examen de admisión del centro pre universitario de la Universidad Nacional del Altiplano que se realizó el 23 de julio del 2022, en el que se recolectaron 143 Gigas de información en Amazon s3 que corresponden a 308 mil archivos (Figura 43), estos provienen del control biométrico, control de identidad, archivos de configuración, archivos de audio y video.

En el cuadro 3, se muestra la cantidad de videos cuando el estudiante observa al lado derecho, izquierda, abajo, arriba, frente e imágenes de estudiantes sin rostro (ANEXO 2); en el cuadro 4 se puede apreciar la cantidad de total de audios y videos registrados.

Tabla 3

Cantidad de videos categorizados.

						Sin
	Derecha	Izquierda	Abajo	Arriba	Frente	Rostro
Cantidad	13770	2871	134131	5630	16249	7373
					Total	180024

Elaborado por el equipo de trabajo.

Tabla 4

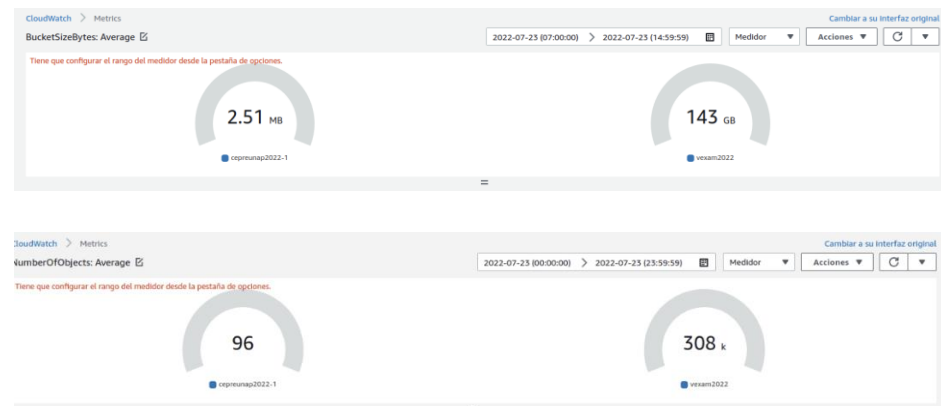
Cantidad total de audios y videos.

	Videos	Audio
Cantidad	180024	115292

Elaborado por el equipo de trabajo.

Figura 43

Cantidad de datos recolectados durante el simulacro.



Elaborado por el equipo de trabajo.

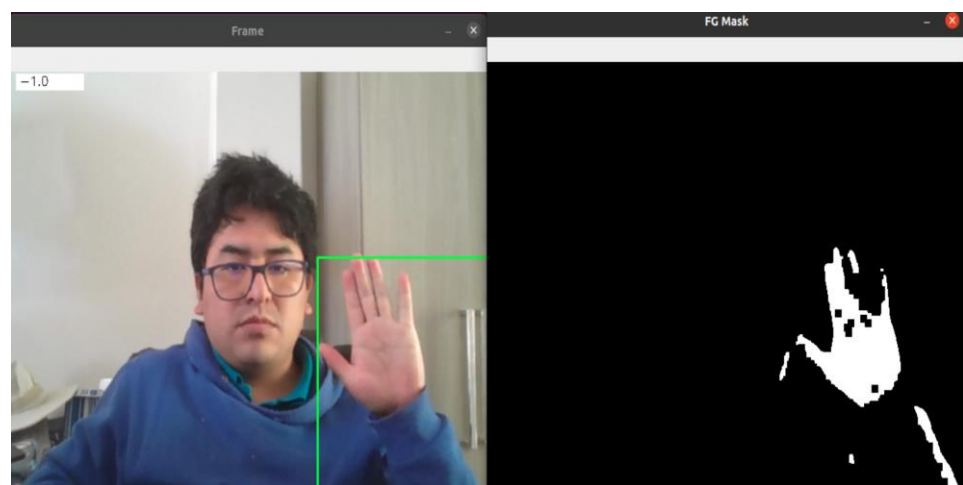
4.1.1.4.1. Dataset de anomalías en video

4.1.1.4.1.1. Dataset de movimiento excesivo

Para la elaboración del dataset de secuencias de vídeo se ha desarrollado un algoritmo que detecta el movimiento para esto se utilizó la técnica llamada sustracción de fondo (Figura 44) la misma que consiste en la resta de imágenes consecutivas y los cambios (movimiento) se muestran como píxeles en blanco, estos píxeles en blanco se contabilizan y sobre esta cantidad se establece un umbral, si el umbral es superado se considera que se ha producido movimiento excesivo.

Figura 44

Sustracción de fondo en secuencias de video.

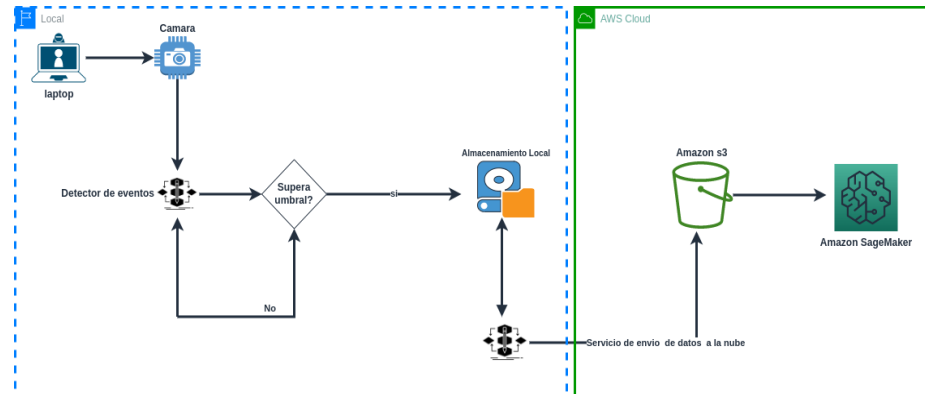


Elaborado por el equipo de trabajo.

Cuando el umbral previamente establecido es superado el algoritmo almacena en un directorio local que posteriormente un servicio se encarga de subir los archivos al servicio de Amazon s3 en los que se almacena, como se ilustra en la Figura 45.

Figura 45

Algoritmo para la construcción y almacenamiento de un dataset de señales de video.



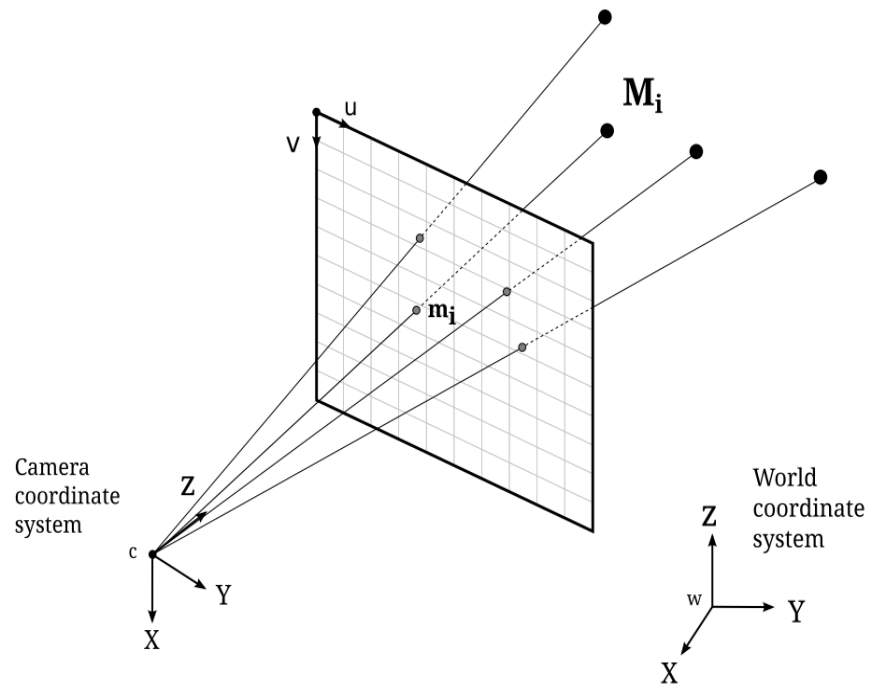
Elaborado por el equipo de trabajo.

4.1.1.4.1.2. Dataset de postura de la cabeza

La calidad del dataset es la parte más crítica para el presente trabajo sea creado un dataset de poses de la cabeza para ello sea resuelto el PnP (Perspective n Point Problem), en el que se trata de estimar la pose de una cámara calibrada dado un conjunto de n puntos 3D en el mundo y sus correspondientes proyecciones 2D en la imagen, como se muestra en la Figura 46.

Figura 46

Esquema del problema PnP, dado un conjunto de puntos 3D y sus proyecciones 2D sobre la imagen, se busca recuperar la pose (R y T).



Nota: Adaptado de Riba Pi, E, 2015.

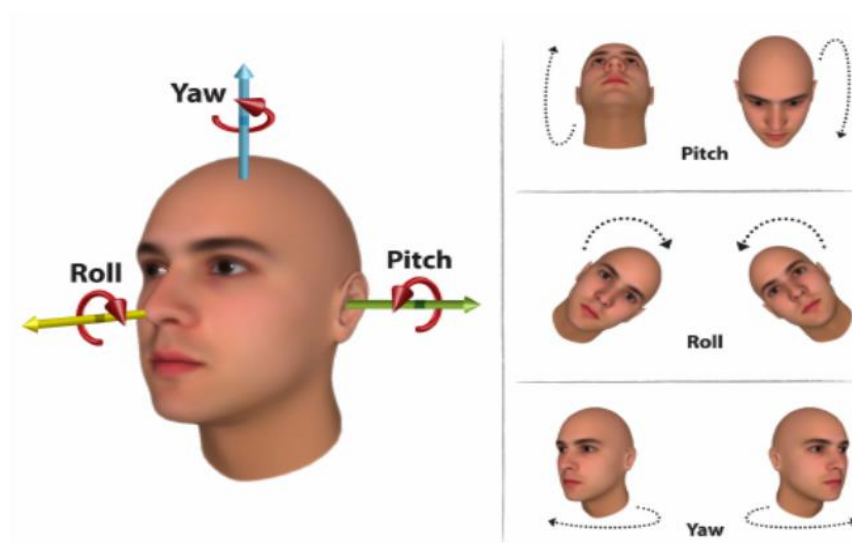
Teniendo en cuenta lo explicado anteriormente se establecieron umbrales respecto a los vectores de rotación que denotan el movimiento de la cabeza dependiendo del movimiento de estos vectores.

4.1.1.4.1.3. Dataset de eventos anómalos

Se establecieron etiquetas a los clips de video cuando estas superaban el umbral previamente establecido (Figura 47), siendo las etiquetas: derecha, izquierda, arriba, abajo y sin rostro (Figura 48, Figura 49, ANEXO 5) Entendiendo que durante un examen de admisión estos movimientos no están permitidos por lo tanto son considerados anómalos.

Figura 47

Orientación de la cabeza con los ángulos yaw, pitch y roll



Nota: Grados de libertad de una cabeza humana Arcoverde Neto, E. N.2014.

Figura 48

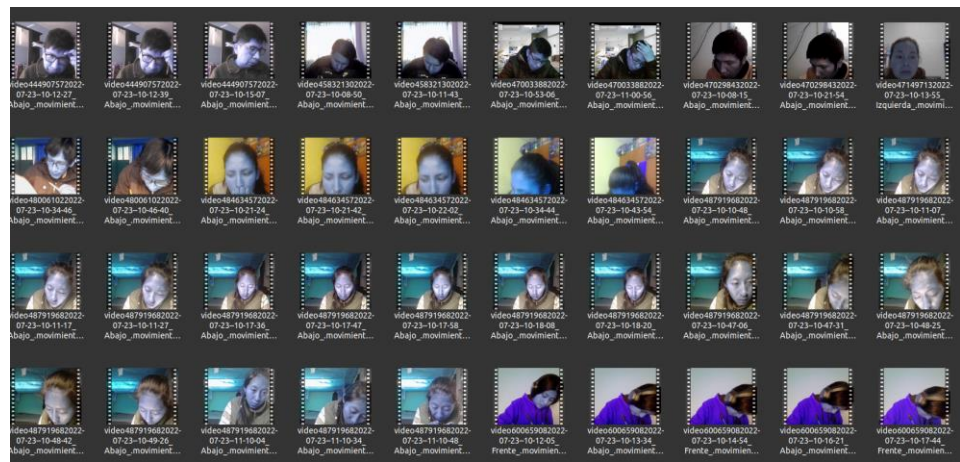
Imágenes de movimientos hacia arriba, abajo, derecha e izquierda, son considerados anómalos



Elaborado por el equipo de trabajo.

Figura 49

Dataset de videos de anomalías en exámenes en línea.



Elaborado por el equipo de trabajo.

4.1.1.4.1.4. Dataset de eventos normales

Así también se creó una etiqueta de frente, esta etiqueta se asigna cuando la cabeza está de frente a la cámara sin movimiento rotatorio (Figura 50), estos movimientos son considerados normales (Figura 51).

Figura 50

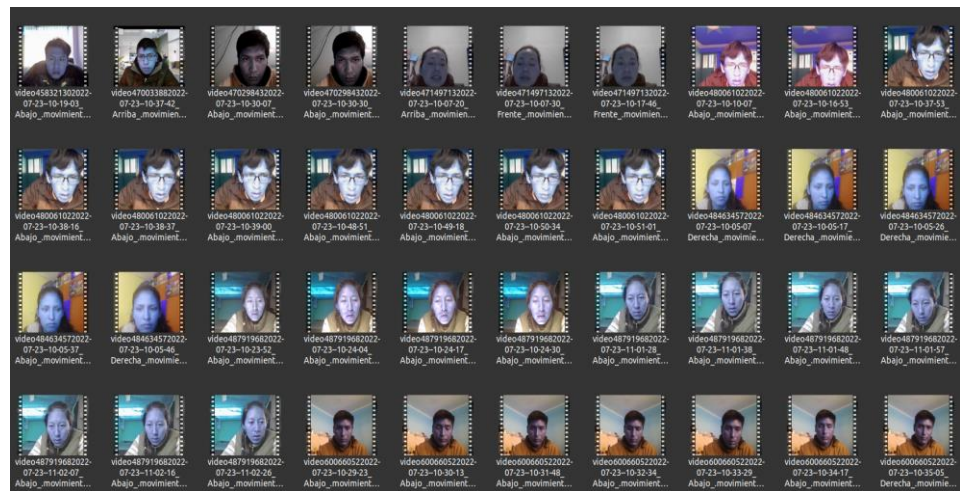
Movimiento de frente considerado normal.



Elaborado por el equipo de trabajo.

Figura 51

Dataset de eventos normales.



Elaborado por el equipo de trabajo.

4.1.1.4.1.5. Construcción del Dataframe

Luego del etiquetado manual las muestras se han alojado en directorios desde los cuales han sido leídos por un algoritmo que se ha desarrollado para la construcción de un dataframe con los ángulos X, Y y Z que representan los movimientos de la postura de la cabeza, el nombre del archivo desde el cual han sido extraídos y una clase en este caso se asignó la clase 1 como anomalías y la clase 2 como normal, como se muestra en la Figura 52.

Figura 52

Dataframe construido del movimiento de la postura de la cabeza

Unnamed: 0	Nombre	x	y	z	tipo
0	dataset/normales/video714865302022-07-23--10-5...	5.197004	-2.151005	-0.127771	2
1	dataset/normales/video714865302022-07-23--10-5...	5.382286	-2.724528	-0.123090	2
2	dataset/normales/video714865302022-07-23--10-5...	5.218100	-3.512717	-0.140746	2
3	dataset/normales/video714865302022-07-23--10-5...	5.205775	-3.324547	-0.130182	2
4	dataset/normales/video714865302022-07-23--10-5...	5.221635	-3.659923	-0.142273	2
5	dataset/normales/video714865302022-07-23--10-5...	5.606001	-3.495542	-0.133704	2
6	dataset/normales/video714865302022-07-23--10-5...	5.431803	-3.515042	-0.130705	2
7	dataset/normales/video714865302022-07-23--10-5...	5.071761	-3.611175	-0.136641	2
8	dataset/normales/video714865302022-07-23--10-5...	5.108971	-3.471833	-0.131493	2
9	dataset/normales/video714865302022-07-23--10-5...	5.237356	-3.509872	-0.137258	2
10	dataset/normales/video714865302022-07-23--10-5...	5.289860	-3.486947	-0.131402	2
11	dataset/normales/video714865302022-07-23--10-5...	5.309353	-3.652214	-0.135214	2
12	dataset/normales/video714865302022-07-23--10-5...	5.072923	-3.783748	-0.135595	2
13	dataset/normales/video714865302022-07-23--10-5...	4.983398	-4.192100	-0.128398	2
14	dataset/normales/video714865302022-07-23--10-5...	5.028680	-3.652382	-0.132748	2

Elaborado por el equipo de trabajo.

El dataset está constituido por un total de 386 videos de eventos normales y anómalos que se distribuyen en la Tabla 5.

Tabla 5

Dataset de anomalías para el entrenamiento

	Anomalías	Normales
Cantidad	96	290

Elaborado por el equipo de trabajo.

4.1.2. Entrenar los modelos en la detección de anomalías en secuencias de video mediante inteligencia artificial

4.1.2.1. Análisis exploratorio de datos

El Análisis exploratorio de datos es un método muy utilizado para el entendimiento de los datos a fin de seleccionar las variables adecuadas,

entender la distribución de las mismas, para ello se utilizan diagramas de caja y bigote, histogramas y series de tiempo. El diagrama de caja y bigote nos muestra los cuartiles, la media de los datos, los histogramas nos dan información sobre la distribución y las series de tiempo nos muestran cómo cambian las señales en el tiempo.

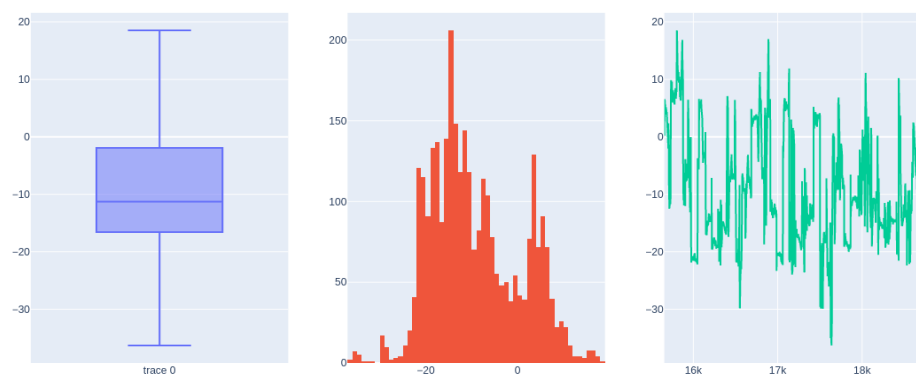
4.1.2.2. Movimientos de la cabeza en los distintos ejes

Entendiendo que ya se ha construido un dataframe con el movimiento de la cabeza en los ejes X, Y, Z, tenemos clases que representan si es un evento anómalo o no, se han realizado los diagramas correspondientes (Figura 53, 54, 55, 56, 57, 58).

Movimiento de la cabeza en el eje X

Figura 53

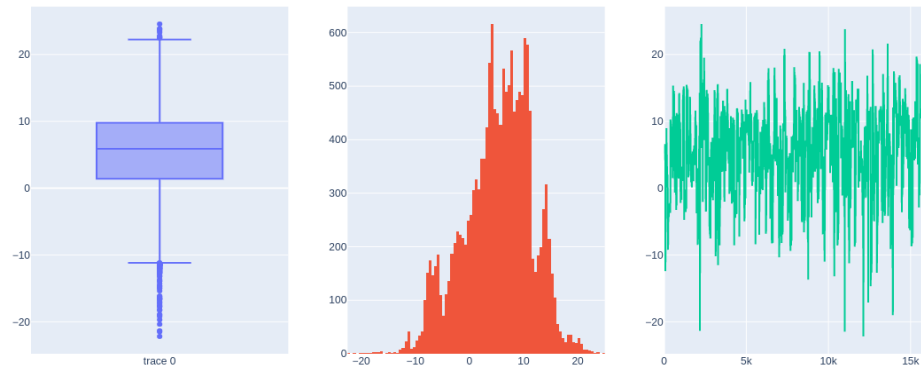
Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje X de muestras anómalas.



Elaborado por el equipo de trabajo.

Figura 54

Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje X de muestras normales.

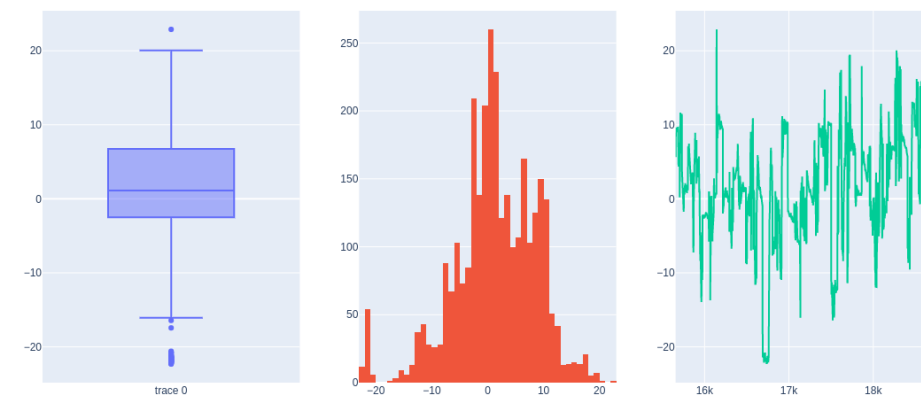


Elaborado por el equipo de trabajo.

Movimiento de la cabeza en el eje Y

Figura 55

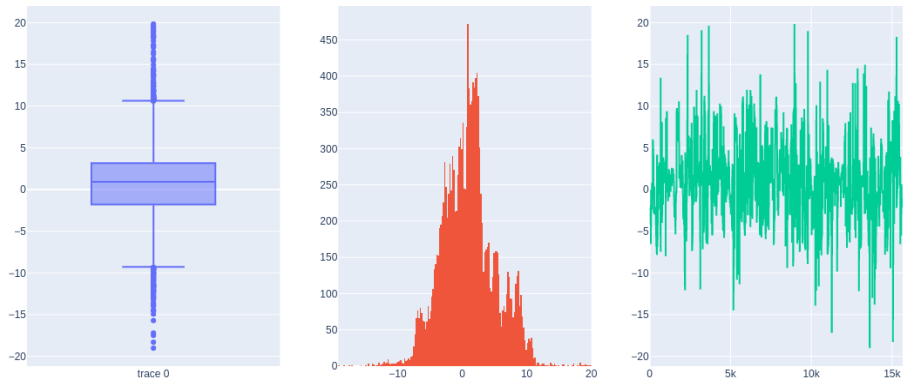
Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras anómalas.



Elaborado por el equipo de trabajo.

Figura 56

Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras normales.

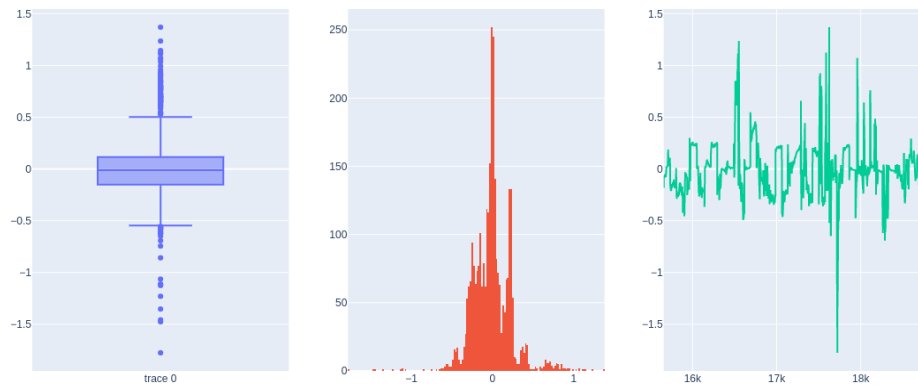


Elaborado por el equipo de trabajo.

Movimiento de la cabeza en el eje Z

Figura 57

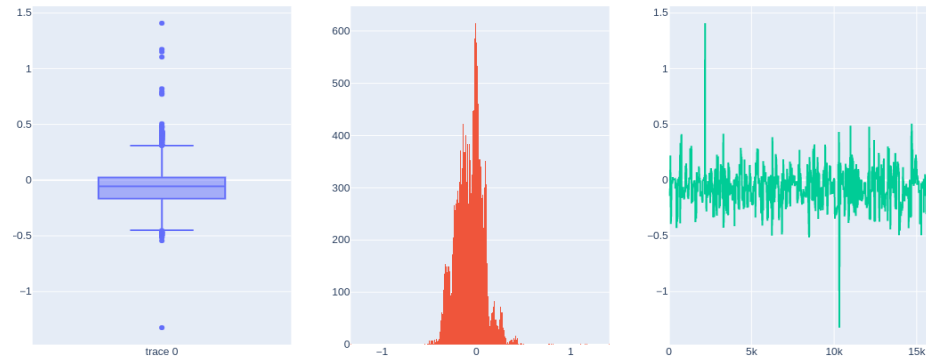
Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Z de muestras anómalas.



Elaborado por el equipo de trabajo.

Figura 58

Diagrama de caja y bigotes, distribución y series de tiempo del movimiento de la cabeza en el eje Y de muestras normales.



Elaborado por el equipo de trabajo.

4.1.2.3. Preprocesamiento y normalización de datos

Uno de los procesos más habituales en los proyectos que conllevan análisis de datos es el preprocesamiento, donde se eliminan, imputan valores nulos o atípicos que puedan afectar el desempeño de los resultados del modelo. Otro proceso importante es la normalización de los datos, generalmente estos datos se convierten en valores entre 0 y 1 para evitar la redundancia de los mismos.

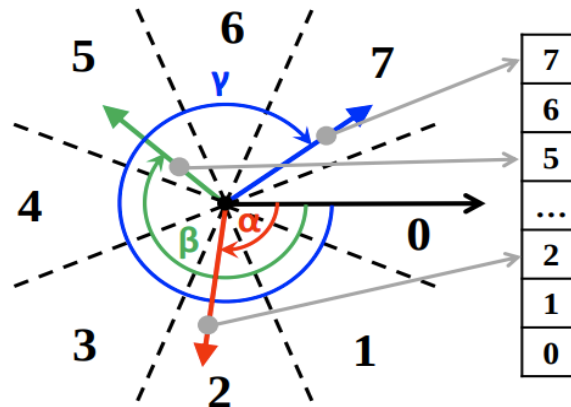
4.1.2.4. Extracción de características de movimiento de la postura de la cabeza

En visión artificial se suelen utilizar algoritmos que ayudan a extraer características de acuerdo al problema que se está abordando pueden ser características de movimiento, forma, etc. En el presente trabajo se tiene el movimiento descrito por la postura de la cabeza los mismos que se dan en tres ejes X, Y, y Z, para ello se ha construido

descriptores de movimiento por cada eje, los cuales tienen 16 bins que tienen un rango de 5 grados, los mismos que se van construyendo, en la Figura 59 se muestra un histograma con 7 bins.

Figura 59

Histograma orientado de flujo óptico con 7 bins que describen 360°



Nota: Adaptado de Liu, et al., 2015.

Una vez construidos los histogramas de movimiento por cada eje, se procede a agrupar los mismo en un dataframe en el que se mantiene el nombre de la muestra, se almacena el histograma de movimiento y se almacena el tipo de muestra siendo las clases 1 una clase que representa a las muestras anómalas y la clase 2 a las muestras normales, como se muestra en la Figura 60.

Figura 60

Dataframe que contiene el nombre, histograma de movimiento y el tipo.

Name	histogram	type
0 dataset/anomalas/video444907572022-07-23--10-1...	[0.0, 0.0, 0.0, 0.0, 34.42906714384863, 19.479...	1
0 dataset/anomalas/video444907572022-07-23--10-1...	[0.0, 0.0, 0.0, 0.0, 47.75378489116466, 89.498...	1
0 dataset/anomalas/video444907572022-07-23--10-1...	[0.0, 0.0, 0.0, 425.6831280459175, 16.35000652...	1
0 dataset/anomalas/video458321302022-07-23--10-0...	[0.0, 0.0, 0.0, 0.0, 569.0675049195837, 275.68...	1
0 dataset/anomalas/video458321302022-07-23--10-1...	[0.0, 0.0, 0.0, 52.468528851658355, 96.9137028...	1
...
0 dataset/normales/video716973702022-07-23--10-3...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 141.018479...	2
0 dataset/normales/video716973702022-07-23--10-3...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 321.321697...	2
0 dataset/normales/video717011422022-07-23--10-2...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 26.288613809915...	2
0 dataset/normales/video717011422022-07-23--10-2...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.9201401188627...	2
0 dataset/normales/video717011422022-07-23--10-3...	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1212....	2

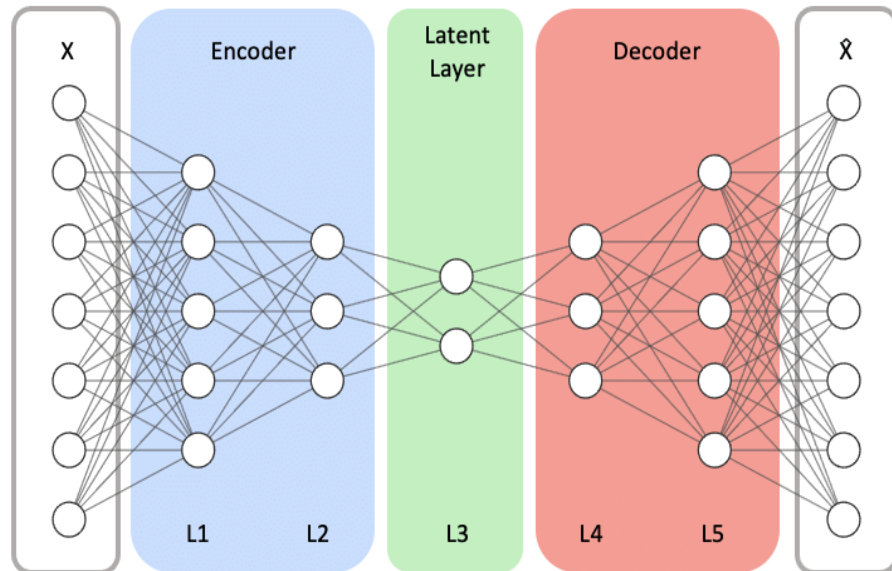
Elaborado por el equipo de trabajo.

4.1.2.5. Modelo

Los algoritmos de inteligencia artificial se utilizan cada vez más en problemas de clasificación, predicción, etc. Para el presente trabajo de se utilizó la arquitectura llamada ISOLATION FOREST, AUTOENCODER (Figura 61) y LSTM-AUTOENCODER (Figura 62), ya que la literatura ha demostrado que estas arquitecturas abordan de mejor manera el problema de la detección de anomalías que usualmente lidian con problemas de desbalance de datos, para lo cual se desarrolló un algoritmo que recibe como entrada los histograma orientados de movimiento por cada eje como una serie de tiempo en este caso compuesto por 48 elementos que describen el movimiento de un clip de video de 76 fotogramas de duración.

Figura 61

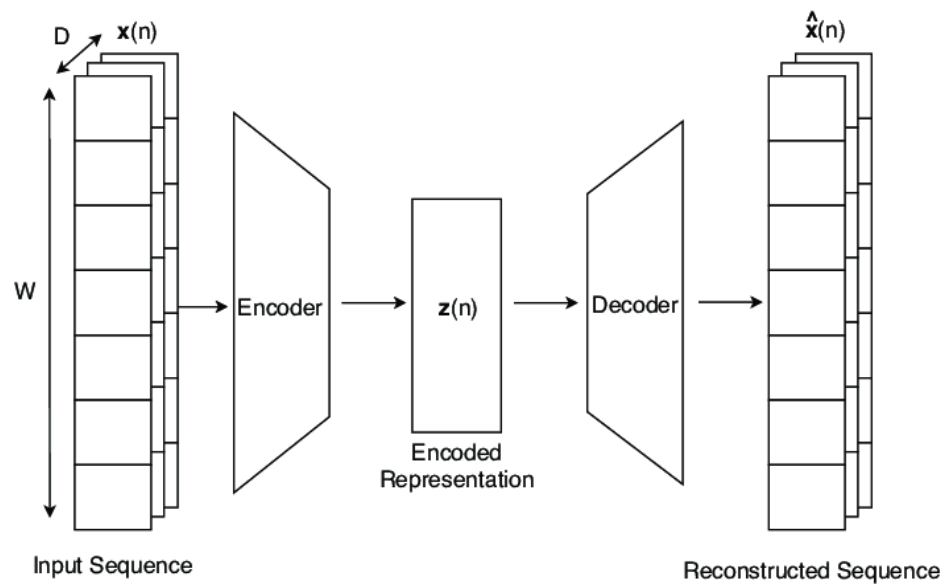
AUTOENCODER



Nota: Adaptado de Song, Y., et al. 2021.

Figura 62

LSTM-AUTOENCODER



Nota: Adaptado de Trinh, H. D., et al., 2019.



4.1.2.6. Entrenamiento de modelos

Una vez se ha construido el dataset de anomalías de exámenes mediante un histograma orientado de movimiento por eje X, Y y Z los cuales constituyen un vector de características de 48 elementos que ingresan a los modelos para el presente trabajo tres modelos un ISOLATION FOREST, AUTOENCODER y un LSTM-AUTOENCODER, estos modelos al ser entrenados tratan de minimizar la función error como se muestra líneas abajo.

4.1.2.6.1. Entrenamiento con LSTM-AUTOENCODER

Se realizó el entrenamiento con LSTM-AUTOENCODER, ya que la literatura ha demostrado que por ser una red neuronal recurrente puede almacenar información y tener mejores resultados cuando se presentan datos que cambian en el tiempo, primero se construyó el modelo como se muestra en la Figura 63.

Figura 63

Arquitectura del modelo 1.

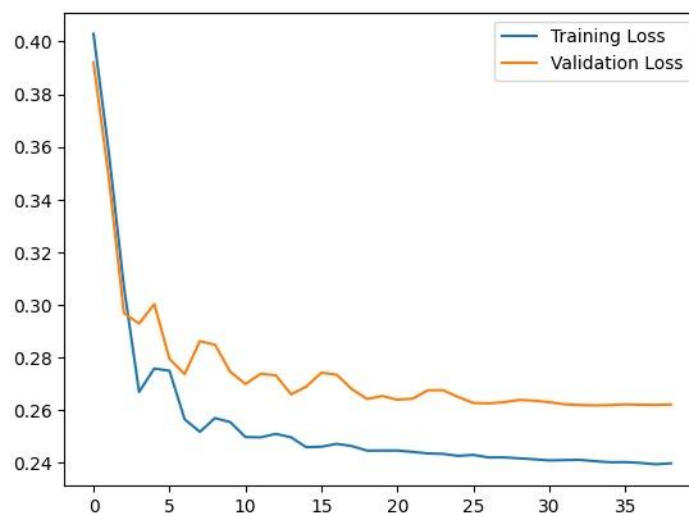
```
Model: "sequential"
Layer (type)                Output Shape                Param #
-----
lstm (LSTM)                  (None, 48)                  9600
dropout (Dropout)           (None, 48)                  0
repeat_vector (RepeatVector) (None, 48, 48)              0
lstm_1 (LSTM)                (None, 48, 48)             18624
dropout_1 (Dropout)         (None, 48, 48)              0
time_distributed (TimeDistributed) (None, 48, 1)              49
-----
Total params: 28,273
Trainable params: 28,273
Non-trainable params: 0
```

Elaborado por el equipo de trabajo.

Luego de haber de construido la arquitectura del modelo se procede a realizar el entrenamiento para lo cual se puede visualizar el comportamiento del entrenamiento en la Figura 64.

Figura 64

Entrenamiento del modelo 1



Elaborado por el equipo de trabajo.

4.1.2.6.1. Entrenamiento con AUTOENCODER

Se utilizó el AUTOENCODER ya que es muy utilizada para la detección de anomalías es por eso que resulta interesante poder experimentar con esta arquitectura, para el presente trabajo se realizó el presente entrenamiento se construyó el modelo como se muestra en la Figura 65.

Figura 65

Arquitectura del modelo

```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv1d (Conv1D)             (None, 24, 32)           256
dropout (Dropout)          (None, 24, 32)           0
conv1d_1 (Conv1D)           (None, 12, 16)           4112
conv1d_transpose (Conv1DTr (None, 24, 16)           1808
anspose)
dropout_1 (Dropout)         (None, 24, 16)           0
conv1d_transpose_1 (Conv1DT (None, 48, 32)           3616
ranspose)
conv1d_transpose_2 (Conv1DT (None, 48, 1)            225
ranspose)

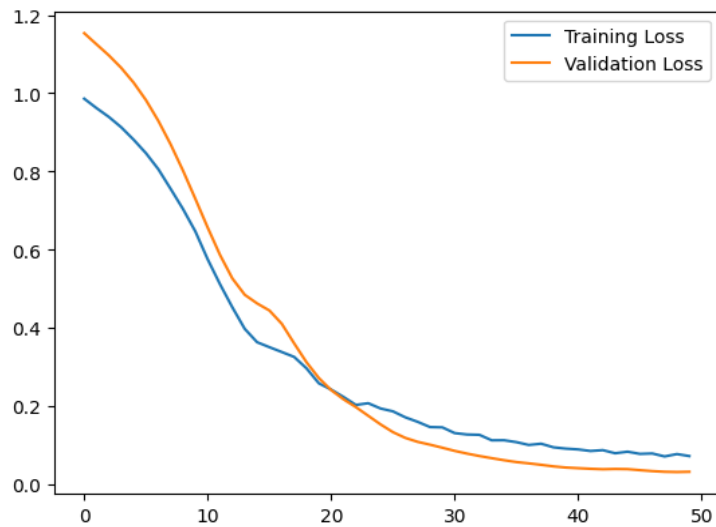
Total params: 10,017
Trainable params: 10,017
Non-trainable params: 0
```

Elaborado por el equipo de trabajo.

Luego de haber de construido la arquitectura del modelo se procede a realizar el entrenamiento para lo cual se puede visualizar el comportamiento del entrenamiento en la Figura 66.

Figura 66

Resultados del entrenamiento del Autoencoder.



Elaborado por el equipo de trabajo.

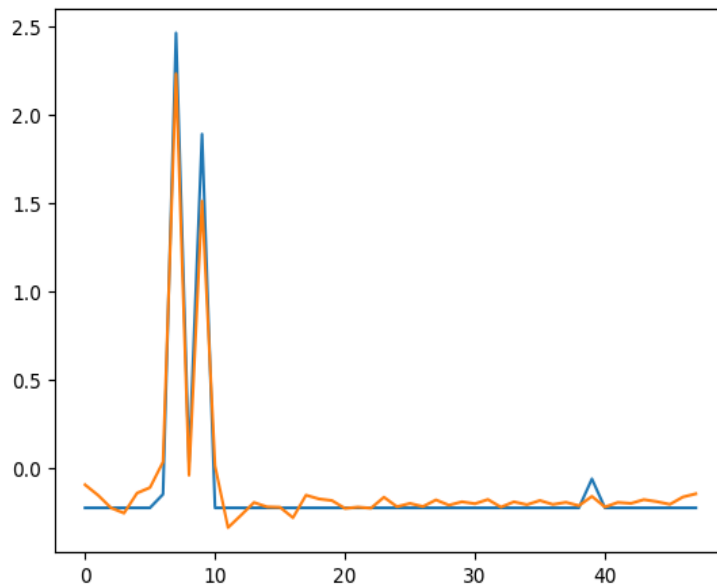
Se puede apreciar de las imágenes que el modelo que mejor a convergido en el entrenamiento es el del AUTOENCODER.

4.1.3. Evaluar y comparar los modelos en la detección de anomalías en secuencias de video en exámenes en línea

Entendiendo que se está abordando un problema con desbalance de datos y se tienen más muestras de eventos normales, el modelo ha sido entrenado con muestras de eventos normales y para comparar la reconstrucción o la salida del modelo se está utilizando la métrica de error de nombre error absoluto medio (MAE), en la que se compara la señal reconstruida y la señal original (Figura 67).

Figura 67

Señal original (color azul) y señal reconstruida (color naranja)

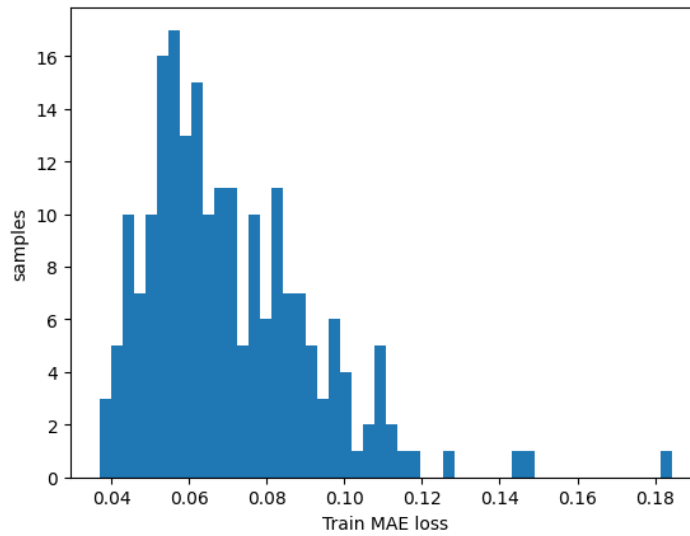


Elaborado por el equipo de trabajo.

Se calcula una distribución con todos los errores de MAE y mediante esta se determina un umbral que posteriormente nos servirá para determinar si una muestra es anómala y de esta manera construir nuestra matriz de confusión en la fase de prueba del modelo.

Figura 68

Distribución del Error medio absoluto.



Elaborado por el equipo de trabajo.

Para los resultados se ha decidido utilizar las métricas Accuracy y Precision, que nos da una idea global del comportamiento de nuestro modelo.

Tabla 5

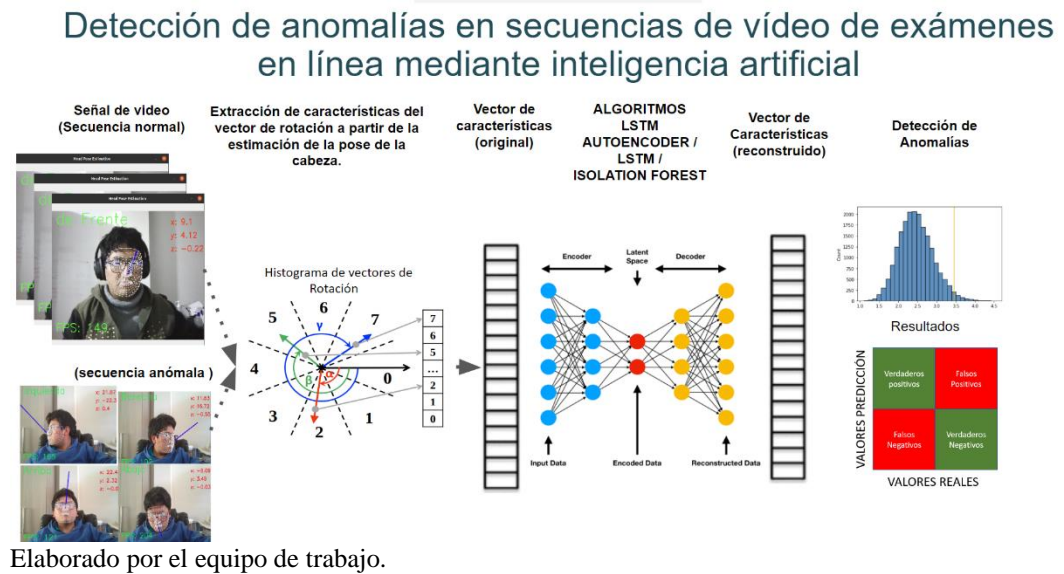
Tabla de resultados

Algoritmo	Accuracy	Precision
ISOLATION FOREST	40.86%	44.18%
LSTM- AUTOENCODER	35.46 %	98.00%
AUTOENCODER	80.08 %	98.00 %

Elaborado por el equipo de trabajo.

Figura 69

Modelo para la detección de anomalías en secuencias de video de exámenes en línea



4.2. DISCUSIÓN

En esta sub sección se muestra la comparación respecto a trabajos similares de detección de anomalías en exámenes en línea

La investigación de Moyano et al. (2021) proporciona una valiosa contribución a la discusión sobre la evaluación en línea, destacando la importancia de garantizar la seguridad y la validez de las evaluaciones en línea, y proporcionando información útil sobre el uso de la plataforma Moodle para lograr estos objetivos.

El estudio de Garg et al. (2020) es una contribución valiosa en la investigación de seguridad de los exámenes en línea mediante la utilización de técnicas de deep learning, y su precisión en la detección de malas prácticas es prometedora. Sin embargo, es importante continuar investigando para mejorar la precisión del sistema y abordar otras posibles amenazas en los exámenes en línea.



El estudio presentado por Martínez Donoso, (2020) es interesante ya que se enfoca en la detección de fraude en cursos en línea mediante el análisis de patrones de comportamiento de los usuarios. Los resultados obtenidos muestran una alta precisión en la detección de anomalías utilizando técnicas de big data e inteligencia artificial. No obstante, el hecho de que en este trabajo se haya utilizado el lenguaje de programación Python para el análisis de los datos y la construcción del modelo en el presente trabajo es importante en el área de inteligencia artificial. Además, la utilización de servicios en la nube, como Amazon S3, permite el almacenamiento y procesamiento eficiente de grandes cantidades de datos.

En general, los estudios realizados y este estudio demuestran el potencial de la aplicación de técnicas de big data e inteligencia artificial para la detección de fraude en exámenes en línea y su relevancia en la actualidad debido al aumento en la demanda de la educación en línea.



V. CONCLUSIONES

- Se presenta un modelo para la detección de anomalías en exámenes en línea, que se obtiene a través del análisis de un umbral dado por el indicador MAE, se utiliza un enfoque de detección de anomalías para abordar el problema del desbalance de datos etiquetados en el que el modelo solo aprende de eventos considerados normales.
- Se obtuvo un dataset eventos de exámenes en línea mediante una aplicación que permite obtener información de la cámara y el micrófono en tiempo real mediante modelos pre-entrenados de inteligencia artificial y servicios en la nube, se ha adquirido datos por un total de 308 mil archivos y con un tamaño de 143 Gigas que han sido almacenados en Amazon s3, a partir de estos datos se han seleccionado 386 muestras de las cuales 290 pertenecen a la clase normal y 96 a la clase de anomalías.
- Se entrenó con éxito tres modelos para la detección de anomalías en exámenes en línea, uno basado en ISOLATION FOREST, LSTM-AUTOENCODER y AUTOENCODER, ya teniendo las muestras etiquetadas, se realizó el análisis exploratorio de datos mediante diagramas de caja y bigotes, distribución y series de tiempo, posteriormente se procedió a extraer características de movimiento de la cabeza por cada eje (X,Y y Z) mediante un histograma de movimiento concatenado por eje, el cual tiene un tamaño de 48 (16 por cada eje), se construyeron las arquitecturas y luego se crearon series de tiempo para que ingresen a las mismas para posteriormente ser entrenadas, Se pudo determinar que el modelo basado en AUTOENCODER converge de mejor manera durante el entrenamiento.
- Se evaluó con éxito 3 modelos, un primer modelo basado en ISOLATION FOREST, con el que se obtuvo un ACCURACY de 40.86 % y una Precisión 44.18 %, un



segundo modelo basado en la arquitectura LSTM-AUTOENCODER, con el que se obtuvo un ACCURACY de 35.46 % y una Precisión 98.00 % y un tercer modelo basado en AUTOENCODERS, con este modelo se obtuvo un ACCURACY de 80.08 % y una Precisión 98.00 %, mostrando mejores resultados lo cual significa que el 80.08% de las predicciones realizadas por este último modelo son correctas, el modelo acierta en la clasificación (anomalía o no anomalía) en el 80.08% de los casos y respecto a la precisión del 98.00% indica que, cuando el modelo clasifica una instancia como anomalía, acierta el 98.00% de las veces.



VI. RECOMENDACIONES

- Para trabajos futuros se recomienda realizar la búsqueda exhaustiva de hiperparámetros del modelo para determinar la configuración óptima que presente mejores resultados. Adicionalmente se recomienda utilizar arquitecturas más modernas como son los TRANSFORMERS o GANS (las redes neuronales adversarias).
- También se recomienda mejorar la calidad del dataset de entrenamiento con un etiquetado más riguroso y exhaustivo.
- Para el desarrollo de posteriores aplicaciones para la supervisión de exámenes en línea se recomienda analizar el tráfico de red y la cantidad de dispositivos conectados a dicha red (Monitoreo de las tablas ARP de los dispositivos de red)
- Para el desarrollo de posteriores evaluaciones se recomienda integrar una cámara externa que tenga gran cobertura de visión del ambiente de evaluación.
- Optimizar el consumo del ancho de banda de la aplicación a fin de que se tenga que analizar un menor volumen de datos.
- Para obtener la detección de anomalías en tiempo real se recomienda utilizar un clúster de computadoras en la nube como *Amazon Elastic Container Service*, a fin de que pueda procesar en tiempo cuasi real la gran cantidad de datos generados durante un examen de admisión.
- Se recomienda desplegar el modelo en un pipeline de *Mlops* que sea *serverless* que utilice los *endpoint* del servicio *Sagemaker* de Amazon.



VII. REFERENCIAS BIBLIOGRÁFICAS

Agile Learning Loops -Combining Agile Approaches in Higher Education Programs - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Illustration-of-a-Sprint-within-a-SCRUM-process-with-some-adaptions-to-the-ALL-methodology_fig2_348336701 [accessed 5 Dec, 2022]

Alonso-Aranda, C. (2019). MODELO-VISTA-CONTROLADOR. LENGUAJE UML.

Amazon web services. (2014). Visión general de Servicios web de Amazon.

Amazon Web Services. (2022). Amazon Timestream Developer Guide (p. 377). <https://docs.aws.amazon.com/timestream/latest/developerguide/what-is-timestream.html>

Amazon.com. (2022). Amazon S3. https://aws.amazon.com/es/s3/?trk=5970b1e9-218b-48cc9862f23c151d81b2&sc_channel=ps&s_kwcid=AL!4422!3!590443989054!e!!g!!amazons3&ef_id=Cj0KCQiApb2bBhDYARIsAChHC9tMjSJdxEw158t_XRKHAAwyJDoqsmPH3OmM5M_N133iSKG1k1g_OoaAvzYEALwwcB:G:s&s_kwcid=AL!4422!3!59

Arias, J., & Covinos, M. (2021). Diseño y metodología de la investigación. In Enfoques Consulting EIRL.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. 1–22.

Barba Guamán, L. R. (2015). Utilización de métodos de visión artificial para pc como apoyo en la automoción (Doctoral dissertation, ETSI_Sistemas_Infor).

Bradski, G., & Kaebler, A. (2016). Learning opencv (Vol. 4, Issue 1).



- Bermejo, D. I. (2021). Estudio comparativo de redes neuronales convolucionales para la clasificación de especies forestales maderables en la Amazonia Peruana. *Tesis*, 1–95.
http://repositorio.unap.edu.pe/bitstream/handle/UNAP/7104/Molleapaza_Mamani_Joel_Neftali.pdf?sequence=1&isAllowed=y
- Caballero, R., & Martion, E. (2021). Las bases de Big Data y de la Inteligencia Artificial. In *Angewandte Chemie International Edition*, 6(11), 951–952. (Vol. 119, Issue 4).
- Concha Llorente, T., & Vega Africano, J. (2022). Uso de Inteligencia Artificial (IA) para mejorar la cultura de la originalidad de las evaluaciones en línea en la UNAD.
- Deemer, Por Pete, Gabrielle Benefield, Craig Larman, B. V. (2009). *Información Básica De Scrum (the Scrum Primer)*. Leo Antoli.
- Delgado Quintero, S. (2022). Aprende Python. Aprende Python, python, 1–488.
https://aprendepython.es/_downloads/907b5202c1466977a8d6bd3a2641453f/aprendepython.pdf
- Deng, J., Roussos, A., Chrysos, G., Ververas, E., Kotsia, I., Shen, J., & Zafeiriou, S. (2019). The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. *International Journal of Computer Vision*, 127(6), 599–624.
- Diginmotion. (11 de Noviembre de 2010). About Amazon Web Services. Diginmotion, 13.
- Datascientes (S/f). ¿Qué es un DataFrame? (2022, mayo 27). Formation Data Science | Datascientest.com. <https://datascientest.com/es/que-es-un-dataframe>
- Doc.Qt.Io. (2022). QML Applications. <https://doc.qt.io/qt-6/qmlapplications.htm>



- Eguíluz Pérez, J. (2018). Introducción a JavaScript. *Pendidikan Fisika*, 1, 4.
<https://uniwebsidad.com/libros/javascript?from=librosweb>
- Enciso, L., Guajala, D., Sarango, J., & Pablo Alejandro, Q. S. (2018). Simulación de envío de paquetes bidireccionales aplicando el protocolo HTTP y TCP/RENO
Simulation of sending bi-directional packets using the HTTP and TCP / RENO protocol.
- Escobar Tafurt, L. F. (2016). Evaluación de algoritmos de sustracción de fondo para conteo de personas.
- Gestión del Aprendizaje mediante la implementación de un Entorno Personalizado de Aprendizaje en la Nube - Scientific Figure on ResearchGate. Available from:
https://www.researchgate.net/figure/Figura-2-Modelos-de-Servicio-de-la-Computacion-en-la-Nube-y-Tipo-de-Usuarios-Asociados_fig1_338753128
[accessed 28 Dec, 2022]
- Grif, M & Kondratenko, Y. (2021). Development of a software module for recognizing the fingerspelling of the Russian Sign Language based on LSTM. *Journal of Physics: Conference Series*. 2032. 012024. 10.1088/1742-6596/2032/1/012024.
- Hernandez, Fabio. (2020). Clasificación de registros ECG de corta duración usando redes neuronales recurrentes. 10.13140/RG.2.2.13943.24488.
- HUGO SANCHEZ CARLESSI, C. R. M. (2019). *METODOLOGIA Y DISEÑOS EN LA INVESTIGACION CIENTIFICA* (2nd ed.). 2019.
- IEEE. (2016). Introduction to simulation using JavaScript. *Winter Simulation Conference (WSC)*.



- Khandkar, Vinod & Hanawal, Manjesh. (2021). Challenges in Net Neutrality Violation Detection: A Case Study of Wehe Tool.
- Kaddoura, S., Popescu, D. E., & Hemanth, J. D. (2022). A systematic review on machine learning models for online learning and examination systems. *PeerJ Computer Science*, 8, e986. <https://doi.org/10.7717/peerj-cs.986>
- Laguna, L. (2021). Detección de opiniones fraudulentas empleando Autoencoders.
- Mainak, C., & Pratap, K. A. (2021). Monitoring Cloud-Native Applications.
- Michael Dawson. (2003). Python Programming for the Absolute Beginner. TeamLib.
- Ofoeda, J., Boateng, R., & Effah, J. (2019). Application Programming Interface (API) Research: A Review of the Past to Inform the Future. doi:10.4018/IJEIS.2019070105
- Owens, M., & Hipp, D. R. (2019). The Definitive Guide to SQLite.
- Patterson, S. (2019). Learn AWS serverless computing : a beginner's guide to using AWS Lambda, Amazon API Gateway, and services from Amazon Web Services (1st ed.). Packt Publishing.
- Pérez Porto, J., Gardey, A. (2015). Definición de API - Qué es, Significado y Concepto. Definicion.de. Recuperado el 9 de noviembre de 2022 de <https://definicion.de/api/>
- Riba Pi, E. (2015). Implementation of a 3D pose estimation algorithm.
- Rodríguez, César & Dorado, Ruben. (2015). ¿Por qué implementar Scrum?. Revista Ontare. 3. 125. 10.21158/23823399.v3.n1.2015.1253.



- Roncagliolo, P. (2008). *Procesamiento Digital de Imágenes*. Universidad Técnica Federico Santa María, 1–15.
- Saffiro, M. (2018). *Tecnologías de la Información y Procesos de Negocios*. Santiago.
- Somma, F. J. (2018). *Cuaterniones y ángulos de Euler para describir rotaciones en* (Doctoral dissertation, UNIVERSIDAD ABIERTA INTERAMERICANA).
- Song, Y., Hyun, S., & Cheong, Y. G. (2021). Analysis of autoencoders for network intrusion detection. *Sensors*, 21(13), 4294.
- Taquía, J. A. (2017). El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital. *Interfases*, 11–30.
<http://repositorio.ulima.edu.pe/handle/ulima/5416>
- Trinh, H. D., Zeydan, E., Giupponi, L., & Dini, P. (2019). Detecting mobile traffic anomalies through physical control channel fingerprinting: A deep semi-supervised approach. *IEEE Access*, 7, 152187-152201.
- Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137-154.
- Wittig, M., & Wittig, A. (2018). *Amazon web services in action* (Second Edi).
[https://books.google.com.pe/books?hl=es&lr=&id=dzczEAAAQBAJ&oi=fnd&pg=PT22&dq=amazon+web+services&ots=mmyCdoC08F&sig=fHC2ixtEi9YJ5Ps32oeAbiS_3AU&redir_esc=y#v=onepage&q=amazon web services&f=false](https://books.google.com.pe/books?hl=es&lr=&id=dzczEAAAQBAJ&oi=fnd&pg=PT22&dq=amazon+web+services&ots=mmyCdoC08F&sig=fHC2ixtEi9YJ5Ps32oeAbiS_3AU&redir_esc=y#v=onepage&q=amazon+web+services&f=false)
- Zaghbani, S., Boujneh, N., & Bouhlel, M. S. (2018). Age estimation using deep learning. *Computers & Electrical Engineering*, 68, 337-347.



- Cadena Moreano, J. A., Montaluisa Pulloquina, R. H., Flores Lagla, G. A., Chancúsig Chisag, J. C., & Guaypatín Pico, O. A. (2017). Reconocimiento Facial Con Base En Imágenes. *Revista Boletín Redipe*, 6(5), 143–151.
- Beraún Barrantes, J. G. (2021). Sistema de reconocimiento facial en línea para prevenir la suplantación y el plagio en el examen de admisión virtual en la Universidad de Huanuco 2020. In *Facultad De Ciencias De La Salud Escuela Académico Profesional De Obstetricia* (Vol. 1). http://repositorio.udh.edu.pe/bitstream/handle/123456789/238/uzuriaga_cespedes_ever_tesis_maestria_2016.pdf?sequence=1&isAllowed=y
- Moyano, E., Moncho, M., Jerez, M., & Izarra, E. (2021). Evaluar en Línea a través de un Entorno Virtual (Moodle). Experiencias en la UNTDF. *Te&Et*, 5. http://sedici.unlp.edu.ar/bitstream/handle/10915/122756/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Paramita, C., Rajesh, R., Subhasish, B., & Sandip, R. (2022). *View of Cloud based LMS Security & Exam Proctoring Solution* (p. 13).
- Rabiha, S. G., Kartowisastro, I. H., Setiawan, R., & Budiharto, W. (2022). Survey of Online Exam Proctoring Model to Detect Cheating Behavior based on Face Recognition. *2022 8th International Conference on Systems and Informatics (ICSAI)*, 1-7. <https://doi.org/10.1109/ICSAI57119.2022.10005488>
- Rodríguez Morales, P., Luzardo Verde, M., Rodríguez Morales, P., & Luzardo Verde, M. (2020). Cómo asegurar evaluaciones válidas y detectar falseamiento en pruebas a distancia síncronas. *Revista Digital de Investigación en Docencia Universitaria*, 14(2). <https://doi.org/10.19083/ridu.2020.1240>



ServiceNow (S/f). ¿Qué es la detección de anomalías? Recuperado el 30 de diciembre de 2022, de <https://www.servicenow.com/es/products/it-operations-management/what-is-anomaly-detection.html>

ANEXOS

ANEXO 1: BackLog de las tareas en Jira

Backlog

VX Sprint 4 6 Jan - 14 Jan (21 issues)

ID	Task Description	Status	FC
VX-115	Implementar la interfaz del semaforo en QML	DONE	FC
VX-121	Cerrar proceso de la aplicacion despues de cualminar el examen	DONE	FC
VX-116	Conectar grafana con AWS Timestream	DONE	FC
VX-120	Crear el proceso de cierre de sesion en la interfaz del examen	DONE	FC
VX-122	Establecer tiempo en el examen	DONE	FC
VX-123	No se puede navegar entre preguntas.	DONE	FC
VX-134	Captura de audio en segundo plano	DONE	FC
VX-117	Añadir temporizador al dar click en el boton SIGUIENTE en la interfaz del examen	DONE	FC
VX-135	Proceso que controla cuando la aplicacion se cierra, ya sea esto porque se culminó el examen o cuando recibe una petición HTTP GET desde el API Gateway	DONE	FC
VX-120	Hay un proceso que envía peticiones HTTP-GET para comprobar el estado de cada etapa en el examen	DONE	FC
VX-133	Captura de video en segundo plano	DONE	FC
VX-126	Funciona para detectar el BlackList	DONE	FC
VX-131	Proceso que verifica internet a cada momento	DONE	FC
VX-118	Crear proceso para recibir mensaje en la interfaz del examen	DONE	FC
VX-124	Funcion para detectar el giro de la cabeza	DONE	FC
VX-125	Funcion para detectar una maquina virtual	DONE	FC

Elaborado por el equipo de trabajo.

ANEXO 2: Datos adquiridos por estudiante

	codigo	video	mic	stage	exam	process	derecha	izquierda	abajo	arriba	frente
2	41605205	201	231	2	2	2	28	22	139	0	12
3	43139019	21	29	2	2	2	0	0	21	0	0
4	44490757	184	157	2	2	2	47	0	133	0	4
5	45744705	180	179	2	2	2	12	0	162	0	6
6	45832130	181	139	2	2	2	4	1	175	0	1
...
2100	79218552	14	50	2	2	2	2	4	1	2	5
2101	79380995	55	12	2	2	2	14	0	37	3	1
2102	90323668	17	23	2	2	2	0	0	17	0	0
2103	90485192	51	46	2	2	2	3	0	46	0	2
2104	92251645	164	37	2	2	2	11	0	127	0	26

2103 rows × 11 columns

Elaborado por el equipo de trabajo.



ANEXO 3: Algoritmo para guardar rostros

```
if self.Save == True or self.NumberFaces == 0:

    if self.iterator_save <= 150:

        self.array_buff.append(self.frame)

        self.Save = True

    if self.iterator_save == 150:

        mat = np.array(self.array_buff)

        date_time = datetime.datetime.now().strftime("%Y-%m-%d-%H-%M-%S")

        pathstr = os.path.abspath(os.getcwd())

        if self.NumberFaces == 0:

            name_video = pathstr+"/services/output/events/video"+ self.__class__.codeEstudent+
date_time+".sin_rostro.mp4"

            elif self.Save == True:

                name_video = pathstr+"/services/output/events/video"+
self.__class__.codeEstudent+date_time+"_"+ self.type_video+"_.movimiento_excesivo.mp4"

            save_video_thread2 = threading.Thread(target=self.save_video,args =(mat,name_video ))

            save_video_thread2.start()

            self.iterator_save = 0

            self.array_buff = []

            self.Save = False

            self.out.release()

            self.iterator_save = self.iterator_save + 1
```

Elaborado por el equipo de trabajo.

ANEXO 4: Algoritmo para Monitoreo de la cámara y audio en el examen

```
if self.status==2:

    self.image = self.frame
```



```
self.frame=cv2.GaussianBlur(self.frame,(3,3),0)

fgMask = self.backSub.apply(self.frame)

th1 = cv2.erode(fgMask, None, iterations=2)

ret1,th1 = cv2.threshold(th1,150,255,cv2.THRESH_BINARY)

th1 = cv2.erode(th1, None, iterations=3)

th1 = cv2.dilate(th1, None, iterations=1)

cnts = cv2.findContours(th1,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)

cnts = imutils.grab_contours(cnts)

whites = cv2.countNonZero(th1)

listDataVideo=whites

for c in cnts:

    whites = cv2.countNonZero(th1)

    if cv2.contourArea(c) > 500:

        window = True

        self.Save = True

        (x, y, w, h) = cv2.boundingRect(c)

        text = "True"

        text = "True"

        text = "True"

    else:

        window = False

        self.Save = False

self.results = self.face_mesh.process(self.image)
```



```
self.image.flags.writeable = True

self.image = cv2.cvtColor(self.image, cv2.COLOR_RGB2BGR)

img_h, img_w, img_c = self.image.shape

self.face_3d = []

self.face_2d = []

if self.results.multi_face_landmarks:

    self.Save = True

    self.NumberFaces = 1

    for face_landmarks in self.results.multi_face_landmarks:

        for idx, lm in enumerate(face_landmarks.landmark):

            if idx == 33 or idx == 263 or idx == 1 or idx == 61 or idx == 291 or idx == 199:

                if idx == 1:

                    self.nose_2d = (lm.x * img_w, lm.y * img_h)

                    self.nose_3d = (lm.x * img_w, lm.y * img_h, lm.z * 3000)

                    x, y = int(lm.x * img_w), int(lm.y * img_h)

                    self.face_2d.append([x, y])

                    self.face_3d.append([x, y, lm.z])

                self.face_2d = np.array(self.face_2d, dtype=np.float64)

                self.face_3d = np.array(self.face_3d, dtype=np.float64)

                focal_length = 1 * img_w

                cam_matrix = np.array([ [focal_length, 0, img_h / 2],

                                        [0, focal_length, img_w / 2],
```



```
[0, 0, 1]])
```

```
dist_matrix = np.zeros((4, 1), dtype=np.float64)

success, rot_vec, trans_vec = cv2.solvePnP(self.face_3d, self.face_2d, cam_matrix, dist_matrix)

rmat, jac = cv2.Rodrigues(rot_vec)

angles, mtxR, mtxQ, Qx, Qy, Qz = cv2.RQDecomp3x3(rmat)

x = angles[0] * 360

y = angles[1] * 360

z = angles[2] * 360

if y < -10:

    self.type_video = "Izquierda"

#print("izquierda")

elif y > 10:

    self.type_video = "Derecha"

#print("derecha")

elif x < 5:

    self.type_video = "Abajo"

#print("abajo")

elif x > 10:

    self.type_video = "Arriba"

#print("arriba")

else:

    self.type_video = "Frente"
```

```
#print("Frente")

# Display the nose direction

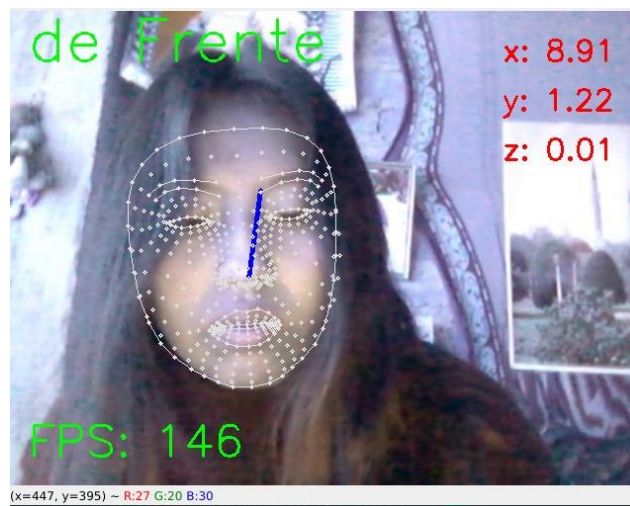
nose_3d_projection, jacobian = cv2.projectPoints(self.nose_3d, rot_vec, trans_vec, cam_matrix,
dist_matrix)

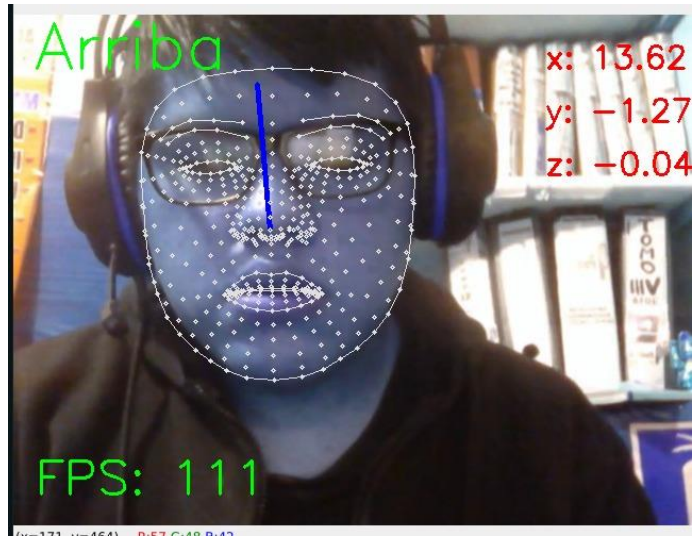
p1 = (int(self.nose_2d[0]), int(self.nose_2d[1]))

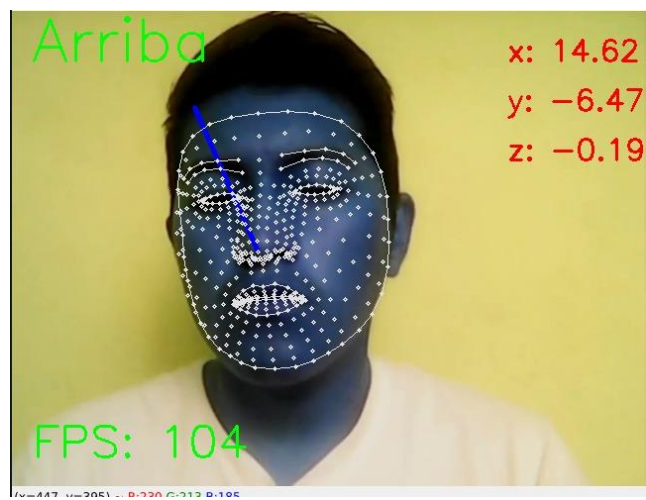
p2 = (int(self.nose_2d[0] + y * 10) , int(self.nose_2d[1] - x * 10))
```

Elaborado por el equipo de trabajo.

ANEXO 5: Imágenes de la extracción de características del movimiento de rostro





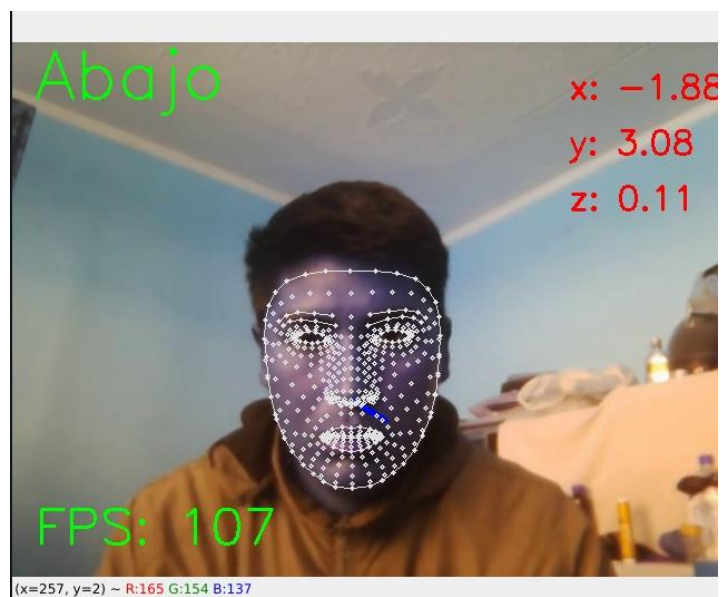


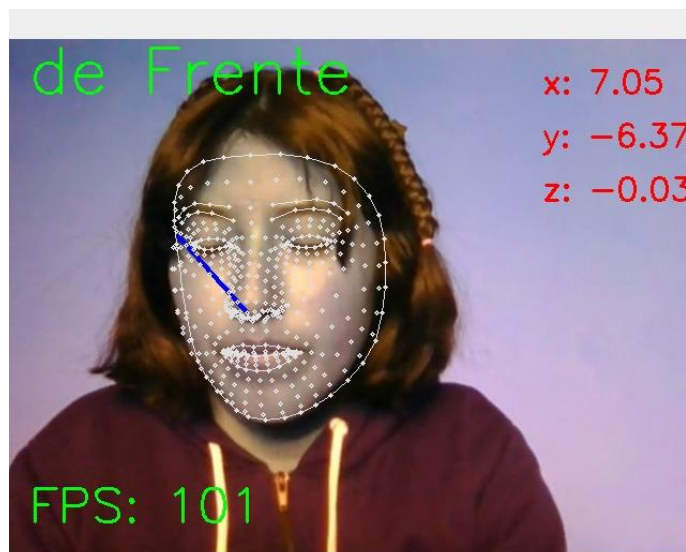
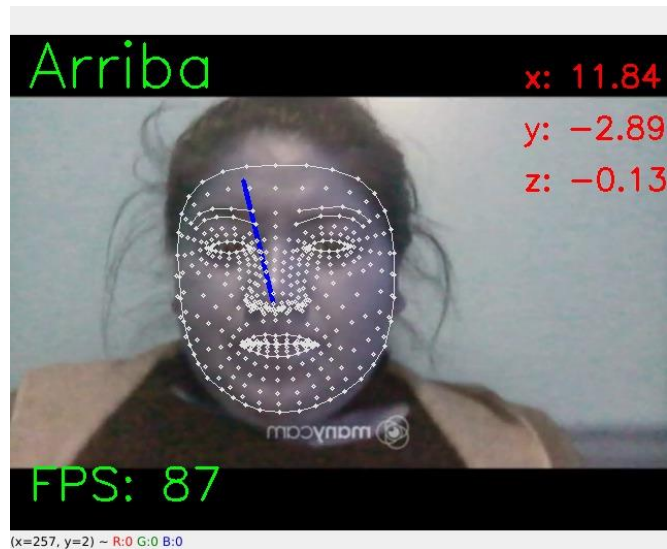
tipo

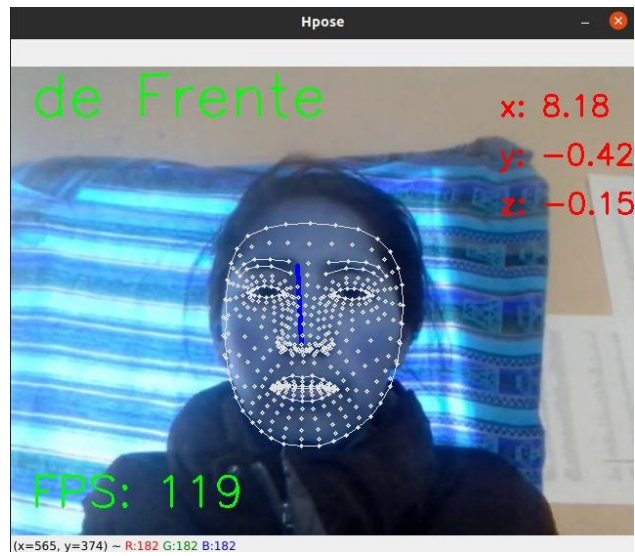


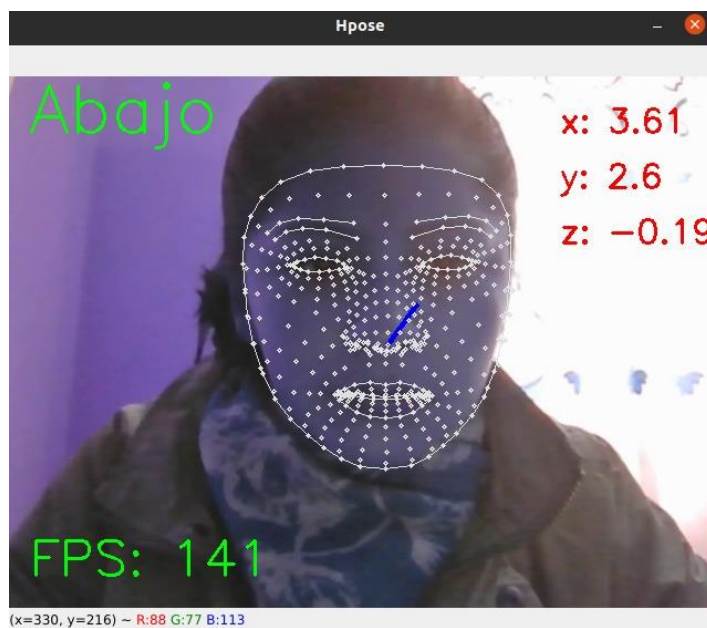














DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, nosotros: Milwart Rossini Calizaya Bobadilla y Fredy Calsin Cari, identificados con DNI N° 72177294 y 44952030, respectivamente, en nuestra condición de egresados de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

de Ingeniería de Sistemas, informamos que hemos elaborado el/la Tesis o Trabajo de Investigación denominada:

“MODELO PARA LA DETECCIÓN DE ANOMALÍAS EN SECUENCIAS DE VIDEOS DE EXÁMENES EN LÍNEA MEDIANTE INTELIGENCIA ARTIFICIAL CASO DE ESTUDIO: UNIVERSIDAD NACIONAL DEL ALTIPLANO”

Es un tema original.

Declaramos que el presente trabajo de tesis es elaborado por nosotros a y no existe plagio/copia de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejamos constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumimos como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratificamos que somos plenamente conscientes de todo el contenido de la tesis y asumimos la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, nos sometemos a las disposiciones legales vigentes y a las sanciones correspondientes, de igual forma nos sometemos a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 15 de noviembre del 2023

Milwart Rossini Calizaya Bobadilla
DNI N° 72177294



Fredy Calsin Cari
DNI N° 44952030





AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, nosotros: Milwart Rossini Calizaya Bobadilla y Fredy Calsin Cari, identificados con DNI N° 72177294 y 44952030, respectivamente, en nuestra condición de egresados de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

de Ingeniería de Sistemas, informamos que hemos elaborado el/la Tesis o Trabajo de Investigación denominada:

MODELO PARA LA DETECCIÓN DE ANOMALÍAS EN SECUENCIAS DE VIDEOS DE EXÁMENES EN LÍNEA MEDIANTE INTELIGENCIA ARTIFICIAL CASO DE ESTUDIO: UNIVERSIDAD NACIONAL DEL ALTIPLANO"

para la obtención de Grado, **Título Profesional** o Segunda Especialidad.

Por medio del presente documento, afirmamos y garantizamos ser los legítimos, únicos y exclusivos titulares de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, damos seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizamos a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración, ni regalía alguna a favor nuestro; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizamos que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribimos el presente documento.

Puno 15 de noviembre del 2023

Milwart Rossini Calizaya Bobadilla
DNI N° 72177294



Fredy Calsin Cari
DNI N° 44952030

