



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

DESARROLLO DE APLICACIÓN DE ENTRENAMIENTO PERSONALIZADO CON LA UTILIZACIÓN DE NUEVAS TECNOLOGÍAS

Alumno: Adolfo Orzaes Céspedes

Tutor: Prof. D. Raúl Mata Campos
Depto.: Departamento de Ingeniería de
Telecomunicaciones

Febrero, 2024

Agradecimientos

Quiero expresar mi profundo agradecimiento a mi tutor, Raúl, por su dedicación y apoyo incondicional a lo largo de este trayecto académico. Su orientación y aliento fueron fundamentales para alcanzar este logro.

A mi amigo Javi, quien ha sido un pilar durante este tiempo, mi más sincero agradecimiento por su constante apoyo y motivación.

En tercer lugar, agradezco a mis tíos por su constante respaldo y consejos sabios que siempre me han orientado en momentos cruciales.

Y por último y más importante, le doy las gracias a mi madre. Su fuerza, amor y sacrificio han sido la base de cada paso que he dado. Su guía y apoyo incondicional han sido cruciales en este camino.

Adolfo Orzaes Céspedes

Febrero, 2024



Universidad de Jaén

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

DESARROLLO DE APLICACIÓN DE ENTRENAMIENTO PERSONALIZADO CON LA UTILIZACIÓN DE NUEVAS TECNOLOGÍAS.

Firma autor:

Firma tutor:

Febrero, 2024

Índice

1	Resumen.....	11
	Abstract	11
2	Introducción	12
3	Objetivos	14
4	Materiales y Métodos	16
4.1	Estado del arte	16
4.1.1	Introducción a la Visión Artificial	17
4.1.2	Análisis de Poses en 3D utilizando Visión Artificial.....	19
4.1.3	Análisis de Poses en 2D utilizando Visión Artificial.....	20
4.1.4	Integración de Realidad Aumentada para Guiar el Ejercicio.....	22
4.1.5	Sistemas de almacenamiento en aplicaciones de seguimiento físico.....	24
4.2	Desarrollo	26
4.2.1	Aplicaciones en el mercado actual	26
4.2.2	Comparativa de las distintas tecnologías	29
4.2.3	Implementación de la aplicación	42
4.3	Metodología	69
4.3.1	Herramientas hardware.....	70
4.3.2	Herramientas software	71
5	Resultados y discusión.....	77
5.1	Interfaz gráfica del usuario	77
5.1.1	Press Militar	79
5.1.2	Sentadillas	80
5.1.3	Peso Muerto	81
5.2	Interfaz web	82
5.2.1	Interfaz web de usuario.....	82
5.2.2	Interfaz web de entrenador	86
5.3	Análisis de resultados	90
5.3.1	Análisis del movimiento articular en tiempo real.....	91

5.3.2	Análisis de velocidad de movimiento.....	94
5.3.3	Análisis de cadencia de movimiento	98
5.3.4	Análisis de los parámetros estadísticos.....	99
6	Conclusiones	103
6.1	líneas de futuro	103
7	Planos y Anexos.....	105
7.1	Anexo I: Documentación de empresa.....	105
7.1.1	Antecedentes de los promotores.....	105
7.1.2	Oportunidad de negocio y análisis del sector	106
7.1.3	El producto o servicio.....	106
7.1.4	El mercado: Los clientes y competencia	108
7.1.5	Plan de Marketing y Comunicación: Estrategia de Posicionamiento y Segmentación del Mercado	109
7.1.6	Recursos materiales y humanos	110
7.1.7	Plan Económico - Financiero	111
7.2	Anexo II: Código fuente (anexo digital).....	118
7.3	Anexo III: Base de Datos Local (anexo digital)	119
8	Bibliografía	120

Índice de figuras

Figura 2.1 Estadística sobre obesidad [2]	12
Figura 4.1. Figura 2. Aplicaciones "Salud y fitness" en PlayStore [5]	16
Figura 4.2. Campos receptivos de neuronas en la corteza estriada del gato [52] ..	18
Figura 4.3. Percepción mecánica de sólidos tridimensionales [7]	18
Figura 4.4. Azure Kinect DK [9]	19
Figura 4.5 Ejemplo de funcionamiento de Azure Kinect DK [53]	20
Figura 4.6 Iteración de DeepPose para una Mayor Precisión [11]	21
Figura 4.7. Keypoints detectados por OpenPose [13]	22
Figura 4.8 Gymnotize Fitness App [56]	23
Figura 4.9 Virtuagym	24
Figura 4.10 Modelos de almacenamiento en base de datos no relacionales [19] ..	25
Figura 4.11 WL Analysis App [21]	27
Figura 4.12 Rep Speed Bar velocity tracker [24]	28
Figura 4.13. Yekar App	29
Figura 4.14 Logotipo de Unity y C Sharp [25]	30
Figura 4.15. Logotipo de JavaScript	31
Figura 4.16. Logotipo de Node.js	32
Figura 4.17. Logotipo de Python	33
Figura 4.18. Logotipo Jupyter	33
Figura 4.19. Logotipo Spyder	34
Figura 4.20. Logotipo de Pycharm	34
Figura 4.21. Logotipo de Visual Studio Code	36
Figura 4.22. Logotipo YOLOv8	36
Figura 4.23. Logotipo de MediaPipe	37
Figura 4.24. Lista de soluciones de Mediapipe [31]	38
Figura 4.25. Implementación de OpenCv y MediaPipe	39
Figura 4.26. Representación de la creación de matrices en NumPy	40
Figura 4.27. Módulos de la Aplicación	42
Figura 4.28. Diagrama de flujo del Usuario	45
Figura 4.29. Zoom 1 del diagrama de flujo del usuario	46
Figura 4.30. Inicialización de Listas	46
Figura 4.31. Configuración de Variables	46
Figura 4.32. Zoom 2 del diagrama de flujo del usuario	47
Figura 4.33. Configuración de la Cámara	47
Figura 4.34. Configuración del Archivo	47

Figura 4.35. Zoom 3 del diagrama de flujo del usuario	48
Figura 4.36. Bucle While captura del fotograma	48
Figura 4.37. Dimensiones y Conversión	49
Figura 4.38. Landmarks en MediaPipe [32]	49
Figura 4.39. Coordenada de articulación y marca de tiempo	50
Figura 4.40. Array de coordenadas.....	50
Figura 4.41. Almacenamiento de punto clave	50
Figura 4.42. Longitud de un vector	51
Figura 4.43. Teorema del Coseno.....	51
Figura 4.44 Contador de repeticiones.....	52
Figura 4.45. Longitudes del esqueleto	54
Figura 4.46. Filtro de media móvil con ventana rectangular.....	55
Figura 4.47. Threshold para ángulos	55
Figura 4.48. Zoom 4 del diagrama de flujo del usuario	57
Figura 4.49. Variable de salida	57
Figura 4.50. Ejemplo de visualización	58
Figura 4.51. Zoom 5 del diagrama de flujo usuario	59
Figura 4.52. Conexión a MongoDB.....	59
Figura 4.53. Creación de diccionario	60
Figura 4.54. Diagrama de flujo de entrenador.....	61
Figura 4.55. Zoom 1 del diagrama de flujo de entrenador.....	62
Figura 4.56. Zoom 2 del diagrama de flujo de entrenador.....	63
Figura 4.57. Contraseña y correo	64
Figura 4.58. Instalación de PyMongo	64
Figura 4.59. Importación de librería	64
Figura 4.60. Agregación de IP	65
Figura 4.61. Base de datos y Colecciones.....	65
Figura 4.62. Configuración de credenciales.....	66
Figura 4.63. URI de conexión	66
Figura 4.64. Agregación de encabezados.....	67
Figura 4.65. Creación de barra lateral y selección.....	67
Figura 4.66. Carga de archivos	67
Figura 4.67. Creación de botón	67
Figura 4.68. Creación de cuenta en Ngrok	68
Figura 4.69. Comando y token de autenticación	68
Figura 4.70. Creación del túnel.....	69
Figura 4.71. URL pública	69

Figura 4.72. Webcam Primux WC187 FullHD	70
Figura 4.73. Selección de puntos clave en Mediapipe Pose	72
Figura 4.74. Arquitectura de red MediaPipe [41]	73
Figura 4.75. Logo oficial de Opencv	74
Figura 4.76. Logo de NumPy	74
Figura 4.77. Logo de Matplotlib	75
Figura 4.78. Logo de Streamlit	75
Figura 4.79. Logo de Python y MongoDB	76
Figura 5.1. Webcam conectada al ordenador	78
Figura 5.2. Funcionamiento de la aplicación de usuario	79
Figura 5.3. Press Militar	80
Figura 5.4. Sentadilla	81
Figura 5.5. Peso Muerto	82
Figura 5.6. Interfaz web de usuario	83
Figura 5.7. Movimientos básicos	83
Figura 5.8. Subir Archivo	84
Figura 5.9. Selección del archivo	84
Figura 5.10. Mensaje de confirmación	85
Figura 5.11. Carga remota de entrenamientos	85
Figura 5.12. Carga exitosa	86
Figura 5.13. Interfaz web de entrenador	87
Figura 5.14. Selección de usuario y ejercicio	87
Figura 5.15. Visualización de datos del usuario	88
Figura 5.16. Confirmación del correo enviado	89
Figura 5.17. Acceso remoto a datos de entrenamiento	89
Figura 5.18. Verificación del canal de comunicación	90
Figura 5.19. Análisis de puntos de referencia	91
Figura 5.20. Análisis del movimiento articular en tiempo real	92
Figura 5.21. Gráfica extraída de A_SV_1 de database	92
Figura 5.22. Variación de la amplitud en el eje horizontal durante el ejercicio	93
Figura 5.23. Trazado óptimo del desplazamiento articular	94
Figura 5.24. Señal velocidad extraída de A_SV_1 de database	94
Figura 5.25. Comparación de filtros	95
Figura 5.26. Señal velocidad filtrada en rango óptimo de movimiento	96
Figura 5.27. Zoom Señal velocidad filtrada de A_SV_1	97
Figura 5.28. Irregularidad en el ejercicio M_SNV_1	98

Figura 5.29. Cadencia extraída de A_SV_1	99
Figura 5.30. Parámetros estadísticos	100
Figura 5.31. Intervalos óptimos de la señal	101
Figura 5.32. Baja desviación en J_SV_2	102
Figura 7.1 Pérdidas y Ganancias.....	113
Figura 7.2. Tesorerías.....	114
Figura 7.3 Tesorería primer año.....	115
Figura 7.4 Punto de equilibrio.....	116
Figura 7.5 Balances.....	117
Figura 7.6 Repositorio Drive con ejercicios.....	119

Índice de tablas

Tabla 7.1 Plan económico.....	111
Tabla 7.2 Fondos propios	112

1 Resumen

El proyecto tiene como objetivo principal desarrollar una aplicación en Python que utilice la Visión Artificial para realizar un seguimiento preciso de las articulaciones durante el entrenamiento de fuerza con pesas. La herramienta proporcionará mediciones en tiempo real y en diferido, utilizando Realidad Aumentada para guiar al usuario. Se ha implementado un sistema de almacenamiento de datos para capturar la información y presentarla de manera integrada en una aplicación web dirigida al entrenador. La herramienta facilita la monitorización de movimientos básicos, permitiendo ajustes precisos en la técnica de movimiento. Por último, se implementará un canal de comunicación para ofrecer retroalimentación por parte del entrenador con el fin potenciar el rendimiento deportivo y prevenir posibles lesiones.

Abstract

The main objective of the project is to develop a Python application that utilizes computer vision to accurately track joints during weightlifting strength training. The tool will provide real-time and deferred joint movement measurements, utilizing Augmented Reality to guide the user. A data storage system has been implemented to capture and present the information seamlessly within a web application designed for coaches. The tool streamlines the monitoring of basic movements, enabling precise adjustments in movement technique. Lastly, a communication channel is implemented to provide feedback from the coach, aiming to enhance athletic performance and prevent potential injuries.

2 Introducción

Actualmente cada vez son más las personas que demandan asistencia en sesiones deportivas, con el objetivo de recibir orientación, evitar lesiones y maximizar los beneficios de los entrenamientos que se llevan a cabo. Sin embargo, la falta de expertos en los centros deportivos para poder dar servicio a todos los clientes puede llevar a prácticas inadecuadas de ejercicios físicos, aumentando el riesgo de lesiones y desmotivando por la falta de progreso. La técnica inapropiada o la insuficiente cantidad de descanso deriva en lesiones comunes en centros deportivos. [1]

Esta frustración suele llevar al abandono de la actividad física, lo que resulta en una población sedentaria, contribuyendo así a elevadas tasas de obesidad. De acuerdo con la Encuesta Europea de Salud en España del año 2020, se registró que un 16,5% de hombres mayores de 18 años y un 15,5% de mujeres sufren de obesidad. Además, en el rango de edades entre 35 y 74 años, se observa un mayor porcentaje de hombres afectados por esta condición. [2]

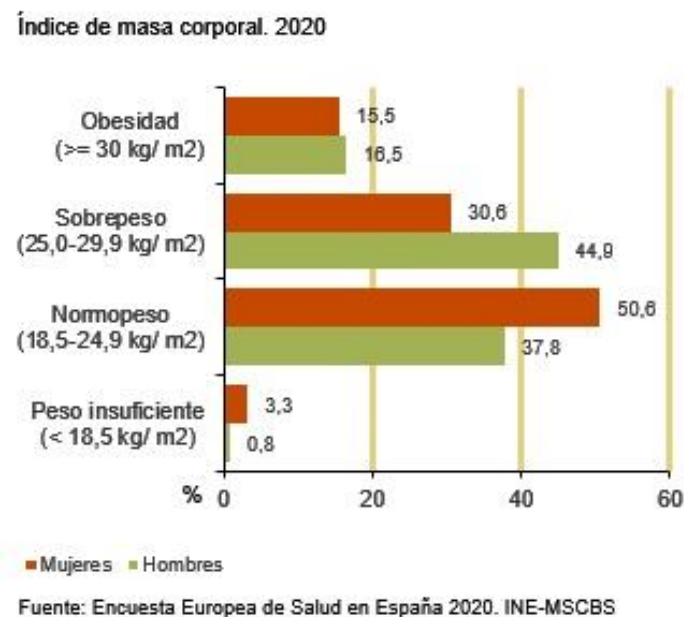


Figura 2.1 Estadística sobre obesidad [2]

Para abordar este desafío, se propone desarrollar una aplicación en Python que utilice la Visión Artificial para realizar un seguimiento preciso de las articulaciones durante los ejercicios deportivos como el entrenamiento de fuerza con pesas. Esta herramienta permitirá obtener mediciones del movimiento articular tanto en tiempo real como en diferido, derivando así parámetros como velocidad, cadencia y otros indicadores relevantes para presentarlos de manera gráfica y analítica.

Se utilizará Realidad Aumentada para proporcionar al usuario medidas en el entorno real, sirviendo de guía durante la ejecución los ejercicios deportivos y desarrollando una experiencia atractiva para evitar la falta de motivación e incentivando a conseguir mejores resultados. Esta tarea se llevará a cabo utilizando la cámara de un dispositivo móvil, integrada en un ordenador o a través de un dispositivo externo.

Este proyecto ofrece una solución innovadora para la monitorización en tiempo real o diferida de los movimientos básicos en el entrenamiento de fuerza con pesas, proporcionando información detallada sobre la ejecución de los ejercicios deportivos. Facilitando al entrenador la tarea de visualizar los parámetros individuales de cada usuario, para realizar ajustes precisos en la técnica de movimiento. Además, se ha implementado un sistema de almacenamiento de datos no relacional que captura la información registrada por los usuarios a través de la aplicación. Estos datos, esenciales para el seguimiento y análisis, se presentan de manera integrada en la aplicación web dirigida al entrenador. Esta información se extrae de un repositorio donde se almacenan todos los entrenamientos realizados, diseñado específicamente para este propósito, asegurando un acceso rápido y efectivo a los registros y métricas pertinentes. Esta estructura garantiza que al interactuar con la interfaz web, el entrenador tenga acceso inmediato a la información crítica sobre el desempeño de cada usuario. Además, se mantiene un canal de comunicación vía email después de revisar los resultados, con el fin de potenciar el rendimiento deportivo y prevenir posibles lesiones.

3 Objetivos

El objetivo principal de este trabajo final de grado es crear una aplicación basada en Python implementando algoritmos de Visión Artificial que permiten hacer un seguimiento en tiempo real o diferido del entrenamiento, obteniendo parámetros como velocidad, cadencia y otros indicadores relevantes, con el propósito de ofrecer mediciones detalladas del movimiento articular durante la ejecución de los ejercicios deportivos.

Adicionalmente, se plantea la integración de Realidad Aumentada con el fin de proporcionar al usuario medidas en tiempo real en su entorno físico, actuando como una guía durante el desarrollo de los ejercicios. Este enfoque tiene como finalidad mejorar la experiencia del usuario y permitir ajustes técnicos por parte del entrenador, consolidando así la eficacia del seguimiento de movimientos articulares.

Por otro lado, debe facilitar el análisis gráfico y analítico de la ejecución de los ejercicios para que el entrenador pueda obtener información detallada sobre la realización del movimiento por parte del deportista. Este enfoque busca proporcionar las herramientas necesarias para realizar ajustes precisos en la técnica de movimiento de cada usuario, con el objetivo de mejorar la calidad del entrenamiento y contribuir a la prevención de lesiones.

En el ámbito del almacenamiento de datos, se considera el diseño de un sistema eficiente y fácil para registrar la información de los usuarios, asegurando la disponibilidad inmediata de los parámetros fundamentales para una toma de decisiones sobre el rendimiento individual de cada participante.

Como parte esencial del proyecto, se propone el desarrollo de una interfaz web dirigida al entrenador, la cual integrará de manera eficiente los datos extraídos del repositorio que contiene todos los entrenamientos. Esto permitirá un acceso remoto y sencillo a través de un dispositivo externo. Por otro lado, se añade una funcionalidad adicional para que el usuario pueda enviar de forma remota sus videos de entrenamiento pregrabados y extraer los datos pertinentes. Estos datos serán subidos al repositorio de entrenamientos para su visualización por parte del entrenador, facilitando así el seguimiento y la evaluación del rendimiento de los deportistas.

Finalmente, se plantea el desarrollo de un canal de comunicación después de revisar los resultados. Este sistema busca potenciar el rendimiento deportivo y prevenir posibles lesiones al ofrecer *feedback* personalizado y recomendaciones, estableciendo así un vínculo efectivo entre el entrenador y los usuarios de la aplicación.

A partir de estos objetivos generales, se ha optado por concretarlos de la siguiente manera:

1. Desarrollar una aplicación basada en Python
 - 1.1. Detección de articulaciones con algoritmo de Visión Artificial
 - 1.2. Generar Mediciones del Movimiento Articular en tiempo real
 - 1.3. Procesado de datos
 - 1.4. Integrar Realidad Aumentada
 - 1.5. Facilitar el Análisis Gráfico y Analítico
 - 1.6. Tratamiento de las señales extraídas en gráficos
2. Implementar un Repositorio para los entrenamientos de los deportistas
3. Desarrollar una Interfaz Web
 - 3.1. Establecer un Canal de Comunicación Post-Análisis

4 Materiales y Métodos

4.1 Estado del arte

El ejercicio físico ha sido fundamental en la historia humana, desde las prácticas de caza hasta los grandes imperios. Los griegos valoraban la armonía entre cuerpo y espíritu, considerando esencial el ejercicio en la educación ciudadana. Creían que no solo fortalecía el cuerpo, sino que también moldeaba el carácter y fomentaba la disciplina, introduciéndolo en la vida diaria a través de competiciones y programas educativos. En la actualidad, el interés por la actividad física crece de manera exponencial, viéndose reflejado en el aumento de personas que se ejercitan y en el desarrollo de gimnasios al alcance de todos. [3]

Dentro de las diferentes actividades físicas que promueven la salud y previenen el deterioro físico asociado con el envejecimiento, el entrenamiento de fuerza ocupa un lugar destacado. Este tipo de entrenamiento se considera fundamental para contrarrestar la pérdida de masa muscular que suele acompañar el proceso de envejecimiento. Entre sus numerosos beneficios se incluyen la prevención de la sarcopenia, la reducción del estrés y la ansiedad, la mejora en la movilidad de las articulaciones y la disminución del nivel de colesterol. [4]

En el transcurso de los últimos años, se ha experimentado un avance exponencial en el desarrollo de numerosas aplicaciones móviles relacionadas con la salud y el deporte. Existe un extenso repertorio de aplicaciones adaptadas a diferentes metas.

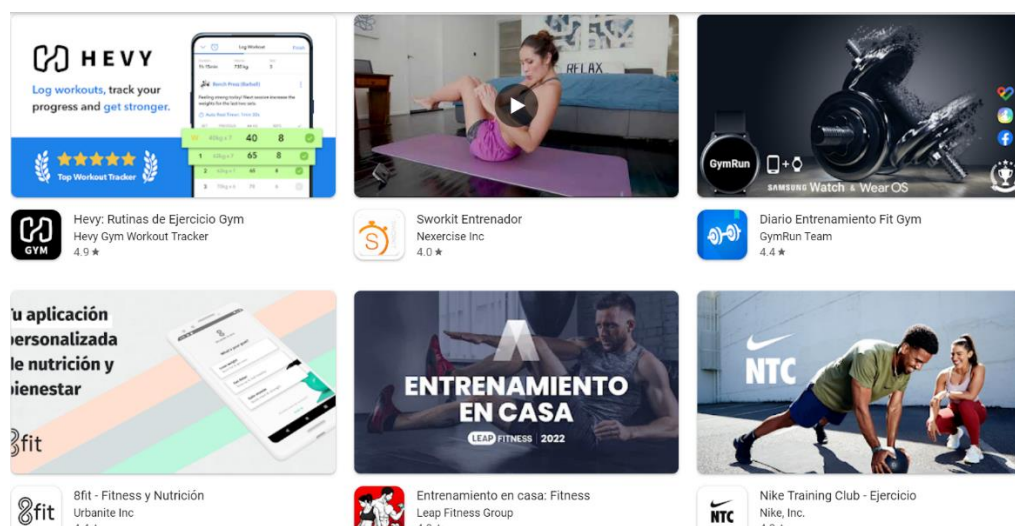


Figura 4.1. Figura 2. Aplicaciones "Salud y fitness" en PlayStore [5]

En la actualidad, el uso de aplicaciones vinculadas a la salud y el deporte se ha vuelto común. Desde la monitorización de ejercicios hasta aplicaciones que registran pasos

y calorías consumidas durante el ejercicio, así como entrenadores virtuales, estas aplicaciones son ampliamente descargadas y utilizadas.

Este proyecto, se enfoca en aplicaciones específicas vinculadas al entrenamiento con pesas. Estas plataformas proporcionan al usuario la capacidad de evaluar la eficacia de sus rutinas de entrenamiento con pesas, permitiéndoles obtener información detallada sobre el movimiento realizado. Además, estas aplicaciones ofrecen instrucciones precisas para llevar a cabo adecuadamente los ejercicios, utilizando diversas tecnologías para brindar orientación detallada.

Cabe destacar que, en el entorno deportivo, especialmente en disciplinas que involucran el uso de pesas, se ha observado un significativo avance en la implementación de tecnologías como la Realidad Aumentada (AR) y la Visión Artificial. [5] Estas tecnologías han transformado la manera en que los atletas y entusiastas del deporte abordan su entrenamiento y evaluación de rendimiento, proporcionando herramientas más precisas y detalladas para analizar y mejorar habilidades. Este fenómeno representa un significativo progreso en la unión de la tecnología y el deporte.

4.1.1 Introducción a la Visión Artificial

La Visión Artificial tiene como objetivo que las computadoras perciban y comprendan imágenes para tomar decisiones, similar a la capacidad humana de ver. [6] Esta tecnología, que combina software y hardware, se aplica en diversos ámbitos, desde entornos industriales hasta otros no industriales.

La historia de la Visión Artificial en el seguimiento de movimientos comprende una evolución que abarca aproximadamente 60 años, donde científicos e ingenieros han dedicado esfuerzos a desarrollar sistemas que permitan a las máquinas percibir y comprender datos visuales. Esta experimentación, iniciada en 1959 con un estudio neurofisiológico que involucraba la presentación de imágenes a un gato para correlacionar respuestas cerebrales, marcó un primer paso revelador. [7]

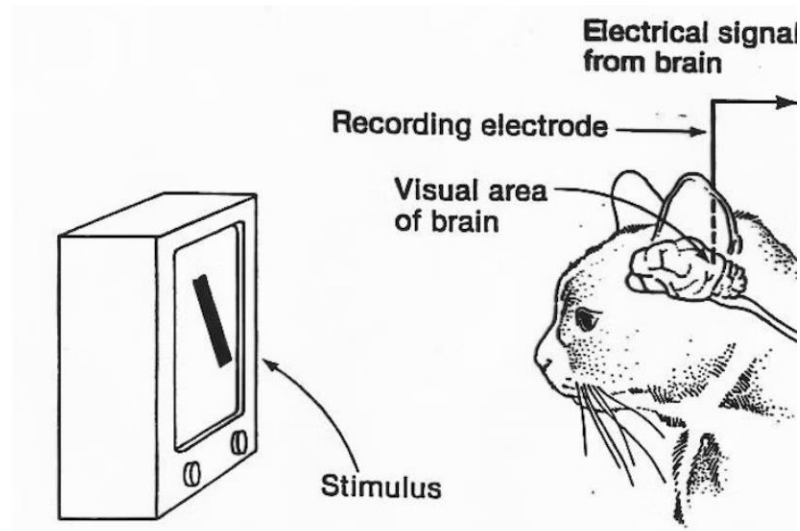


Figura 4.2. Campos receptivos de neuronas en la corteza estriada del gato [52]

Descubrieron que el procesamiento de imágenes inicia con formas simples, como bordes rectos, mostrando el fundamento inicial de cómo la Visión Artificial puede interpretar elementos visuales fundamentales.

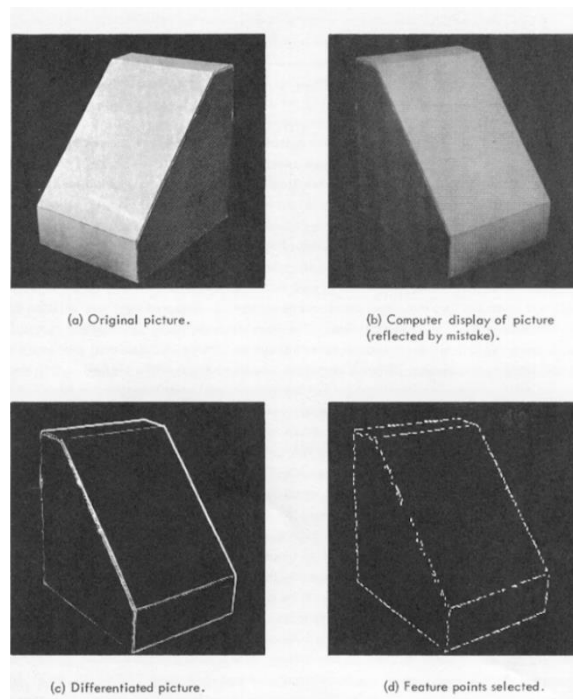


Figura 4.3. Percepción mecánica de sólidos tridimensionales [7]

El avance hacia la Visión Artificial, impulsado por el desarrollo de la informática y la capacidad de procesado de imágenes, marcó un momento decisivo en la precisión y la accesibilidad del seguimiento de movimientos. [8] La implementación de algoritmos para el análisis en tiempo real de imágenes ha permitido la precisa detección y rastreo de movimientos y articulaciones. Este progreso ha allanado el camino para el desarrollo de

aplicaciones más avanzadas en el ámbito de la actividad física, permitiendo una comprensión más profunda y detallada de la ejecución de los ejercicios.

La integración de técnicas de aprendizaje automático, como las redes neuronales, ha ampliado significativamente las capacidades de la visión por computadora en el seguimiento de movimientos, permitiendo un análisis detallado de la postura, la movilidad de las articulaciones y la biomecánica en tiempo real.

Este progreso histórico establece el fundamento para implementar las técnicas modernas de Visión Artificial en proyectos contemporáneos, como esta aplicación desarrollada en Python que persigue la precisa monitorización de las articulaciones durante las rutinas de ejercicios, estableciendo un punto de no retorno hacia un análisis exhaustivo del movimiento humano, prescindiendo por completo de sensores corporales.

4.1.2 Análisis de Poses en 3D utilizando Visión Artificial

Los sistemas que se fundamentan en imágenes 3D demuestran una mayor precisión al identificar articulaciones, dado que la coordenada de profundidad se detecta directamente en lugar de ser calculada.

4.1.2.1 Azure Kinect

Azure Kinect DK es un dispositivo que incorpora sensores de inteligencia artificial, ofreciendo modelos avanzados en Visión Artificial y reconocimiento de voz. El sistema consta de una cámara RGB que capta imágenes bidimensionales y un sistema de luz para detectar la profundidad en la escena. De este modo, emite un patrón de luz infrarroja y una cámara sensible a este espectro observa las deformaciones, permitiendo calcular la profundidad de cada punto. Por otro lado, presenta una variedad de modos, opciones y herramientas de desarrollo de software (SDK)



Figura 4.4. Azure Kinect DK [9]

El entorno de desarrollo de Azure Kinect DK ofrece el SDK de Sensor que permite el acceso a dispositivos y sensores, proporcionando datos fundamentales. Por otro lado, el SDK de Body Tracking posibilita el seguimiento tridimensional del cuerpo, ofreciendo información detallada sobre la postura y los movimientos. Además, se encuentra el Azure AI Voice SDK, que permite el acceso a través del micrófono y aprovecha los servicios de voz basados en la nube de Azure, proporcionando una interfaz eficaz para aplicaciones de voz y control por voz, como se indica en [10].

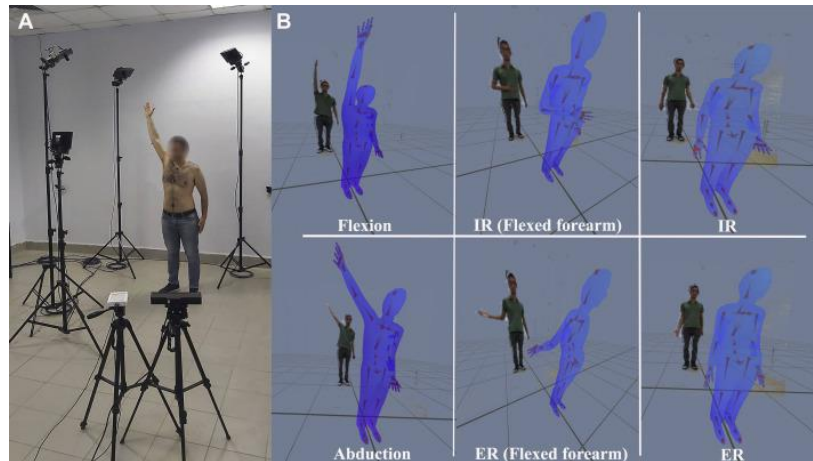


Figura 4.5 Ejemplo de funcionamiento de Azure Kinect DK [53]

4.1.3 Análisis de Poses en 2D utilizando Visión Artificial

A pesar de los numerosos beneficios que ofrece la visión 3D, especialmente en términos de precisión, su implementación se ve limitada en la mayoría de los dispositivos que comúnmente cuentan solo con una cámara RGB tradicional. Por esta razón, resulta interesante explorar alternativas, como la detección de pose mediante visión 2D.

El surgimiento y desarrollo de las redes neuronales han tenido un impacto positivo, generando una variedad de opciones que abordan de manera efectiva el desafío de la detección de poses. En este contexto, la aplicación de Convolutional Neural Networks (CNN) [7] en el campo de la visión por computadora ha sido fundamental. Las CNN tienen la capacidad de aprender características complejas en imágenes, lo que ha mejorado significativamente la capacidad de detectar patrones.

4.1.3.1 DeepPose

DeepPose es un método de estimación de la pose humana que se basa en redes neuronales profundas (DNN), y específicamente, utiliza Convolutional Neural Networks (CNN). Las CNN son ampliamente conocidas y empleadas en tareas de visión por computadora, y en el caso de DeepPose, desempeñan un papel esencial. Desarrollado por investigadores de Google y presentado en la conferencia de Visión por Computadora y

Reconocimiento de Patrones (CVPR) en 2014, este enfoque se centra en la representación de la pose humana a través de puntos de referencia o articulaciones en el cuerpo, de acuerdo a [11].

DeepPose demuestra su versatilidad en distintos campos. En Realidad Aumentada y Virtual, posiciona objetos de manera precisa en entornos reales. En Seguimiento de Movimiento, rastrea la posición y movimiento de personas para aplicaciones de monitoreo. En Reconocimiento de Gestos, identifica gestos ejecutados por individuos. Además, se emplea en estudios para entender cómo las personas se mueven y realizan tareas diversas. Su aplicación abarca desde la integración de tecnologías interactivas hasta el análisis detallado del comportamiento humano.



Figura 4.6 Iteración de DeepPose para una Mayor Precisión [11]

4.1.3.2 OpenPose

OpenPose es un método de estimación de la pose humana basado en redes neuronales profundas (DNN). Se caracteriza por ser pionero en abordar la detección de *keypoints* del cuerpo humano, utilizando inicialmente C++ y Redes Neuronales Convolucionales (CNN). Propuesto por investigadores de la Universidad de Zaragoza en la CVPR de 2016, este método de estimación de la pose humana basado en DNN presenta una arquitectura más compleja que DeepPose, utilizando una CNN de cinco capas con dos capas de convolución, dos capas de *max pooling* y una capa totalmente conectada. Aunque ofrece mayor precisión que DeepPose, OpenPose es más lento en términos de tiempo de ejecución. [12]

OpenPose, al igual que DeepPose, destaca por su versatilidad en diversas aplicaciones, desde la colocación precisa de objetos virtuales en Realidad Aumentada y Virtual hasta el seguimiento y reconocimiento de gestos. Además, juega un papel crucial en el análisis detallado de la actividad humana, facilitando el estudio de cómo las personas se desplazan y llevan a cabo tareas diversas.

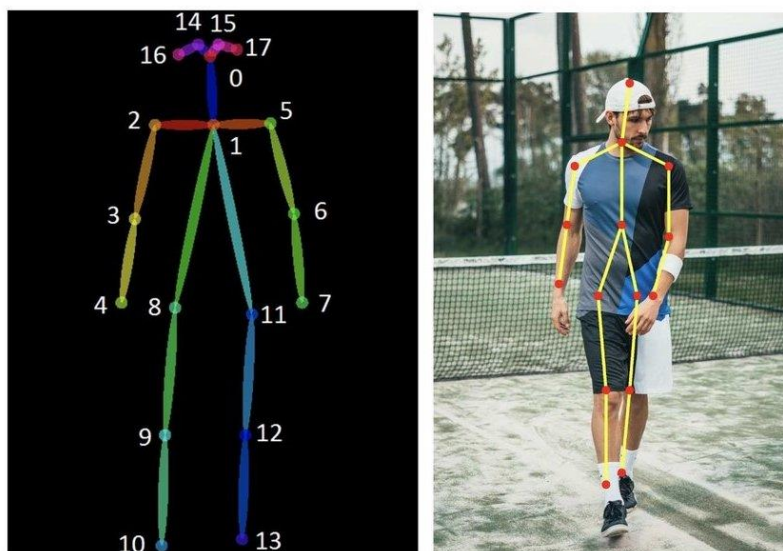


Figura 4.7. Keypoints detectados por OpenPose [13]

4.1.4 Integración de Realidad Aumentada para Guiar el Ejercicio

La Realidad Aumentada (AR) está adquiriendo una importancia cada vez mayor en el ámbito del ejercicio físico al fusionar elementos virtuales con la realidad. Esta integración permite que la AR desempeñe un papel crucial al ofrecer orientación y mediciones en tiempo real, beneficiando a los usuarios. Una de estas ventajas radica en la mejora de la forma durante el ejercicio, mediante indicaciones visuales en tiempo real, lo que no solo contribuye a evitar lesiones, sino que también eleva el rendimiento. Además, la AR aporta un factor estimulante al ejercicio, transformándolo en una experiencia más atractiva y divertida, lo que, a su vez, fomenta la persistencia en la práctica del ejercicio físico.

4.1.4.1 Tipos de aplicaciones de AR centradas en la orientación deportiva

Se presentan ejemplos de aplicaciones que incorporan Realidad Aumentada, pero carecen de Visión Artificial; en los siguientes apartados, se examinarán aplicaciones que ya integran esta tecnología.

4.1.4.1.1 Gymnotize

Gymnotize Technologies, fue fundada en 2022 por expertos en Realidad Aumentada y fitness, su lanzamiento oficial tuvo lugar en el año 2023. Es una aplicación centrada en el aprendizaje seguro de técnicas de levantamiento de pesas, mediante el uso

de la Realidad Aumentada, proporcionando orientación en tiempo real durante el ejercicio. Ofreciendo una biblioteca diversa que incluye más de 50 ejercicios organizados por nivel de dificultad, tipo de ejercicio y objetivo. Algunas características distintivas incluyen avatares personalizables, adaptados según la altura, peso y género del usuario, permitiendo visualizar cómo se ejecutan los ejercicios en su propio cuerpo.

Esta aplicación garantiza la correcta ejecución de cada ejercicio, proporcionando detalles sobre los músculos trabajados en cada sesión de actividad física. Además, la aplicación realiza un seguimiento detallado del progreso del usuario, registrando repeticiones, peso levantado y tiempo invertido en cada ejercicio. La combinación de estas características pone a disposición del usuario una herramienta completa y efectiva para aquellos que buscan mejorar su rendimiento en el levantamiento de pesas mediante la Realidad Aumentada. [14]

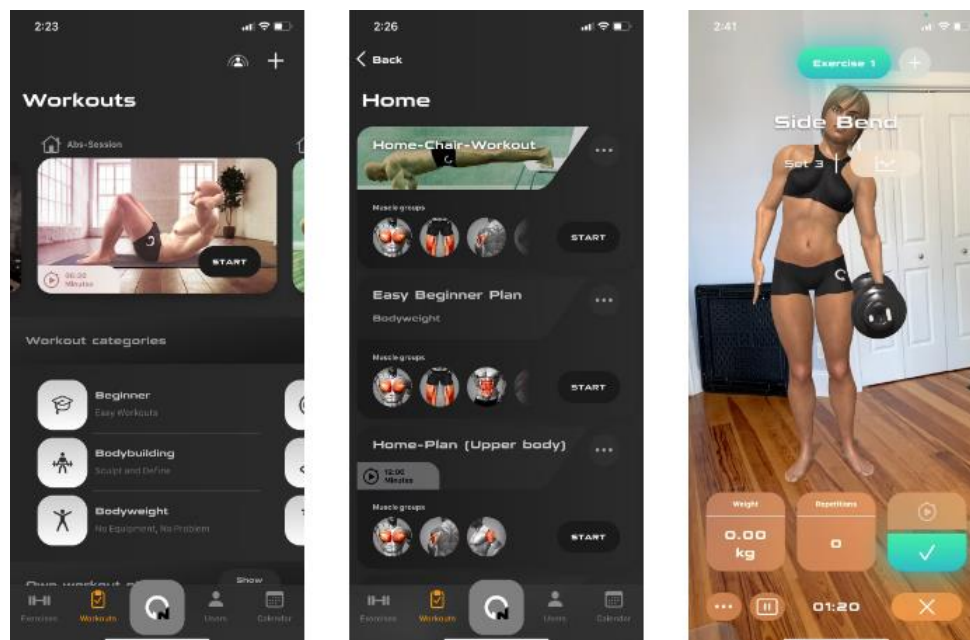


Figura 4.8 Gymnotize Fitness App [56]

4.1.4.1.2 Virtuagym Fitness

Esta aplicación fue fundada en 2008 por Hugo Braam y Paul Braam, en los Países Bajos, experimentando desde entonces un crecimiento significativo. La aplicación se ha desarrollado para adaptarse a las cambiantes necesidades de los usuarios en busca de soluciones integrales para la salud y el fitness.

Esta aplicación está diseñada para fomentar un estilo de vida saludable a través del fitness, la nutrición y la salud, destacándose por su innovadora integración de Realidad Aumentada. Ofrece planes de entrenamiento personalizados adaptados a metas individuales y permite el seguimiento diario de la actividad física, incluyendo pasos, distancia y calorías quemadas. Facilita la gestión de la nutrición mediante el registro de la ingesta de alimentos para mantener una dieta equilibrada. Además, con funciones sociales, los usuarios pueden conectarse, compartir logros y recibir apoyo. Virtuagym también incorpora desafíos y programas de entrenamiento específicos, y su integración de AR ofrece una experiencia única, mejorando la participación y la interactividad de los usuarios en su búsqueda de una vida activa y saludable. [15]

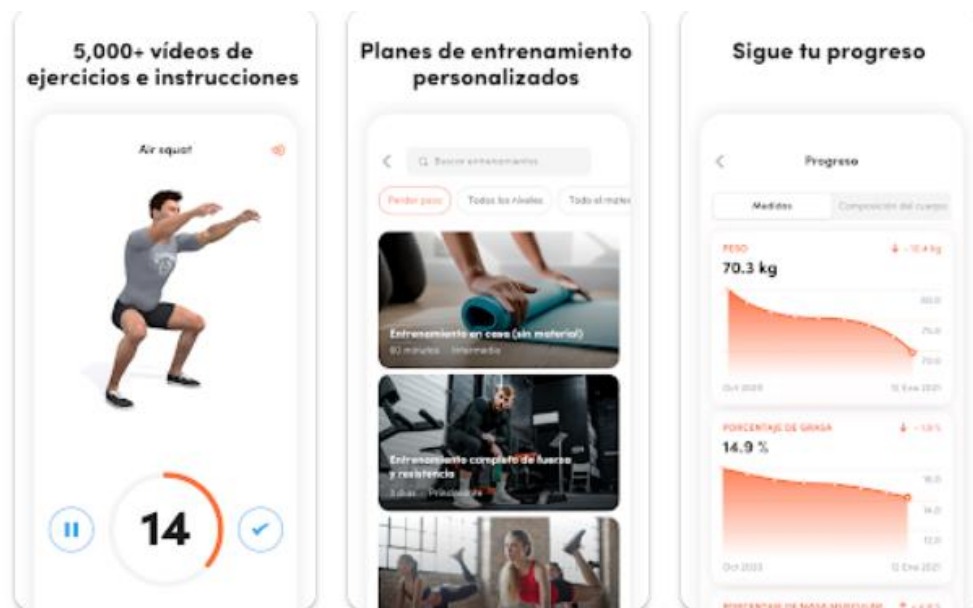


Figura 4.9 Virtuagym

4.1.5 Sistemas de almacenamiento en aplicaciones de seguimiento físico

Con los avances actuales en tecnología, las aplicaciones de seguimiento físico son fundamentales en diversos campos, desde la eficiencia a niveles de alto rendimiento hasta el cuidado de la salud y el fitness. Estas aplicaciones recopilan datos sobre la ubicación y la condición física mediante una variedad de sensores y dispositivos. La administración eficaz y segura de estos datos es fundamental para garantizar la funcionalidad y la utilidad de dichas aplicaciones.

Este contexto plantea la necesidad de explorar y comprender los diferentes sistemas de almacenamiento de datos que respaldan estas aplicaciones de seguimiento físico. Desde bases de datos geospaciales hasta soluciones de almacenamiento en tiempo real. La elección de un sistema adecuado puede tener un impacto significativo en la precisión, la eficiencia y la seguridad de la información recopilada. En esta parte se

examinarán los tipos de sistemas de almacenamiento de datos más importantes para las aplicaciones de seguimiento físico, destacando sus características y su aplicación práctica en entornos concretos.

4.1.5.1 Base de datos no relacionales

En la década de 1990, las bases de datos no relacionales surgieron como una respuesta a las limitaciones que las bases de datos relacionales presentaban para manejar datos no estructurados o semiestructurados. [16]

MongoDB lidera este tipo de bases de datos, seguido por Redis, Elasticsearch y Cassandra.

Los modelos no relacionales utilizan enfoques distintos para almacenar y organizar la información. Algunos modelos comunes incluyen el de documentos, donde los datos se almacenan en unidades independientes; grafos, que emplean una red de nodos y relaciones; y key-value, que almacenan datos en pares de clave-valor [17]. Como se muestra en la figura 4.10.

Columnar:

1	Things	A	foo	B	bar	C	baz
2	Things	C	bam	E	coh	People	A Emmanuel
3	Languages	A	C	B	Java	C	Ceylon

Documental:

```
{ "user": {
  "id": "124",
  "name": "Emmanuel",
  "addresses": [
    { "city": "Paris", "country": "France" },
    { "city": "Atlanta", "country": "USA" }
  ]
}
```

Key-value:

key	value
123	Address@23
126	"Booya"

Graph:

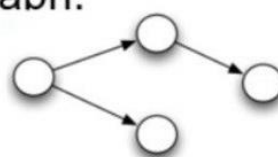


Figura 4.10 Modelos de almacenamiento en base de datos no relacionales [19]

Las aplicaciones de bases de datos no relacionales abarcan diversas áreas, como el almacenamiento de datos no estructurados, siendo una opción adecuada para textos, imágenes, audio y video. También se han integrado cada vez más en aplicaciones web y móviles, donde la necesidad de escalabilidad y rendimiento es crucial. En el campo del análisis de *Big Data*, este tipo de base de datos son ventajosas frente grandes volúmenes de datos.

4.2 Desarrollo

En esta parte, se lleva a cabo un análisis detallado de la aplicación desarrollada, explorando aplicaciones existentes en el mercado con objetivos similares al propuesto en este trabajo final de grado. Además, se realiza una investigación y comparación de las herramientas actualmente disponibles. Por último, se expone la implementación del proyecto.

Es importante destacar que las figuras que muestran las representaciones de los ejercicios en este proyecto son de elaboración propia, obtenidas específicamente a través de la aplicación que se ha desarrollado.

4.2.1 Aplicaciones en el mercado actual

En la actualidad, se dispone de aplicaciones en el mercado que incorporan tecnologías de Visión Artificial y Realidad Aumentada en el contexto deportivo, tal como se ha mencionado con anterioridad. En este punto, se explorarán las diferentes aplicaciones existentes en el mercado vinculadas al objeto de estudio de este trabajo final de grado.

4.2.1.1 WL Analysis

Desarrollada por WL Analysis Technologies en 2021, la aplicación ha revolucionado el análisis de levantamiento de pesas mediante un enfoque pionero en Realidad Aumentada (AR). Ofrece a los usuarios un análisis detallado de la trayectoria, velocidad y potencia de la barra durante el levantamiento. Esta innovación proporciona datos precisos en tiempo real, permitiendo la identificación y corrección de errores de manera eficiente.

Por otro lado, la aplicación ofrece herramientas, como la realimentación visual en tiempo real. Esto habilita al usuario recibir indicaciones en directo durante sus levantamientos, lo que resulta útil para corregir la técnica y optimizar el rendimiento. La posibilidad de comparar levantamientos con otros usuarios añade un extra de competencia entre participantes, permitiendo examinar el ejercicio equiparándose a otros usuarios de la aplicación y de ese modo descubrir áreas específicas donde necesitan mejorar. [18]

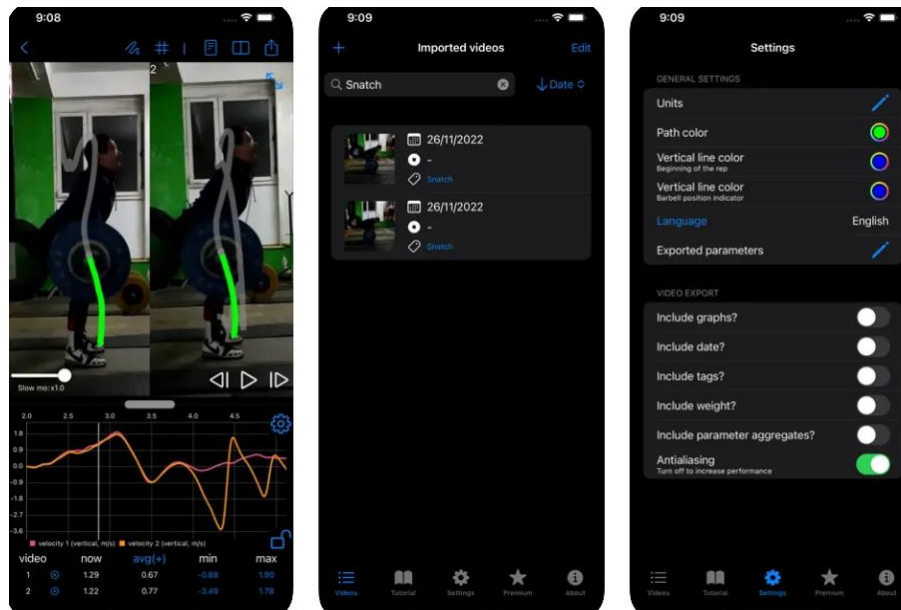


Figura 4.11 WL Analysis App [21]

4.2.1.2 RepSpeed: Bar velocity tracker

Desarrollada por el ingeniero de software de Google, Sreyas Rangaraju, es una aplicación gratuita disponible para dispositivos iOS y Android que utiliza la Visión artificial para monitorear la velocidad, trayectoria de la barra y el rango de movimiento durante los levantamientos con barra. La aplicación ofrece características clave, como la detección automática de la barra, la capacidad de recortar vídeos para enfocarse en secciones relevantes, y la visualización detallada de repeticiones a lo largo del tiempo. [19]

Por otro lado, incorpora funcionalidades técnicas como el seguimiento del índice de esfuerzo percibido (RPE), estimación de la repetición máxima (1RM), el seguimiento de la barra mediante realidad aumentada para ver el recorrido en el entorno real. Este enfoque técnico y avanzado permite a los levantadores de pesas mejorar su rendimiento, corregir la forma y reducir el riesgo de lesiones. [20]

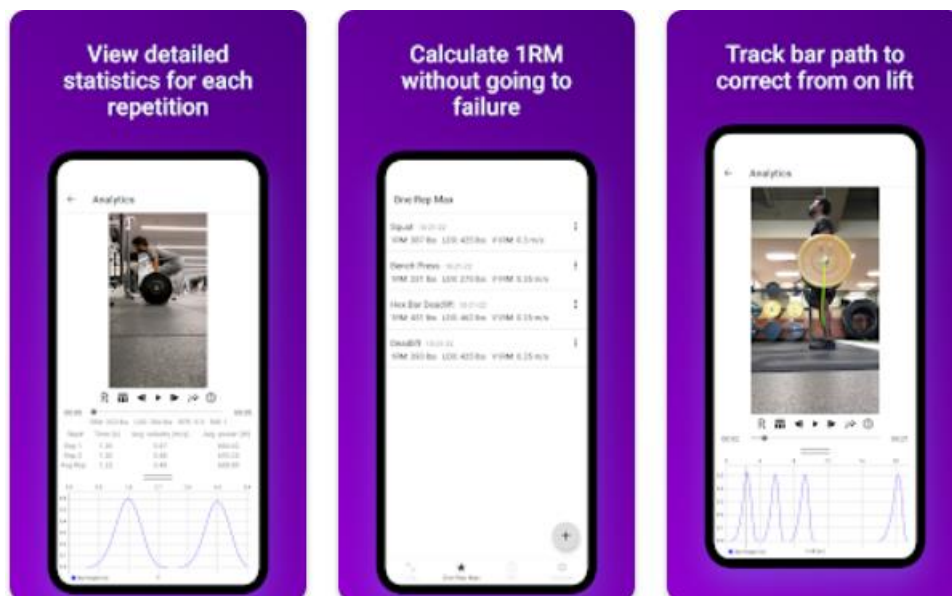


Figura 4.12 Rep Speed Bar velocity tracker [24]

4.2.1.3 Yekar

Desarrollada por un equipo de ingenieros pertenecientes a la Universidad del País Vasco y Mondragón, la aplicación utiliza Visión Artificial para asistir a los atletas en la mejora de sus marcas personales. Actualmente en fase de prelanzamiento, se puede acceder a la aplicación a través de la web mediante registro en un enlace específico. Dado su fase previa a la comercialización, la información disponible es limitada, por lo que se llevó a cabo un estudio de campo con el propósito de ampliar el conocimiento sobre el tema. Durante este proceso, se estableció contacto inicial a través de correo electrónico, seguido por una sesión de videoconferencia con el equipo de desarrollo. Esta interacción permitió obtener detalles específicos, como por ejemplo la herramienta de Visión Artificial empleada, identificada como MediaPipe, la cual se utiliza para la estimación de poses. La aplicación ha estado en desarrollo durante dos años y se enfoca exclusivamente en el nicho de mercado de personas involucradas en el *Powerlifting*.

En términos sencillos, el *Powerlifting* es una disciplina deportiva que se centra en levantamientos de peso máximo en tres movimientos específicos: Sentadilla, Press de Banca y Peso Muerto.

Entre las funciones clave de Yekar se incluyen la detección y corrección de errores en la técnica mediante cámaras y sensores, el conteo automático de repeticiones, el registro del tiempo de las repeticiones, la visualización en 3D de la trayectoria de la barra, la notificación de la profundidad adecuada y la presentación de un resumen en gráficos del rendimiento de los atletas



Figura 4.13. Yekar App

Esta aplicación, se encuentra en constante evolución gracias al trabajo del equipo de ingenieros, tiene el objetivo de transformar el entrenamiento del *Powerlifting*, ofreciendo a los atletas herramientas valiosas para mejorar su técnica y alcanzar sus objetivos de rendimiento.

4.2.2 Comparativa de las distintas tecnologías

Antes de abordar este apartado, es necesario señalar las necesidades del proyecto y, en función de ello, analizar los lenguajes de programación y entornos existentes para seleccionar los que mejor se adapten al desarrollo de este trabajo.

- Se requiere un lenguaje eficiente para trabajar con algoritmos de Visión Artificial para un seguimiento preciso, manipular datos y representar diversos tipos de graficas.
- Desarrollar funciones de Realidad Aumentada para proporcionar al usuario una experiencia inmersiva, ofreciendo medidas en tiempo real y motivándolo a alcanzar mejores resultados.

- Implementar un sistema de almacenamiento de datos no relacionales para almacenar la información registrada por los usuarios.
- Garantizar un acceso efectivo de las métricas mediante el repositorio de entrenamientos realizados.
- Creación de una interfaz web que facilite la interacción tanto para el usuario como para el entrenador con la aplicación. Implementando un canal para la comunicación con el cliente.

4.2.2.1 Lenguajes de programación

4.2.2.1.1 C# (Unity)

En primer lugar, se considera C# por ser globalmente utilizado en el desarrollo de aplicaciones de Realidad Aumentada junto con el motor de desarrollo de juegos conocido como Unity.

Unity destaca por su popularidad y potencia en el desarrollo de aplicaciones con Realidad Aumentada (RA) y Realidad Virtual (VR). Es una plataforma para la creación de experiencias interactivas en 2D y 3D. Su capacidad para trabajar con diversos dispositivos y una amplia comunidad de desarrolladores, lo convierte en una elección atractiva para proyectos que buscan integrar funcionalidades de RA y VR de manera efectiva.

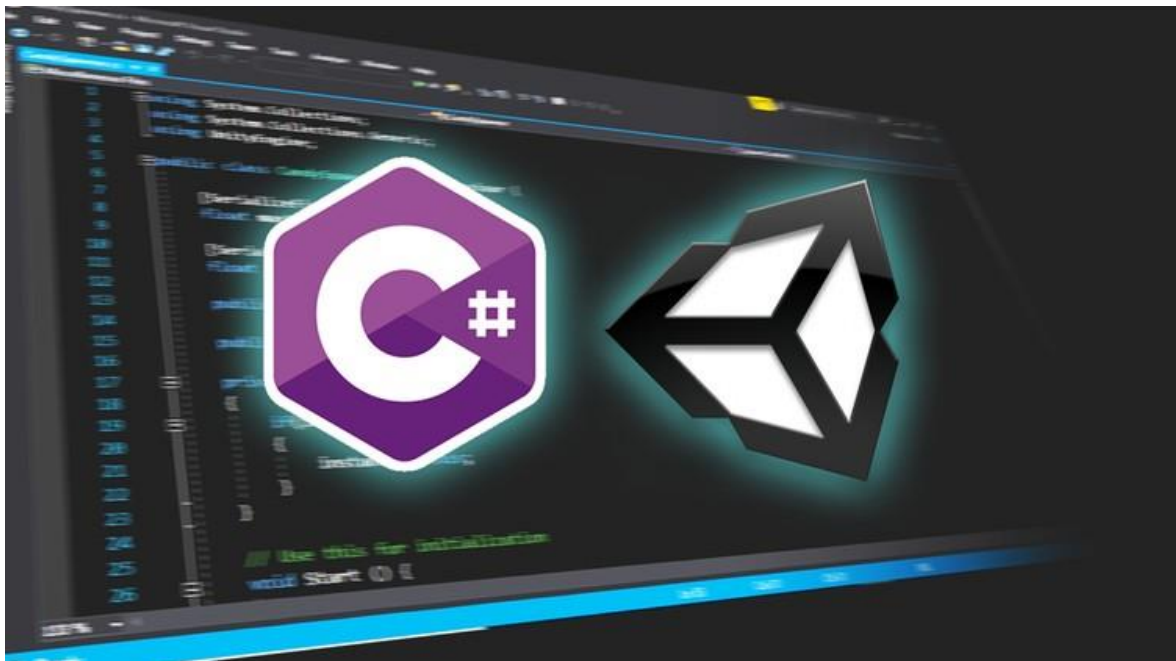


Figura 4.14 Logotipo de Unity y C Sharp [25]

El lenguaje C#, también conocido como C sharp. C# es un lenguaje orientado a objetos que fue diseñado por Microsoft en el año 2000 y posteriormente estandarizado. [21]

Este lenguaje comparte algunas similitudes con C++ y Java, aunque es considerado menos complejo.

Se ha observado que Unity junto a C++ representa una opción idónea para aquellos desarrolladores interesados en la creación de gráficos en 3D, entornos de juego o experiencias inmersivas. Sin embargo, no se alinea con el objetivo principal de este trabajo, dado que se requiere eficiencia para trabajar con algoritmos de Visión Artificial, manipular grandes cantidades de datos y representar diversos tipos de gráficos.

4.2.2.1.2 JavaScript (Node.js)

JavaScript se evalúa como una opción viable gracias a su uso común en el desarrollo web y la disponibilidad de bibliotecas como AR.js [22] para Realidad Aumentada. Por otro lado, permite desarrollar tanto el lado del cliente como el del servidor con Node.js. La combinación de ambos, utilizando JavaScript tanto en el cliente como en el servidor, facilita la creación de aplicaciones web completas. [23]



Figura 4.15. Logotipo de JavaScript

Node.js es un entorno de tiempo de ejecución para JavaScript, posibilita la ejecución en tiempo real de programas, ampliando el alcance del lenguaje más allá de los navegadores para funcionar como aplicaciones independientes en computadoras. Desarrollado por los creadores originales de JavaScript, permite no solo la creación de sitios web interactivos, sino también la realización de tareas comparables a otros lenguajes. Basados en el motor de tiempo de ejecución JavaScript V8 de Google Chrome, tanto JavaScript como Node.js optimizan la velocidad de ejecución convirtiendo el código en eficiente código de máquina. [24]



Figura 4.16. Logotipo de Node.js

Pero el objetivo primordial reside en el uso de herramientas de Visión Artificial, manipulación de datos y visualización de gráficos para estudiar y comparar valores. Aunque JavaScript es válido, su eficiencia se cuestiona en este ámbito por lo que se considera la exploración de otras alternativas más adecuadas para el propósito específico.

4.2.2.1.3 Python

Se termino la búsqueda por el tipo de lenguaje de código que mejor se ajusta a las necesidades del proyecto. Se requiere un lenguaje eficiente para manipular datos, trabajar con algoritmos de Visión Artificial y representar diversos tipos de gráficas.

Se ha seleccionado Python por varias razones:

- Python cuenta con una amplia colección de bibliotecas y módulos para diversas tareas, desde la manipulación de datos hasta el desarrollo web. Bibliotecas como NumPy y Pandas proporcionan herramientas poderosas para estas tareas.
- Python es reconocido por su facilidad para integrarse con otros sistemas y tecnologías. Si el código necesita interactuar con sistemas externos o servicios web, Python ofrece opciones flexibles.
- Python cuenta con una comunidad grande y activa, lo que facilita encontrar soporte, tutoriales y soluciones a problemas comunes.
- En el caso de utilizar herramientas de Visión Artificial en combinación con tareas de procesamiento de datos, Python dispone de una amplia variedad de bibliotecas, como NumPy, OpenCV y TensorFlow, ampliamente utilizadas en tareas de visión por computadora y aprendizaje automático. [25]
- En el entorno de Python, existen recursos que posibilitan el trabajo tanto con Realidad Aumentada como en el desarrollo de aplicaciones web, aunque su

competencia en estos aspectos podría ser menor en comparación con algunos otros lenguajes.



Figura 4.17. Logotipo de Python

4.2.2.2 Entornos de desarrollo

Después de la elección del lenguaje para el proyecto, se requiere identificar un entorno de desarrollo para escribir, editar, depurar y compilar el código, así como para implementar la aplicación. Se exploraron diversas alternativas, considerando inicialmente Jupyter como una opción versátil que puede ejecutarse en línea en plataformas como Google Colab [26], facilitando la colaboración y el acceso remoto, aunque también se puede utilizar sin conexión, aunque esto no es esencial para el proyecto.



Figura 4.18. Logotipo Jupyter

Por otro lado, entornos *offline* como Spider son preferidos en el ámbito científico debido a su interfaz simple. Sin embargo, para desarrollar proyectos que buscan entornos más robustos, no limitándose únicamente a la ciencia de datos, se deben considerar otras opciones.

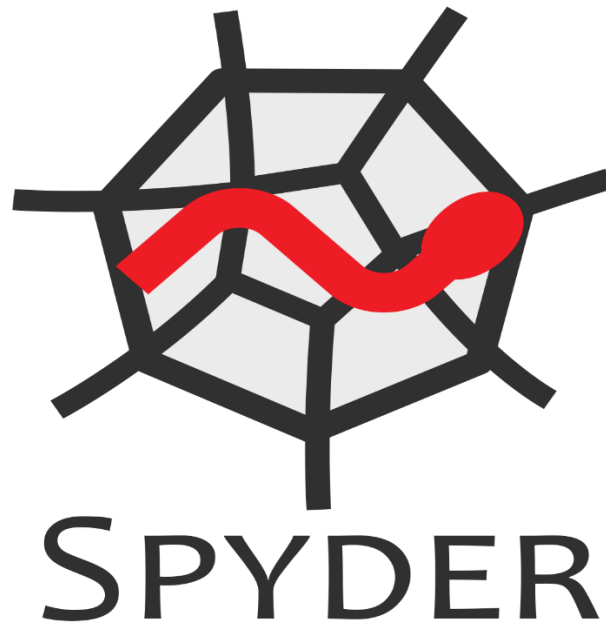


Figura 4.19. Logotipo Spyder

Por ejemplo, Pycharm es un entorno que destaca por su enfoque en proyectos de gran escala, pero es importante tener en cuenta que, a pesar de sus características potentes, puede tener una curva de aprendizaje más pronunciada y consumir más recursos en comparación con entornos más ligeros.



Figura 4.20. Logotipo de Pycharm

Por último, es necesario hablar de Visual Studio Code, conocido por su ligereza y flexibilidad. La elección de este entorno responde a diversos motivos, los cuales se detallan en la siguiente sección.

4.2.2.2.1 Visual Studio Code

Visual Studio Code es un editor de código fuente de código abierto desarrollado por Microsoft, presentado en la conferencia Build de Microsoft en 2015 y lanzado oficialmente en abril de ese mismo año. Desde entonces, ha experimentado un rápido aumento en popularidad, consolidándose como uno de los editores de código más utilizados a nivel mundial. [27]

Este editor, clasificado como un Entorno de Desarrollo Integrado (IDE) [28], ofrece un conjunto de características potentes, versátiles y gratuitas que lo han convertido en una herramienta esencial para numerosos desarrolladores. Su interfaz de usuario intuitiva, rendimiento rápido y amplia gama de funciones lo posicionan como una elección destacada para desarrolladores de todos los niveles de experiencia.

La decisión de utilizar Visual Studio Code como entorno de desarrollo para este proyecto se basa en varias razones específicas:

- En primer lugar, Visual Studio Code proporciona un sólido respaldo para Python, facilitando la escritura de código, la depuración y la gestión de proyectos en este lenguaje.
- Incluye características como resaltado de sintaxis, autocompletado, sugerencias de código y una interfaz de depuración eficiente, mejorando la productividad durante el desarrollo en Python.
- La capacidad de Visual Studio Code para trabajar con entornos virtuales de Python y gestionar dependencias mediante herramientas como *pip* contribuye a una configuración más simple y eficiente del entorno de desarrollo.



Figura 4.21. Logotipo de Visual Studio Code

4.2.2.3 Herramientas para la detección de poses utilizando Visión Artificial

Se llevó a cabo un análisis para determinar las herramientas actuales más adecuadas para la estimación de poses y la detección de articulaciones de manera eficiente y sencilla. En la evaluación inicial, se consideró YOLOv8 (*You Only Look Once*), un modelo de detección de objetos que utiliza una sola red neuronal [29] para predecir clases y coordenadas de objetos directamente en la imagen. No obstante, se observó que este modelo se enfoca específicamente en la detección de objetos, siendo más apropiado para situaciones que requieren entrenamiento personalizado del modelo, de cara a tareas específicas en el ámbito de la detección de objetos y una precisión detallada en dichas tareas.



Figura 4.22. Logotipo YOLOv8

En el ámbito de la inteligencia artificial y el aprendizaje automático, el término "modelos" se refiere a archivos entrenados para reconocer patrones específicos. Estos modelos pueden ser entrenados con conjuntos de datos y utilizar algoritmos para obtener información de esos datos. [30]

Por otro lado, se analiza la herramienta MediaPipe en comparación con YOLOv8, destacando su mayor versatilidad y facilidad de uso. Ambas herramientas proporcionan modelos que son redes neuronales entrenadas para tareas específicas. Durante el entrenamiento, ajustan sus parámetros para minimizar la diferencia entre predicciones en los datos de entrenamiento. Una vez entrenados, estos modelos pueden generalizarse y aplicar sus capacidades a nuevos datos, permitiendo la identificación y localización de objetos o puntos de articulación en imágenes o videos.



Figura 4.23. Logotipo de MediaPipe

La elección de MediaPipe se basa en el empleo de arquitecturas de redes neuronales específicas para cada tarea. La lista siguiente presenta las soluciones disponibles para cada plataforma compatible, donde cada solución incluye uno o más modelos y ofrece la posibilidad de personalizar modelos para algunas soluciones.

Solution	Android	Web	Python	iOS	Customize model
Object detection	●	●	●	●	●
Image classification	●	●	●	●	●
Image segmentation	●	●	●		
Interactive segmentation	●	●	●		
Hand landmark detection	●	●	●	●	
Gesture recognition	●	●	●	●	●
Image embedding	●	●	●		
Face detection	●	●	●	●	
Face landmark detection	●	●	●		
Face stylization	●	●	●		●
Pose landmark detection	●	●	●		
Image generation	●				●
Text classification	●	●	●	●	●
Text embedding	●	●	●		
Language detector	●	●	●		
Audio classification	●	●	●		

Figura 4.24. Lista de soluciones de Mediapipe [31]

Esta herramienta proporciona implementaciones eficientes y modelos previamente entrenados, lo que facilita a los desarrolladores la integración de estas capacidades en sus aplicaciones de manera más sencilla. En este caso se utiliza el modelo de *Pose Landmark Detection*.

4.2.2.4 Análisis de las librerías

En este proyecto, se destaca el peso de OpenCV, ya que se utiliza para llevar a cabo operaciones esenciales en el procesamiento de video. La librería desempeña un papel crucial al realizar tareas como la captura de *frames* y la conversión del formato de color de BGR (*Blue, Green, Red*) a RGB (*Red, Green, Blue*). Esta conversión, tal como se muestra en la figura 4.25, resulta necesaria debido a la diferencia en la representación del color entre OpenCV y otras bibliotecas, como MediaPipe, que se emplea para la estimación de poses.

Por otro lado, OpenCV desempeña un papel fundamental en la visualización en tiempo real de la imagen resultante, así como en el análisis de movimiento y la representación de datos en el entorno real de la aplicación.

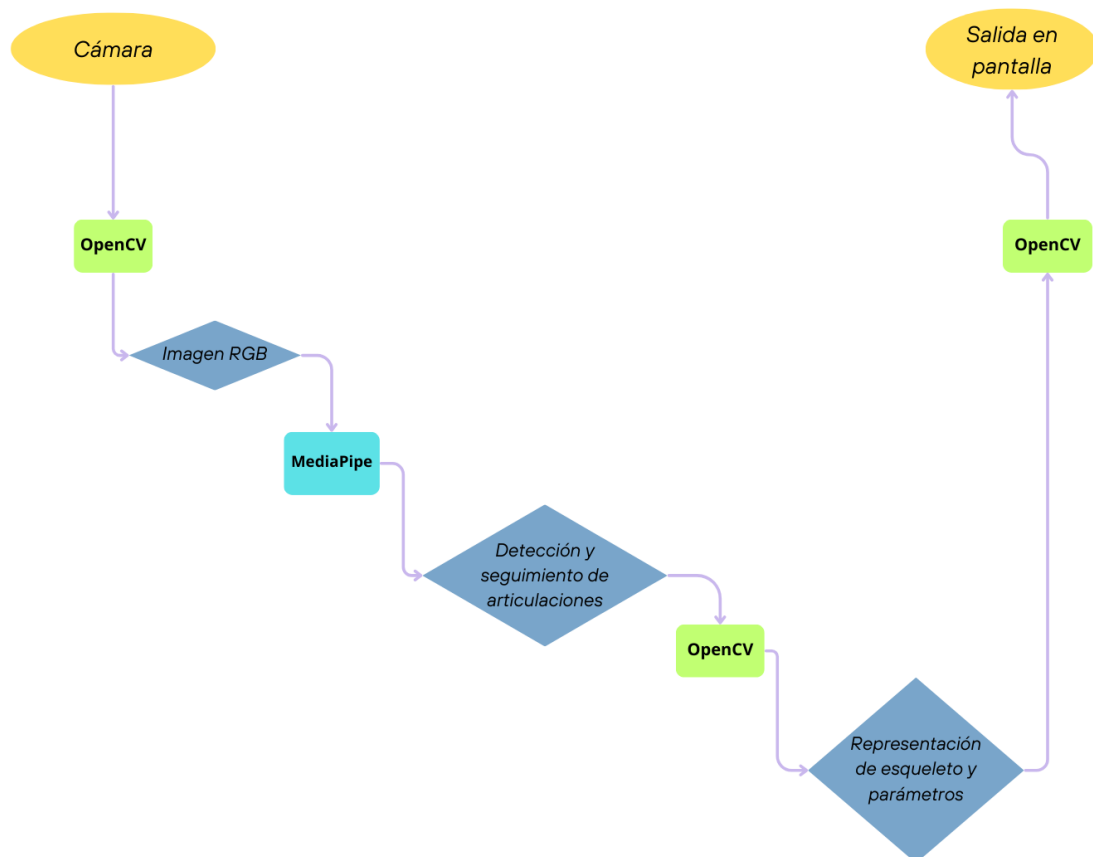


Figura 4.25. Implementación de OpenCv y MediaPipe

Por otro lado, en la búsqueda de bibliotecas para el proyecto, se exploraron opciones como NumPy, que permite manejar operaciones matriciales y cálculos numéricos de manera eficiente. Esta biblioteca posibilita la creación de matrices bidimensionales a partir de las coordenadas “X” e “Y” de los puntos. Estas matrices bidimensionales resultan más eficientes para llevar a cabo operaciones numéricas en comparación con las listas estándar de Python, especialmente al trabajar con grandes cantidades de datos.

La utilización de NumPy ofrece beneficios adicionales, ya que proporciona una amplia variedad de funciones diseñadas para realizar operaciones matemáticas y estadísticas en estas matrices. Esto facilita el procesamiento de datos en contextos científicos y de análisis de datos. En el contexto de NumPy, se representan estructuras de datos que contienen elementos del mismo tipo y pueden ser unidimensionales (vectores), bidimensionales (matrices).

```
>>> a[0,3:5]
array([3,4])
```

```
>>> a[4:,4:]
array([[44, 45],
       [54, 55]])
```

```
>>> a[:,2]
array([2,12,22,32,42,52])
```

```
>>> a[2::2,::2]
array([[20,22,24]
       [40,42,44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Figura 4.26. Representación de la creación de matrices en NumPy

Para la creación de gráficos y animaciones, se ha seleccionado la biblioteca Matplotlib debido a su capacidad para generar visualizaciones de datos de manera sencilla y eficaz. Matplotlib es ampliamente utilizada en la comunidad de Python para la creación de gráficos, ofreciendo una amplia gama de funciones que permiten personalizar y representar datos de forma clara.

La elección de la biblioteca se justifica no solo por su versatilidad, sino también por su compatibilidad con NumPy, simplificando así la integración de datos y cálculos en el proceso de visualización. En resumen, Matplotlib proporciona una herramienta poderosa y flexible para la representación visual de datos en entornos científicos, aspecto crucial en aplicaciones como la presente.

La interfaz gráfica de usuario se ha desarrollado utilizando Tkinter, una biblioteca estándar de Python reconocida por su simplicidad y facilidad de uso. La elección se basó en su éxito para implementar gráficos en tiempo real. La capacidad para gestionar actualizaciones en tiempo real y su flexibilidad en la creación de elementos gráficos, fueron factores esenciales en su elección como herramienta integral en la interfaz gráfica del proyecto científico.

En el contexto de la interfaz web, se ha empleado la biblioteca de Streamlit, posibilitando la creación eficiente de aplicaciones interactivas. La claridad de su sintaxis ha facilitado un desarrollo ágil de la interfaz web, evitando la necesidad de una curva de aprendizaje extensa. En resumen, la elección de Streamlit para el proyecto se justifica por su facilidad de uso, rapidez en el desarrollo, capacidad de visualización de datos y facilidad en la implementación y distribución.

En el ámbito del almacenamiento, la elección de MongoDB se fundamenta en su fácil implementación con Python, especialmente a través del uso de la biblioteca PyMongo. La integración sin problemas facilitada por la librería permite una conexión eficiente y una manipulación de datos sin complicaciones en el proyecto. La combinación de la flexibilidad de esquema de MongoDB y su interoperabilidad directa con Python mediante PyMongo ha simplificado de manera significativa la gestión de datos deportivos, brindando así una solución eficaz y de fácil implementación en la aplicación.

4.2.3 Implementación de la aplicación

La aplicación se ha fragmentado en distintos módulos, cada uno con funciones específicas, con el objetivo de facilitar la comprensión del funcionamiento del sistema. Estos módulos se muestran en la figura 4.27.

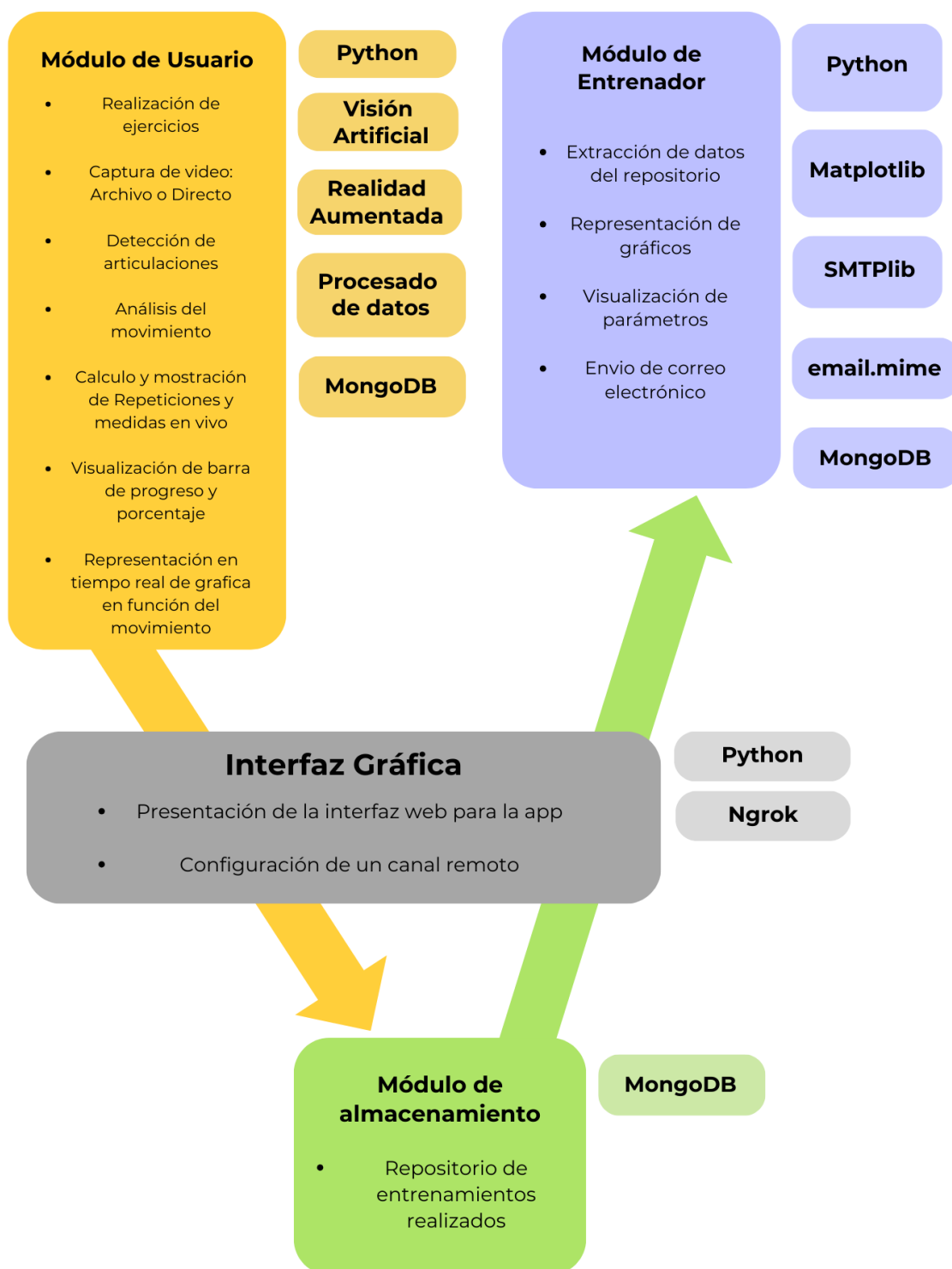


Figura 4.27. Módulos de la Aplicación

4.2.3.1 Módulos

4.2.3.1.1 Módulo de Usuario

En el módulo de usuario, se llevan a cabo los ejercicios deportivos mediante la elección del tipo y la selección de la entada, ya sea a través de un archivo previamente grabado para el análisis del movimiento o en tiempo real mientras el usuario realiza el ejercicio de pesas y se graba para el monitoreo instantáneo. Este módulo es responsable de detectar los puntos clave en el esqueleto y analizar el movimiento. Calcula y muestra en el entorno real el número de repeticiones, grados de flexión y cadencia del movimiento, así como una barra de progreso con su porcentaje. Si el usuario ejecuta el movimiento en tiempo real, se genera una representación gráfica en vivo de la articulación clave en el tiempo, previniendo posibles lesiones por oscilaciones. Todos los datos deportivos procesados se almacenan en un repositorio de entrenamientos.

4.2.3.1.2 Módulo de entrenador

La función principal de este módulo es la extracción de datos previamente procesados del repositorio, presentándolos al entrenador mediante gráficos y tablas. Esto simplifica la tarea de revisar individualmente las grabaciones de todos los clientes, permitiendo al entrenador realizar un análisis deportivo comparando marcas. Posteriormente, se posibilita el contacto a través de correo electrónico para informar sobre posibles ejercicios mal ejecutados que puedan ocasionar lesiones articulares y daños musculares.

4.2.3.1.3 Módulo de almacenamiento

En este módulo se realiza la tarea de archivar todos los datos esenciales (velocidad, cadencia, tiempo de duración, repeticiones realizadas...)

De este modo, se establece un repositorio que almacena los registros de los entrenamientos deportivos ejecutados por los usuarios de la aplicación. Esta elección prescinde de una base de datos local, adoptando un enfoque diferenciado que facilita el acceso desde cualquier dispositivo.

4.2.3.1.4 Módulo gráfico

Es parte integral de la estructura, ya que facilita la visualización de una fase beta de la aplicación.

En primer lugar, posibilita que el usuario elija remotamente el tipo de ejercicio a realizar y lo cargue para su procesamiento, extrayendo datos relevantes que se almacenarán en el repositorio de entrenamientos.

Por otro lado, permite al entrenador deportivo acceder de manera remota, por ejemplo, mediante un dispositivo móvil, a sus clientes que almacenan toda esta información en el repositorio, facilitando el acceso a parámetros deportivos por nombre de ejercicio y fecha de registro. Finalmente, en la interfaz misma, el entrenador se comunica por correo electrónico con el cliente seleccionado.

4.2.3.2 Arquitectura de la aplicación

En el siguiente apartado, se presenta la arquitectura de la aplicación mediante la división de los módulos previamente comentados. El desarrollo se explica de manera detallada, abordando cada paso de forma sistemática.

4.2.3.2.1 Arquitectura del módulo de usuario

En primer lugar, se expone la estructura completa, como se muestra en la figura 4.28, con el fin de proporcionar una visión general de su funcionamiento. Posteriormente, se procederá a desglosar el diagrama, explicando la implementación llevada a cabo en cada sección.

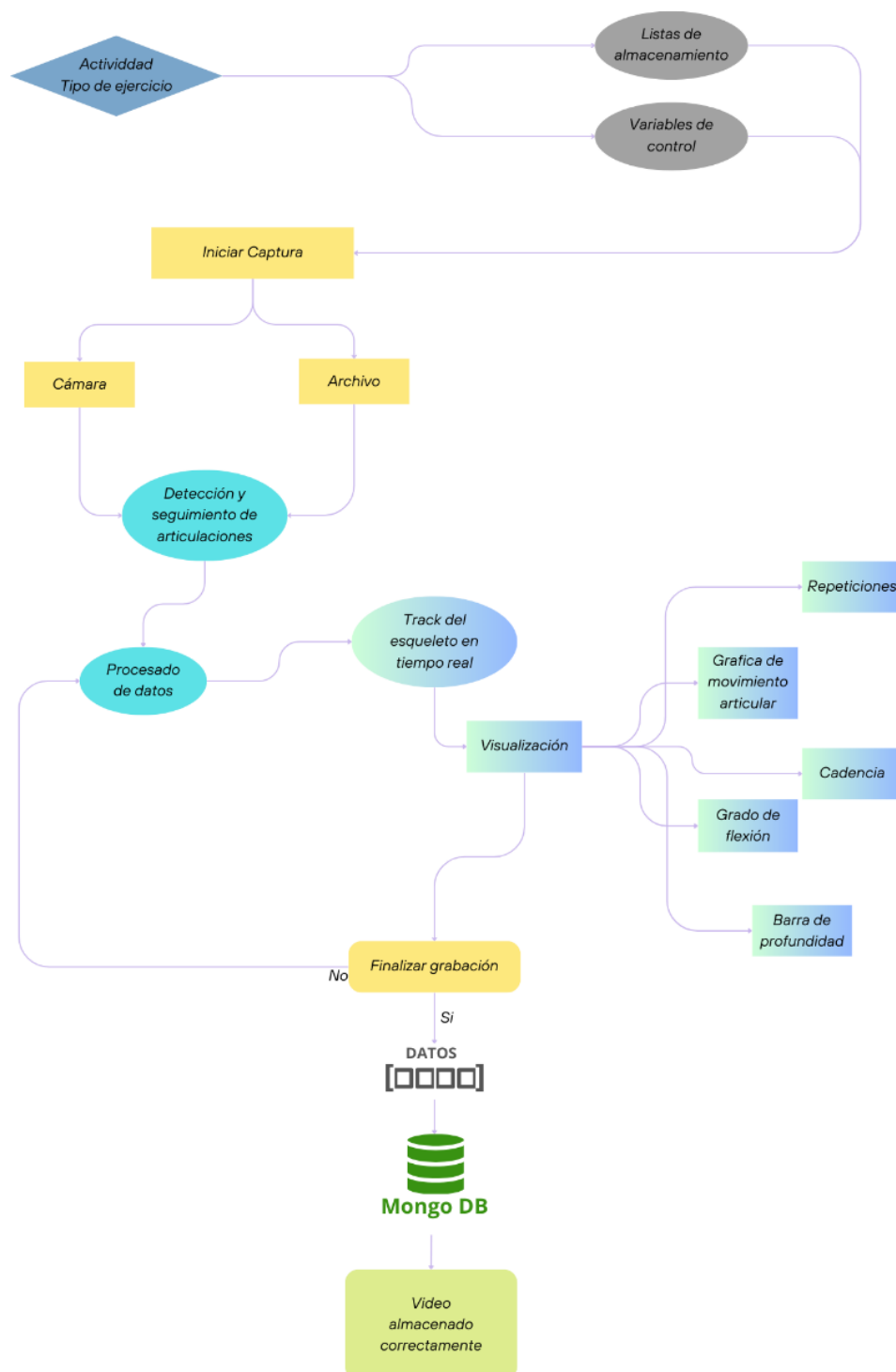


Figura 4.28. Diagrama de flujo del Usuario

4.2.3.2.1.1 Listas y variables de control

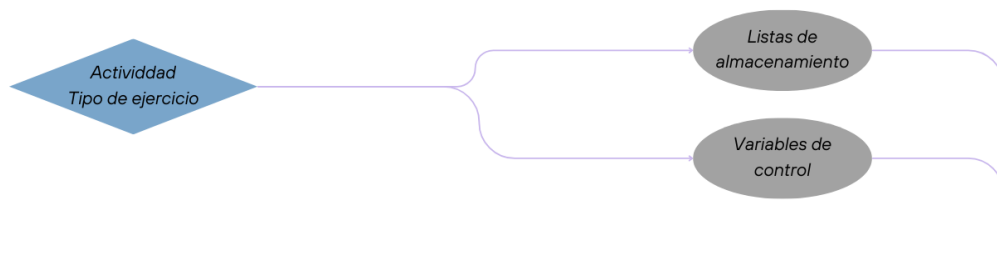


Figura 4.29. Zoom 1 del diagrama de flujo del usuario

En esta sección, se definen diversas listas vacías destinadas a almacenar información vinculada al procesamiento de la aplicación, tales como puntos clave, tiempos, distancias, ángulos, velocidades y cadencias, como se muestra en la figura 4.30.

```
## Se crean listas para almacenar información relacionada con el procesamiento de la aplicación.
tiempos = []
puntos = []# guardo x1 e y1:
TX_porfin = []
Velocidades_2_1 = []
distancias = []
angulos = []
PuntosY1 = []#solo guardo y1:
PuntosX1 = []
# ... (otras listas)
```

Figura 4.30. Inicialización de Listas

Además, se procede a la inicialización de variables encargadas de gestionar el conteo de repeticiones, la cadencia y otros parámetros asociados al rendimiento físico, como se ilustra en la figura 4.31.

```
79 # Se inicializan variables y listas específicas
80 #para gestionar el conteo de repeticiones, la cadencia y otros parámetros.
81 up = False # arriba
82 down = False # abajo
83 count = 0 # contador de repeticiones
84 #Barra
85 dir = 0
86 count2 = 0
87
```

Figura 4.31. Configuración de Variables

4.2.3.2.1.2 Inicializar captura de imagen

Una vez que se cuenta con todos los elementos listos para almacenar datos y variables definidas, se procede a la inicialización de la captura de imagen, como muestra la figura 4.32. En este punto, se encuentran dos modos de entrada: el primero involucra el uso de la cámara, mientras que el segundo implica un archivo contenido en la carpeta de

video locales denominada “*database*”. El acceso a esta carpeta se facilita a través del enlace depositado en la sección de anexos.

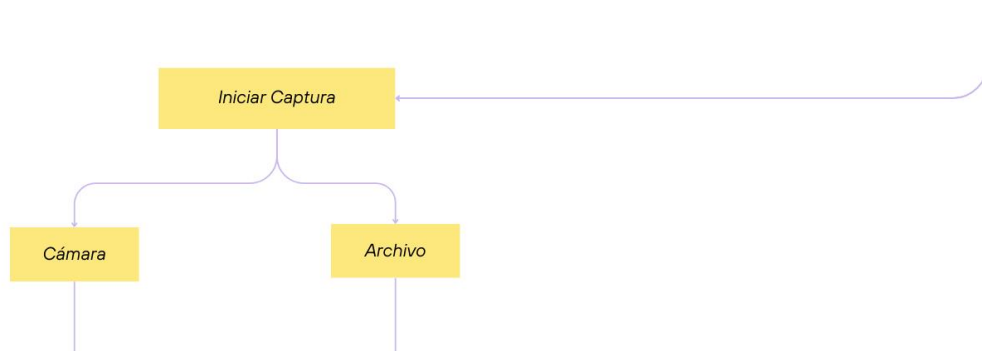


Figura 4.32. Zoom 2 del diagrama de flujo del usuario

Haciendo uso de la biblioteca OpenCV, tal como se detalló en la sección de análisis, se inicia la grabación de video utilizando la cámara conectada al sistema, generando así una variable que se convierte en un objeto de captura de video, referenciada en la figura 4.33. Este objeto será empleado posteriormente dentro de un bucle para la lectura y procesamiento de los fotogramas provenientes de la cámara.

```
89 # Inicia la grabación desde dispositivo conectado al sistema
90 cap = cv2.VideoCapture(1)
91
```

Figura 4.33. Configuración de la Cámara

En segundo lugar, se propone la opción de seleccionar la alternativa de cargar un archivo específico, indicando la ruta del video que se desea abrir. A modo de ejemplo, en la figura 4.34 se hace referencia al archivo de video disponible en la carpeta enlazada en los anexos.

```
89 # Inicia la grabación desde del archivo "M_PHNV_1.mp4" ubicado en la carpeta "database".
90 cap = cv2.VideoCapture("database/M_PHNV_1.mp4")
```

Figura 4.34. Configuración del Archivo

4.2.3.2.1.3 Detección de articulaciones y procesamiento de datos

Esta etapa constituye el punto central de la aplicación donde se inicia el proceso de detección de poses, la extracción de puntos clave y el procesamiento.

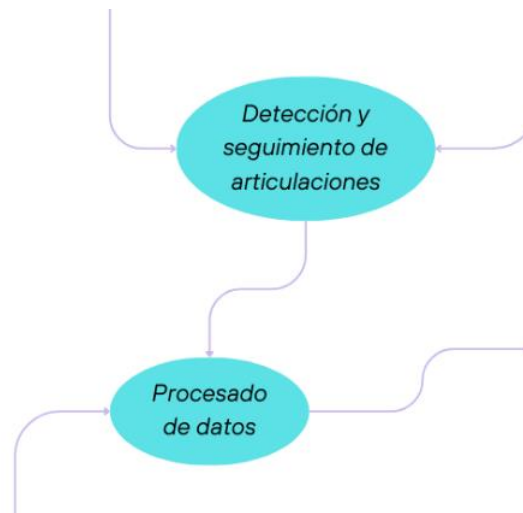


Figura 4.35. Zoom 3 del diagrama de flujo del usuario

En el proceso de detección de pose y extracción de puntos claves, se hace uso del modelo de MediaPipe para identificar puntos claves en el cuerpo humano, tales como las articulaciones. En primer lugar, se importan los módulos de MediaPipe y se genera una instancia de la clase, representada en la figura 4.36. Luego, se inicia un bucle que captura continuamente fotogramas del flujo de video, verificando la correcta captura de cada fotograma.

```
64 mp_drawing = mp.solutions.drawing_utils
65 mp_pose = mp.solutions.pose
66
67 with mp_pose.Pose(static_image_mode=False) as pose:
68     while True:
69         ret, frame = cap.read()
70
71         if ret == False:
72             break
```

Figura 4.36. Bucle While captura del fotograma

A continuación, se adquieren las dimensiones del fotograma actual y se realiza la conversión de BGR a RGB, requisito esencial para el procesamiento de MediaPipe. Los resultados obtenidos se almacenan en una variable. Estos pasos habilitan el análisis en tiempo real de la pose en un video, suministrando información acerca de la posición de puntos claves en el cuerpo humano en cada fotograma.

Los formatos BGR y RGB son utilizados para representar el color en imágenes. BGR significa *Blue, Green, Red* y es el utilizado por OpenCV para mostrar el color de los

píxeles de una imagen. RGB, por otro lado, significa *Red, Green, Blue* y es el formato estándar utilizado en la mayoría de las aplicaciones y dispositivos para representar el color. La conversión se ilustra en la figura 4.37.

```

77
78     height, width, _ = frame.shape # el metodo shape de opencv se utiliza para obtener
79     # la dimensiones de la matriz altura ancho y caanles(no es relevante por eso -)
80     frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
81     results = pose.process(frame_rgb)
82

```

Figura 4.37. Dimensiones y Conversión

A continuación, se presenta un ejemplo de implementación para una articulación, extrapolable a cualquier otra. Se obtienen las coordenadas (X, Y) de una articulación específica, en este caso, la articulación de la cadera identificada por el índice 23 en la lista de *landmarks*, tal como se visualiza en la figura 4.38. Estas coordenadas se convierten a píxeles según las dimensiones del marco de la imagen. Posteriormente, se emplean estas coordenadas para realizar operaciones relacionadas con la postura del cuerpo.

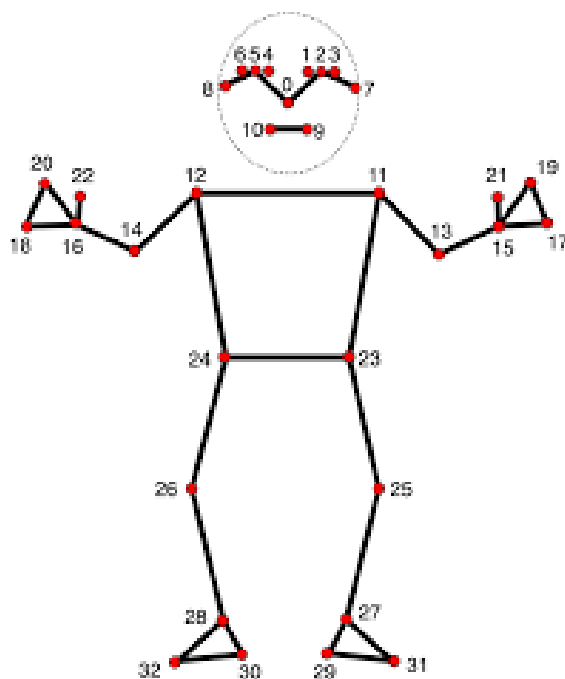


Figura 4.38. Landmarks en MediaPipe [32]

Además, se realiza el cálculo de la marca de tiempo del fotograma actual en microsegundos y se guarda en una lista después de ser convertida a segundos. El fragmento de la figura 4.39 resulta crucial para la captura de información específica sobre la pose, como las coordenadas de ciertas articulaciones, y puede ser empleado para llevar a cabo acciones adicionales o análisis basados en estos datos. Se destaca la posibilidad

de ejecutar acciones con la marca de tiempo, como el cálculo de la velocidad o la cadencia asociada al movimiento.

```

84     if results.pose_landmarks is not None: # si la variable pose_landmarks no esta vacia
85         x1 = int(results.pose_landmarks.landmark[23].x * width) # multiplica los pixeles horizontales
86         #print(results.pose_landmarks.landmark[23])
87         y1 = int(results.pose_landmarks.landmark[23].y * height) # multiplica los pixeles verticales
88         # Obtener la marca de tiempo del marco actual en microsegundos
89         ##Piedra Angular!!!!!!!!!!!!
90         timestamp_microseconds = int(cap.get(cv2.CAP_PROP_POS_MSEC) * 10)
91         TX_porfin.append(timestamp_microseconds/10000)
92         # Realizar acciones con la marca de tiempo (por ejemplo, imprimir)
93         #print("Marca de tiempo en microsegundos:", timestamp_microseconds/10000)
94         # ... (otros puntos de distintas articulaciones)
95

```

Figura 4.39. Coordinada de articulación y marca de tiempo

En el procesado, se abordarán los cálculos realizados con los datos extraídos del proceso de detección de posturas.

En el procedimiento inicial, se realiza el cálculo del ángulo para evaluar el grado de flexión posible en la articulación y proporcionar un conteo de las repeticiones basado en este parámetro. Después de extraer las coordenadas clave de las articulaciones, como codos, rodillas, muñeca y cadera, se lleva a cabo el cálculo de los puntos correspondientes a estas coordenadas. Este proceso implica la utilización de la biblioteca NumPy para estructurar un array, como se ejemplifica en el fragmento del código en la figura 4.40.

```

107     # formalizar los puntos con un array
108     p1 = np.array([x1, y1])#23
109

```

Figura 4.40. Array de coordenadas

Después de estructurar todos los puntos relevantes, es crucial almacenar en una lista el punto clave según el tipo de ejercicio en curso. En la imagen 4.41 de ejemplo, se ilustra el punto 1, que se corresponde con la cadera, ya que el ejercicio seleccionado para el ejemplo es una Sentadilla.

```

110     puntos.append(p1)
111     PuntosY1.append(y1)
112     PuntosX1.append(x1)

```

Figura 4.41. Almacenamiento de punto clave

Posteriormente, se determinan las longitudes (distancias euclidianas) de diversos segmentos corporales (I1, I2, I3, I4 y I5) mediante la norma euclidiana entre dos puntos en un espacio bidimensional. Estas distancias se registran en listas, reflejando la magnitud entre los respectivos puntos en el plano. El cálculo de la distancia euclidiana se lleva a cabo mediante la fórmula correspondiente en la ecuación 1.

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

La función `np.linalg.norm()` de NumPy desempeña el papel de calcular la longitud de un vector, siendo dicho vector la diferencia entre los puntos. En el contexto de la Figura 4.42, esto se aplica específicamente a los puntos `p2` y `p3`.

```
119     l1 = np.linalg.norm(p2 - p3)
120     #print("estoy representando l1:", l1)
121     long_1.append(l1)
122
```

Figura 4.42. Longitud de un vector

Cuando se tienen las longitudes, se recurre al uso del Teorema del Coseno. Esta herramienta matemática se emplea para calcular el ángulo generado por la flexión de la articulación, como se ilustra en la figura 4.43. En este caso, la articulación que flexiona es la rodilla, debido a la ejecución de una Sentadilla. La finalidad de este cálculo es la obtención posterior de diversos parámetros que dependen directamente del valor angular resultante.

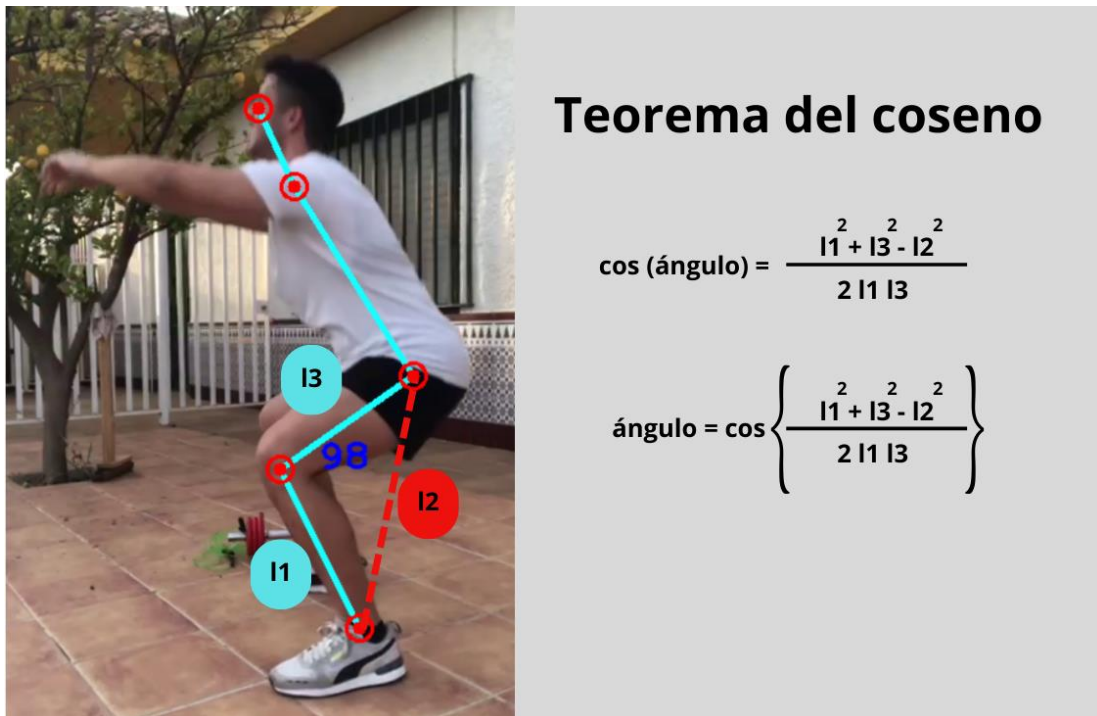


Figura 4.43. Teorema del Coseno

La información se almacena en una lista con el propósito de generar gráficos que permitan observar y analizar la coherencia y fluidez del movimiento. Este grado de flexión

obtenido a través de los cálculos realizados permite detectar la correcta realización del ejercicio desde una perspectiva biomecánica.

Dentro de la implementación, se captura el momento previo al inicio de una repetición, y a lo largo del proceso, se monitorean de manera constante los ángulos que caracterizan la postura corporal. La detección del inicio y final de cada repetición se logra mediante la definición de umbrales específicos de ángulos, los cuales se ajustan según el tipo de ejercicio realizado. Esto se ejemplifica en el siguiente caso para la Sentadilla, como se muestra en la figura 4.44.

```
if angle >= 150:
    up = True # persona esta parada arriba
if up == True and down == False and angle <= 80 : # cuando la persona esta bajando y pasa el rango
    down = True
if up == True and down == True and angle >= 150: # si up y down han sido true se incrementa el contador
    count += 1
    up = False
    down = False
timestamp_microseconds_Contador = int(cap.get(cv2.CAP_PROP_POS_MSEC) * 10)
```

Figura 4.44 Contador de repeticiones

Este enfoque adaptable facilita un seguimiento preciso de las repeticiones, permitiendo así una evaluación detallada de la ejecución del ejercicio deportivo y proporcionando información valiosa para el análisis del rendimiento del usuario.

A continuacion se calcula la velocidad del movimiento asociado al punto articular clave. La selección de este punto puede variar según el tipo de ejercicio realizado. Por ejemplo, durante la Sentadilla, se evalúa el punto ubicado en la cadera; en el Peso Muerto, se identifica en el hombro; y en el Press Militar, se elige la muñeca. Esta selección se basa en consideraciones deportivas y biomecánicas, ya que estos puntos son los que experimentan más oscilaciones y balanceo durante estos ejercicios.

Para determinar la velocidad, es necesario calcular el desplazamiento realizado por unidad de tiempo. Este proceso se realiza en tiempo real dentro del bucle de captura de fotogramas.

En primer lugar, se determinan las diferencias (Δt) entre las marcas de tiempo consecutivas como muestra la ecuación 2.

$$\Delta_{t_i} = T_{[i+1]} - T_{[i]} \quad (2)$$

Posteriormente, se calculan las diferencias de posición entre puntos consecutivos, como se indica en la ecuación 3. Esto se lleva a cabo utilizando el método `np.diff()` sobre el vector de puntos almacenado durante la detección de poses, P1 en el caso del ejemplo, como se visualiza en la figura 4.40.

$$\Delta x_i = x_{[i+1]} - x_{[i]} \quad (3)$$

$$\Delta y_i = y_{[i+1]} - y_{[i]}$$

Luego, se calculan las distancias euclidianas entre estos puntos consecutivos, y finalmente, se obtienen las velocidades promedio dividiendo las distancias por las diferencias de tiempo, como ilustra la ecuación 4. El resultado se almacena en una lista.

$$d_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2} \quad (4)$$

$$V_{Promedio_i} = \frac{d_i}{\Delta t_i}$$

Es importante destacar que se verifica si la lista de velocidades tiene elementos antes de intentar acceder al último valor. Si la lista está vacía, imprime un mensaje indicando que no tiene elementos aún. La velocidad resultante se presenta en tiempo real y será la unidad de medida de la distancia dividida por la unidad de medida del tiempo.

Si las coordenadas X e Y están en píxeles y las marcas de tiempo están en segundos, entonces la unidad de medida para la velocidad sería $Píxeles/s$.

Con el objetivo de proporcionar parámetros más comprensibles al monitor deportivo, se ha integrado la conversión a m/s . Esta adaptación busca ofrecer una medida

más realista al abordar el tema del movimiento, permitiendo una interpretación más intuitiva de las velocidades.

Inicialmente, se realiza el cálculo de la medida en metros correspondiente a un píxel, tomando en cuenta la altura predefinida del individuo y la suma de las longitudes máximas que abarcan el esqueleto desde el tobillo hasta la cabeza en la imagen. Este proceso proporciona una relación que permite la conversión de unidades, estableciendo una escala realista para interpretar medidas.

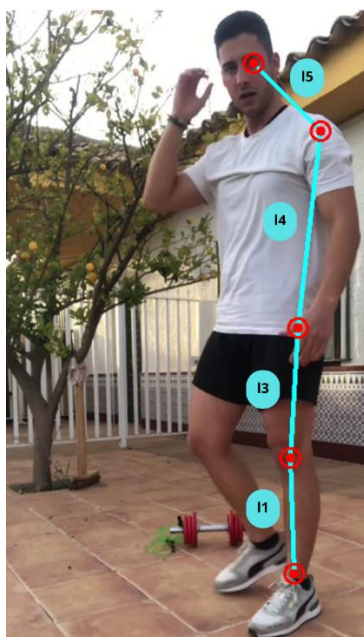


Figura 4.45. Longitudes del esqueleto

Luego, se multiplica cada velocidad en píxeles por segundo por esta escala para obtener las velocidades correspondientes en m/s . Finalmente, se calcula la velocidad media en metros por segundo a partir de todas las velocidades registradas.

En la implementación del código, disponible a través del enlace proporcionado en los anexos, se ha calculado la aceleración como la derivada de la velocidad con respecto al tiempo. Este cálculo se realiza utilizando la librería NumPy para obtener las diferencias entre las velocidades en dos *frames* consecutivos y las diferencias de tiempo correspondientes. Luego, se divide la diferencia de velocidades entre la diferencia de tiempo para obtener la aceleración. Aunque se ha incorporado esta medida, se ha decidido no mostrarla en la versión final de la aplicación, ya que su contribución actual es limitada en términos de valor. No obstante, se reconoce que podría ser interesante para evaluar la fuerza aplicada en el ejercicio. Por esta razón, aunque se haya descartado en esta primera versión, no se descarta su posible inclusión en versiones futuras del proyecto.

Se ha implementado un proceso de filtrado de la señal de velocidad para mejorar la representación gráfica. Al analizar la gráfica de velocidad con respecto al tiempo, se observó un considerable ruido debido a las oscilaciones del punto seleccionado en la imagen. Para abordar este problema, se aplicaron técnicas de tratamiento digital de señales, utilizando filtros de media móvil con diferentes ventanas (Rectangular y Hamming). Tras evaluar los resultados, se optó por la ventana rectangular, como se ilustra en la figura 4.46, ya que proporcionaba resultados válidos para el propósito establecido.

```
131 def media_movil_rectangular(signal, ventana):
132     # Crear el filtro de ventana rectangular
133     weights = np.ones(ventana) / ventana
134
135     # Aplicar filtro de media móvil con convolución
136     signal_filtrada = np.convolve(signal, weights, mode='same')
137     return signal_filtrada
138
139
```

Figura 4.46. Filtro de media móvil con ventana rectangular

Además, con el fin de eliminar la influencia de los datos que están fuera del rango deseado, es decir, aquella información que se registra cuando el ángulo de flexión es mayor 150 y menor a 81, se llevó a cabo una limpieza de la señal. Restringiendo el movimiento al grado de flexión óptimo de la articulación, durante la repetición y eliminando los datos atípicos. Estos ajustes contribuyeron a una representación más clara y coherente de la señal de velocidad como se observa en la figura 4.47

```
141 def Threshold_angulos(signal, ang):
142     umbral= 0
143     signal_filtrada = []
144
145     for idx, muestra in enumerate(signal):
146         if muestra < umbral or ang[idx] > 150 or ang[idx] < 81:
147             signal_filtrada.append(0)
148         else:
149             signal_filtrada.append(muestra) # Agregar el valor original en lugar de la media
150
151     return signal_filtrada
152
```

Figura 4.47. Threshold para ángulos

En la implementación del cálculo de la cadencia, se procesa un array que almacena los períodos en segundos de cada repetición. En cada ciclo, se suma este tiempo y se incrementa el contador de muestras. Cuando se completan dos repeticiones, se calcula la cadencia media en repeticiones por minuto, y se añade a la lista de cadencias. Luego, se reinicia el tiempo total para acumular el siguiente conjunto de repeticiones. Se presenta una cadencia en tiempo real actualizada con base en las últimas dos repeticiones, permitiendo visualizar la cadencia instantánea y estimar el ritmo de manera más precisa.

Posteriormente, se obtiene la cadencia media de las repeticiones por minuto en la serie completa.

Por último, también se calculan parámetros estadísticos de estos valores para lograr una interpretación del movimiento más detallada. Parámetros como la media, que representa el valor promedio de la señal. Este valor es útil para comprender la tendencia general de la señal registrada durante el ejercicio.

Además, se obtienen los valores máximos y mínimos, proporcionando así una visión completa de la señal registrada y su variabilidad durante el rendimiento deportivo. Ofreciendo una comprensión detallada de la amplitud de los datos, enriqueciendo la evaluación de los movimientos durante la actividad física.

Adicionalmente, se calcula la desviación estándar, que mide la dispersión de los datos en relación con la media. En el contexto del análisis de movimiento, la desviación estándar revela la uniformidad del movimiento registrado. Por ejemplo, una baja desviación estándar en la señal de velocidad indica que los valores están cercanos a la media, sugiriendo un movimiento constante. De manera análoga, una desviación baja en la cadencia señala un ritmo de repetición constante, sin variaciones significativas durante el ejercicio, lo cual refleja una ejecución efectiva del movimiento.

Estos cálculos se realizan mediante funciones estadísticas disponibles en bibliotecas como NumPy.

4.2.3.2.1.4 Visualización y dibujo de elementos en el fotograma

En esta sección, ilustrada en la figura 4.48, se presentan elementos en el entorno real que cumplen la función de guía para el usuario durante la ejecución del ejercicio. La visualización de estos elementos busca proporcionar una referencia visual continua y en tiempo real para ayudar al usuario a seguir correctamente el movimiento.

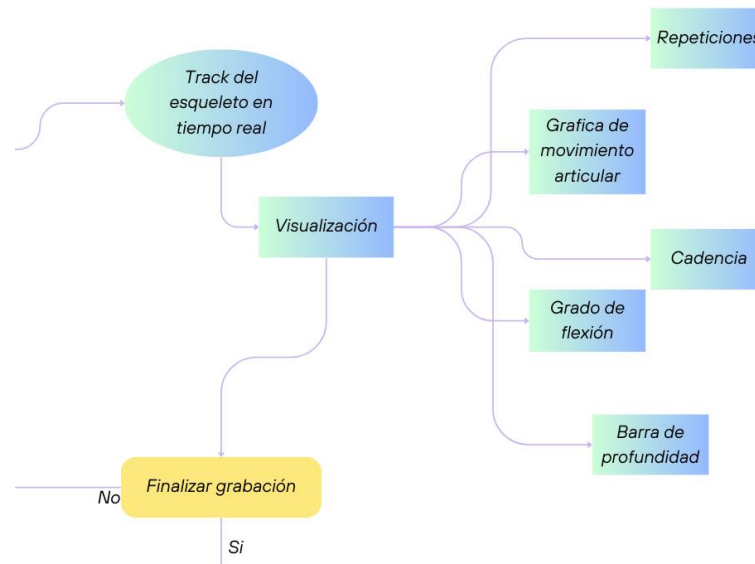


Figura 4.48. Zoom 4 del diagrama de flujo del usuario

Para ello se crea una imagen de fondo negra con las mismas dimensiones que el *frame* actual. Luego, utilizando la función `addWeighted()` de OpenCV, se combina esta imagen de fondo con el *frame* original, aplicando un efecto de transparencia. El resultado se almacena en la variable de salida como indica la figura 4.49.

251	<code>aux_image = np.zeros(frame.shape, np.uint8)</code>
252	<code>output = cv2.addWeighted(frame, 1, aux_image, 0.8, 0) # 0.8 muestra transparencia</code>

Figura 4.49. Variable de salida

Posteriormente, se incorporan elementos visuales en la imagen combinada, con el propósito de resaltar características en el entorno real. Estos elementos incluyen líneas que conectan los puntos clave (Longitudes), círculos alrededor de las articulaciones, barra de progreso y texto que muestra información como el número de repeticiones, la cadencia y el ángulo de la articulación (Grado de flexión).

Por otro lado, cabe destacar la implementación de una gráfica en tiempo real que ilustra el punto que ocupa la articulación, estableciendo una relación directa con la medida en píxeles. Utilizando la biblioteca Matplotlib y Tkinter para la interfaz gráfica, la función `grafica_live()` limpia y actualiza la gráfica mostrando la evolución temporal del punto articular. La coordenada X describe la posición horizontal del píxel, mientras que la coordenada Y indica su posición vertical. Por ejemplo, en una imagen de 100 píxeles de ancho y 50 píxeles de alto, la coordenada (50,25) se correspondería con el centro de la imagen, representando la mitad del ancho y la mitad del alto.

Esta representación gráfica se expone con el propósito de proporcionar al usuario una herramienta adicional para la corrección de movimientos durante la ejecución de los

ejercicios. La visualización detallada de las coordenadas articular ofrece al usuario la posibilidad de identificar cualquier oscilación notable en el patrón de movimiento. Esta función resulta crucial, ya que permite una evaluación más precisa y en tiempo real de la alineación y estabilidad de la articulación, brindando así la oportunidad de ajustar y perfeccionar la técnica de manera inmediata, contribuyendo de este modo a la optimización del rendimiento durante la sesión de entrenamiento.

Por último, se ha integrado un botón en la interfaz para detener la captura de imágenes, proporcionando una herramienta adicional para la finalización del proceso de registro de datos. En la siguiente figura 4.50 se muestra un ejemplo de visualización.

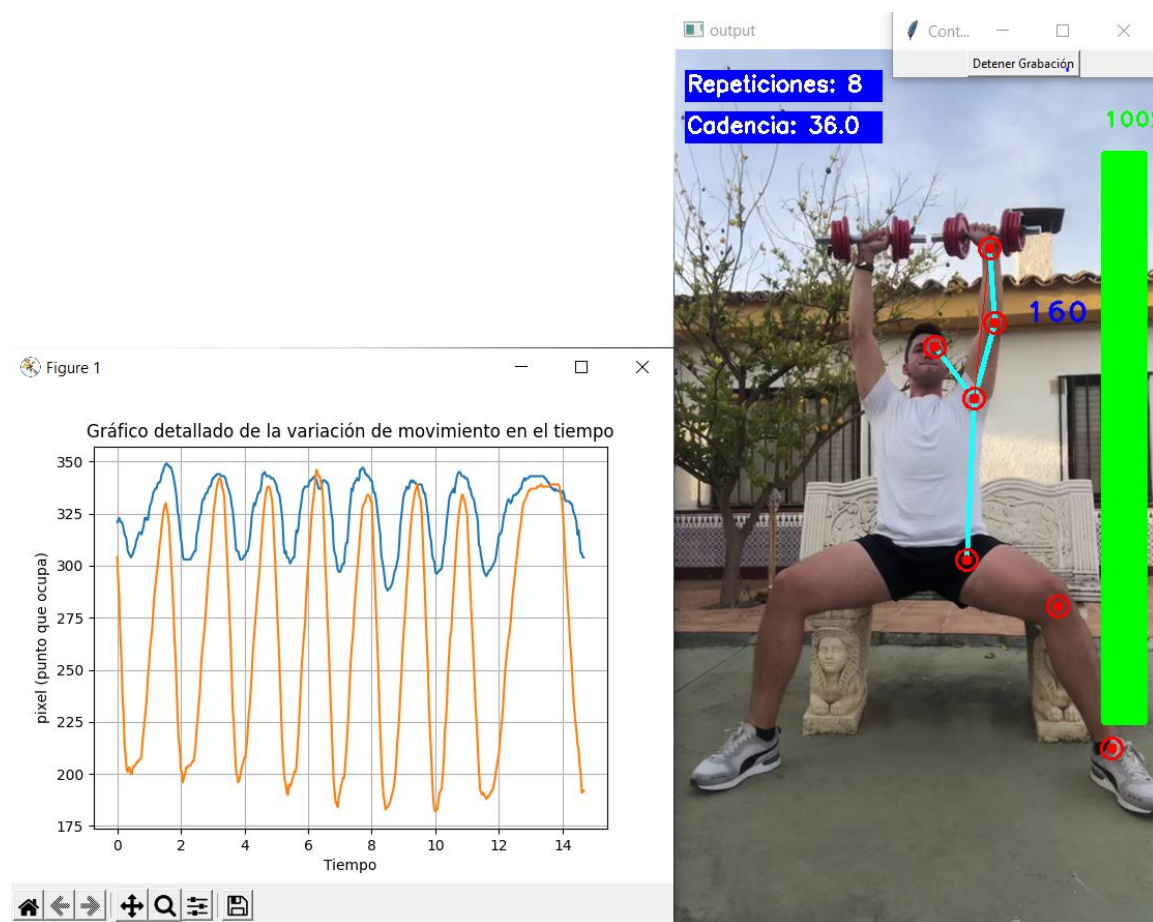


Figura 4.50. Ejemplo de visualización

4.2.3.2.1.5 Envío de datos al repositorio

Una vez que se detiene la grabación se implementa la función llamada `insertar_datos_mongodb_usuarios()` la cual se encarga de almacenar datos relacionados con el rendimiento de un ejercicio deportivo en una base de datos MongoDB, tal como se presenta en el esquema de la figura 4.51. Esto facilita un seguimiento y análisis posteriores.



Figura 4.51. Zoom 5 del diagrama de flujo usuario

En esta fase beta, se solicita confirmación al usuario por el terminal para guardar los datos en la base. Esto se debe a que se realizaron diversas pruebas para lograr un procesamiento eficiente y una obtención de valores coherentes.

En caso de una respuesta afirmativa, se realiza la conexión con MongoDB mediante PyMongo. La cadena de conexión contiene el nombre de usuario, la contraseña y la dirección del *cluster*. Se establece un enlace y se elige la base de datos, como se muestra en la figura 4.52. Es importante destacar que se ha creado previamente un nuevo *cluster* en la plataforma, como se observará en el módulo de almacenamiento.

```
# Conexión a MongoDB
cluster = MongoClient("mongodb+srv://aoc00025:12345@cluster0.blj29c3.mongodb.net/?retryWrites=true&w=majority")
db = cluster["RUTINAS"]
```

Figura 4.52. Conexión a MongoDB

Posteriormente, se emplea el nombre de usuario para generar una colección exclusiva destinada a ese cliente dentro del repositorio. Luego, se genera la fecha y hora actuales y se construye un diccionario, como en la imagen 4.53, el cual incluye información estructurada como el tipo de ejercicio, velocidades, ángulos, movimiento, cadencias, entre otros.

```
# Crear un diccionario con los datos del ejercicio
datos_ejercicio = {
    "tipo_ejercicio": tipo_ejercicio,
    "Peso": Peso,
    "velocidades": [{ 'velocidad': v } for v in velocidades],
    "resultado_filtro_angulo_Rectangular": [{ 'velocidad_filtrada': vf } for vf in resultado_filtro_angulo_Rectangular],
    "Desviacion_estandar_velocidad" : Desviacion_estandar_velocidad,
    "angulos": [{ 'angulo': a } for a in angulos],
    "angulos_divididos": [{ 'angulo_dividido': ad } for ad in angulos_divididos],
    "movimiento": [{ 'movimiento': m } for m in movimiento],
    "PuntosX1": [{ 'puntox1': px } for px in PuntosX1],
    "velocidad_media": velmediams,
    "cadencias": [{ 'Cadencia': c } for c in cadencias],
    "cadencia_media": cadencia_media,
    "desviacion_estandar_cadencia": desviacion_estandar_cadencia,
}
```

Figura 4.53. Creación de diccionario

Finalmente, se introduce este diccionario en la colección correspondiente del usuario en MongoDB mediante la función `insert_one()`. La implementación detallada del código está disponible en los anexos.

4.2.3.2.2 Arquitectura del módulo de entrenador

En primer lugar, se presenta la estructura completa en la figura 4.54 con el objetivo de ofrecer una visión general de su funcionamiento. Luego, se procederá a desglosar el diagrama, detallando la implementación realizada en cada sección.

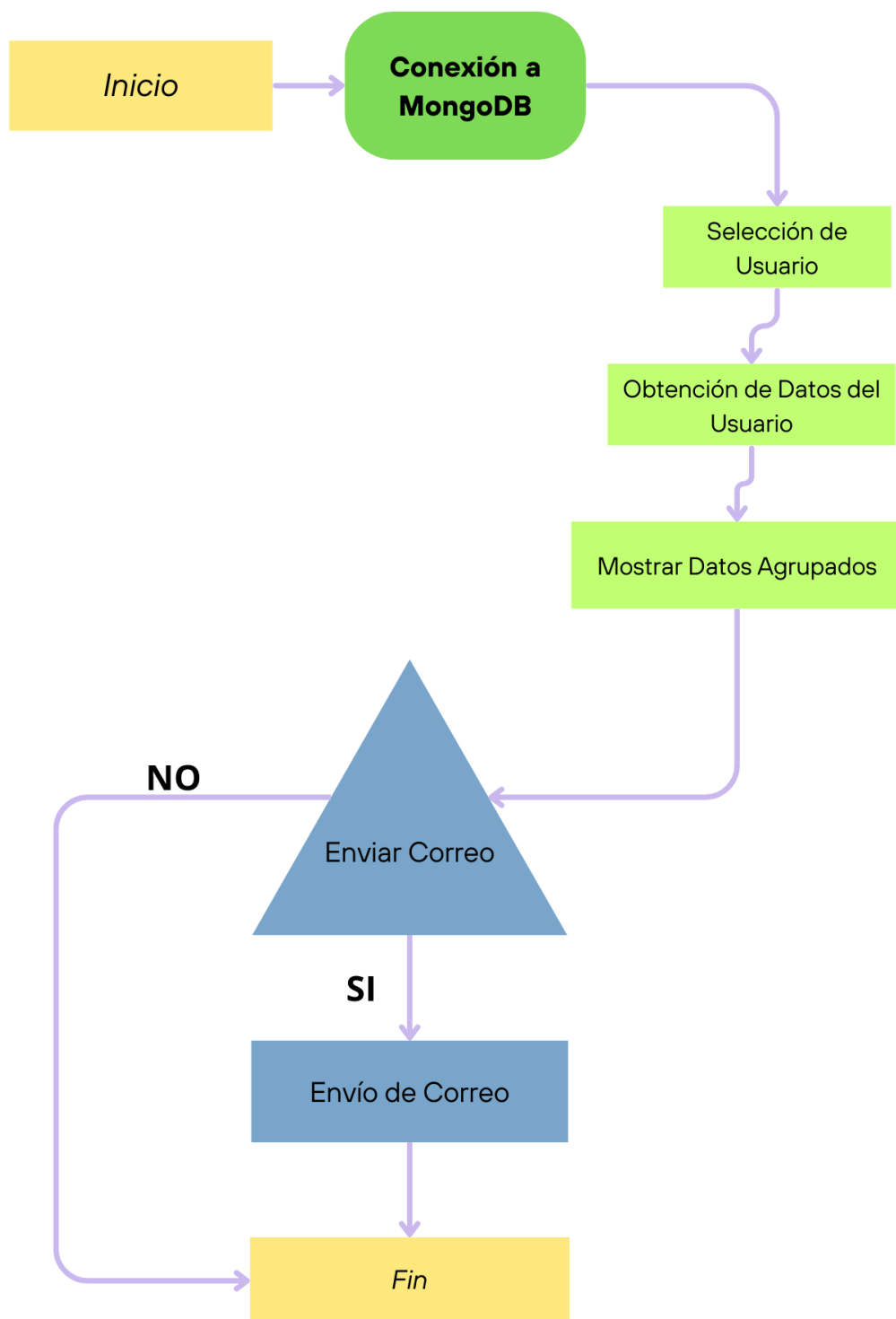


Figura 4.54. Diagrama de flujo de entrenador

4.2.3.2.2.1 Conexión a MongoDB y obtención de datos del usuario

En la sección inicial, representada en la figura 4.55, se establece la conexión con el repositorio que almacena los parámetros deportivos junto con sus respectivos usuarios y fechas de registro. Se pretende seleccionar un deportista específico para obtener todos sus datos deportivos, con la finalidad de presentarlos al entrenador para que pueda analizar el entrenamiento.

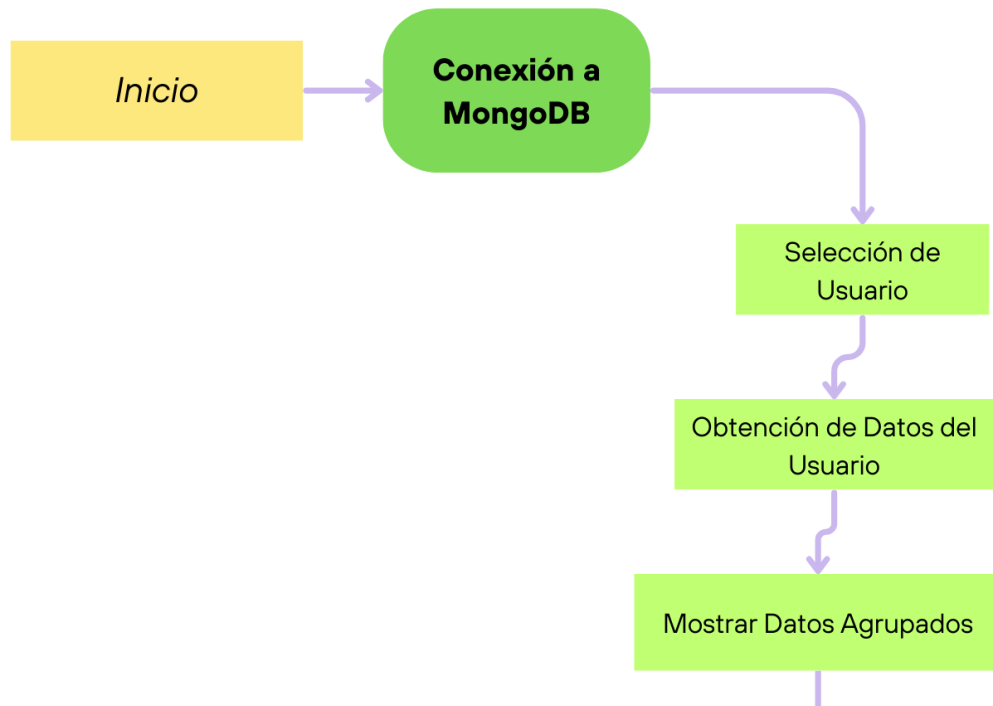


Figura 4.55. Zoom 1 del diagrama de flujo de entrenador

En la implementación del código se utiliza la función `obtener_usuarios()` para crear un objeto MongoClient con la librería de Pymongo y así establecer una conexión con el *cluster* creado en la plataforma. Posteriormente se selecciona la base de datos y se obtiene con el metodo `list_collection_names()` la lista de nombres de las colecciones (usuarios) en la base de datos.

En la obtención de datos, se emplea la función `obtener_datos_usuario()`. Similar a la función anterior, se establece una conexión utilizando un objeto MongoClient y se elige la base de datos. Luego, se accede a la colección del usuario específico. Para obtener todos los parámetros deportivos en esa colección, se utiliza la función `find()`. Este proceso permite recuperar la información relevante del usuario de manera estructurada.

4.2.3.2.2.2 Canal de comunicación con el usuario

En esta parte, el esquema representado en la figura 4.56 muestra el seguimiento de la función para enviar correos electrónicos al usuario. Utilizando credenciales SMTP (Protocolo Simple de Transferencia de Correo) para autenticarse en un servidor de correo, en este caso, el servidor de Gmail.

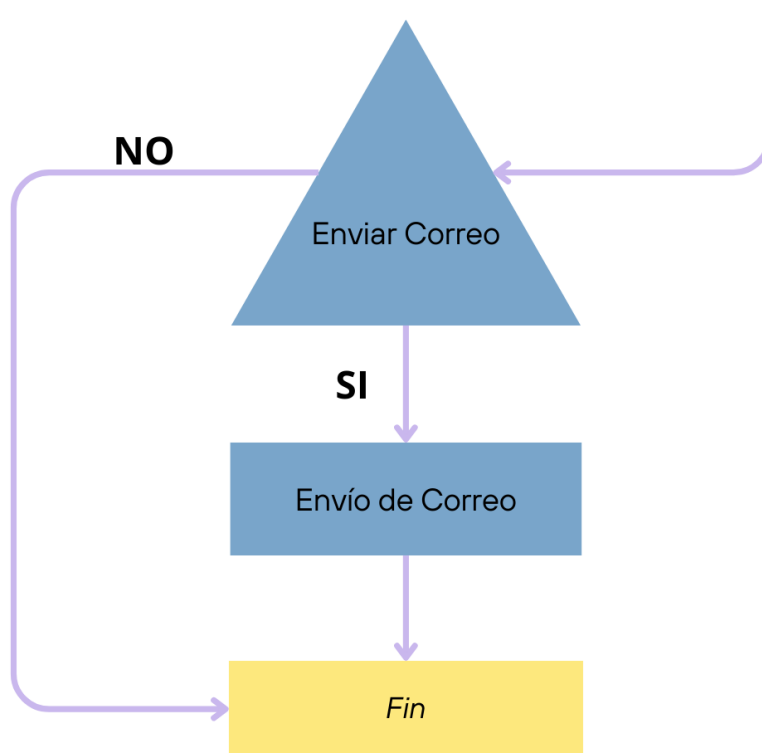


Figura 4.56. Zoom 2 del diagrama de flujo de entrenador

Para lograrlo, se ha procedido a configurar previamente una cuenta de Gmail designada al entrenador; sin embargo, surge un inconveniente. Desde el 30 de mayo de 2022, Google ha modificado su política de soporte, restringiendo el acceso a aplicaciones consideradas menos seguras en comparación con los estándares previos, según se describe en [33]. Por ejemplo, si se desea utilizar el servidor SMTP de Gmail para enviar correos electrónicos, no es posible hacerlo.

Por consiguiente, es necesario llevar a cabo la configuración, inicialmente accediendo a la cuenta de Gmail asignada al entrenador, creada específicamente para esta fase beta de la aplicación. Dentro de la cuenta principal, en la sección de seguridad, se debe activar la verificación en dos pasos. Una vez completado este paso, es necesario configurar la contraseña de aplicación. Esta acción permitirá conectar el servicio de SMTP utilizando una contraseña de 16 dígitos proporcionada por Google, junto con el nombre de

usuario de Gmail, desde el código implementado como se muestra en la figura 4.57. De esta manera, se posibilitará el envío de correos electrónicos a través del SMTP de Gmail.

```
11     usuario_smtp = 'correoentrenador6@gmail.com'
12     contraseña_smtp = 'srgd dige hlco gqlx'
```

Figura 4.57. Contraseña y correo

En el código se desarrolla la función `enviar_correo()` utilizando las bibliotecas `Smtplib` y `Email.mime.text`. Estas bibliotecas son esenciales para establecer una conexión segura con el servidor SMTP de Gmail, configurar los elementos del mensaje, como el asunto, remitente y destinatario, y estructurar el contenido del mensaje. Con estas herramientas, se logra una integración eficiente y segura para el envío de correos electrónico al usuario. Con este proceso se finaliza la ejecución del módulo.

4.2.3.3 Integración

En esta sección de integración, es importante señalar que, aunque los aspectos de programación pueden ser más livianos, la integración de la aplicación con otras herramientas externas requiere un análisis detallado de la documentación correspondiente. Además, es necesario llevar a cabo múltiples pruebas hasta lograr una integración adecuada y sin fallos. Este proceso garantiza que la aplicación funcione de manera óptima y se pueda aprovechar al máximo su potencial en conjunto con otras herramientas externas.

4.2.3.3.1 Integración del módulo de almacenamiento

En el proceso inicial, se eligió MongoDB [34] como la base de datos no relacional para el repositorio de entrenamientos deportivos. Para integrar esta tecnología con el entorno virtual en Visual Studio Code, se llevó a cabo la instalación del controlador de MongoDB para Python mediante el comando `pip install pymongo` en la terminal, seguido de su importación en el script. Como muestran las figuras 4.58 y 4.59.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Universidad\TFG_AdolfoOrzaes\Apto> pip install pymongo
>> █
```

Figura 4.58. Instalación de PyMongo

```
18     from pymongo import MongoClient
19     #/.....
```

Figura 4.59. Importación de librería

Posteriormente, se procedió a registrar una cuenta en MongoDB Atlas con el objetivo de configurar un proyecto y generar un *cluster* dentro de dicho proyecto. Es necesario incluir previamente la dirección IP en la lista autorizada, estableciendo reglas de seguridad para permitir la conexión desde la aplicación.

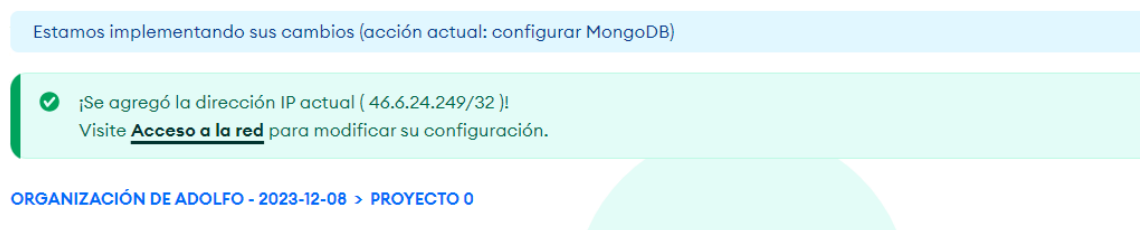


Figura 4.60. Agregación de IP

Una vez creado el *cluster*, se procede a configurar una base de datos denominada RUTINAS, referenciada en la figura 4.61. En esta base de datos, se generan colecciones individuales, que representan a cada usuario. Estas colecciones son generadas desde el módulo de usuario después de procesar los datos de entrenamiento.

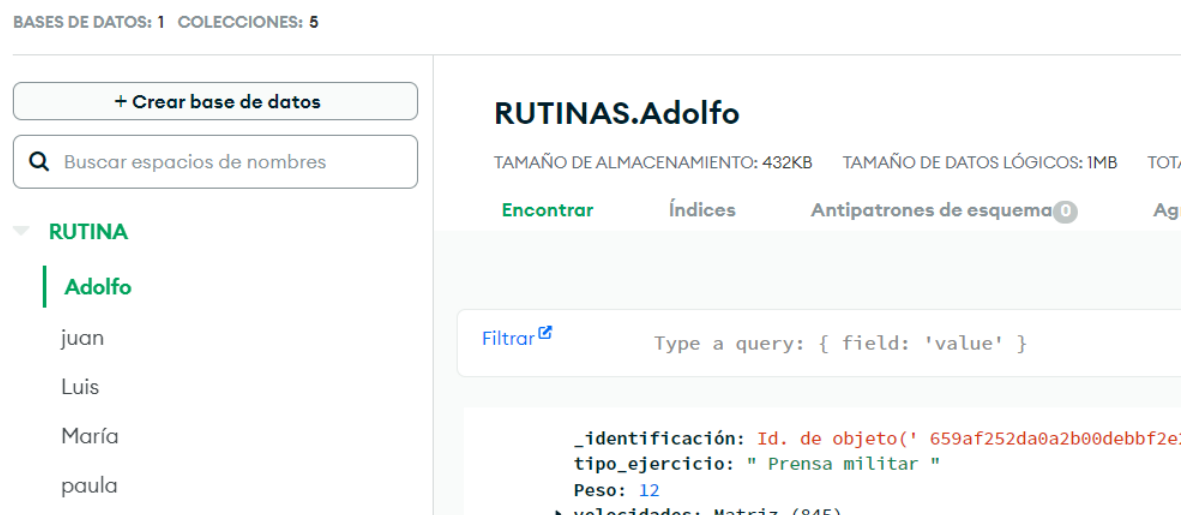


Figura 4.61. Base de datos y Colecciones

Es esencial resaltar que los módulos utilizan el repositorio después de una configuración inicial. Esto implica acceder a la pestaña *Clusters* en MongoDB Atlas, seleccionar *Connect* y obtener la *URL* de conexión.

Conéctese al clúster0



Debe proteger su clúster MongoDB Atlas antes de poder usarlo. Establezca qué usuarios y direcciones IP pueden acceder a su clúster ahora. [Leer más](#)

1. Agregar una dirección IP de conexión

✓ Se ha agregado una dirección IP a la lista de acceso IP. Agregue otra dirección en la [pestaña Lista de acceso IP](#).

2. Crear un usuario de base de datos

Este primer usuario tendrá [atlasAdmin](#) permisos para este proyecto.

Tenga sus credenciales a mano, las necesitará para el siguiente paso.

Nombre de usuario

aoc00025

Contraseña

12345

ESCONDER



Generar automáticamente una contraseña
segura



Copiar

Figura 4.62. Configuración de credenciales

Posteriormente, estas credenciales son sustituidas en el script de Python para garantizar la conexión adecuada con el repositorio. Se puede observar en la figura 4.63 la utilización de la *URL*.

```
45 # Conexión a MongoDB
46 cluster = MongoClient("mongodb+srv://aoc00025:12345@cluster0.blj29c3.mongodb.net/?retryWrites=true&w=majority")
47 db = cluster["RUTINAS"]
48
```

Figura 4.63. URI de conexión

4.2.3.3.2 integración del módulo gráfico

La implementación inicial de la interfaz gráfica se realizó con el objetivo de proporcionar una representación visual de la aplicación, permitiendo al entrenador leer los datos deportivos de todos sus clientes. En una etapa posterior de la fase beta de la aplicación, se incorporó una interfaz web de usuario, facilitando así la carga de archivos pregrabados en la aplicación.

En la implementación del código, se integra la biblioteca Streamlit, la cual provee utilidades para la creación de elementos de interfaz de usuario. En el *main*, se emplean elementos de interfaz como `st.markdown()`, `st.sidebar()`, `st.selectbox()`, `st.button()` y `st.file_uploader()` para la creación de la interfaz gráfica. Un

ejemplo de implementación de estos elementos en el código se visualiza en las figuras 4.64, 4.65, 4.66 y 4.67.

```
185 st.markdown('<h1 class="main-title">MOVEMENTOR AI</h1>', unsafe_allow_html=True)
186 st.markdown('<h2 class="h2">Consulta de Datos del Usuario</h2>', unsafe_allow_html=True)
187
```

Figura 4.64. Agregación de encabezados

```
193 st.sidebar.write("Selecciona un usuario:")
194 nombre_usuario_seleccionado = st.sidebar.selectbox("Nombres de Usuario", nombres_usuarios)
195
```

Figura 4.65. Creación de barra lateral y selección

```
43 if option == 'Subir Archivo':
44     uploaded_file = st.file_uploader("Subir el archivo", type=['mp4', 'mov', 'avi'])
```

Figura 4.66. Carga de archivos

```
210 if st.button('Enviar'):
```

Figura 4.67. Creación de botón

En la sección dedicada a la generación de gráficos, se implementan funciones que desarrollan representaciones visuales basadas en los datos obtenidos, utilizando la librería Matplotlib. En conclusión, la interfaz gráfica ofrece una experiencia sencilla y accesible para procesar y analizar videos de entrenamiento de manera interactiva.

Una vez que la interfaz de la aplicación está configurada, se debe establecer un canal para permitir conexiones externas a la aplicación. En esta fase beta, para ilustrar un ejemplo de cómo sería la aplicación web, se ha empleado la herramienta Ngrok. Esta herramienta crea un túnel seguro desde una URL pública hacia un servidor local, posibilitando el acceso a la aplicación que se ejecuta en el localhost mediante el puerto 8501.

Para llevar a cabo este proceso, se debe comenzar por visitar el sitio web oficial de Ngrok [35] y registrarse para obtener una cuenta, como se muestra en la figura 4.68. Luego, se debe descargar el archivo ejecutable de Ngrok compatible con el sistema operativo y extraerlo en una ubicación conveniente del sistema.



Figura 4.68. Creación de cuenta en Ngrok

En la ubicación donde se ha extraído el programa Ngrok, se debe abrir una terminal o línea de comandos. Posteriormente, se realiza la autenticación de la cuenta de Ngrok mediante el comando se ilustra en la figura 4.69 y el correspondiente token de autenticación, el cual puede ser obtenido desde la cuenta de Ngrok. Este proceso es esencial para garantizar la seguridad y acceso autorizado al servicio.

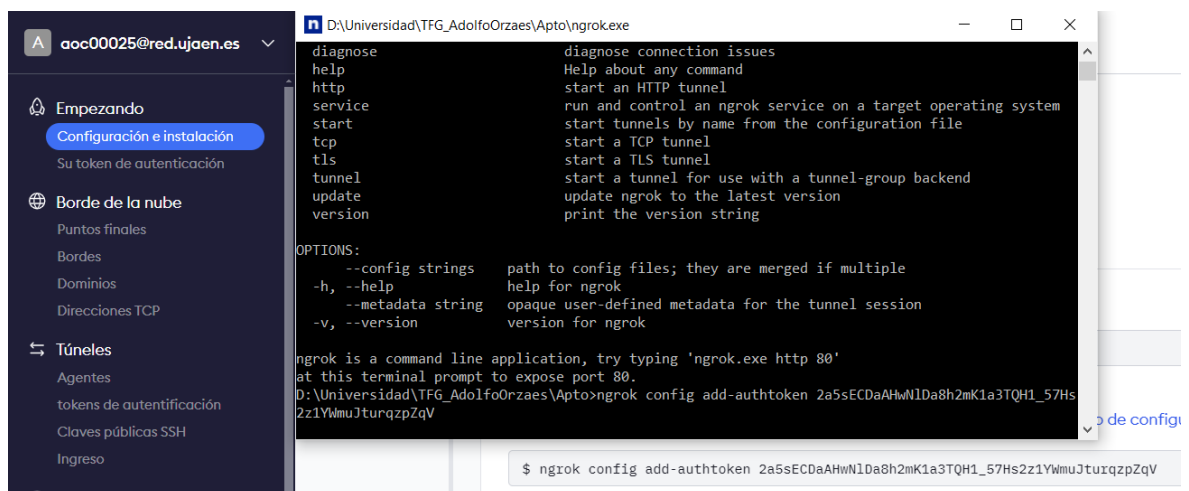


Figura 4.69. Comando y token de autenticación

Para establecer un túnel para un puerto local, como el utilizado por la aplicación Streamlit, se debe emplear el ultimo comando referenciado en la figura 4.70. En este caso la aplicación está utilizando el puerto 8501.

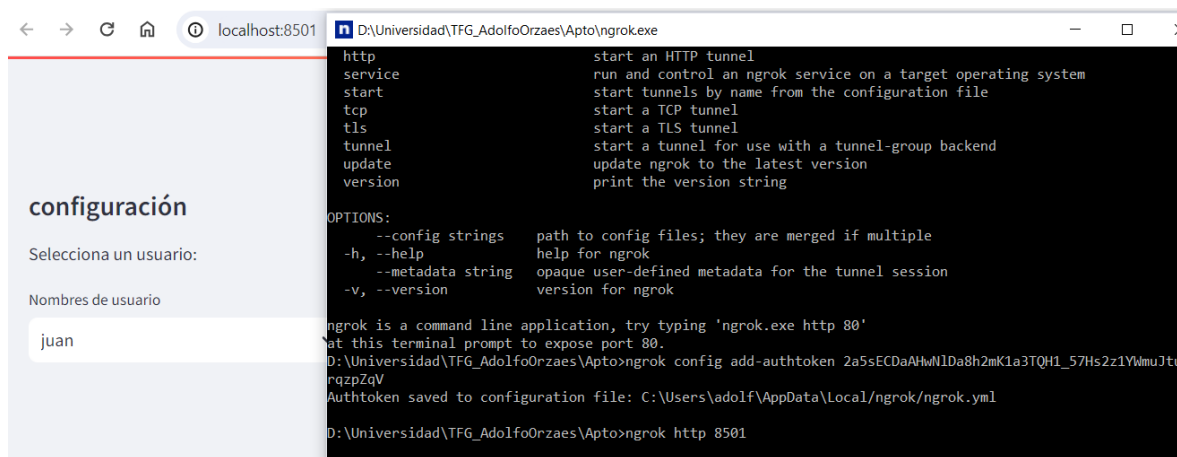


Figura 4.70. Creación del túnel

Tras la ejecución del comando, se generará una URL pública para la aplicación mediante Ngrok. De esta manera, la aplicación se encuentra accesible desde cualquier ubicación utilizando la URL proporcionada. En la siguiente imagen se presenta de forma clara la URL generada.

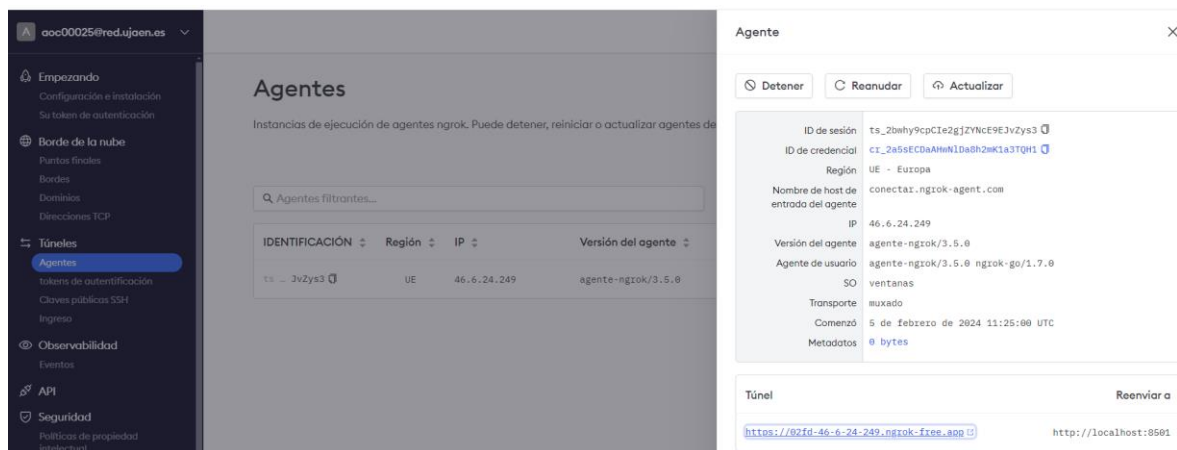


Figura 4.71. URL pública

Esta incorporación posibilita la conexión del entrenador desde otro dispositivo para acceder a los entrenamientos deportivos de sus clientes. Al mismo tiempo, ofrece a los usuarios la capacidad de cargar sus entrenamientos pregrabados de forma remota, brindando una visión del potencial que puede desarrollar la aplicación.

4.3 Metodología

En este apartado de metodología, se realiza una recopilación del hardware y software empleado en el desarrollo de la aplicación. Se proporciona una descripción detallada de los elementos técnicos utilizados, junto con el entorno de realización y los recursos utilizados en la implementación de la aplicación.

4.3.1 Herramientas hardware

4.3.1.1 Cámara

Para garantizar el óptimo desempeño del sistema, se requiere la integración de una cámara RGB que capture la representación bidimensional del usuario durante la ejecución de los ejercicios. Se requiere de una calidad suficiente para captar con nitidez la figura del individuo y sus articulaciones.

En el transcurso del proyecto, se ha hecho uso tanto de la cámara web incorporada en el ordenador como de una cámara externa Webcam Full HD Primux, adquirida del Laboratorio de Multimedia.

Es importante señalar que, tras un análisis comparativo, se ha observado que la cámara Full HD Primux ofrece resultados superiores en términos de calidad de imagen y precisión en la captura de detalles anatómicos. Este rendimiento mejorado la posiciona como la opción preferida para el seguimiento y análisis del usuario y sus articulaciones en tiempo real durante las sesiones de ejercicio.



Figura 4.72. Webcam Primux WC187 FullHD

Con el objetivo de realizar un análisis más completo, se ha implementado un enfoque dual. Además de la captura en tiempo real con las cámaras mencionadas, se ha llevado a cabo la grabación de sesiones de ejercicio utilizando la cámara integrada en dispositivos móviles. Posteriormente, se han analizado minuciosamente estas grabaciones, ampliando así la variedad de datos disponibles para una evaluación exhaustiva del rendimiento del usuario durante los ejercicios deportivos.

4.3.2 Herramientas software

4.3.2.1 Python

Python, concebido por Guido van Rossum a finales de los 80, ha evolucionado para convertirse en uno de los lenguajes de programación más populares y versátiles. [36]

Hoy, Python es esencial en campos como desarrollo web, inteligencia artificial y análisis de datos, demostrando su capacidad para adaptarse y crecer a lo largo del tiempo.

Destaca como un lenguaje de programación poderoso y de aprendizaje accesible. Ofrece estructuras de datos eficientes de alto nivel y un sistema de programación orientado a objetos simple pero efectivo. Su elegante sintaxis y tipado dinámico, combinados con su naturaleza interpretada, lo posicionan como un lenguaje óptimo para scripting y desarrollo ágil de aplicaciones en diversas áreas y plataformas.

Tanto el intérprete de Python como la amplia librería estándar están accesibles de manera gratuita en formato de código fuente y binario para la mayoría de las plataformas, disponibles en el sitio web oficial de Python [37]. Se permite la distribución libre de estos recursos. El mismo portal alberga distribuciones y enlaces a diversos módulos, programas, herramientas y documentación adicional de Python de terceros de forma gratuita.

4.3.2.2 Mediapipe

MediaPipe, una iniciativa de código abierto respaldada por Google que ofrece una colección de bibliotecas y herramientas para el seguimiento de objetos, el reconocimiento de gestos y la detección de rostros. [38]

Dentro de las capacidades de MediaPipe, destaca MediaPipe Pose, una solución de detección de puntos en el cuerpo. Esta herramienta utiliza un modelo de aprendizaje automático para identificar 33 puntos de referencia en el cuerpo de una persona. Estos puntos de referencia se emplean para estimar la postura del cuerpo, abarcando aspectos como la posición de las articulaciones, la orientación del cuerpo y las distancias entre diferentes partes de este.

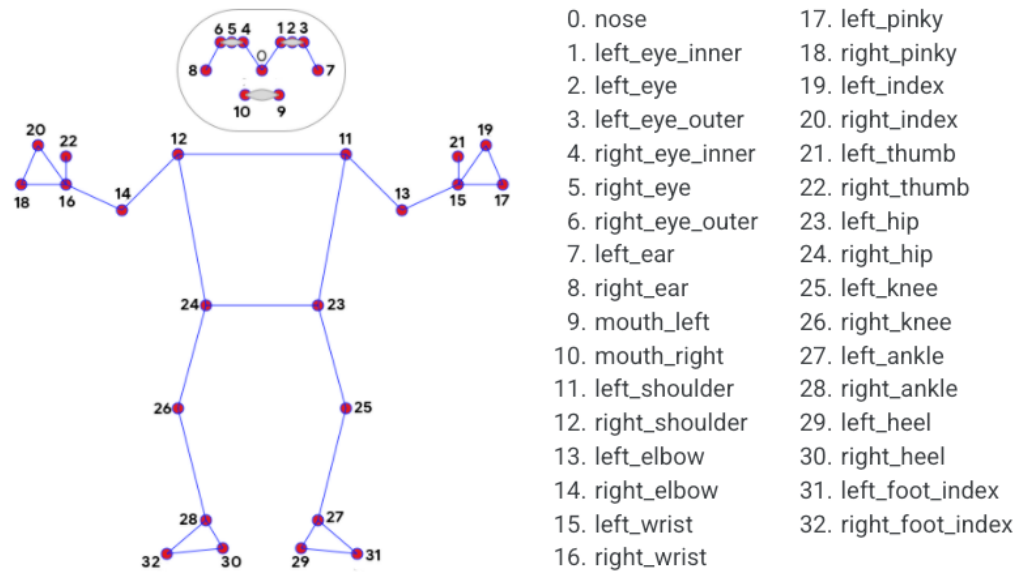


Figura 4.73. Selección de puntos clave en Mediapipe Pose

BlazePose se utiliza como el modelo de predicción de pose 3D dentro del marco de trabajo de Mediapipe según se describe en [39] . Este modelo emplea una combinación de mapa de calor y regresión en su enfoque. La pérdida asociada al mapa de calor se limita exclusivamente a la fase de entrenamiento, y su respectiva capa de salida se elimina durante la inferencia. La eliminación de los mapas de calor es crucial, ya que contribuye a que el modelo sea lo suficientemente liviano como para ejecutarse eficientemente en dispositivos móviles.

En el proceso de inferencia, se lleva a cabo la ejecución del detector hasta que se logra la localización de una persona. Una vez detectada, se activa el modelo de estimación de pose 3D. La salida de este modelo proporciona la ubicación de los *keypoints* del cuerpo, la confirmación de la detección o no de una persona, la generación de una nueva Región de Interés (ROI) derivada de la ROI previa y la predicción de la pose. En caso de que la red señale la ausencia de una persona, se procede a volver a ejecutar el detector como se indica en [40] .

En la arquitectura de la red, se emplean conexiones de salto (skip-connections) con el propósito de incorporar características tanto de alto como de bajo nivel, como se ilustra en la figura 4.74. Esto aporta una dimensión adicional a la red, permitiendo la integración de información detallada y general a lo largo de las capas, como se puede observar en la representación gráfica de la arquitectura.

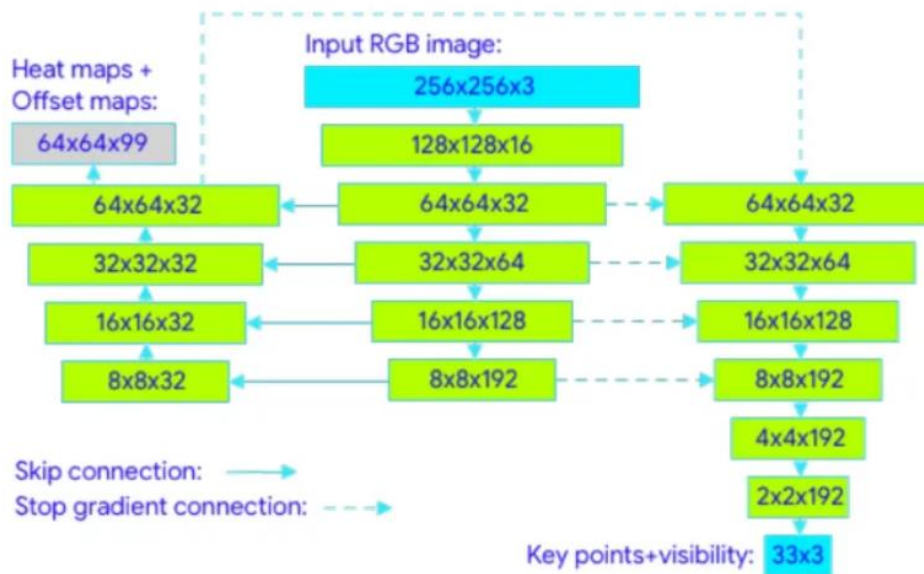


Figura 4.74. Arquitectura de red MediaPipe [41]

4.3.2.3 Open CV

OpenCV, originalmente desarrollada por Intel, es una biblioteca de software de aprendizaje automático y Visión Artificial de código abierto, está implementada en C/C++. Con más de 2500 algoritmos optimizados según se describe en [42], abarca desde la Visión Artificial convencional hasta los algoritmos más avanzados de aprendizaje automático. Su denominación proviene de Open Computer Vision (Visión Artificial Abierta).

Desde su versión alfa en enero de 1999, ha sido ampliamente utilizada en diversas aplicaciones y hasta el año 2020 se mantiene como la biblioteca más popular en el ámbito de la Visión Artificial. Entre sus aplicaciones se incluyen la detección de movimiento, reconocimiento de objetos y reconstrucción tridimensional a partir de imágenes [43].



Figura 4.75. Logo oficial de OpenCV

En este proyecto, OpenCV se empleó específicamente para realizar el preprocesamiento de las imágenes adquiridas antes de pasarlas a los modelos de MediaPipe. Estas imágenes se utilizarán para la estimación de la pose de la persona.

4.3.2.4 Numpy

NumPy fue inicialmente concebido por Jim Hugunin con contribuciones de diversos desarrolladores. En 2005, Travis Oliphant refinó y expandió NumPy, integrando características de la competencia Numarray en Numeric, lo que resultó en una biblioteca especializada en Python para el cálculo numérico y análisis de datos, eficaz especialmente para conjuntos de datos extensos [44].



Figura 4.76. Logo de NumPy

NumPy introduce la categoría de objetos *arrays* que posibilita la representación de colecciones de datos homogéneos en múltiples dimensiones, junto con funciones altamente eficientes para su manipulación. La principal ventaja frente a las listas predefinidas en Python es su capacidad para procesar *arrays* de manera significativamente más rápida, convirtiéndola en la elección ideal para el manejo de vectores y matrices de amplias dimensiones en el procesamiento de datos.

4.3.2.5 Matplotlib

Matplotlib es una librería de Python de código abierto, fue desarrollada inicialmente en 2002 por el neurobiólogo John Hunter con el objetivo de visualizar señales eléctricas cerebrales en personas epilépticas y replicar las capacidades de creación gráfica de MATLAB en Python. Se ha convertido en una herramienta ampliamente utilizada para crear gráficos y diagramas de alta calidad [45].



Figura 4.77. Logo de Matplotlib

Esta alternativa MATLAB permite la creación de diversos tipos de gráficas, como trazados, histogramas y diagramas de barra, con solo unas pocas líneas de código. Matplotlib se destaca por su versatilidad y detalle en la generación de visualizaciones de datos, siendo especialmente valiosa para aquellos que trabajan con Python o NumPy.

4.3.2.6 Streamlit

Streamlit, una biblioteca Python de código abierto, facilita una solución robusta para la creación de aplicaciones web enfocadas al ámbito del aprendizaje automático y la ciencia de datos [46].



Figura 4.78. Logo de Streamlit

Adicionalmente, proporciona una variedad de herramientas para las visualizaciones de datos. Ya sea que se necesite generar gráficos de dispersión, presentar histogramas o

diseñar mapas, Streamlit aborda diversas necesidades. Incluso se puede utilizar para desarrollar paneles dinámicos que posibiliten a los usuarios explorar datos en tiempo real.

4.3.2.7 Smtplib

Es un módulo esencial en Python, brindando una interfaz para la transmisión de correos electrónicos mediante el protocolo Simple Mail Transfer Protocol (SMTP).

Su función primordial radica en establecer conexiones con servidores de correo saliente, posibilitando el envío de mensajes con configuraciones detalladas, incluyendo asunto, cuerpo del mensaje y destinatarios. Este componente resulta fundamental en la implementación de funcionalidades de correo electrónico en aplicaciones y scripts desarrollados en Python, facilitando la comunicación eficiente y segura a través de mensajes electrónicos.

4.3.2.8 MongoDB (PyMongo)

MongoDB, un sistema de gestión de bases de datos NoSQL, ha surgido como una alternativa notable a MySQL en los últimos años de acuerdo con [47]. Su flexibilidad para trabajar con diversos lenguajes de programación, permitiendo la visualización de objetos, lo posiciona como una opción atractiva. En este contexto, PyMongo destaca como el paquete oficial de controladores que simplifica la interacción con MongoDB a través de Python, ofreciendo una integración eficaz para la gestión de datos en el proyecto científico.



Figura 4.79. Logo de Python y MongoDB

5 Resultados y discusión

En este capítulo se presentarán los resultados y la discusión del proyecto. Los resultados incluyen una interfaz gráfica diseñada para mejorar la experiencia del usuario durante la realización de ejercicios deportivos, así como el procesamiento en tiempo real de datos de una cámara para guiar el movimiento. Por otro lado, se detalla tanto la interfaz web de usuario, que permite cargar archivos de entrenamiento de forma remota, como la del entrenador, que posibilita el acceso externo y la comunicación con el deportista. Posteriormente, se realizará un análisis detallado de los resultados en función de los tres ejercicios implementados, considerando aspectos como el movimiento articular, la velocidad, la cadencia y parámetros estadísticos para evaluar el rendimiento deportivo.

Todas las visualizaciones de los ejercicios analizados son de elaboración propia y los videos correspondientes pueden encontrarse en el enlace proporcionado en la sección de anexos.

5.1 Interfaz gráfica del usuario

Con el objetivo de simplificar la experiencia de los usuarios, se ha diseñado una interfaz gráfica, empleando Tkinter [48], que resulta tanto sencilla como intuitiva, permitiendo a cualquier persona utilizarlo sin dificultades. En esta etapa, se lleva a cabo el procesamiento de datos y la presentación de elementos que facilitan la guía del ejercicio deportivo en tiempo real, haciendo uso de la información captada por la cámara adquirida en el laboratorio, tal como se muestra en la figura 5.1.



Figura 5.1. Webcam conectada al ordenador

Al iniciar el programa con la cámara conectada al ordenador, se desplegará una pantalla de salida que permitirá al usuario visualizarse en tiempo real mientras realiza el ejercicio. En esta visualización, se presenta el seguimiento del esqueleto del usuario, y se incluye una barra en la parte derecha que indica, en porcentaje, la profundidad del ejercicio, mientras en la esquina superior derecha se muestran las repeticiones realizadas junto con la cadencia del movimiento. La cadencia refleja el ritmo de repeticiones que el usuario es capaz de llevar a cabo por minuto, basándose en su frecuencia de movimiento. Los diferentes grados de flexión de la articulación se presentan al lado de esta, siendo una medida útil para prevenir el estrés articular. En la figura 5.2 se observa el desarrollo de la aplicación durante la realización del ejercicio.

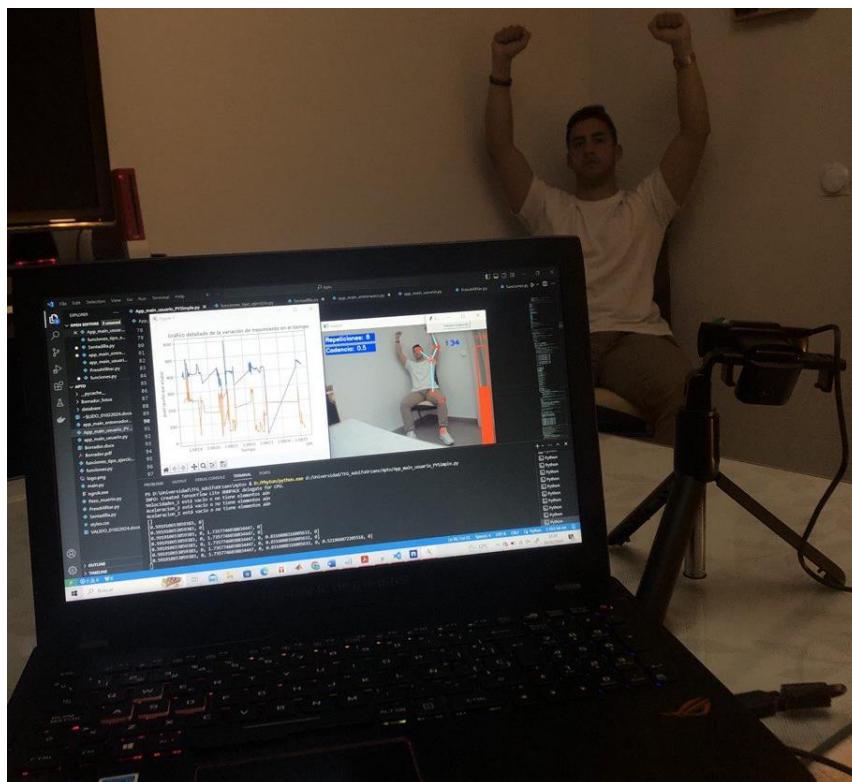


Figura 5.2. Funcionamiento de la aplicación de usuario

Durante la fase de procesamiento, se lleva a cabo la extracción de datos fundamentales, tales como velocidad, aceleración, cadencia, duración del ejercicio, grados de flexión y coordenadas ocupadas en píxel por la articulación en la imagen. Cuando se detiene la grabación, se realizan cálculos estadísticos basados en las medidas obtenidas, incluyendo desviaciones estándar de la cadencia y velocidad, medias, así como valores máximos y mínimos.

Estos datos se estructuran en un diccionario de manera ordenada y luego se almacenan en un repositorio, MongoDB en este caso, como se explicó detalladamente en el capítulo de desarrollo, específicamente en el proceso de envío de datos al repositorio. Esta elección se basa en la necesidad de gestionar eficientemente la información adquirida, aprovechando la flexibilidad y escalabilidad que MongoDB proporciona para almacenar y recuperar datos.

Se ha implementado tres tipos de ejercicios para esta fase experimental: Press militar, Sentadilla y Peso Muerto.

5.1.1 Press Militar

Es un ejercicio fundamental, forma parte de la categoría de ejercicios básicos. El Press Militar es un ejercicio de entrenamiento focalizado en fortalecer los músculos deltoides, tríceps y trapecios. Consiste en levantar una barra desde los hombros hasta una

posición elevada sobre la cabeza, con variantes como el uso de mancuernas o la posición tras nuca. Estas opciones proporcionan flexibilidad para adaptar el ejercicio a diversos objetivos de entrenamiento.

La propia aplicación se utiliza para presentar una visualización del ejercicio, como se muestra en la figura 5.3, donde se observa una ejecución correcta del movimiento articular en tiempo real. Este seguimiento se mantiene constante y sin grandes variaciones en amplitud, como se detallará más adelante en el análisis. Además, se muestran al usuario medidas en el entorno real que lo motivan a completar el ejercicio, proporcionándole una experiencia inmersiva.

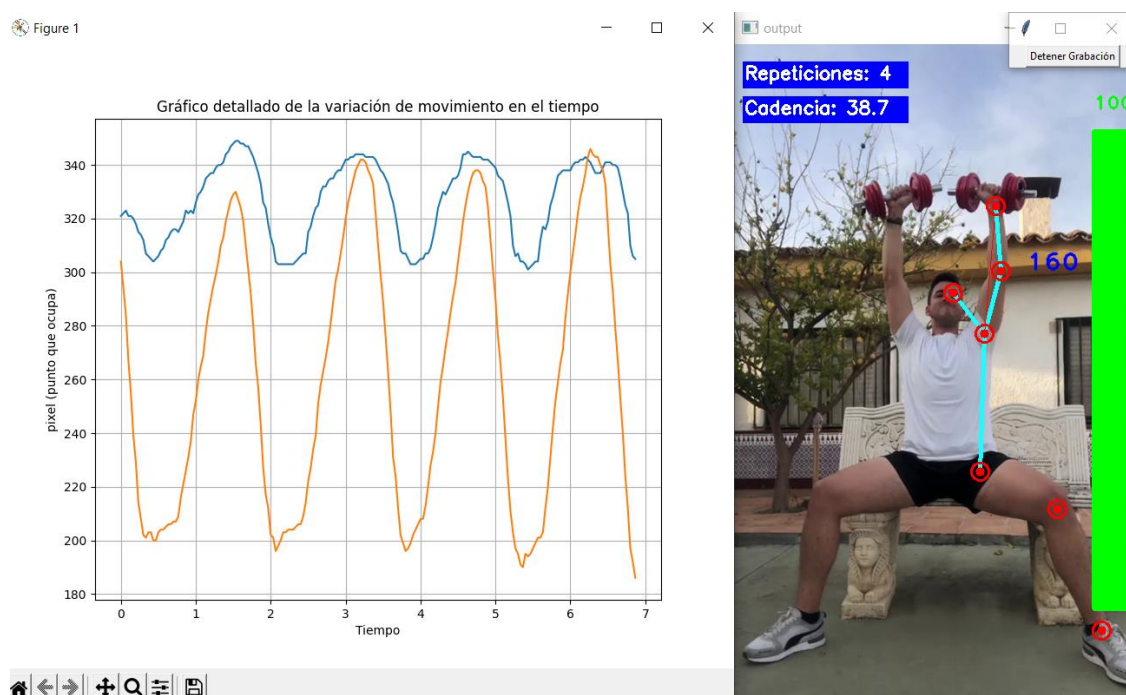


Figura 5.3. Press Militar

5.1.2 Sentadillas

La Sentadilla es un ejercicio destinado a fortalecer los músculos, tendones y ligamentos de las piernas, proporcionando beneficios adicionales como la tonificación de los glúteos y mejorando la salud de la cadera. Su ejecución implica flexionar las rodillas y descender el cuerpo manteniendo la verticalidad, seguido de un retorno a la posición erguida. Normalmente, se realiza sosteniendo algún peso con los brazos o el cuello para intensificar el ejercicio.

La aplicación misma se emplea para presentar la realización del ejercicio, conforme se ilustra en la figura 5.4, donde se observa una ejecución precisa del movimiento articular en tiempo real. Este seguimiento se mantiene estable y sin grandes variaciones en amplitud, como se detallará minuciosamente más adelante en el análisis.

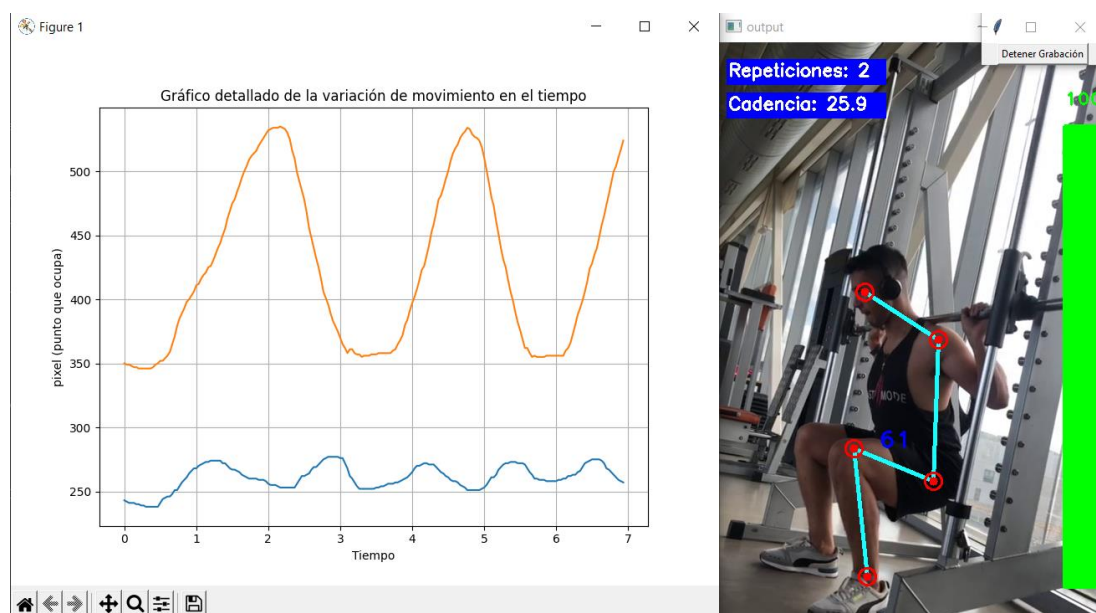


Figura 5.4. Sentadilla

5.1.3 Peso Muerto

El Peso Muerto constituye un ejercicio esencial para fortalecer diversos grupos musculares, centrándose en la espalda baja, glúteos, isquiotibiales y la zona lumbar. Durante su realización, se eleva una barra desde el suelo hasta la posición vertical, con variantes como el uso de mancuernas. Manteniendo la espalda en posición recta y los hombros hacia atrás.

Al igual que en el resto de los ejercicios, en este caso también se utiliza la propia aplicación para mostrar el desarrollo del ejercicio, como se ilustra en la figura 5.5. Esta registra una ejecución correcta del ejercicio, monitorizando el movimiento articular y proporcionando parámetros que ayudan al deportista a guiarse durante el entrenamiento.



Figura 5.5. Peso Muerto

5.2 Interfaz web

Durante la fase inicial del desarrollo, se implementó la interfaz web de la aplicación con el título "Movementor IA" con el propósito de proporcionar una visión preliminar de la fase beta al entrenador deportivo, permitiéndole revisar a todos sus usuarios y sus respectivos entrenamientos. Más recientemente, se añadió una interfaz web adicional para los usuarios finales, permitiéndoles cargar archivos pregrabados de forma remota, los cuales son procesados eficientemente.

5.2.1 Interfaz web de usuario

Esta parte se añade como extra para que el usuario pueda enviar de forma remota sus videos de entrenamiento pregrabados y extraer los datos, con el fin de subirlos al repositorio de entrenamientos para su visualización por parte del entrenador.

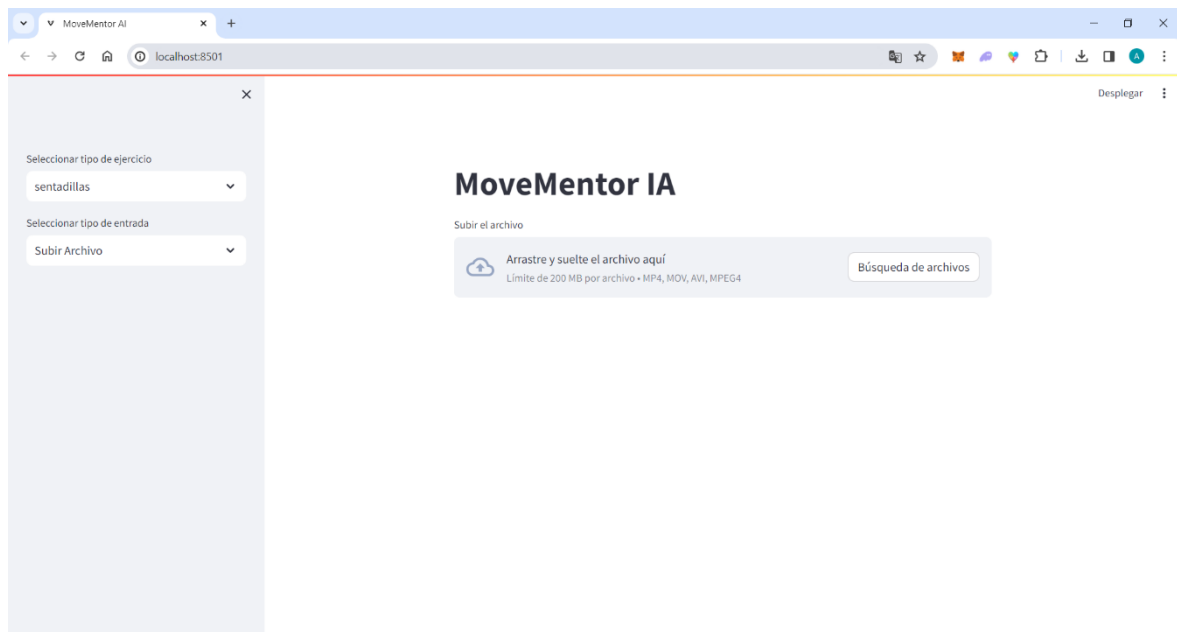


Figura 5.6. Interfaz web de usuario

Al iniciar la aplicación, se debe seleccionar el tipo de ejercicio a realizar. En esta fase beta, se ofrece la opción entre tres movimientos básicos para el entrenamiento de fuerza con pesas, como se ha mencionado anteriormente.

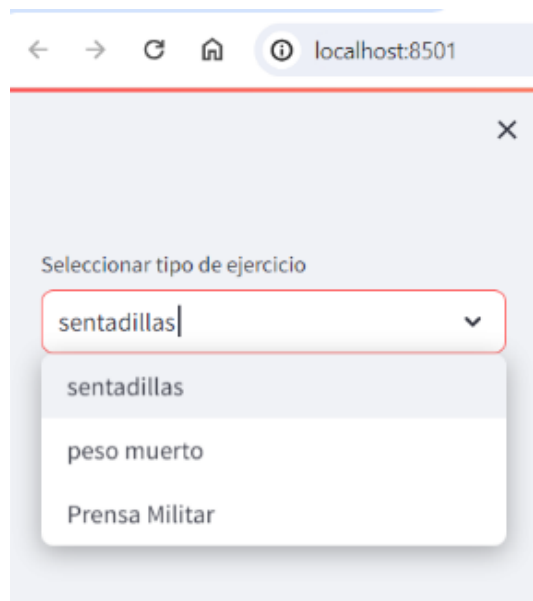


Figura 5.7. Movimientos básicos

Luego, se elige subir el archivo como tipo de entrada desde el dispositivo donde esté almacenado.

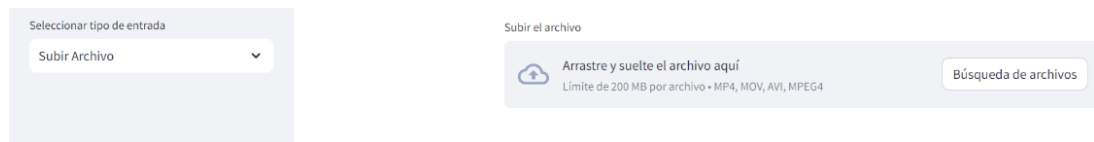


Figura 5.8. Subir Archivo

A continuación, se desplegará una ventana adicional en el lateral derecho que facilitará la búsqueda del archivo mediante la opción “*Browse file*”. Al hacer clic, se habilitará la selección del archivo correspondiente desde el dispositivo del usuario. Los videos se encuentran en una carpeta local denominada “*database*”.

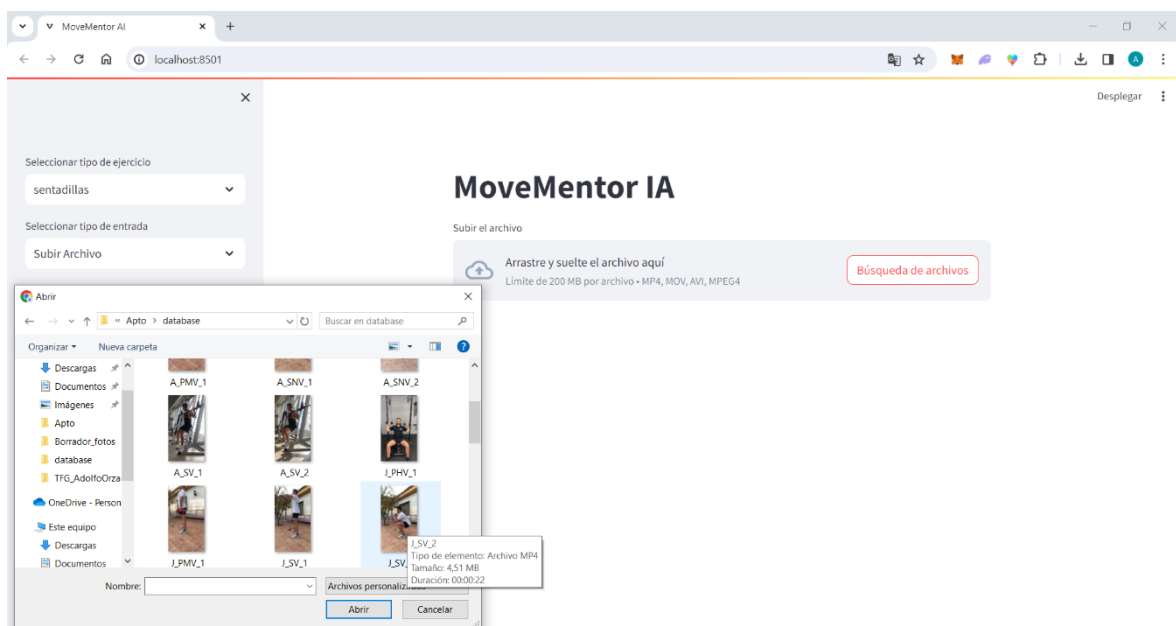


Figura 5.9. Selección del archivo

Finalmente, una vez terminado aparece un mensaje de confirmación de que el video ha sido procesado correctamente. Toda esta información se estructura en forma de diccionario y posteriormente se almacenará en el repositorio de entrenamientos realizados.

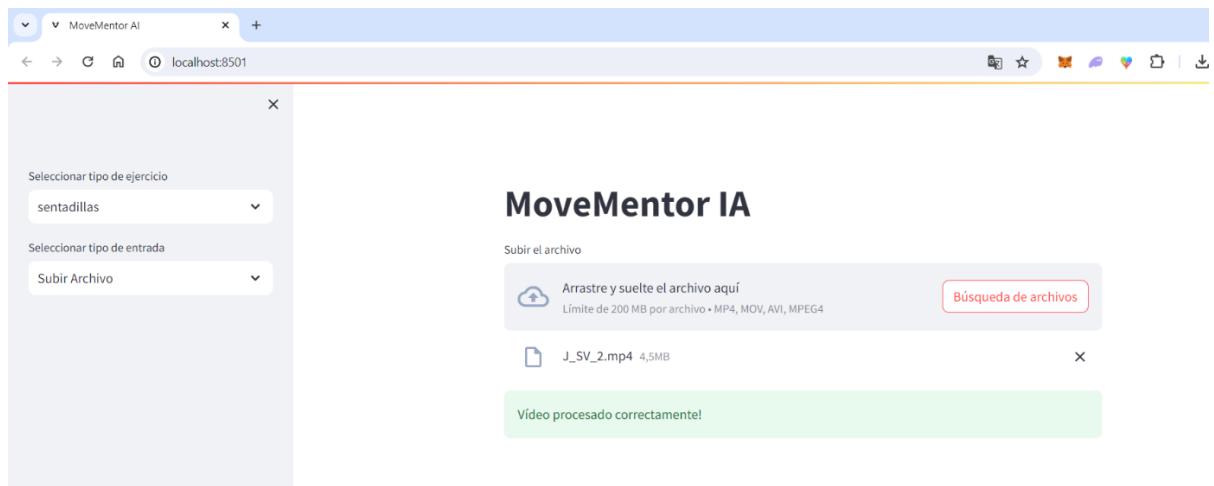


Figura 5.10. Mensaje de confirmación

A continuación, se describe la funcionalidad de la aplicación, que permite al usuario cargar sus entrenamientos de forma remota mediante un dispositivo externo, como una Tablet, tal y como se ilustra en este ejemplo de la figura 5.11.

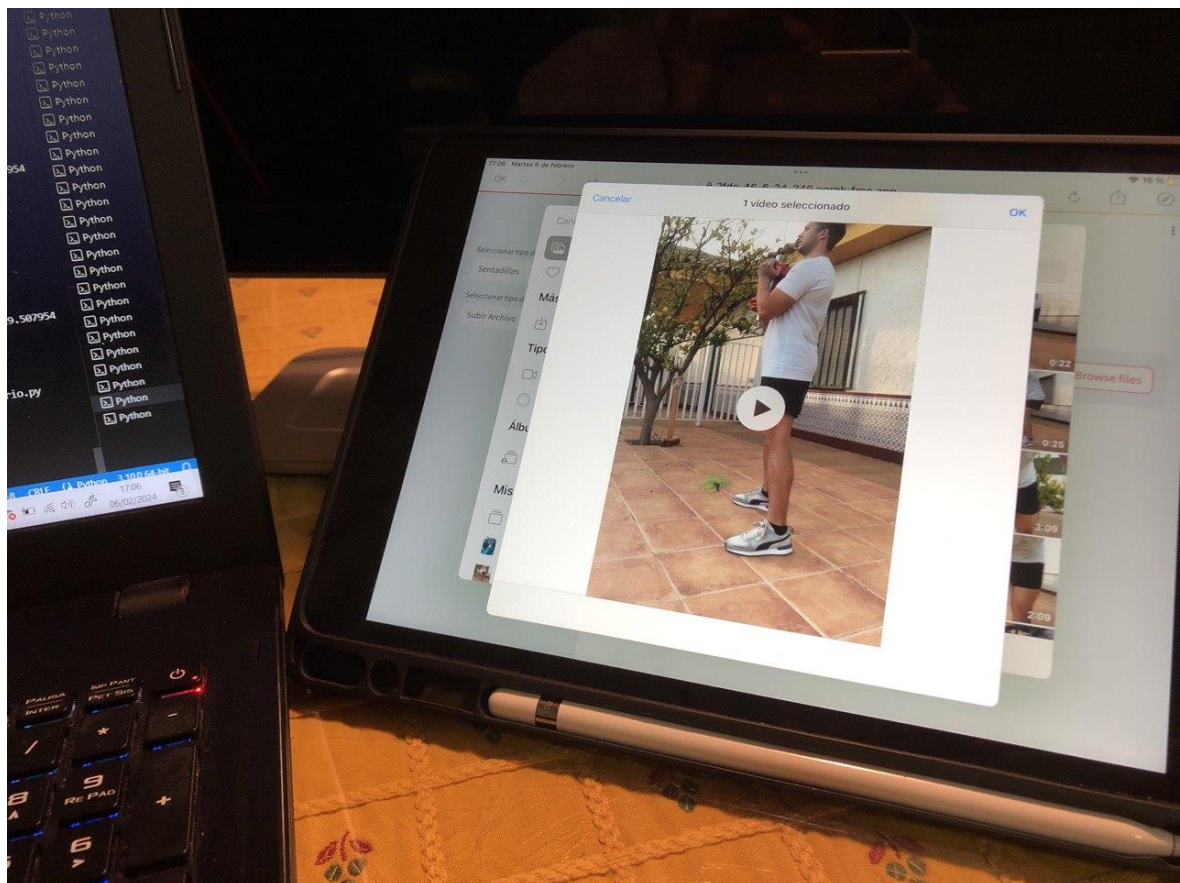


Figura 5.11. Carga remota de entrenamientos

Posteriormente, se confirma que la extracción de los datos necesarios para caracterizar el tipo de ejercicio ha sido exitosa, como se ilustra en la imagen 5.12.

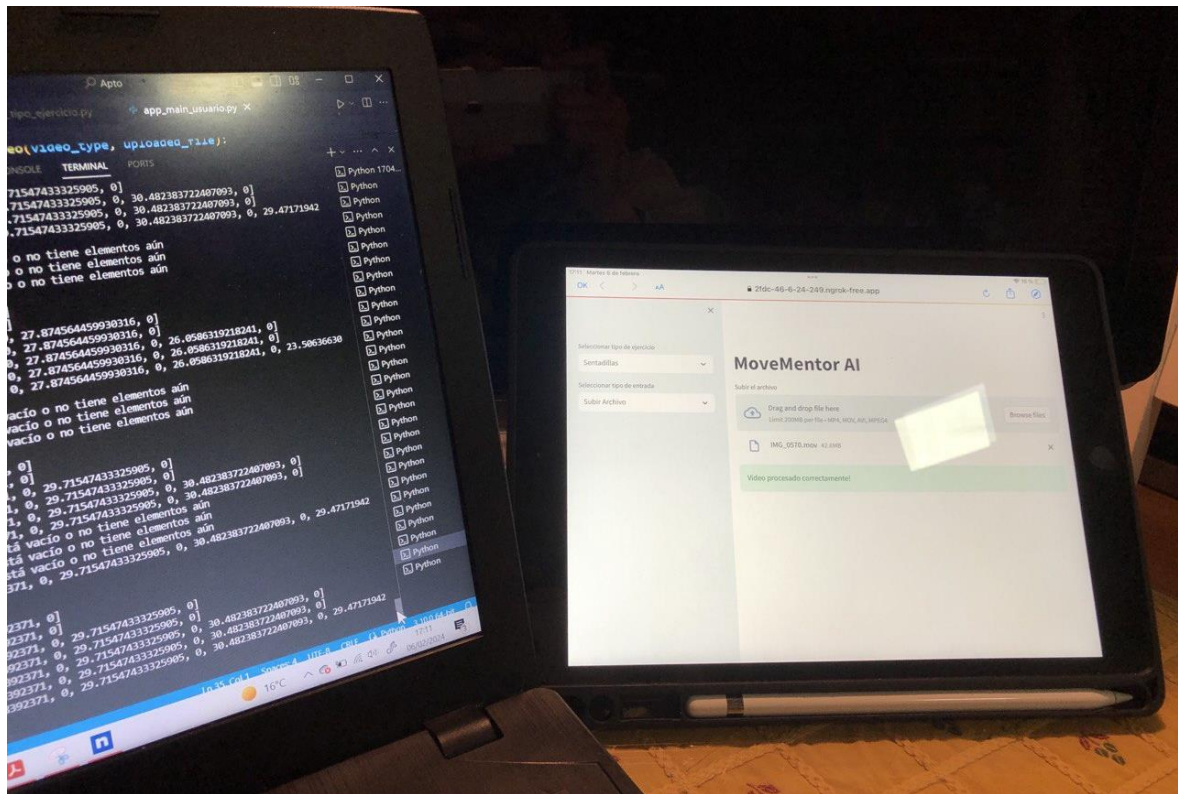


Figura 5.12. Carga exitosa

5.2.2 Interfaz web de entrenador

Se presenta la interfaz del entrenador como una herramienta completa para el seguimiento, análisis y comunicación, con el propósito de optimizar el proceso de entrenamiento deportivo y mejorar el rendimiento de los deportistas. Esto simplifica la labor, eliminando la necesidad de revisar cada video del usuario de manera individual. Además, reduce la posibilidad de errores en la visualización que podrían surgir por fatiga acumulada al revisar numerosos videos, lo cual podría afectar la percepción del entrenador respecto a posibles riesgos de lesiones.

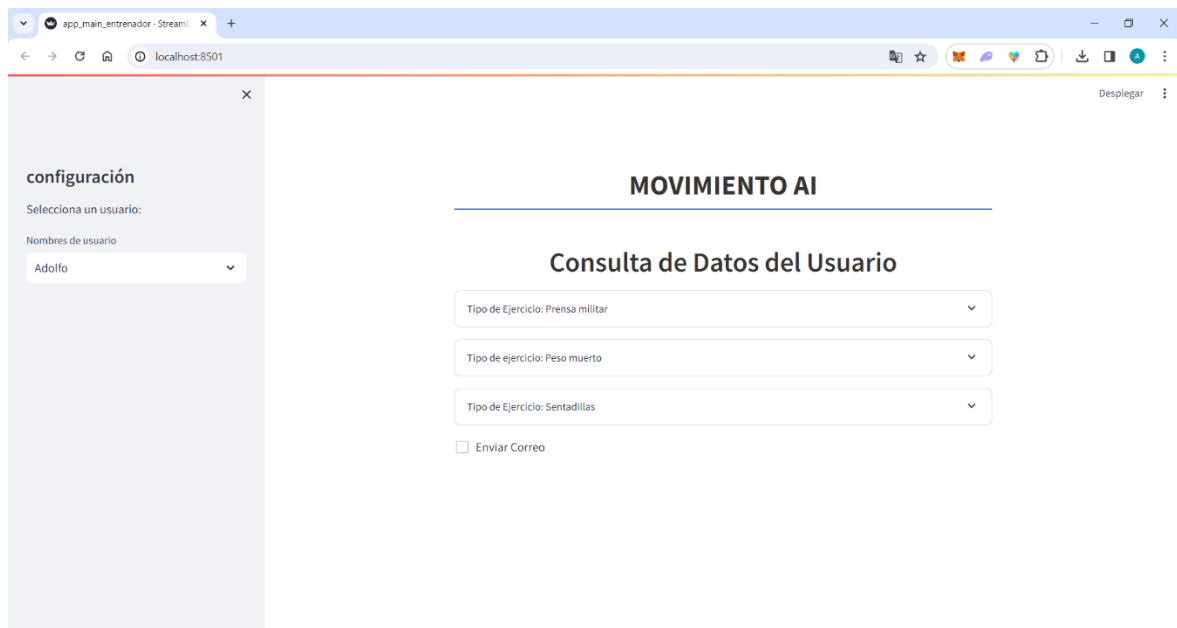


Figura 5.13. Interfaz web de entrenador

En el inicio de la configuración, se despliega inicialmente el menú para la elección de un usuario. Una vez seleccionado, se habilita la posibilidad de que el entrenador examine los distintos tipos de ejercicios realizados. Al optar por un ejercicio específico, se muestran las fechas registradas junto con sus respectivos horarios.



Figura 5.14. Selección de usuario y ejercicio

Posteriormente, al hacer clic en una fecha específica se accede a una visualización gráfica de la velocidad en m/s , acompañada de un gráfico que representa el grado de flexión de la articulación a lo largo del tiempo. También se exhibe un gráfico de barras que detalla la cadencia cada dos series en repeticiones por minuto, junto con una tabla que

contiene datos adicionales, incluidos parámetros estadísticos y el recuento total de repeticiones efectuadas. Esta funcionalidad permite monitorizar al deportista de manera eficiente, estructurando los parámetros de manera ordenada y visual, simplificando el proceso de interpretación por parte del instructor, como se representa en la figura 5.15.

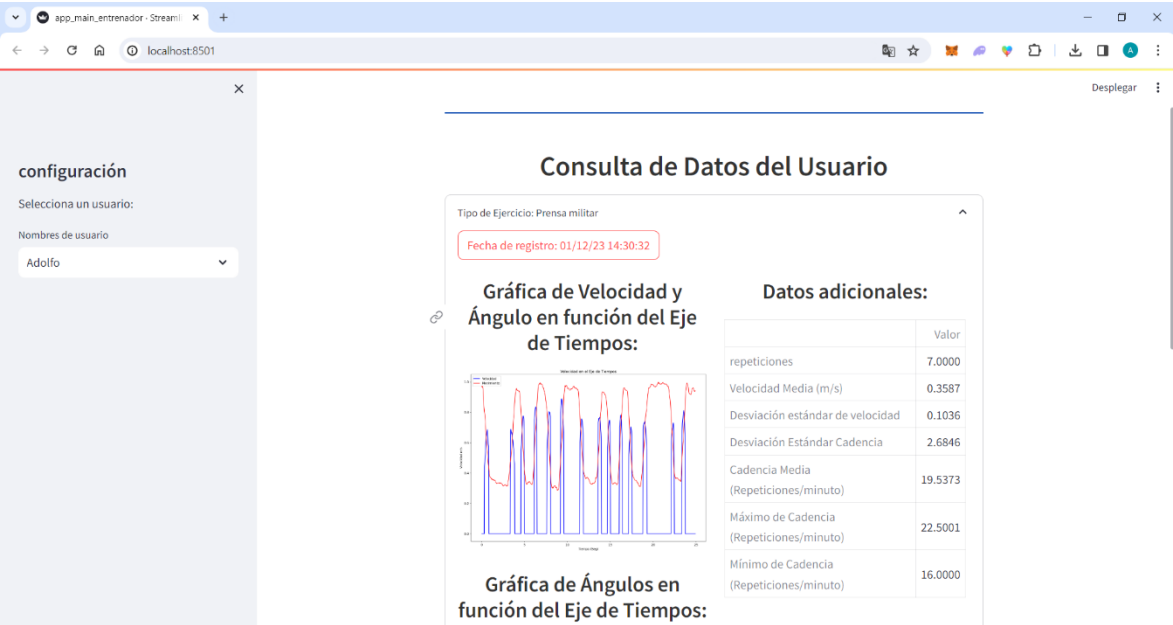


Figura 5.15. Visualización de datos del usuario

Después de revisar y analizar los datos, se ofrece la posibilidad de emitir correcciones al atleta mediante correo electrónico. En la aplicación, se encuentra una casilla destinada al envío. En caso de determinar que no son necesarias correcciones, el proceso simplemente se finaliza. En cambio, si se decide enviar rectificaciones, se solicita al entrenador ingresar la dirección de correo del deportista y completar el asunto del mensaje. Tras rellenar esta información, se puede activar el botón "Enviar". Por último, si la acción se realiza de manera exitosa, se muestra un mensaje de confirmación indicando que el correo se ha enviado con éxito, como se ilustra en la figura 5.16.

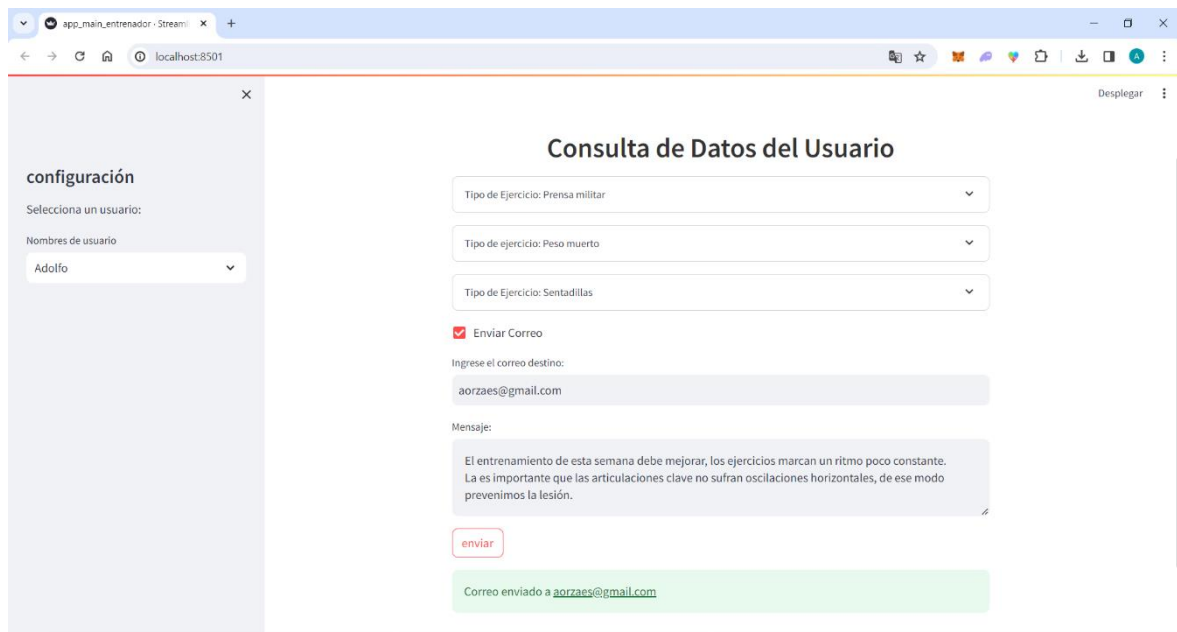


Figura 5.16. Confirmación del correo enviado

A continuación, se presentan los resultados de la comunicación remota proporcionada por la primera fase beta de la aplicación, que permite al entrenador acceder a su repositorio de clientes y a cada uno de los ejercicios que han realizado de manera externa, utilizando dispositivos como, en el caso de la figura 5.17, una *Tablet*.

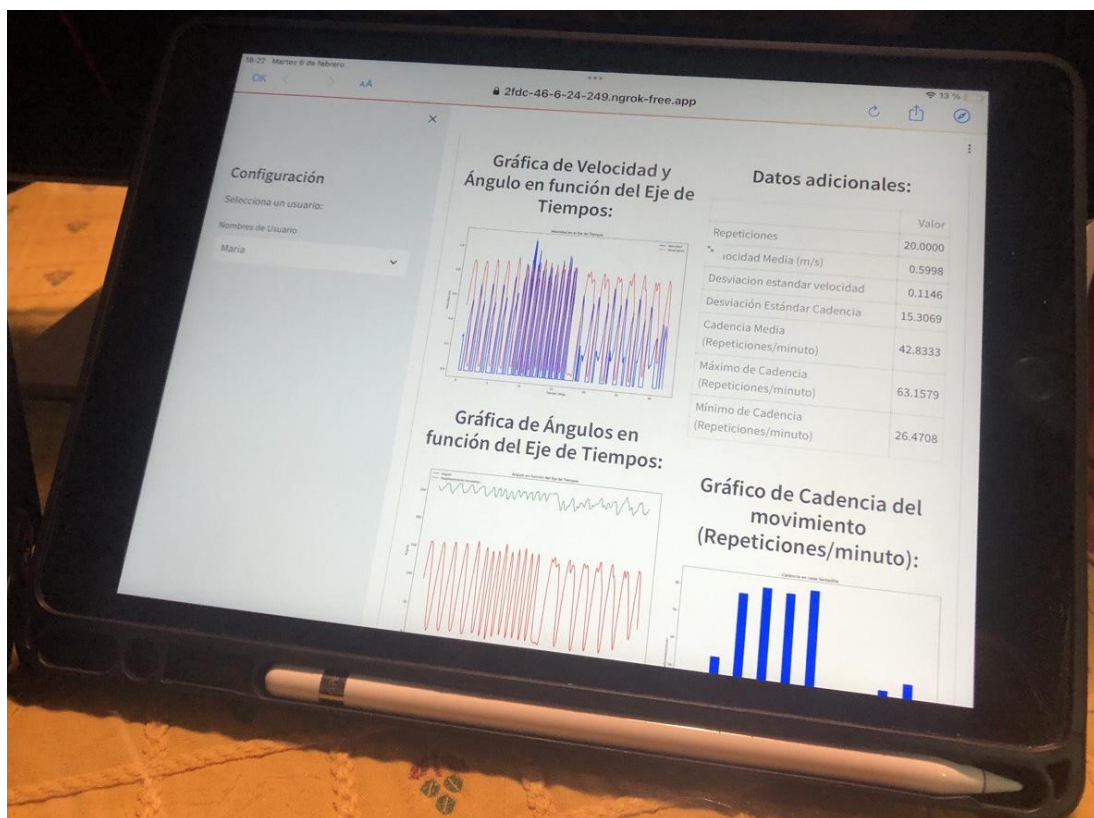


Figura 5.17. Acceso remoto a datos de entrenamiento

Por último, para verificar la calidad de la comunicación, se envía un mensaje al cliente a través de la opción de correo electrónico integrada en la aplicación, proporcionándole indicaciones sobre el desarrollo de su entrenamiento. Como se puede observar en la figura 5.18, el mensaje llega al correo electrónico de destino, que en este caso corresponde al configurado en el reloj inteligente.

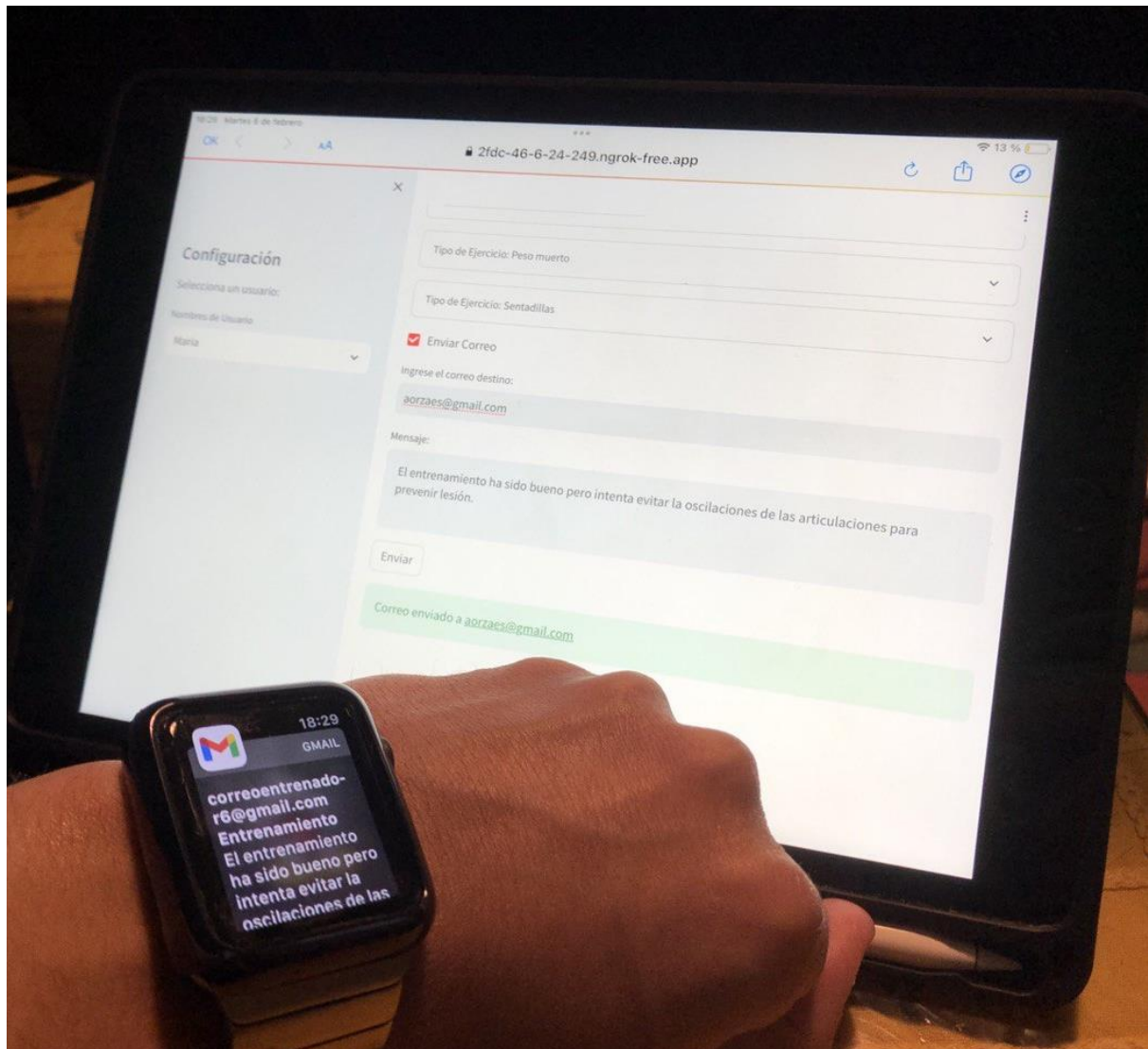


Figura 5.18. Verificación del canal de comunicación

5.3 Análisis de resultados

En esta sección, se procede a analizar los resultados obtenidos mediante el procesamiento de datos. Se inicia explicando las gráficas generadas a partir de los datos procesados, detallando su contenido y exponiendo el procedimiento de tratamiento de señal empleado con el fin de mejorar la calidad de los datos y obtener mediciones precisas que reflejen fielmente la realidad.

En el proceso inicial, se identifica el punto articular clave del cual se extraen los datos del movimiento. La elección de este punto varía dependiendo del tipo de ejercicio realizado. Por ejemplo, durante la ejecución de la Sentadilla, se analiza el punto situado en la cadera; en el Peso Muerto, se localiza en el hombro; y en el Press Militar, se selecciona la muñeca. Además, para cada ejercicio se determina un punto de flexión articular que proporciona un valor angular en grados. Se puede visualizar de manera clara en la figura 5.19.

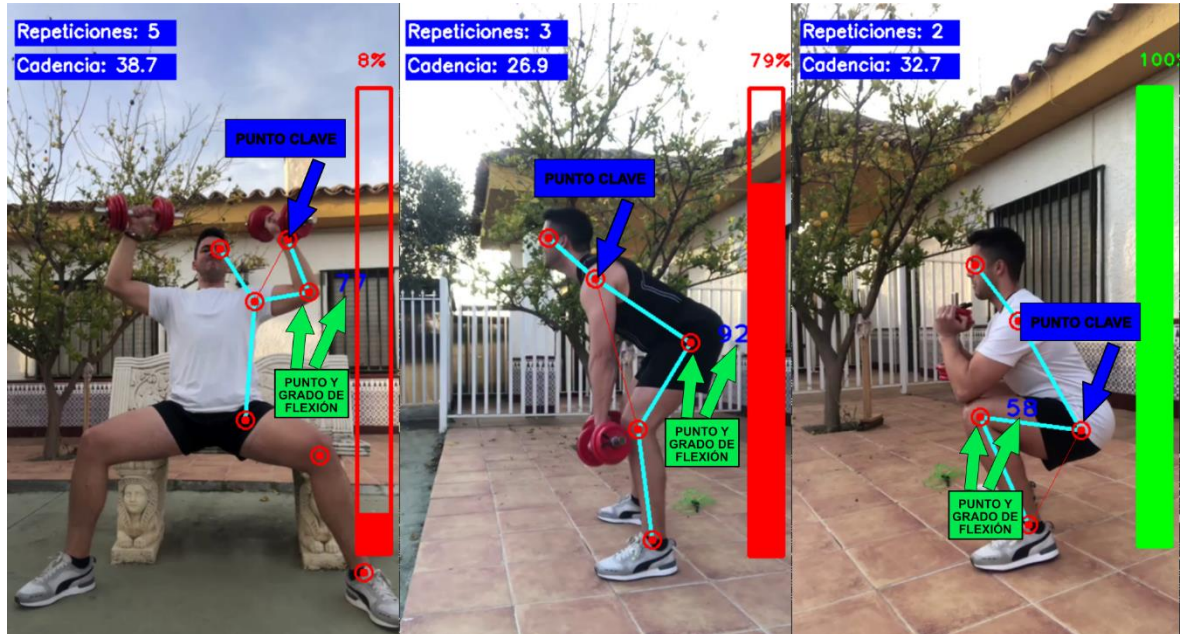


Figura 5.19. Análisis de puntos de referencia

5.3.1 Análisis del movimiento articular en tiempo real

En función del punto clave, se registra el movimiento reflejado en píxeles a lo largo del tiempo. Esta medida se proporciona al usuario en tiempo real, permitiendo la observación del movimiento del punto articular, el cual está definido por dos coordenadas: X e Y. La gráfica naranja muestra el rango de movimiento del punto en el eje vertical a lo largo del tiempo, mientras que la gráfica azul representa el rango de movimiento del punto en el eje horizontal. Esta representación se observa en la figura 5.20.

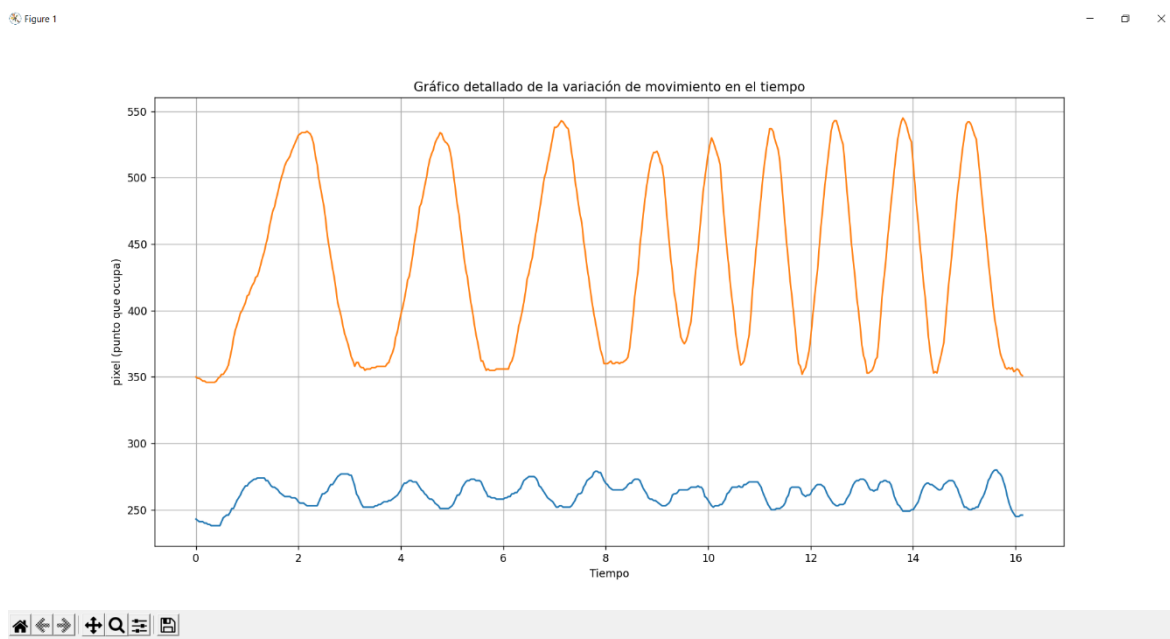


Figura 5.20. Análisis del movimiento articular en tiempo real

Esta gráfica es una herramienta útil para ofrecer una comprensión tanto al usuario como al entrenador sobre las oscilaciones que puede experimentar la articulación durante la ejecución del ejercicio, lo que permite prevenir lesiones. Por ejemplo, en el ejercicio es crucial que el movimiento sea fluido y continuo, y esto puede ser observado a través de la representación gráfica del punto en el eje Y. En el ejemplo mostrado en la figura 5.21, se aprecia una mayor interrupción en las tres primeras repeticiones, indicadas con un círculo a continuación.

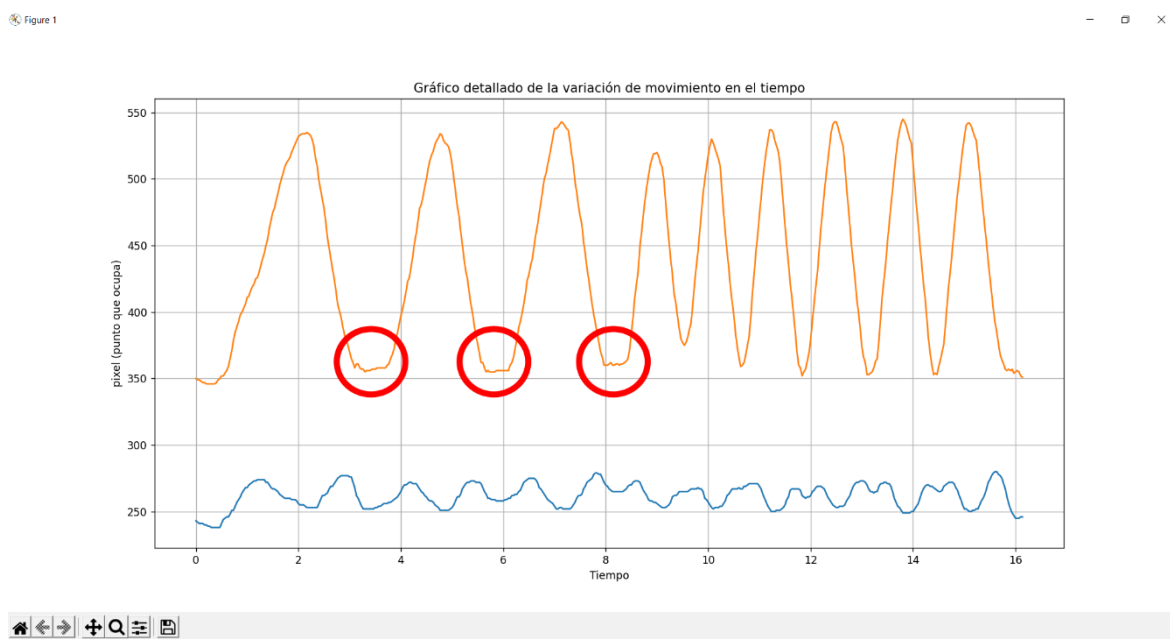


Figura 5.21. Gráfica extraída de A_SV_1 de database

La representación gráfica en azul, que muestra la variación del punto en el eje horizontal a lo largo del tiempo, resulta extremadamente útil para la prevención de lesiones. A continuación, se muestra la figura 5.22, donde se realiza el ejercicio de manera incorrecta para destacar esta variación notable en la amplitud en comparación con la figura anterior. Este aumento en la amplitud del punto en el eje horizontal puede aumentar el riesgo de lesiones en el hombro al abrir demasiado los brazos durante el movimiento con pesas en alto, como indica la flecha azul en la imagen siguiente.

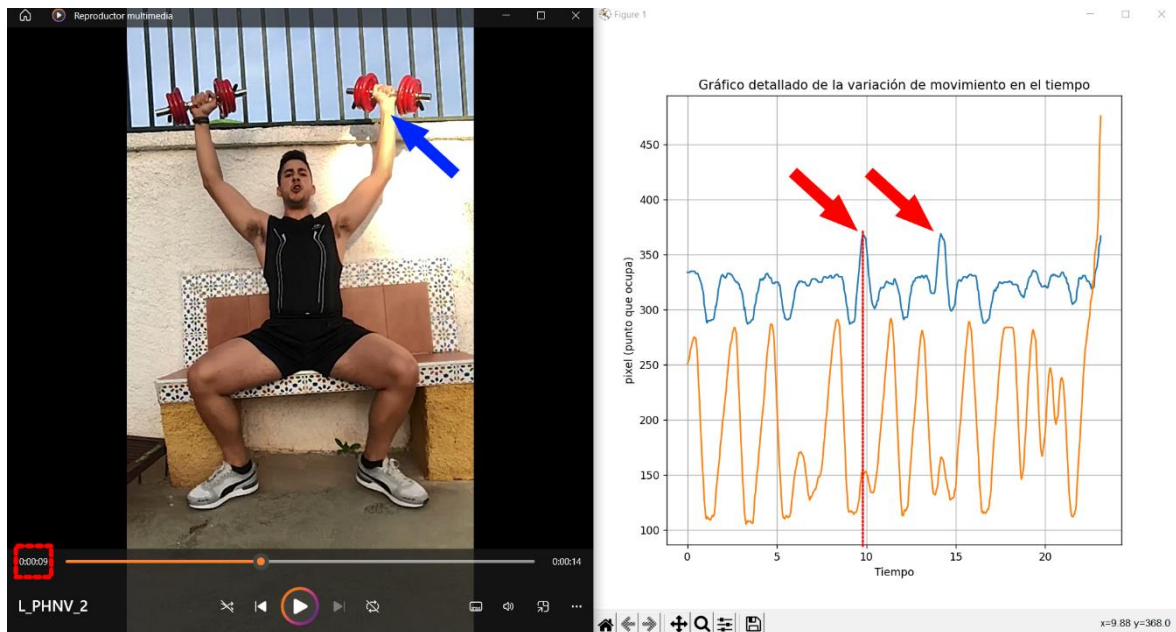


Figura 5.22. Variación de la amplitud en el eje horizontal durante el ejercicio

Sin embargo, un ejercicio bien ejecutado debe mantener la gráfica del desplazamiento del punto en el eje horizontal lo más constante posible a lo largo del tiempo, sin picos de amplitud. Esto se ilustra en la imagen 5.23 siguiente.



Figura 5.23. Trazado óptimo del desplazamiento articular

5.3.2 Análisis de velocidad de movimiento

Se calcula la velocidad de movimiento asociada a este punto clave, lo que proporciona una medida del cambio de posición con el tiempo. En las primeras pruebas, se observaron gráficas de velocidad con un nivel considerable de ruido, lo cual se debe a las pequeñas oscilaciones que experimentaba el punto en la imagen durante la realización del ejercicio. Se muestra la señal en la figura 5.24.

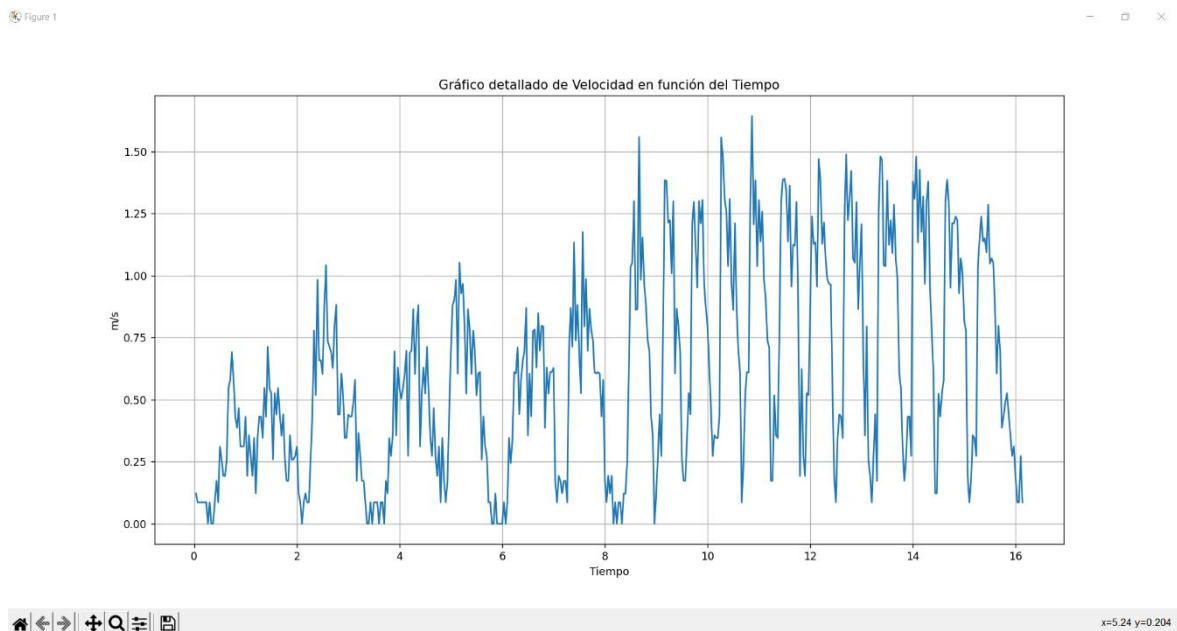


Figura 5.24. Señal velocidad extraída de A_SV_1 de database

Aplicando los conocimientos adquiridos durante el periodo de formación en los estudios de Grado en Ingeniería de Tecnologías de Telecomunicación, se consideró la opción de filtrar la señal utilizando un filtro de media móvil, tanto con una ventana rectangular como con una de Hamming. A pesar de aplicar ambos filtros, el objetivo era suavizar la señal. Los resultados no variaron considerablemente.

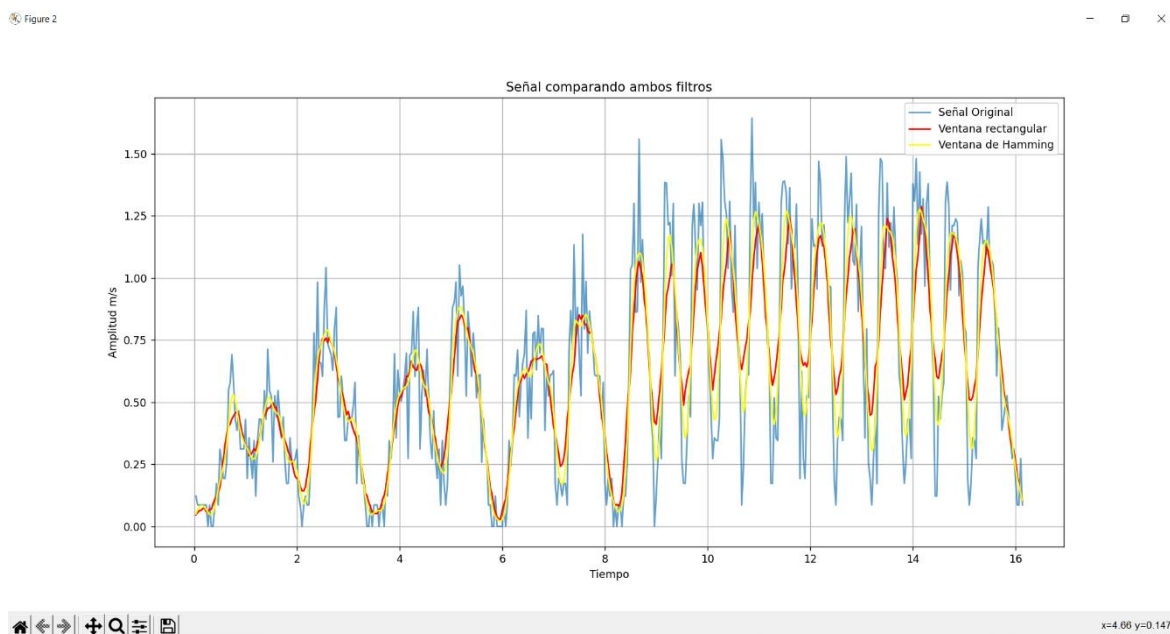


Figura 5.25. Comparación de filtros

Se optó por utilizar el filtro de media móvil con ventana rectangular, considerándolo más adecuado para el propósito debido a su simplicidad y validez en la aplicación.

Por otro lado, como se señaló en el apartado de procesamiento (mencionado anteriormente), se determina el ángulo de flexión de la articulación, mediante el teorema del coseno para representar el grado en función del tiempo durante el ejercicio. No obstante, es crucial no confundir este ángulo del punto de flexión con el punto clave de donde se extrae la información esencial, lo cual se evidencia claramente en la figura 5.19.

Este parámetro resulta muy útil para restringir el movimiento dentro de un rango de valores considerado óptimo para la ejecución del ejercicio, eliminando todos los datos fuera de este intervalo. Se estableció un umbral que filtra todo movimiento que no se encuentre en el intervalo de 81° a 150° durante la realización del ejercicio. Además, este ángulo, que indica el grado de flexión de la articulación durante el ejercicio, se presenta de manera superpuesta en la gráfica de la figura 5.26, donde se aplica un filtro a la señal de velocidad y se elimina cualquier componente fuera del rango predefinido. Esta representación permite obtener una orientación clara sobre la dirección del movimiento y proporciona información valiosa sobre la actividad realizada.

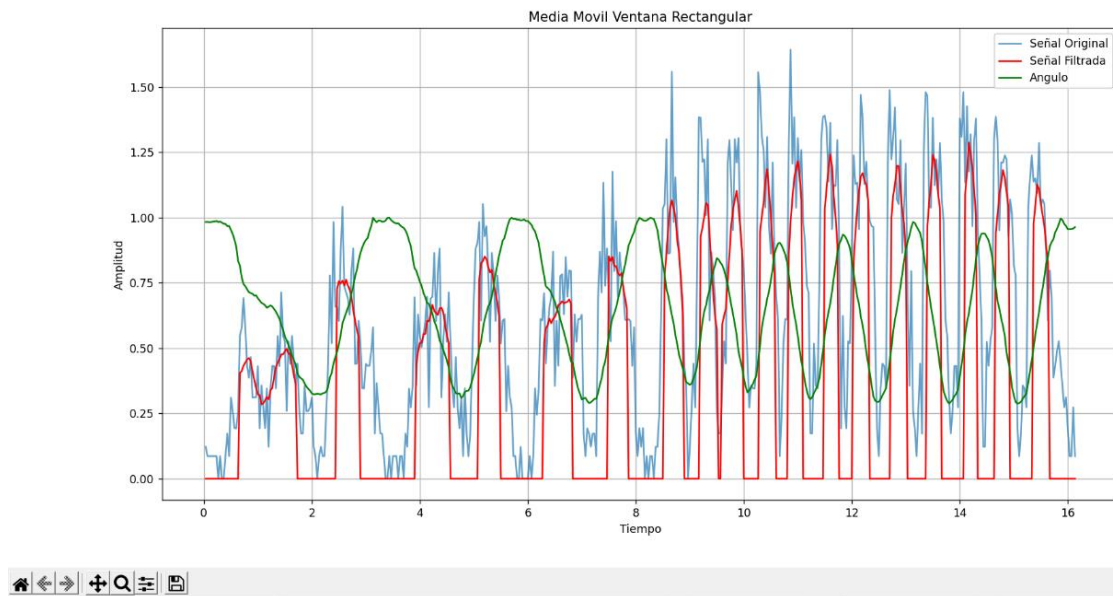


Figura 5.26. Señal velocidad filtrada en rango óptimo de movimiento

En la imagen anterior se muestra la señal filtrada correspondiente al ejercicio de Sentadillas, etiquetado como (A_SV_1) en la base de datos local de videos denominada "database", la cual está disponible en el enlace adjunto en los anexos. Se observa que, durante las tres primeras repeticiones, la velocidad registrada es menor que en las últimas, donde se incrementa. Al hacer un acercamiento en la imagen, esta puede ser analizada con mayor detalle. Por ejemplo, se puede realizar un *zoom* entre los segundos 6 y 8 para cubrir un periodo completo de la señal del ángulo, representada en verde en la figura 5.27.

En la figura se presenta un período completo de la señal que indica una repetición total. La señal "Ángulo" (representada en verde) refleja la dirección del movimiento, comenzando con un descenso de la gráfica que indica el inicio en la bajada del ejercicio (A_SV_1) analizado. Durante el descenso, donde el ángulo se encuentra dentro del rango de trabajo, se analiza la velocidad (representada en rojo), observando un aumento de velocidad de aproximadamente 0.5 m/s a 0.7 m/s , seguido de una disminución en la velocidad durante la fase de ascenso, pasando de 0.8 m/s a 0.6 m/s .

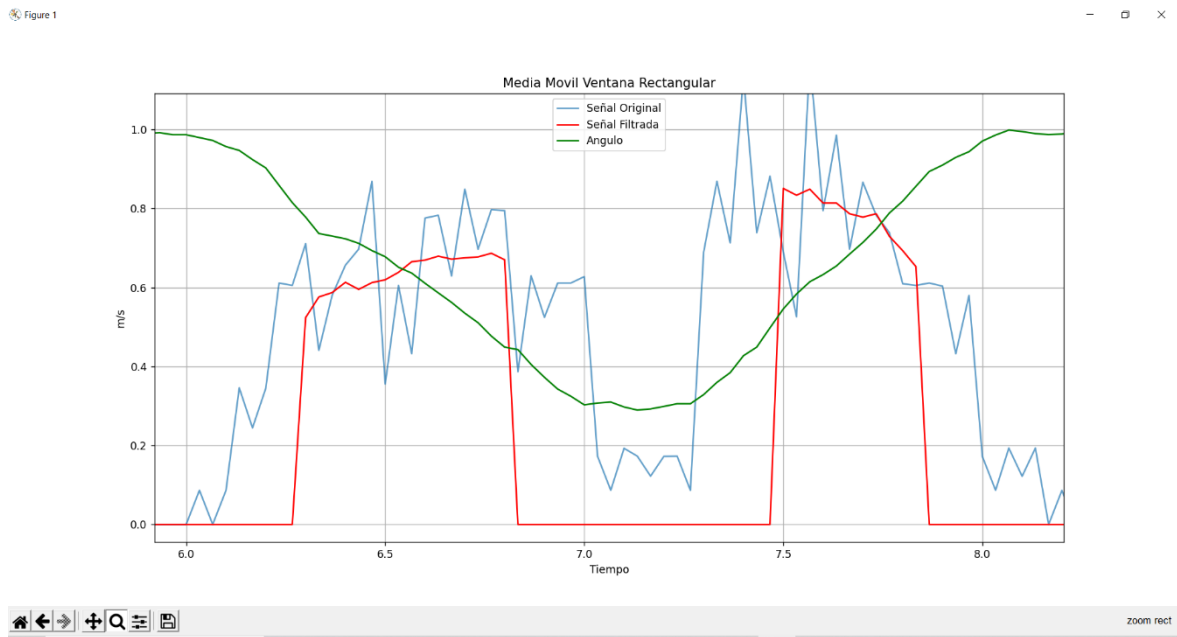


Figura 5.27. Zoom Señal velocidad filtrada de A_SV_1

Estos resultados son coherentes con la realidad, ya que es común que el movimiento descendente, especialmente en ejercicios con carga, sea más rápido que el movimiento ascendente debido al peso involucrado en la ejecución del ejercicio. Este análisis nos proporciona información útil para controlar el descenso en ejercicios con pesas, como las Sentadillas, ya que una caída brusca, reflejada en un aumento abrupto en la gráfica de velocidad, podría resultar en lesiones graves en la región lumbar del individuo.

Por lo tanto, sería recomendable que las gráficas de los ejercicios siguieran un estilo similar al mostrado en la figura anterior, ya que proporciona una representación óptima y coherente del movimiento a lo largo de las repeticiones. A continuación, se procede al análisis de una gráfica que ilustra un ejercicio mal ejecutado, evidenciando irregularidades en la lectura.

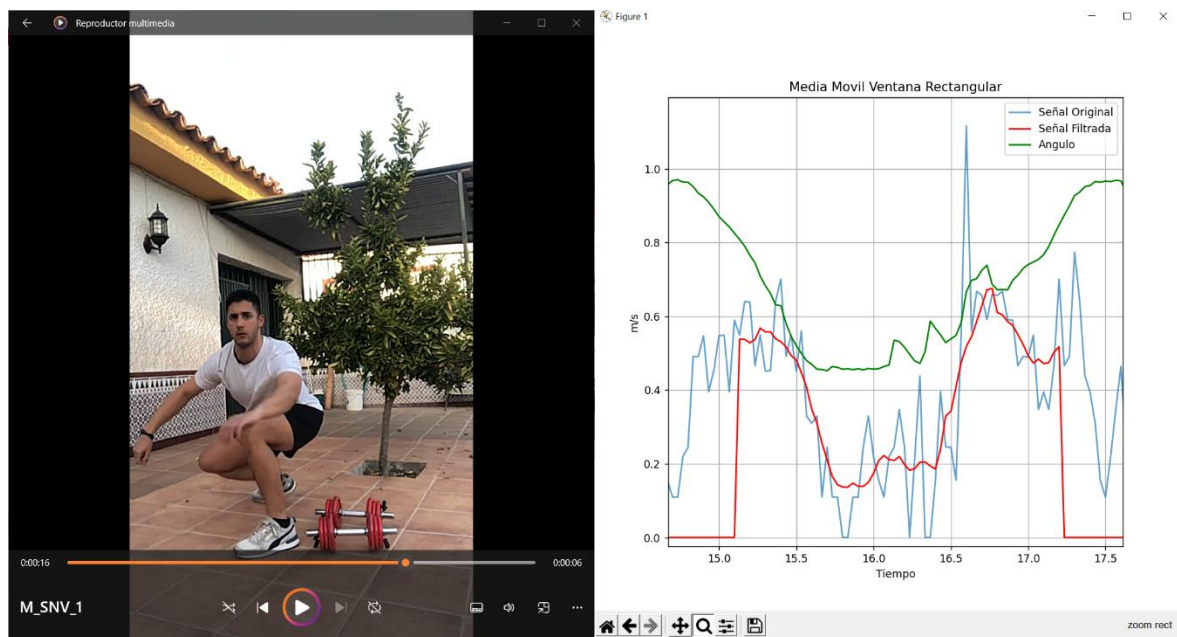


Figura 5.28. Irregularidad en el ejercicio *M_SNV_1*

En la figura 5.28, se observa en el intervalo de tiempo comprendido entre los 14.5 y los 17.5 segundos una irregularidad. La señal de ángulo indica que el movimiento no sobrepasa el rango de 81° , lo que provoca una irregularidad en la velocidad, representada por la señal roja. Durante este ciclo de repetición, se mantiene una velocidad del orden de 0.2 m/s , lo que sugiere un estancamiento en el movimiento.

Esta situación puede deberse a una pérdida de equilibrio, como en este caso particular donde el sujeto de la imagen experimenta desestabilización al flexionar. Por ello, la velocidad permanece muy baja durante este periodo hasta que el sujeto recupera el equilibrio, momento en el cual la señal de velocidad vuelve a aumentar.

5.3.3 Análisis de cadencia de movimiento

A continuación, se lleva a cabo la evaluación de la cadencia en repeticiones por minuto que el sujeto es capaz de realizar, en relación con la frecuencia del movimiento. Para este propósito, se presenta un diagrama de barras que muestra la cadencia cada dos repeticiones. Por lo tanto, si el ejercicio realizado comprende 8 repeticiones, el total de barras será 4, indicando la cadencia en repeticiones por minuto para cada par de repeticiones. Esta representación proporciona una idea bastante precisa del ritmo que el usuario está manteniendo durante la realización del ejercicio.

En la figura 5.29 se observan nueve repeticiones, sin embargo, en el diagrama de barras solo se muestran cuatro conteos. Esto se debe a que la cadencia se registra cada dos últimas repeticiones, por lo tanto, si el número de repeticiones es impar, aún se necesita agregar una repetición más para mostrar en la cadencia una barra adicional. En este caso,

al tener 9 repeticiones, se muestran cuatro barras. Cada barra en el diagrama representa el ritmo en repeticiones por minuto que el deportista es capaz de realizar en ese instante. Al observar la imagen, se puede notar que el último registro coincide con la última cadencia mostrada al usuario.

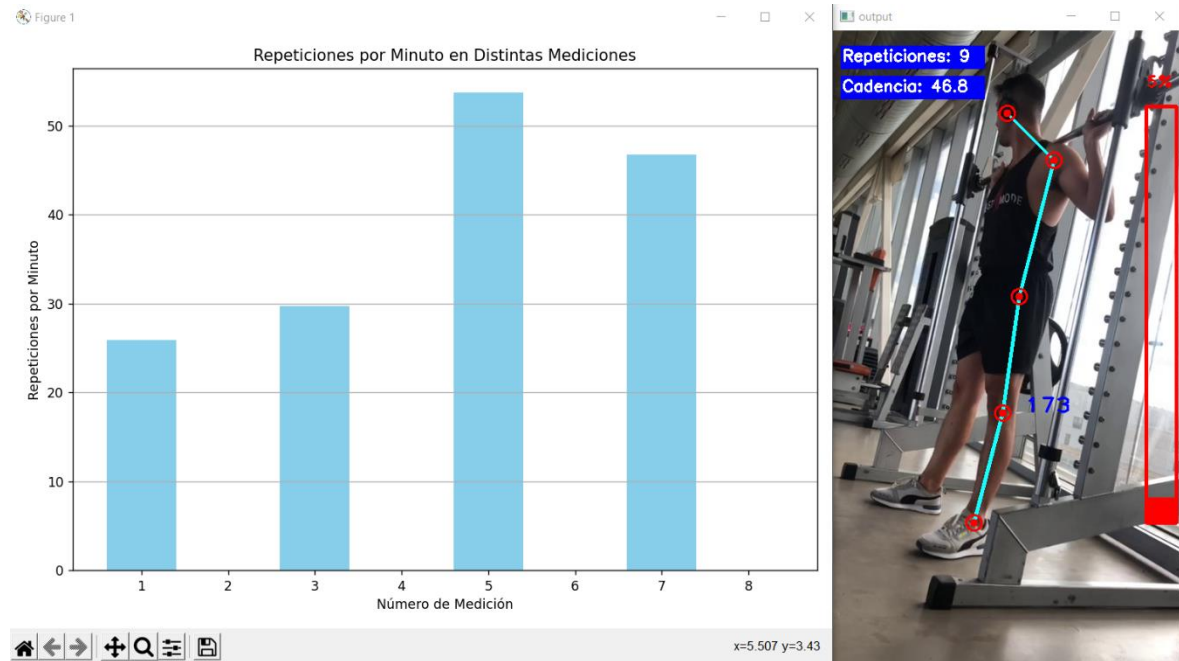


Figura 5.29. Cadencia extraída de A_SV_1

Esta medida es crucial para permitir al entrenador analizar el ritmo del movimiento y determinar si ha habido una progresión gradual o cambios abruptos en la cadencia de las repeticiones.

5.3.4 Análisis de los parámetros estadísticos

Finalmente, se presenta una tabla de parámetros estadísticos calculados en función de las medidas anteriormente explicadas. Estos parámetros estadísticos permiten seguir y comparar el rendimiento a lo largo del tiempo. Al observar tendencias en la velocidad y la cadencia, los atletas y entrenadores pueden evaluar el progreso, identificar mejoras y áreas que requieran trabajo adicional.

Se incluyen parámetros estadísticos como la media de la velocidad y la cadencia, desviaciones estándar, máximos y mínimos. Estos son especialmente útiles para la interpretación del rendimiento deportivo. A continuación, se muestran en la figura 5.30.

Datos adicionales:

	Valor
repeticiones	9.0000
Velocidad Media (m/s)	0.6093
Desviación estándar de velocidad	0.1035
Desviación Estándar Cadencia	11.5574
Cadencia Media (Repeticiones/minuto)	39.0340
Máximo de Cadencia (Repeticiones/minuto)	53.7321
Mínimo de Cadencia (Repeticiones/minuto)	25.8995

Figura 5.30. Parámetros estadísticos

La media es útil para entender el nivel de esfuerzo promedio durante el ejercicio en términos de velocidad y cadencia de las repeticiones. Puede utilizarse como referencia para comparar el rendimiento en diferentes sesiones de entrenamiento o entre diferentes individuos. Si la media aumenta o disminuye con el tiempo, podría indicar cambios en el rendimiento.

Los máximos y los mínimos valores proporcionan información sobre el rango en el que varían los datos de velocidad y cadencia. Puede utilizarse el máximo para identificar el mejor rendimiento alcanzado y el mínimo para identificar el peor rendimiento.

Finalmente, se examina la desviación estándar, la cual indica la dispersión de los datos alrededor de la media. Este parámetro se calcula tanto para la cadencia como para la velocidad. Sin embargo, al calcular la desviación estándar de la velocidad, surgen dificultades debido a que, en momentos específicos durante la realización del ejercicio, la velocidad tiende a cero cuando el cuerpo está en reposo. Por lo tanto, es de interés calcular la desviación estándar de la velocidad únicamente cuando nos encontramos dentro del rango de movimiento, lo que nos proporciona una medida más precisa del rendimiento deportivo. Como se explicó anteriormente, este rango de movimiento está determinado por el ángulo de flexión de la articulación.

Para abordar esta tarea, se ha implementado una función denominada *calcular_desviaciones()*, la cual se encarga de identificar y segmentar los datos en intervalos continuos, como se indica en la figura 5.31. Posteriormente, se calcula la desviación estándar de cada uno de estos intervalos no nulos. Una vez que se dispone del array con las desviaciones estándar de cada segmento de señal, es necesario sumar los valores y dividir entre la longitud para obtener la desviación estándar media.

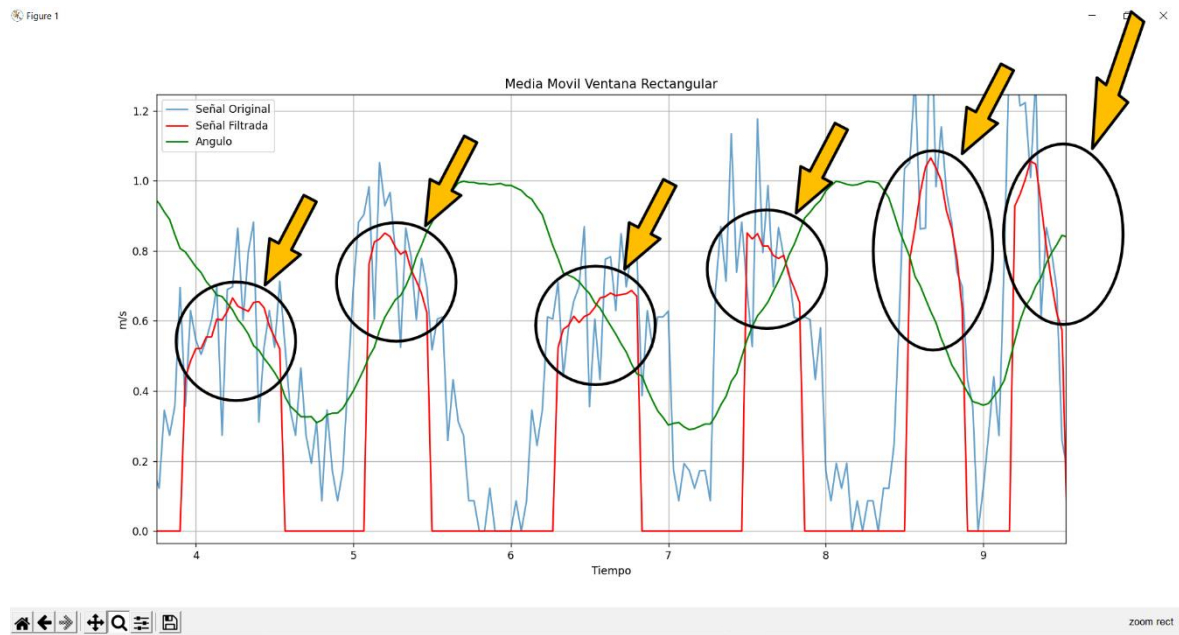


Figura 5.31. Intervalos óptimos de la señal

Una alta desviación estándar indica una variabilidad considerable en el rendimiento de las repeticiones, mientras que una baja garantiza un movimiento más constante y uniforme a lo largo de todo el ejercicio, lo cual puede ser deseable en ciertos casos para evaluar la calidad de la ejecución del ejercicio.

A continuación, se presenta un ejercicio en la figura 5.32 encontrado en el conjunto de videos local “*database*”, que muestra una baja desviación estándar en comparación con otros. Esto sugiere que la repetición se realizó con una velocidad constante y sin alteraciones significativas en el ritmo. Esto revela que el ejercicio fue realizado de manera óptima.

Datos adicionales:

	Valor
repeticiones	8.0000
Velocidad Media (m/s)	0.4106
Desviación estándar de velocidad	0.1080
Desviación Estándar Cadencia	1.2695
Cadencia Media (Repeticiones/minuto)	30.6240
Máximo de Cadencia (Repeticiones/minuto)	32.7279
Mínimo de Cadencia (Repeticiones/minuto)	29.5080

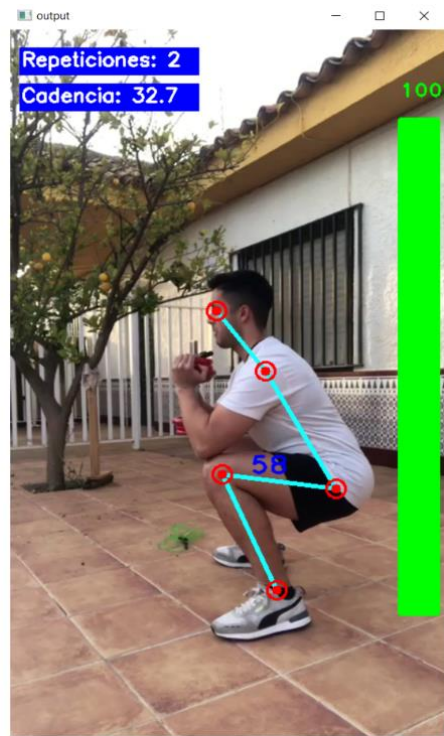


Figure 1

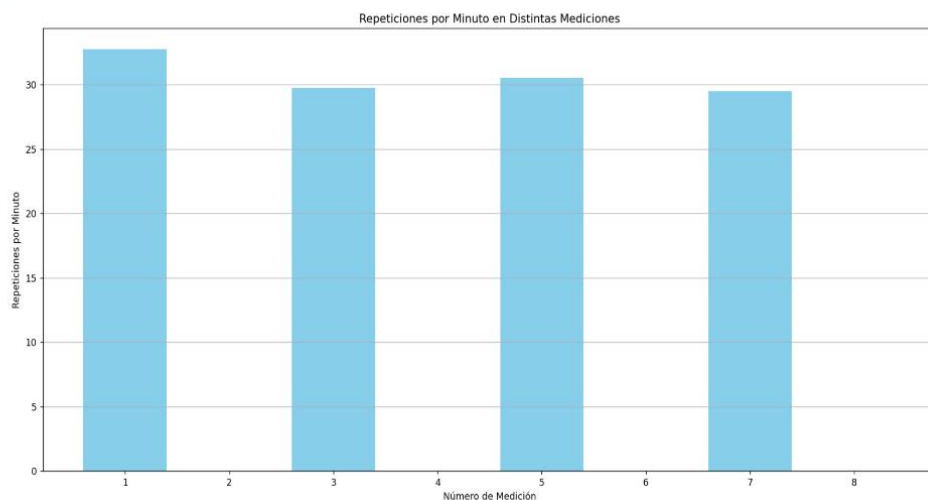


Figura 5.32. Baja desviación en J_SV_2

Como se observa en el gráfico de barras de la cadencia, la desviación indica claramente que no hay mucha dispersión con respecto a la media, proporcionando así un indicador clave para determinar que cuando este parámetro tiende a cero, el ejercicio no ha experimentado variaciones intensas en el ritmo.

6 Conclusiones

Después de concluir este proyecto, se destaca la obtención exitosa de una fase beta de la aplicación, la cual ha demostrado ser versátil y funcional, cumpliendo así con los objetivos generales marcados en la propuesta del proyecto, incluyendo el desarrollo de una aplicación basada en Python que integra detección de articulaciones mediante Visión Artificial y mediciones del movimiento articular tanto en tiempo real como en diferido. También se han realizado modificaciones sobre el planteamiento inicial de la aplicación, dotándola de una mayor componente en tiempo real al suministrar información en formato AR y superponerla a la realización del ejercicio. Además, se ha implementado un repositorio para almacenar los entrenamientos de los deportistas y se ha creado una interfaz web para facilitar la comunicación entre entrenadores y deportistas.

Es importante destacar que se ha logrado no solo la implementación de una aplicación completa, sino también la integración de múltiples tecnologías, algunas de las cuales eran desconocidas. Este proceso implicó enfrentar y superar los desafíos habituales asociados con el aprendizaje y la integración de nuevas herramientas y tecnologías.

Este logro confirma la capacidad para identificar una necesidad específica, en este caso, abordar la falta de conocimiento en el entrenamiento con pesas, especialmente entre los principiantes, con el objetivo de prevenir lesiones. Se ha demostrado que, mediante el uso de los conocimientos y una búsqueda resolutive, es posible encontrar soluciones efectivas para los problemas recurrentes en los entornos deportivos.

Este caso representa un ejemplo exitoso de cómo abordar y resolver los desafíos que enfrentan las instalaciones deportivas, contribuyendo así de manera positiva a la mejora del rendimiento y la seguridad de los usuarios.

Además, se destaca que esta aplicación no solo beneficia a los deportistas, sino que también ayuda a los entrenadores al agilizar su trabajo. Ahora, no necesitan revisar cada video de los usuarios para analizar si están ejecutando los ejercicios correctamente. Esta herramienta les proporciona datos precisos y detallados sobre la ejecución de los ejercicios, lo que les permite identificar áreas de mejora de manera rápida y eficiente.

6.1 líneas de futuro

Después de analizar exhaustivamente el trabajo realizado, se plantea un desarrollo futuro de la aplicación con un enfoque más orientado a dispositivos móviles. Se busca mejorar la eficiencia y adaptar los estilos para ofrecer una experiencia más fluida y coherente en entornos móviles, eliminando así la percepción de una aplicación web.

Además, se busca mejorar la interfaz de la aplicación para los usuarios, brindándoles la opción de utilizar la cámara a través de la aplicación móvil, en lugar de depender exclusivamente de la opción remota de cargar un archivo pregrabado durante el entrenamiento.

Se considera la implementación de funcionalidades adicionales en la aplicación, tales como la capacidad de crear y guardar rutinas de entrenamiento personalizadas, el seguimiento del progreso a lo largo del tiempo y la generación de informes detallados sobre el rendimiento deportivo.

Se plantea explorar posibles colaboraciones con instituciones deportivas o empresas del sector del fitness para integrar la aplicación en programas de entrenamiento existentes, aprovechando así sinergias y ampliando el alcance de la herramienta.

En consideración de estas líneas futuras, se ha elaborado un anexo empresarial que expone la iniciativa empresarial con todos sus objetivos claramente definidos. Para obtener más información, se recomienda consultar los anexos adjuntos.

7 Planos y Anexos

7.1 Anexo I: Documentación de empresa

El presente anexo sobre la creación de empresas ha sido elaborado con el valioso apoyo del técnico CADE de Linares, quien ha brindado orientación experta, facilitado documentación relevante y proporcionado un formato estructurado para la creación de empresas, lo cual ha sido fundamental para el desarrollo y la organización de este documento.

7.1.1 Antecedentes de los promotores

7.1.1.1 Motivos de la iniciativa empresarial

La iniciativa empresarial surge de la creciente demanda de herramientas tecnológicas que mejoren el seguimiento y análisis de movimientos en el ámbito deportivo y de la salud. Los promotores están motivados por la oportunidad de desarrollar una solución innovadora que combine la Visión Artificial, la Realidad Aumentada y el análisis de datos para proporcionar a entrenadores y usuarios una herramienta efectiva y fácil de usar.

7.1.1.2 Objetivos personales para poner en marcha la empresa

Los promotores tienen como objetivo principal crear una empresa líder en el desarrollo de aplicaciones tecnológicas para el seguimiento y análisis de movimientos. Buscan contribuir al avance en el campo del deporte y la salud, brindando soluciones que mejoren el rendimiento deportivo y prevengan lesiones.

7.1.1.3 Indicar en que ayudara la experiencia profesional previa en la presente iniciativa

La experiencia profesional adquirida durante el Trabajo de Fin de Grado (TFG) proporciona a los promotores una base sólida de conocimientos y habilidades técnicas relevantes para abordar la iniciativa empresarial. Este trabajo les habrá permitido obtener experiencia práctica en el diseño, desarrollo e implementación de aplicaciones basadas en la Visión Artificial y la Realidad Aumentada, así como en el análisis de datos relacionados con el seguimiento de movimientos. Gracias a esta experiencia, estarán equipados para enfrentar los desafíos técnicos y aprovechar las oportunidades emergentes en la creación de una herramienta innovadora para el seguimiento y análisis de movimientos en el ámbito deportivo y de la salud.

7.1.1.4 Indicar como se resolverán los problemas del trabajo en grupos

Los promotores tienen una sólida experiencia en el trabajo en equipo, lo que les permite colaborar de manera efectiva y resolver cualquier conflicto que pueda surgir. Se

comprometen a establecer una comunicación abierta y transparente, asignar roles y responsabilidades claras, y fomentar un ambiente de trabajo colaborativo y motivador. Además, contarán con un plan de gestión de conflictos y mecanismos de retroalimentación para abordar cualquier problema de manera constructiva y garantizar el éxito del equipo.

7.1.2 Oportunidad de negocio y análisis del sector

7.1.2.1 Ineficiencia en la supervisión de los entrenamientos

Los gimnasios enfrentan el desafío de supervisar adecuadamente a todos los clientes durante sus entrenamientos, especialmente en horas pico, lo que puede llevar a una atención deficiente o incluso a la falta de supervisión.

7.1.2.2 Necesidad de ampliar la cobertura con menos personal

Existe una necesidad de ampliar la capacidad de supervisión y atención a los clientes sin aumentar el personal, lo que implica encontrar formas más eficientes de gestionar y supervisar las sesiones de entrenamiento.

7.1.2.3 Valor económico de la oportunidad

Al ofrecer una solución que permite a los gimnasios supervisar y guiar a sus clientes de manera más efectiva con menos personal, se crea un valor económico significativo. Esto se traduce en una mayor eficiencia operativa, un mayor número de clientes atendidos simultáneamente y una reducción en los costos asociados a la contratación de entrenadores adicionales.

7.1.2.4 Descripción del sector y su potencial de crecimiento

El sector de los gimnasios es altamente competitivo y en constante crecimiento, con una demanda creciente de servicios que mejoren la experiencia del cliente y la eficiencia operativa. Sin embargo, existen barreras tecnológicas y legales que pueden influir en la entrada y salida del mercado, como la implementación de tecnologías innovadoras y la regulación de la privacidad de los datos de los usuarios. Es crucial comprender y abordar estas barreras para aprovechar el potencial de crecimiento del sector.

7.1.3 El producto o servicio

7.1.3.1 Generación de la idea. Objetivos por alcanzar

7.1.3.1.1 Origen de la idea y características de la idea empresarial

La idea surge de la necesidad de los gimnasios de optimizar la supervisión de los entrenamientos de sus clientes sin aumentar su personal. La aplicación se basa en la tecnología de Visión Artificial y Realidad Aumentada para ofrecer una solución innovadora

que permite a los gimnasios mejorar la eficiencia en la supervisión de los entrenamientos, reducir costos y aumentar la capacidad de atención a los clientes.

7.1.3.1.2 Descripción del producto/servicio

La aplicación ofrece una plataforma que utiliza la cámara de los dispositivos móviles para supervisar y guiar a los usuarios durante sus entrenamientos. Utiliza la Visión Artificial para analizar y corregir la técnica de movimiento, mientras que la Realidad Aumentada proporciona indicaciones visuales en tiempo real para mejorar la ejecución de los ejercicios.

7.1.3.1.3 Necesidades del mercado que se satisfacen

La aplicación satisface la necesidad de los gimnasios de optimizar la supervisión de los entrenamientos y mejorar la experiencia del cliente. Permite a los gimnasios ofrecer un servicio más personalizado y efectivo sin aumentar su personal, lo que les ayuda a retener y atraer a más clientes.

7.1.3.1.4 Temporalización del Proyecto y fuentes de financiación

El proyecto se divide en varias etapas, que incluyen el desarrollo y prueba del prototipo, la implementación piloto en gimnasios seleccionados, y la expansión a nivel nacional. Las fuentes de financiación incluyen inversores privados, subvenciones gubernamentales y capital propio de los promotores.

7.1.3.1.5 Metodologías de trabajo, patrocinadores y colaboradores

Se utilizará una metodología ágil de desarrollo de software para iterar rápidamente y adaptarse a las necesidades del mercado. Se buscarán patrocinadores en la industria del fitness y la tecnología, así como colaboraciones con gimnasios y expertos en entrenamiento deportivo.

7.1.3.1.6 Recursos del Proyecto y mediciones/evaluaciones iniciales

Los recursos del proyecto incluyen un equipo de desarrollo de software, expertos en Visión Artificial y Realidad Aumentada, y asesores en el sector del fitness. Se realizarán mediciones y evaluaciones iniciales para evaluar la eficacia de la aplicación en la mejora del rendimiento deportivo y la satisfacción del cliente.

7.1.3.1.7 Objetivos por alcanzar y establecimiento de indicadores

Los objetivos incluyen el lanzamiento exitoso de la aplicación, la adquisición de clientes piloto en gimnasios seleccionados, y la expansión a nivel nacional en un plazo determinado. Los indicadores incluyen el número de descargas de la aplicación, la satisfacción del cliente y la retención de clientes en los gimnasios piloto.

7.1.3.2 Novedades de la idea

La aplicación propuesta para gimnasios introduce innovaciones significativas en el mercado al emplear tecnologías avanzadas como la Visión Artificial y la Realidad Aumentada para ofrecer una experiencia de supervisión de entrenamientos altamente precisa y personalizada. A diferencia de otras soluciones, la aplicación proporciona *feedback* instantáneo y corrección de la técnica en tiempo real, lo que mejora la ejecución de los ejercicios y aumenta la motivación del usuario. Además, la capacidad de supervisar y guiar a múltiples usuarios simultáneamente con una sola aplicación optimiza los recursos y la eficiencia operativa para los gimnasios, ofreciendo ventajas competitivas significativas en el mercado.

7.1.4 El mercado: Los clientes y competencia

7.1.4.1 El mercado

En 2022, los alrededores de 4.000 gimnasios repartidos por todo el territorio español contaban con aproximadamente 5,5 millones de socios. Según el European Health & Fitness Market Report de Deloitte y EuropeActive, los gimnasios españoles registraron una facturación agregada de 2.100 millones de euros en 2022, un incremento del 32,9% respecto al año anterior, pero aún un 10,7% menos que en 2019 de acuerdo con [49]. Aunque el número de clientes alcanzó un récord de 5,4 millones, el sector no ha recuperado completamente los niveles de negocio previos a la pandemia, dejando de facturar 2.036 millones de euros en el trienio 2020-2022 en comparación con 2019. A pesar de esta recuperación del 11,8% de clientes en un año, aún persiste un déficit de 100.000 clientes respecto a las cifras prepandemia, aunque se ha logrado un aumento de 70.000 abonados más en la actualidad [50]. En cuanto al volumen de facturación en aplicaciones de seguimiento de ejercicios y salud similares, no se dispone de datos específicos, pero dada la importancia y crecimiento del sector del fitness en España, se estima que el mercado de aplicaciones de este tipo presenta un volumen significativo.

7.1.4.2 Identificación de la competencia

7.1.4.2.1 Empresas competidoras

WL Analysis Technologies, RepSpeed y Yekar.

7.1.4.2.2 Implantación en el mercado de la competencia

WL Analysis y RepSpeed tienen una presencia establecida en el mercado, mientras que Yekar se encuentra en fase de prelanzamiento.

7.1.4.2.3 Publicidad, antigüedad y nicho de mercado

WL Analysis y RepSpeed cuentan con una publicidad activa en línea y una base de clientes consolidada en el sector del fitness y levantamiento de pesas. Yekar se encuentra en una etapa inicial, con una estrategia de publicidad centrada en el prelanzamiento y un enfoque específico en el nicho de mercado de powerlifting.

7.1.4.2.4 Potenciales clientes de la iniciativa empresarial

Los gimnasios pueden ser potenciales clientes de la aplicación debido a su enfoque en la supervisión de movimientos en el ámbito deportivo, lo que les permite ofrecer un servicio más personalizado y efectivo a sus usuarios.

7.1.4.2.5 Puntos fuertes y débiles de la competencia

Los puntos fuertes incluyen la presencia establecida en el mercado y las características técnicas avanzadas. Los puntos débiles pueden ser la falta de adaptación al nicho específico de los gimnasios o la limitada disponibilidad de funcionalidades en fase de prelanzamiento.

7.1.4.3 Identificación de los clientes (Target)

Los clientes potenciales de la aplicación son los propietarios y gerentes de gimnasios y centros deportivos que buscan mejorar la experiencia y el servicio ofrecido a sus usuarios, valorando la innovación tecnológica. Se espera atender aproximadamente el 10-15% del número total de centros deportivos en España en los primeros años de actividad, centrándonos en gimnasios medianos a grandes. Esta solución les ayudará a optimizar la experiencia de entrenamiento, permitiéndoles supervisar y corregir la técnica de sus usuarios en tiempo real, lo que les permitirá diferenciarse en el mercado, aumentar la satisfacción y fidelización de sus clientes, y mejorar su reputación como centro deportivo de referencia.

7.1.5 Plan de Marketing y Comunicación: Estrategia de Posicionamiento y Segmentación del Mercado

7.1.5.1 Producto/Servicio

EL nombre de la aplicación será: *Movementor IA*

Se priorizará el desarrollo de una marca distintiva que transmita la innovación y calidad de la aplicación para gimnasios, asegurando una experiencia de usuario intuitiva y atractiva. El compromiso con la satisfacción del cliente se reflejará en un servicio de soporte completo y atención continua, respaldado por una sólida protección legal de la propiedad intelectual. Se establecerán indicadores clave de rendimiento para evaluar el éxito de la aplicación, aprovechando el conocimiento generado para mejoras continuas. La estrategia

de marketing se centrará en resaltar los beneficios únicos y la propuesta de valor, posicionando la aplicación como líder en el mercado fitness.

7.1.5.2 Precios

La política de precios se basa en una cuidadosa consideración de los costos de producción y en una comparación con los precios de la competencia en el mercado del fitness. La cuota mensual que paga el gimnasio se establece en función de las funcionalidades y el valor que la aplicación ofrece, adaptándose a las necesidades y presupuestos individuales de cada cliente. Además, como parte de la estrategia de promoción, se ofrece a los nuevos gimnasios una oferta especial de 3 meses de uso gratuito al ingresar en la plataforma, permitiéndoles experimentar plenamente los beneficios de la aplicación antes de comprometerse con la suscripción mensual, que asciende a 159.99 euros a partir del cuarto mes.

7.1.5.3 Promoción / Comunicación

La empresa se dará a conocer a la clientela a través de una estrategia integral que incluirá presencia en la web, en redes sociales y visitas comerciales. En la web, se desarrollará un sitio profesional y atractivo que destaque los beneficios y características únicas de la aplicación para gimnasios. En las redes sociales, se implementará una estrategia de contenido enfocada en el fitness y la tecnología, utilizando plataformas como Instagram, Facebook y LinkedIn para interactuar con la audiencia y promocionar la marca. Además, se realizarán visitas comerciales a gimnasios y centros deportivos para presentar personalmente la solución y establecer relaciones directas con los potenciales clientes. Esta combinación de estrategias permitirá llegar de manera efectiva a la clientela objetiva y generar interés en la oferta.

7.1.6 Recursos materiales y humanos

7.1.6.1 Recursos materiales

- Equipos informáticos: 2 ordenadores de alta gama para desarrollo y pruebas, estimado en 1800 euros cada uno.
- Servidor: Se requiere un servidor potente para alojar la aplicación y gestionar la base de datos de manera eficiente.
- Herramientas software: Licencias de software específicas para el desarrollo en Python, herramientas de Visión Artificial, gestión de bases de datos y diseño web, entre otros.

7.1.6.2 Recursos humanos

- 2 programadores Python: Encargados del desarrollo, mantenimiento y actualización de la aplicación en Python.
- 1 experto en Visión Artificial: Responsable de implementar y optimizar los algoritmos de Visión Artificial utilizados en la aplicación.
- 1 especialista en bases de datos: Encargado de diseñar, gestionar y optimizar la base de datos que respalda la aplicación.
- 1 diseñador CSS: Responsable de la parte visual y la experiencia de usuario en la interfaz web de la aplicación, utilizando CSS y otras tecnologías *front-end*.

Una vez puesta en el mercado la app, se contratarán dos informáticos para mantenimiento y soporte.

7.1.7 Plan Económico - Financiero

7.1.7.1 Cálculo de la inversión inicial y su detalle

Inversiones	Horas	Coste (€)
Equipos informáticos		3.600
Desarrollo app		6.840
• 2 programadores • 3 especialistas en Visión Artificial, base de datos y diseño CSS • Licencias de software • Gestión de bases de datos	320 horas 96h	
Gastos de constitución		170
Publicidad de lanzamiento		6.000
IVA soportado inversiones iniciales		3.484
Tesorería inicial		4.906
TOTAL INVERSIÓN INICIAL		25.000

Tabla 7.1 Plan económico

7.1.7.2 FINANCIACIÓN

FONDOS PROPIOS	25.000 €
----------------	----------

Tabla 7.2 Fondos propios

7.1.7.3 Cuenta de resultados

VOYAGER

Cuentas de Pérdidas y Ganancias

MOVEMENTOR

Conceptos	2025	% / Ventas	2026	% / Ventas	Variación	2027	% / Ventas	Variación	2028	% / Ventas	Variación	2029	% / Ventas	Variación
Ingresos por ventas	66.396	100,0%	125.215	100,0%	88,6%	152.027	100,0%	21,4%	162.821	100,0%	7,1%	174.382	100,0%	7,1%
Imputación de subvenciones	0	0,0%	0	0,0%		0	0,0%		0	0,0%		0	0,0%	
Ingresos	66396	100,0%	125215	100,0%	1	152027	100,0%	0	162821	100,0%	0	174382	100,0%	0
Costes variables (directos)	6.640	10,0%	12.522	10,0%	88,6%	15.203	10,0%	21,4%	16.282	10,0%	7,1%	17.438	10,0%	7,1%
Margen Bruto s/Ventas	59756	90,0%	112694	90,0%	1	136825	90,0%	0	146539	90,0%	0	156944	90,0%	0
Gastos de Personal y Seq. Social	39.000	58,7%	56.160	44,9%	44,0%	58.406	38,4%	4,0%	60.743	37,3%	4,0%	63.172	36,2%	4,0%
Seguridad Social	7.680	11,6%	17.321	13,8%	126,5%	18.820	12,4%	8,7%	19.318	11,9%	2,6%	19.837	11,4%	2,7%
Otros Gastos Fijos	15.720	23,7%	10.012	8,0%	(36,3%)	10.312	6,8%	3,0%	10.621	6,5%	3,0%	10.940	6,3%	3,0%
Dotación Amortizaciones	3.000	4,5%	3.000	2,4%	0,0%	3.000	2,0%	0,0%	720	0,4%	(76,0%)	720	0,4%	0,0%
Total Gastos Explotación	65400	98,5%	86492	69,1%	0	90538	59,6%	0	91402	56,1%	0	94669	54,3%	0
Res. antes Int. e Imp. (BAII)	-5644	-8,5%	26201	20,9%	-6	46286	30,4%	1	55137	33,9%	0	62274	35,7%	0
Ingresos Financieros	0	0,0%	0	0,0%		0	0,0%		0	0,0%		0	0,0%	
Gastos Financieros	0	0,0%	0	0,0%		0	0,0%		0	0,0%		0	0,0%	
Resultado Financiero	0	0,0%	0	0,0%		0	0,0%		0	0,0%		0	0,0%	
Res. antes Impuestos (BAI)	-5644	-8,5%	26201	20,9%	-6	46286	30,4%	1	55137	33,9%	0	62274	35,7%	0
Impuesto de Sociedades	0	0,0%	5.139	4,1%		11.572	7,6%	126,2%	13.784	8,5%	19,1%	15.569	8,9%	12,9%
Resultado Neto	(5.644)	(8,5%)	21.062	16,8%	(473,2%)	34.715	22,8%	64,8%	41.353	25,4%	19,1%	46.706	26,8%	12,9%

Todas las cantidades están expresadas en Euros

Figura 7.1 Pérdidas y Ganancias

7.1.7.4 Resumen tesorerías

Resumen de Tesorerías

MOVEMENTOR



VOLVER

	Año 0 2025	Año 1 2026	Año 2 2027	Año 3 2028	Año 4 2029
COBROS					
Cobro de ventas	80.339	151.510	183.953	197.014	211.002
Otras entradas	0	0	0	0	0
Devoluciones de IVA	0	0	0	0	0
Ingresos financieros	0	0	0	0	0
Total COBROS	80.339	151.510	183.953	197.014	211.002
PAGOS					
Pagos de Gastos Fijos (IVA incl.)	64.441	85.595	89.704	92.913	96.246
Pago de compras y Costes Variables (IVA incl.)	8.034	15.151	18.395	19.701	21.100
Pagos de Gastos Financieros	0	0	0	0	0
Otros pagos NO gastos					
Devolución de principal del préstamo	0	0	0	0	0
Otras salidas de caja	0	0	0	0	0
Liquidaciones de IVA	2.295	19.947	25.745	28.012	30.091
Otros pagos (Imp. Soc./IRPF)	0	0	5.139	11.572	13.784
Total PAGOS	74.770	120.693	138.984	152.197	161.222
Cash Flow	5.569	30.818	44.969	44.816	49.780
Tesorería acumulada	10.906	16.475	47.293	92.262	186.858
Máximo dispuesto de póliza de crédito	0	0	0	0	0
mes máxima disposición de póliza de crédito					
Máxima tesorería mensual	16.475	47.293	92.262	137.078	186.858
mes máxima tesorería mensual	diciembre	diciembre	diciembre	diciembre	diciembre

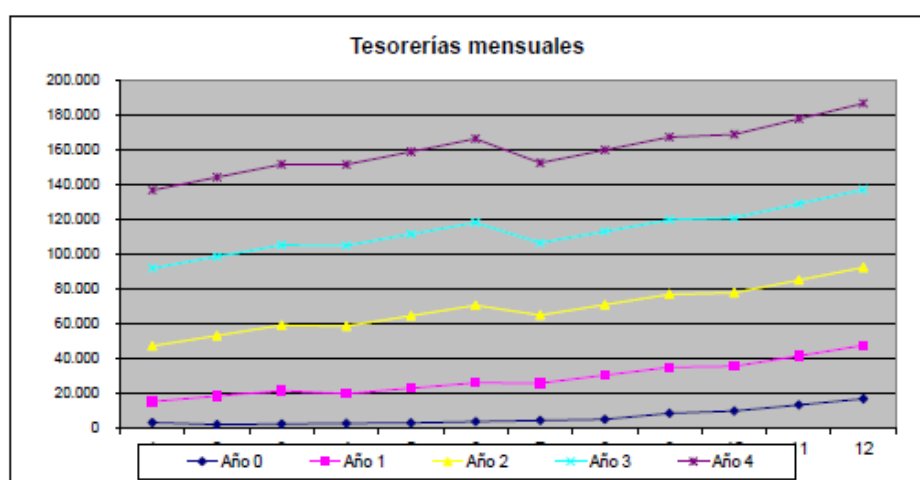


Figura 7.2. Tesorerías

Plan de Tesorería MOVEMENTOR

2025

Concepto	enero	febrero	marzo	abril	mayo	junio	julio	agosto	septiembre	octubre	noviembre	diciembre	Total
COBROS													
Cobro de ventas	479	1,597	3,194	3,194	3,194	7,986	7,986	7,986	11,180	11,180	11,180	11,180	80,339
Otras entradas	0	0	0	0	0	0	0	0	0	0	0	0	0
Devoluciones de IVA	0	0	0	0	0	0	0	0	0	0	0	0	0
Ingresos financieros	0	0	0	0	0	0	0	0	0	0	0	0	0
Total COBROS	479	1,597	3,194	3,194	3,194	7,986	7,986	7,986	11,180	11,180	11,180	11,180	80,339
PAGOS													
Pagos de Gastos Fijos (IVA incl.)	8,560	2,560	2,560	2,560	2,560	6,520	6,520	6,520	6,520	6,520	6,520	6,520	64,441
Pago de compras y Costes Variables (IVA incl.)	48	160	316	319	319	799	799	799	1,118	1,118	1,118	1,118	8,034
Pagos de Gastos Financieros	0	0	0	0	0	0	0	0	0	0	0	0	0
Pagos IVA cuotas Leasing / Renting	0	0	0	0	0	0	0	0	0	0	0	0	0
Otros pagos NO gastos													
Devolución de principal (préstamos y leasing)	0	0	0	0	0	0	0	0	0	0	0	0	0
Otras salidas de caja	0	0	0	0	0	0	0	0	0	0	0	0	0
Liquidaciones de IVA	0	0	0	0	0	0	0	0	0	2,295	0	0	2,295
Otros pagos (Imp. Soc./RPF)	0	0	0	0	0	0	0	0	0	0	0	0	0
Total PAGOS	8,608	2,720	2,880	2,880	2,880	7,319	7,319	7,319	7,638	9,933	7,638	7,638	74,770
Tesorerías mensuales													
		(8,129)	(1,123)	315	315	315	667	667	667	3,542	1,247	3,542	5,569
Dispuesto de póliza de crédito	0	0	0	0	0	0	0	0	0	0	0	0	0
Tesorería acumulada	10,906	2,777	1,655	1,969	2,284	2,599	3,267	3,934	4,601	8,143	9,391	12,933	16,475
Máximo dispuesto de póliza de crédito													
Flujo de caja del periodo (Cash Flow)													
Máxima tesorería mensual													

Figura 7.3 Tesorería primer año

Estimación del Umbral de Rentabilidad o Punto de Equilibrio MOVEMENTOR



VOLVER

	Datos Año 0 2025	Datos Año 1 2026	Datos Año 2 2027	Datos Año 3 2028	Datos Año 4 2029	
Gastos laborales	46.680	73.481	77.226	80.061	83.009	€/ año
Otros Gastos	15.720	10.012	10.312	10.621	10.940	"
Amortizaciones	3.000	3.000	3.000	720	720	"
Total Gastos Fijos	65.400	86.492	90.538	91.402	94.669	"
Resultados Financieros	0	0	0	0	0	€/ año
Margen Medio de Contribución	90.0%	90.0%	90.0%	90.0%	90.0%	
Punto de Equilibrio (sin gastos financ.)	72.667	96.103	100.598	101.558	105.188	€ de venta al año
P.E. Mensual medio	6.056	8.009	8.383	8.463	8.766	€/ mes de media
Punto de Equilibrio (con gastos financ.)	72.667	96.103	100.598	101.558	105.188	€ de venta al año
P.E. Mensual medio	6.056	8.009	8.383	8.463	8.766	€/ mes de media
Ventas anuales previstas	66.396	125.215	152.027	162.821	174.382	€/ año de venta previstos
Venta mensual media prevista	5.533	10.435	12.669	13.568	14.532	€/ mes de media
Beneficio Bruto Estimado (B.A.I.I.)	(5.644)	26.201	46.286	55.137	62.274	€/ año
Beneficio Estimado (B.A.I.)	(5.644)	26.201	46.286	55.137	62.274	€/ año
Coef. de seguridad (sin gastos financ.)	0,91	1,30	1,51	1,60	1,66	= Prev. Ventas / Pto. Equilibrio
Coef. de seguridad (con gastos financ.)	0,91	1,30	1,51	1,60	1,66	
Incrementos de ventas necesarios	9,44%	0,00%	0,00%	0,00%	0,00%	0

Figura 7.4 Punto de equilibrio

Balances Anuales Previsionales

MOVEMENTOR



	enero de 2025		diciembre 2025		diciembre 2026		diciembre 2027		diciembre 2028		diciembre 2029	
	Importe	%	Importe	%	Importe	%	Importe	%	Importe	%	Importe	%
Activos No Corrientes	10440,00	42,0%	7440,00	31,1%	4440,00	8,6%	1440,00	1,5%	720,00	0,5%	0,00	0,0%
Immovilizado Material	3.600	14,5%	3.600	15,1%	3.600	7,0%	3.600	3,8%	3.600	2,6%	3.600	1,9%
Amort. Acum. Innov. Material	0	0,0%	(720)	(3,0%)	(1.440)	(2,8%)	(2.160)	(2,3%)	(2.880)	(2,1%)	(3.600)	(1,9%)
Immovilizado Inmaterial	6.840	27,5%	6.840	28,6%	6.840	13,2%	6.840	7,3%	6.840	5,0%	6.840	3,7%
Amort. Acum. Innov. Inmaterial	0	0,0%	(2.280)	(9,5%)	(4.560)	(8,8%)	(6.840)	(7,3%)	(6.840)	(5,0%)	(6.840)	(3,7%)
Innov. Financiero Largo Plazo	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Activos Corrientes	14.390	58,0%	16475,18	68,9%	47292,75	91,4%	92262,07	98,5%	137078,42	99,5%	186858,10	100,0%
Innov. Financiero Corto Plazo	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Existencias	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Clientes	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
H.P. Pagos y retenciones IRPF	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
H.P. Deudora I. Sociedades	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
H.P. Deudora IVA	3.484	14,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Tesorería	10.906	43,9%	16.475	68,9%	47.293	91,4%	92.262	98,5%	137.078	99,5%	186.858	100,0%
Total Activo	24.830	100,0%	23.915	100,0%	51.733	100,0%	93.702	100,0%	137.798	100,0%	186.858	100,0%

Patrimonio Neto	24.830	100,0%	19.186	80,2%	40.248	77,8%	74.963	80,0%	116.316	84,4%	163.021	87,2%
Capital Social	25.000	100,7%	25.000	104,5%	25.000	48,3%	25.000	26,7%	25.000	18,1%	25.000	13,4%
Reservas	(170)	(0,7%)	(170)	(0,7%)	(5.814)	(11,2%)	15.248	16,3%	49.963	36,3%	91.316	48,9%
Credito de accionistas	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Subvenciones	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Resultados pend. aplicación	0	0,0%	(5.644)	(23,6%)	21.062	40,7%	34.715	37,0%	41.353	30,0%	46.706	25,0%
Pasivos No Corrientes	0,00	0,00%	0,00	0,00%	0,00	0,00%	0,00	0,00%	0,00	0,00%	0,00	0,00%
Acreedores L.P. Financieros	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Otros Acreedores L.P.	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Pasivos Corrientes	0,00	0,00%	4728,78	19,77%	11484,49	22,20%	18739,10	20,00%	21482,82	15,59%	23836,69	12,76%
Acreedores C.P. Financieros	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Credito financiero C.P.	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Proveedores	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
H.P. acreedora IVA	0	0,0%	4.729	19,8%	6.345	12,3%	7.168	7,6%	7.699	5,6%	8.268	4,4%
H.P. acreedora Imp. Soc./IRPF	0	0,0%	0	0,0%	5.139	9,9%	11.572	12,3%	13.784	10,0%	15.569	8,3%
Otros Acreedores C.P.	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%
Total Recursos Permanentes	24.830	100,0%	19.186	80,2%	40.248	77,8%	74.963	80,0%	116.316	84,4%	163.021	87,2%
Total Recursos Ajenos	0	0,0%	4.729	19,8%	11.484	22,2%	18.739	20,0%	21.483	15,6%	23.837	12,8%
Total Patrimonio Neto y Pasivos	24.830	100,0%	23.915	100,0%	51.733	100,0%	93.702	100,0%	137.798	100,0%	186.858	100,0%

Fondo de Maniobra	14.390	58,0%	11.746	49,1%	35.808	69,2%	73.523	78,5%	115.596	83,9%	163.021	87,2%
--------------------------	---------------	--------------	---------------	--------------	---------------	--------------	---------------	--------------	----------------	--------------	----------------	--------------

Figura 7.5 Balances

7.2 Anexo II: Código fuente (anexo digital)

Se proporciona un enlace al archivo que contiene el código fuente completo de la aplicación, que se encuentra alojado en un repositorio en la nube, específicamente en Google Drive. Este enlace será incluido como parte de los anexos del trabajo de fin de grado (TFG). El código está organizado en módulos y funciones que se implementan en el procesamiento de datos para distintos ejercicios, la interfaz web de usuario y la interfaz web de entrenador. La disponibilidad del código en formato electrónico facilita su revisión y comprensión.

Se facilita el enlace correspondiente para su visualización, seleccionando el siguiente vínculo:

- [Código fuente en Google Drive](#)

7.3 Anexo III: Base de Datos Local (anexo digital)

Con el propósito de obtener resultados concretos y llevar a cabo diversas pruebas para el desarrollo de una fase beta funcional, se ha creado una base de datos propia, como se muestra en la figura 7.3. Esta base de datos contiene múltiples repeticiones de diversos tipos de ejercicios y ha sido denominada "*database*". Se ha integrado en el repositorio mencionado previamente, aunque en una carpeta distinta. Se proporciona el enlace correspondiente para su visualización y acceso, seleccionando el siguiente vínculo:

- [Repositorio de ejercicios deportivos en Google Drive](#)

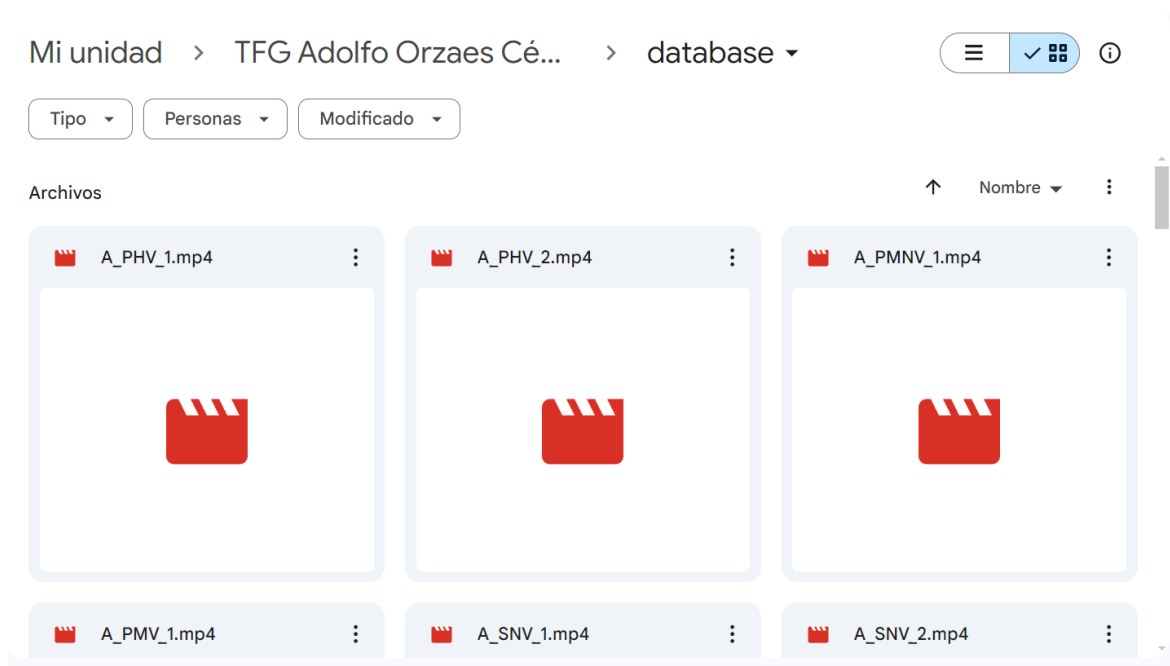


Figura 7.6 Repositorio Drive con ejercicios

8 Bibliografía

- [1] Matarogroc, «Lesiones en la sala de musculación,» 20 Septiembre 2021. [En línea]. Available: <https://anefead.com/fitness/lesiones-en-la-sala-de-musculacion/>.
- [2] I. N. d. Estadística, «Instituto Nacional de Estadística,» 9 Enero 2024. [En línea]. Available: https://www.ine.es/ss/Satellite?L=es_ES&c=INESeccion_C&cid=1259926457058&p=%255C&pagename=ProductosYServicios%2FPYSLayout¶m1=PYSDetalle¶m3=1259924822888.
- [3] «Historia de la educación física.,» de *Historia de la educación física. Cuadernos de sección*, 1992, pp. 27-47.
- [4] C. J. P. C. P. S. & C. M. J. Colón, «Benefits of Strength Training for the Prevention and Treatment of Sarcopenia.,» de *Benefits of Strength Training for the Prevention and Treatment of Sarcopenia.*, Directory of Open Access Journals, 2014, pp. 979-988.
- [5] «Transforming fitness apps,» [En línea]. Available: <https://vsight.io/blog/transforming-fitness-apps-through-augmented-reality/>.
- [6] «What is Computer Vision?,» [En línea]. Available: <https://www.ibm.com/es-es/topics/computer-vision>.
- [7] «History of computer vision and convolutional neural networks,» [En línea]. Available: <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3>.
- [8] O. Porta, «Historia de la visión artificial: así ha evolucionado esta tecnología.,» Marzo 24 2023. [En línea]. Available: <https://infaimon.com/blog/vision-2d-3d/historia-evolucion-vision-artificial/>.
- [9] J. A. P. Estapé, «Vuelve Kinect, con la nueva cámara Azure Kinect solo para profesionales. Computer Hoy.,» 27 Febrero 2019. [En línea]. Available: <https://computerhoy.com/noticias/tecnologia/vuelve-kinect-nueva-camara-azure-kinect-solo-profesionales-380703>.

- [10] Tesych., «Acerca de Azure Kinect DK. Microsoft Learn.,» 31 Julio 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/kinect-dk/about-azure-kinect-dk#azure-kinect-body-tracking-sdk>.
- [11] T. a. C. Szegedy., «Deeppose: Human pose estimation via Deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition.,» 2014, p. 1653–1660.
- [12] A. F. D. M. V. R. J. & L. A. M. Alvarez, 2016, pp. 2961-2969.
- [13] G. Boesch, «The complete guide to OpenPose in 2024,» Enero 4 2024. [En línea]. Available: <https://viso.ai/deep-learning/openpose/>.
- [14] Gymnotize, «Fitness App (Gym & Home Workouts),» [En línea]. Available: <https://www.gymnotize.com/>.
- [15] «Virtuagym,» [En línea]. Available: <https://virtuagym.uptodown.com/android>.
- [16] «Historia de las bases de datos - desde SQL a NoSQL,» [En línea]. Available: <https://www.tecnologias-informacion.com/historiabbdd.html>.
- [17] K. D. Foote, «A Brief History of Non-Relational Databases,» 13 Junio 2018. [En línea]. Available: <https://www.dataversity.net/a-brief-history-of-non-relational-databases/>.
- [18] «WL analysis,» [En línea]. Available: <https://wlanalysis.com/>.
- [19] «Velocity Based Training for everyone,» [En línea]. Available: <https://www.repspeed.net/>.
- [20] «Fit Generation,» [En línea]. Available: <https://fitgeneration.es/rpe-vs-rir/>.
- [21] «Microsoft,» [En línea]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>.
- [22] «AR.js - Realidad Aumentada en la Web,» Documentación AR.js, [En línea]. Available: <https://ar-js-org.github.io/AR.js-Docs/>.
- [23] «Java,» [En línea]. Available: <https://www.java.com/es/>.

- [24] «OpenWbinars,» [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>.
- [25] «OpenWbinars,» [En línea]. Available: <https://openwebinars.net/blog/python-en-ia-aplicaciones-practicas-y-bibliotecas-clave/>.
- [26] «IONOS,» [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/>).
- [27] «Microsoft VisualStudio,» [En línea]. Available: <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>.
- [28] «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/brand>.
- [29] «YOLOv8,» [En línea]. Available: <https://docs.ultralytics.com/es>.
- [30] A. y. T. A. Halder, «Reconocimiento de lengua de signos vernácula en tiempo real mediante mediapipe y aprendizaje automático.,» 2021.
- [31] «MediaPipe Solutions guide,» MediaPipe, [En línea]. Available: <https://developers.google.com/mediapipe/solutions/guide>.
- [32] «Mediapipe Solutions Vision Pose Landmarker,» Mediapipe , [En línea]. Available: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker.
- [33] «Genexus Noticias,» Genexus, [En línea]. Available: <https://www.genexus.com/es/noticias/leer-noticia/anuncio-el-30-de-mayo-google-deja-de-soportar-el-mecanismo-habitual-de-autenticacion-para-mails>.
- [34] «Compass. The GUI for MongoDB.,» MongoDB, [En línea]. Available: <https://www.mongodb.com/products/tools/compass>.
- [35] «Ngrok Unified Ingress Platform,» Ngrok , [En línea]. Available: <https://ngrok.com/> .

- [36] «Pythoninstitute,» Python, [En línea]. Available: <https://pythoninstitute.org/about-python>.
- [37] «Python tutorial,» Python, [En línea]. Available: <https://docs.python.org/es/3/tutorial/>.
- [38] «Developers Google Mediapipe,» Mediapipe, [En línea]. Available: <https://developers.google.com/mediapipe>.
- [39] «Pose Landmark Detection Guide Google for Developers.,» Google for Developers, [En línea]. Available: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker.
- [40] C. W. W. C. C. Y. T. Z. S. S. J. .. y. S. Zheng, «Deep Learning-Based Human Pose Estimation: A Survey,» 2023.
- [41] «Selecting your 2d real time pose estimation models,» Maureentkt Medium, [En línea]. Available: <https://maureentkt.medium.com/selecting-your-2d-real-time-pose-estimation-models-7d0777bf935f>.
- [42] «Que es opencv,» cursodeinstalador, [En línea]. Available: <https://cursodeinstalador.com/que-es-opencv/>.
- [43] «estimacion-de-posicion-con-opencv-y-python/,» Laboratorio Gluon, [En línea]. Available: <https://www.laboratoriogluon.com/estimacion-de-posicion-con-opencv-y-python/>.
- [44] «Numpy,» Numpy, [En línea]. Available: <https://numpy.org/>.
- [45] «Todo Sobre Matplotlib,» Datascientest, [En línea]. Available: <https://datascientest.com/es/todo-sobre-matplotlib>.
- [46] «Streamlit,» Streamlit , [En línea]. Available: <https://streamlit.io/> .
- [47] «Desarrollo Web MongoDB Python,» Ionos, [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/mongodb-python/>.
- [48] «Python Library Module Tkinter,» Python, [En línea]. Available: <https://docs.python.org/es/3/library/tkinter.html#module-tkinter>.

- [49] «Informe european health and fitness market-report 2023 de europeactive,» Valgo, [En línea]. Available: <https://www.valgo.es/blog/publicado-el-informe-european-health-and-fitness-market-report-2023-de-europeactive-y-deloitte?elem=299572>.
- [50] «El sector del Fitness en Espana,» Statista, [En línea]. Available: <https://es.statista.com/temas/6047/el-sector-del-fitness-en-espana/#topicOverview>.
- [51] «PlaySotre,» [En línea]. Available: https://play.google.com/store/search?q=salud+y+fitness&c=apps&hl=es_419&gl=US.
- [52] V. A. P. B. Victoriaruss, «Vi Feature Detectors/ Hubel & Wiesel Experiment/ Prosopagnosia,» 4 Abril 2014. [En línea]. Available: <https://goodpsychology.wordpress.com/2013/03/13/235/?ref=hackernoon.com>.
- [53] U. Y. Y. K. S. Ş. R. & S. L. B. Özsoy, «Reliability and agreement of Azure Kinect and Kinect V2 depth sensors in the shoulder joint range of motion estimation. Journal of Shoulder and Elbow Surgery,» 2022. [En línea]. Available: <https://doi.org/10.1016/j.jse.2022.04.007>.
- [54] GeeksforGeeks., «DeepPose human pose estimation via deep neural networks.,» 8 Septiembre 2021. [En línea]. Available: <https://www.geeksforgeeks.org/deeppose-human-pose-estimation-via-deep-neural-networks/>.
- [55] «The complete guide to OpenPose in 2024,» [En línea]. Available: <https://viso.ai/deep-learning/openpose/>.
- [56] L. E. Gymnotice Mack, « 4 fun augmented reality apps for fitness.,» 21 Junio 2023. [En línea]. Available: <https://www.makeuseof.com/augmented-reality-apps-fitness/>.
- [57] «PlayStore VirtuaGym,» [En línea]. Available: https://play.google.com/store/apps/details?id=digifit.virtuagym.client.android&hl=es_CL.

- [58] G. L. T. y. R. F. Casal, «Conceptos y tipos de bases de datos NoSQL (documental, columnar, clave/valor y de grafos) | Prácticas de tecnologías de gestión y manipulación de datos.,» Noviembre 3 2023. [En línea]. Available: <https://gltaboada.github.io/tgdbook/conceptos-y-tipos-de-bases-de-datos-nosql-documental-columnar-clavevalor-y-de-grafos.html>.
- [59] «FotoWLANalisi:App Store.,» Diciembre 2 2020. [En línea]. Available: <https://apps.apple.com/us/app/wl-analysis/id1541855037>.
- [60] «PlayStore. Bar velocity tracker,» [En línea]. Available: https://play.google.com/store/apps/details?id=com.sire.repspeed&hl=en_US.
- [61] «Udemy,» [En línea]. Available: <https://www.udemy.com/course/unity-c-sharp-scripting/>.
- [62] «Damianavilla. Numpy,» [En línea]. Available: Reconocimiento de lengua de signos vernácula en tiempo real mediante mediapipe y aprendizaje automático..