



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA ELECTRÓNICA Y COMUNICACIONES

Tema:

**“SISTEMA ELECTRÓNICO DE MONITOREO PARA LA DETECCIÓN Y
CORRECCIÓN DE POSTURAS EN EJERCICIOS DE LEVANTAMIENTO
DE POTENCIA PARA DEPORTISTAS.”**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

ÁREA: Comunicaciones

LÍNEA DE INVESTIGACIÓN: Tecnologías de Comunicación

AUTOR: Byron Rodrigo Palacios Sánchez

TUTOR: Ing. Mario Geovanny García Carrillo Mg.

Ambato – Ecuador

Marzo – 2021

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA ELECTRÓNICO DE MONITOREO PARA LA DETECCIÓN Y CORRECCIÓN DE POSTURAS EN EJERCICIOS DE LEVANTAMIENTO DE POTENCIA PARA DEPORTISTAS, desarrollado bajo la modalidad Proyecto de Investigación, por el señor Byron Rodrigo Palacios Sánchez, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2021.



Firmado electrónicamente por:
**MARIO GEOVANNY
GARCIA CARRILLO**

Ing. Mario Geovanny García Carrillo Mg.

TUTOR

AUTORÍA

El presente trabajo de investigación titulado “SISTEMA ELECTRÓNICO DE MONITOREO PARA LA DETECCIÓN Y CORRECCIÓN DE POSTURAS EN EJERCICIOS DE LEVANTAMIENTO DE POTENCIA PARA DEPORTISTAS” es absolutamente original, autentico y personal, en tal virtud el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2021.



BYRON RODRIGO PALACIOS SÁNCHEZ

CC: 1805180526

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Byron Rodrigo Palacios Sánchez, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado “SISTEMA ELECTRÓNICO DE MONITOREO PARA LA DETECCIÓN Y CORRECCIÓN DE POSTURAS EN EJERCICIOS DE LEVANTAMIENTO DE POTENCIA PARA DEPORTISTAS”, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora presidenta del Tribunal.

Ambato, marzo 2021.



Firmado electrónicamente por:
**ELSA PILAR
URRUTIA**

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL



Firmado electrónicamente por:
**CARLOS DIEGO
GORDON GALLEGOS**

PhD. Carlos Diego Gordón Gallegos
PROFESOR CALIFICADOR



Firmado electrónicamente por:
**MARCO ANTONIO
JURADO LOZADA**

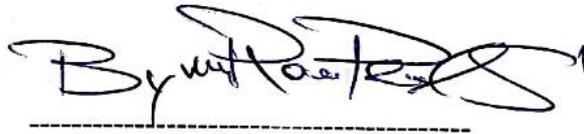
Ing. Marco Antonio Jurado Lozada
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2021.



BYRON RODRIGO PALACIOS SÁNCHEZ

CC: 1805180526

AUTOR

AGRADECIMIENTO

A Dios por darme la fuerza y fe para seguir adelante y no dejarme derrotar en los momentos difíciles de mi formación profesional, a mis Padres por su total apoyo, consejos diarios y aliento económico, a mis hermanos y hermanas que han sido un pilar fundamental y un ejemplo para seguir, a mis compañeros de estudios, que me ayudaron a superar cada etapa en la realización de mi proyecto de titulación

Byron Rodrigo Palacios Sánchez

DEDICATORIA

Dedico esta tesis a mis queridos padres Rigo y María Lucila por ser el ejemplo de superación y liderazgo, por el apoyo incondicional para no decaer en momentos de angustias de mi travesía, por darme la fuerza y ganas de seguir alcanzando metas y cosechar éxitos, a mis hermanas Nancy, Lilian, Mónica, Silvia y hermanos Diego y Willian por su total apoyo, su empuje a seguir por el camino recto de la vida que han sido pilares fundamentales en las batallas diarias que luchamos cada día para salir adelante en los momentos difíciles que necesitaba de apoyo.

A toda mi familia que es lo más valioso que Dios me ha dado

Byron Rodrigo Palacios Sánchez

INDICE GENERAL

PORTADA.....	i
APROBACIÓN DEL TUTOR	ii
AUTORIA.....	iii
APROBACION TRIBUNAL DE GRADO.....	iv
DERECHOS DE AUTOR	v
AGRADECIMIENTO	vi
DEDICATORIA	vii
RESUMEN EJECUTIVO.....	xiv
ABSTRACT.....	xv
GLOSARIO DE TÉRMINOS Y ACRÓNIMOS.....	1
CAPITULO I.- MARCO TEÓRICO	2
1.1 TEMA DE INVESTIGACIÓN	2
1.2 ANTECEDENTES INVESTIGATIVOS.....	2
1.2.1 CONTEXTUALIZACION DEL PROBLEMA	4
1.2.2 JUSTIFICACIÓN	4
1.2.3 FUNDAMENTACION TEORICA.....	5
1.2.3.1 Sistema de Visión	5
1.2.3.2 Sistema de visión artificial	6
1.2.3.3 OPENCV	9
1.2.3.4 Preprocesamiento de la imagen Filtro espacial	11
1.2.3.5 Procesamiento.....	11
1.2.3.6 Software de aplicaciones de visión por computadora	14
1.2.3.7 Microcontroladores.....	15

1.2.3.9	SENSORES	17
1.2.3.10	Clasificación de sensores	18
1.2.3.11	CAMARA DIGITAL	19
1.2.3.12	Entorno De Desarrollo Integrado (IDE)	20
1.2.3.13	Comparación sistema de visión artificial y humano.....	21
1.2.3.14	Aplicaciones de la visión por computadora.....	22
1.2.3.15	Ventajas de los sistemas de visión artificial	22
1.2.3.16	Tipos de software de Inteligencia Artificial (IA)	23
1.2.3.17	Ejercicio Físico con peso muerto.....	26
1.2.3.18	Posición de los pies, la barra y el agarre.....	26
1.2.3.19	Posición de la espalda.....	27
1.3	OBJETIVOS	43
CAPITULO II.- METODOLOGIA		43
2.1.	MATERIALES	43
2.2	Métodos	45
2.2.1	MODALIDAD DE INVESTIGACIÓN	45
2.2.2	RECOPIACIÓN DE LA INFORMACIÓN.....	45
CAPITULO III.- RESULTADOS Y DISCUSIÓN		46
3.1	Análisis y discusión de los resultados	46
3.2	Desarrollo de la propuesta.....	47
3.2.1	Estudio de Factibilidad	47
3.2.1.1	Factibilidad Técnica	47
3.2.1.2	Factibilidad Económica.....	47
3.2.1.3	Factibilidad Bibliográfica.....	47

3.2.2 Requerimientos o especificaciones para el desarrollo del sistema.....	47
2.2.2 Selección de elementos para la elaboración del sistema	50
3.2.2.3 Cámara.....	50
3.2.2.4 Sensor de monitoreo de pulso Cardiacó	53
3.2.2.5 Microcontrolador	55
3.2.2.6 Procesador	57
3.2.2.7 Lenguaje de programación	60
3.2.2.8 Software IDE Proton	63
3.2.3 Diseño del prototipo	64
3.2.4 Implementación	65
3.3 Pruebas y Resultados.....	97
3.3.1 Análisis de Errores.....	100
3.4 RECURSOS.....	103
3.4.1 Institucionales.....	103
3.4.2 Humanos.....	103
3.4.3 Materiales	103
3.5 Económicos	104
CAPITULO IV.....	106
4.1 Conclusiones	106
4.2 RECOMENDACIONES	107
BIBLIOGRAFÍA.....	108
ANEXOS.....	112

INDICE DE FIGURAS

Figura 1: Anatomía del ojo humano.....	6
Figura 2: Sistema de Visión Artificial.....	6
Figura 3: Proceso de visión artificial.....	7
Figura 4: Diagrama de bloques Visión Artificial.....	8
Figura 5: Imágenes con posibilidad de ser digitalizadas.....	12
Figura 6: Ejemplo de imágenes con alta probabilidad de éxito.....	12
Figura 7: Imágenes divididas en partes sustituyentes.....	13
Figura 8: Imagen segmentada para el procesamiento.....	13
Figura 9: Imagen asignada para el reconocimiento y la interpretación de información.....	14
Figura 10: Elementos Básicos de un microcontrolador.....	17
Figura 11: Clasificación de sensores.....	19
Figura 12: Comparativa entre el ojo humano y una cámara fotográfica.....	22
Figura 13: Correcta posición en levantamiento de potencia.....	28
Figura 14: Ejecución de levantamiento de potencia.....	28
Figura 15: Pasos para Ejecución de levantamiento de potencia.....	29
Figura 16: Ejecución de levantamiento de potencia, posición de cabeza.....	31
Figura 17: Ejecución de levantamiento de potencia, posición de columna.....	31
Figura 18: Aplicación del levantamiento de potencia en el deporte.....	32
Figura 19: Frecuencia Cardíaca en Levantamiento de Potencia.....	33
Figura 20: Referencia para personas que ejecutan Levantamiento de Potencia.....	34
Figura 21: Referencia para personas que ejecutan Levantamiento de Potencia.....	34
Figura 22: Comportamiento de Linealidad ReLU.....	42
Figura 23: Etapa convolucional neuronal.....	42
Figura 24: Diagrama de flujo del desarrollo del sistema.....	49
Figura 25: ELP HD 1.3Megapixel USB cámara (modelo: ELP-X000QSOMLP).....	53
Figura 26: Diagrama PIC 12F683.....	62
Figura 27: Ventanilla lista para crear programas en Proton IDE.....	63
Figura 28: Propuesta de solución.....	64
Figura 29: Botón de descarga Anaconda.....	65
Figura 30: Elegir la versión de descarga Anaconda.....	65
Figura 31: Ejecutable/ instalador automático Anaconda.....	66
Figura 32: Instalador automático Anaconda.....	66
Figura 33: Instalador para todos los usuarios.....	67
Figura 34: Ruta de instalación de Anaconda.....	67
Figura 35: Paso Importante para la partición y compatibilidad de Anaconda.....	68
Figura 36: Configuración del sistema.....	68
Figura 37: Instalación Completa de Anaconda con entorno de Python.....	69
Figura 38: Versión de anaconda en el entorno Python.....	69
Figura 39: Prueba de verificación de OpenCV correctamente instalado.....	70
Figura 40: Elegir construir en Python.....	72
Figura 41: Figura que proporciona el algoritmo desarrollado del cuerpo humano.....	74
Figura 42: Figura que muestra los puntos de principales en el cuerpo humano.....	75

Figura 43: Figura que muestra la resolución para la salida de muestreo de imágenes	76
Figura 44: Estimación de pose usando Tensorflow y OPENCV	77
Figura 45: Mapeo de imagen.....	78
Figura 46: Ejemplo de funcionalidad del algoritmo.....	78
Figura 47: Diagrama del circuito para el sensor de pulso	80
Figura 48: Etapa de amplificación	81
Figura 49: Características de un filtro pasa alto	82
Figura 50: Amplificador operacional con filtro de paso bajo.	82
Figura 51: Etapa de Filtrado con un Amplificador	83
Figura 52: Amplificadores seguidor no inversor.....	83
Figura 53: Amplificadores seguidor no inversor en el Circuito.....	84
Figura 54: Conexión PIC 12F683 en el circuito	84
Figura 55: Estructura de código en PROTON IDE, PIC 12F683	87
Figura 56: Detector sensor de pulso	88
Figura 57: Medición óptica de Diodos LED	88
Figura 58: Señal de pulso cardíaco obtenida de diferentes partes del cuerpo.....	89
Figura 59: Configuración para obtener la señal de pulso.....	89
Figura 60: Placa electrónica del sistema electrónico para el monitoreo de pulso cardíaco ...	90
Figura 61: Pistas de la placa electrónica prototipo 1 del sistema electrónico para el monitoreo de pulso cardíaco.....	90
Figura 62: Prototipo mejorado del sistema Electrónico Implementado en guante de entrenamiento deportivo parte posterior con una caja de protección.....	91
Figura 63: Prototipo Mejorado del sistema Electrónico Implementado en guante de entrenamiento deportivo parte Interior.....	91
Figura 64: Prueba para el agarre de levantamiento de potencia.....	92
Figura 65: Prueba del Sistema en Ambiente 2	92
Figura 66: Mala postura monitoreada en tiempo real	93
Figura 67: Prueba real del circuito en entrenamiento de potencia	94
Figura 68: Prueba real del sistema Electrónico	94
Figura 69: Diagrama de interacción del sistema	95
Figura 70: Diagrama de Flujo: Sistema para corrección de posturas.....	96
Figura 71: Estimación de pose usando Tensorflow y OPENCV con incorrecta posición	97
Figura 72: Estimación de pose usando Tensorflow y OPENCV con correcta posición	98
Figura 73: Porcentaje de aciertos en las pruebas del sistema.....	99

INDICE DE TABLAS

Tabla 1: Clasificación de Open CV.....	10
Tabla 2: Ventajas y Desventajas de Cámaras.....	20
Tabla 3:Proporción del torso y brazos en relación con la altura de la persona.....	33
Tabla 4:Recomendaciones de peso muerto sumo o convencional en relación con la tabla anterior.....	33
Tabla 5:Matriz de Kernell.....	36
Tabla 6: Matriz de entrada.....	37
Tabla 7:Convolución del Kernel.....	38
Tabla 8: Resultado de convolución.....	40
Tabla 9: Matriz de convolución.....	41
Tabla 10: Resultado ReLU.....	41
Tabla 11: Comparación de cámaras web.....	51
Tabla 12:Comparación de las características de sensores de proximidad.....	54
Tabla 13: Clasificación de Microcontroladores.....	56
Tabla 14: Comparación de procesadores.....	59
Tabla 15: Comparación de lenguajes de programación.....	61
Tabla 16: Tabla de descripción de pines PIC 12F683.....	62
Tabla 17: Respuesta promedio a la carga de entrenamiento con pesas.....	98
Tabla 18: Datos (Repeticiones y porcentaje) de repeticiones correctas en pruebas.....	100
Tabla 19: Tabla de evaluación para la técnica del Agarre.....	100
Tabla 20: Tabla de evaluación en la respiración antes de realizar el movimiento.....	101
Tabla 21:Tabla de evaluación de la posición de la espalda durante el movimiento.....	102
Tabla 22: Presupuesto del proyecto.....	105

RESUMEN EJECUTIVO

Este proyecto de investigación presenta el desarrollo de un sistema electrónico que implementa la visión artificial mediante el uso del software de programación de lenguaje libre Python y el procesamiento de señales obtenidas por un sensor de pulso cardiaco siendo estas herramientas de utilidad en el análisis y el reconocimiento de movimientos objetivando las principales articulaciones del cuerpo al momento de realizar el ejercicio físico apoyando a los deportistas a ejecutar el levantamiento de peso muerto de forma correcta para que los músculos que intervienen sean estimulados de una forma adecuada evitando posibles lesiones o consecuencias negativas a corto o largo plazo.

El levantamiento de potencia es un deporte en el que se presenta una alta notoriedad de lesiones osteomusculares especialmente en rodillas, hombros y columna lumbar siendo estas zonas donde ocurren la mayoría de las lesiones lo cual se atribuye a la mala postura al realizar dicho ejercicio, siendo este prototipo capaz de ser utilizado por cualquier persona de cualquier edad o condición física

Hoy en día el mundo moderno y la tecnología ha querido igualar los sentidos humanos, capacidades y acciones de manera artificial por ejemplo la vista con el reconocimiento de patrones y algoritmos, para ello se hace uso de herramientas computacionales y electrónicas como lenguajes de programación y componentes electrónicos como es una tarjeta microprocesador Raspberrypi la cual servirá para la comunicación entre los datos obtenidos y el algoritmo desarrollado lo cual las máquinas intentan dar solución a estos requerimientos.

Palabras Clave: Visión Artificial, Monitoreo, Tiempo Real, Pulso Cardiaco, posturas, Ejercicio, Levantamiento de Potencia

ABSTRACT

This research project presents the development of an electronic system that implements artificial vision through the use of Python free language programming software and the processing of signals obtained by a heart pulse sensor, these being useful tools in the analysis and the recognition of movements objectifying the main joints of the body when performing physical exercise supporting athletes to perform dead weight lifting correctly so that the muscles involved are stimulated avoiding possible short-term or long-term negative injuries or consequences.

Power lifting is a sport in which there is a high prevalence of osteomuscular lesions especially in the knees and lumbar spine these areas where most injuries manifest themselves, which is attributed to poor posture when performing such exercise, being this prototype capable of being used by anyone of any age or physical condition

Today the world of technology has artificially matched human actions, capacities and actions in an artificial way, for example the sight with the recognition of patterns and algorithms, for this use is made of computational and electronic tools such as programming languages and electronic components such as a Raspberrypi microprocessor card which will serve for communication between the data obtained and the algorithm developed which the machines try to solve these requirements.

Keywords: Artificial Vision, Monitoring, Real Time, Heart Pulse, Postures, Exercise, Power Lift

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

IA: La inteligencia artificial es una rama de la informática dedicada a la creación de sistemas informáticos inteligentes.

RGB: Es un modelo de color que consiste en representar distintos colores a partir de los colores primarios.

CI: Un circuito integrado es un chip donde se integra varios componentes electrónicos

SD: Abreviatura para representar señales discretas

CNN: Son las redes neuronales convolucionales

OpenCV: Proviene de las siglas en ingles Open Source Computer Vision Library (Librerías de visión por Computadora en Código Abierto)

A/D: Abreviatura de señales Análogo/Digital

CAPITULO I.- MARCO TEÓRICO

1.1 TEMA DE INVESTIGACIÓN

SISTEMA ELECTRÓNICO DE MONITOREO PARA LA DETECCIÓN Y CORRECCIÓN DE POSTURAS EN EJERCICIOS DE LEVANTAMIENTO DE POTENCIA PARA DEPORTISTAS.

1.2 ANTECEDENTES INVESTIGATIVOS

Actualmente se han hecho muchos avances con respecto a la visión artificial y a los sistemas de procesamiento de imágenes, por el hecho de que este es un campo que tiene mucho potencial y muchas instituciones, tanto universidades como industrias, han estado investigando como aplicar y como mejorar los actuales sistemas de visión artificial y procesamiento de imagen; en esta sección se mostrarán algunos trabajos relacionados con el tema de la visión artificial y su utilidad en el desarrollo de la investigación.

En la Universidad Tecnológica de Pereira (Colombia) en el año 2012 se ha realizado un estudio por parte de Nicolás López Pérez y Juan José Toro Agudelo con el tema “Técnicas de biometría basadas en patrones faciales del ser humano”. Esta técnica utiliza el algoritmo de Eigenfaces, que mapea las características de la cara de una persona en un espacio multidimensional del rostro, debido a esto se puede realizar la identificación y su posterior reconocimiento de personas, para permitir su acceso a ciertos recursos, pero este tipo de sistemas no se limita únicamente a controlar la identificación de personal, sino que permite una evolución en el campo de la videovigilancia.[4]

En la Escuela Politécnica Nacional del Ecuador en el año 2014 los estudiantes María Teresa Calderón y Diego Javier Maldonado con el tema:“ Control por visión de un cuadrúpedo utilizando ROS (Sistema Operativo de Robot)” empleando OpenCV concluyeron; que mediante el uso de recursos y herramienta contenidos en el framework ROS es posible desarrollar diferentes aplicaciones para el seguimiento de objetos , ya sea por su color o forma básica, pues el framework ROS se caracteriza por apoyar a la reutilización de código para la investigación y

desarrollo de la robótica; también recomienda implementar un algoritmo de procesamiento muy robusto y sensible que permita el desarrollo de aplicaciones en entornos poco controlados.[5]

En la Universidad Autónoma de México en el año 2018, José Martínez y Juan Manuel con el tema “Sistema de Visión Artificial para la Detección y Corrección de Posturas en Ejercicios realizados por Fisicoculturistas” Entre algunos ejemplos se contemplan: la vista, con el reconocimiento de objetos o patrones, el seguimiento del cuerpo humano y la captura de movimiento (Mo Cap). Para ello se hace uso de herramientas computacionales, como los llamados exergame, que incitan el movimiento del usuario ante entornos virtuales, y la programación de algoritmos que intentan dar solución a estos problemas, el objetivo de dicho proyecto de tesis es desarrollar un sistema de Visión Artificial interactivo, que permita reconocer y seguir los movimientos de los fisicoculturistas para determinar si sus posturas son correctas en tiempo real, mediante la posición de sus articulaciones en el sistema de ejes de tres dimensiones. [6]

En la Institución Universitaria de Salazar y Herrera en el año 2016, S. Amaya, I. Torres y D. Velásquez con el tema “Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura” utilizando una cámara web integrada al software libre Python, que trabaja en conjunto con las librerías Open CV. El principio de funcionamiento del seleccionador está basado en un posicionador de objetos, previamente configurado por el usuario usando el sistema de colores frente a la lente de la cámara, la cual se encargará de capturar la imagen y procesarla por medio del lenguaje de software y el sistema permite informarle al usuario si la pieza ingresada al sistema es de los colores y la forma deseada por él, logrando una clasificación de objetos eficaz [3]

En la Pontificia Universidad Javeriana de Bogotá en 2005, David Martínez en su proyecto de investigación con el tema “Interceptor de Trayectorias Basado en visión Artificial” utiliza la estrategia de control basado en visión artificial con

lenguaje c++ y Open CV y adiciona además los conceptos básicos de la visión humana después de cada imagen adquirida se repite redundantemente el procesamiento para emulación y cálculo de mundo exterior, genera un sistema capaz de soportar variaciones, no solo en las características intrínsecas del entorno, sino también al método inicial del proceso haciéndolo comparable con las estrategias que involucran controladores adaptables o inteligencia artificial. [7]

1.2.1 CONTEXTUALIZACION DEL PROBLEMA

En la actividad física existen riesgos al ejecutar de forma incorrecta los ejercicios del programa de entrenamiento establecido según las necesidades en el levantamiento de pesas, de esta forma no se aprovecha y mucho menos se obtiene ventaja de una manera eficaz el trabajo y por lo tanto el efecto que tiene sobre el cuerpo humano dicho ejercicio es nulo, o incluso perjudicial sabiendo que existe una alta prevalencia de lesiones especialmente en rodillas y columna lumbar en un 10% y 32% respectivamente, los atletas que realizaron el levantamiento de pesas, participantes en los Juegos Deportivos Nacionales en 2004 datos obtenidos gracias al estudio que realizó el Laboratorio Integral de Análisis para el Movimiento de la Escuela Superior Politécnica del Ecuador. Como contribución de la Ingeniería en Electrónica y Comunicaciones para la solución de este problema, se busca la implementación y desarrollo de una interfaz de software interconectado con un sensor que monitorea el pulso cardiaco ya que el incremento acelerado de pulsos en el corazón alerta que algo no va bien, mediante una placa electrónica utilizada para la comunicación con el sistema de visión artificial que permita la identificación, procesamiento y apoyo en la corrección de las posturas en ejercicios físicos, para deportistas en sus diversos niveles, enfocado primordialmente en el nivel de levantamiento de peso muerto o levantamiento de potencia.

1.2.2 JUSTIFICACIÓN

El presente tema corresponde a la aplicación de visión artificial mediante el uso de software libre, este sistema electrónico de monitoreo logra la detección de la correcta posición del cuerpo para la ejecución de ejercicios físicos con pesas

denominados "peso muerto", dicho ejercicio se divide en 3 variantes importantes (peso muerto Rumano, peso muerto Push Press y peso muerto Convencional), este sistema de monitoreo está relacionado con un control estético de la correcta postura del cuerpo, con la ayuda del sistema de visión artificial el individuo estará en constante monitoreo al ejecutar el levantamiento de potencia haciendo correcciones si el caso lo amerita.

Su principal objetivo es evitar lesiones corporales debido al gran peso que soporta el cuerpo, gracias al estudio realizado para la Federación Deportiva de Tungurahua por factores de riesgo posturales dinámicos y de la ejecución de la técnica para lesiones Osteomusculares en levantadores de pesas concluye que las lesiones lumbares y sintomatologías dolorosas del 100% de la población de deportistas que realizan el levantamiento de potencia un 57% presenta lesiones y la aparición de dolor post competencia según este estudio por algún factor de riesgo postural y/o en la ejecución de la técnica. [24]

Este sistema de monitoreo puede ser aplicado para que las personas realicen correctamente la ejecución del ejercicio evitando lesiones y además aprovechando al máximo la eficacia del ejercicio. Principalmente puede ser aplicado en los centros deportivos, clubs de actividades físicas, centros de alto rendimiento deportivo, como es el caso de la posible implementación en el Gimnasio de la Universidad Técnica de Ambato campus Huachi si la situación actual de estado de pandemia cambie favorablemente.

1.2.3 FUNDAMENTACION TEORICA

1.2.3.1 Sistema de Visión

Sistema de visión es uno de los sentidos más complejos y especiales que forman parte del ser humano, el cual es capaz de identificar la forma, el volumen, los colores, el tamaño y la luminosidad de la mayoría de los objetos que rodean al ser humano, lo cual es la mayor parte de percepción de la persona, como se ve en la Figura 1 la anatomía del ojo humano que cumplen con la función específica de permitir la visión al ser humano.[8]



Figura 1: Anatomía del ojo humano

Fuente: Ciencias Naturales/órganos del cuerpo humano [42]

1.2.3.2 Sistema de visión artificial

Los sistemas de visión artificial relacionados al campo de la inteligencia artificial, tiene la finalidad de programar un computador el cual sea capaz de procesar información y entender características de una escena o imagen, por lo cual se considera el proceso de extraer información del mundo físico que rodea a la persona a partir de imágenes y procesándolas en un computador. En la Figura 2 muestra a un sistema de visión Artificial desde la toma de la imagen, su procesamiento digital hasta llegar al actuador externo del proceso. [9]

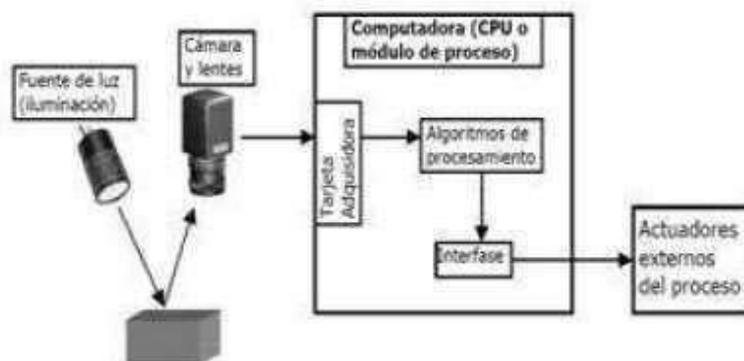


Figura 2: Sistema de Visión Artificial

Fuente: Cognex, Visión Artificial [4]

Algunas de las tareas que se pueden realizar mediante la Visión Artificial incluyen Automatizar tareas repetitivas, como las siguientes:

- Detectar colores, patrones, señales, objetos o movimiento.
- Realizar controles de calidad de productos que no es posible verificar por métodos tradicionales.
- Realizar inspecciones de objetos sin contacto físico y generar una calidad casi en su totalidad a una gran velocidad.
- Reducir el tiempo de ciclos en procesos automatizados.
- Determinar la posición de objetos en el espacio.
- Establecer relaciones espaciales entre varios objetos (guiado de robots).
- Mediciones tridimensionales y angulares.
- Determinar coordenadas importantes de un objeto.

El proceso global de un sistema de visión artificial se presenta en la Figura 3, en la cual se distinguen 4 módulos:

1. Digitalización, el cual se apoya con la cámara para la obtención de la imagen y se convierte la señal analógica proporcionada por la cámara a una señal digital.
2. Procesamiento se encarga de obtener información de la imagen y su tratamiento con instrucciones precisas.
3. Memoria, que permite el almacenamiento de la información.
4. Visualización, en el cuál convierte la señal digital residente en la memoria, en señal de video analógica para poder ser visualizada.

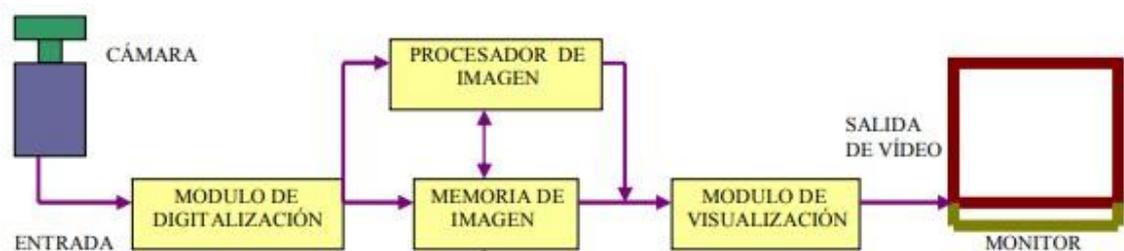


Figura 3: Proceso de visión artificial.

Fuente: Centro Integrado Politécnico ETI Tudela, 2017. [5]

El proceso general de un sistema de visión artificial puede ser visualizado en la Figura 4, en la que las cajas representan datos y las burbujas procesos. Se parte de la escena tridimensional, pasando por la toma de muestras o imágenes para su posterior segmentación de los bordes y regiones, finalmente terminando con su procesamiento para su aplicación. [6]

En el área de visión artificial se pueden distinguir tres procesos generales: Análisis, Procesamiento y Aplicaciones. El análisis consiste en identificar todas aquellas estructuras elementales de la imagen, tales como los bordes o regiones, así como la relación que hay entre ellas. El procesamiento se enfoca en la manipulación de las imágenes como señales digitales, para extraer la información más elemental de la misma. Finalmente, las aplicaciones tratan de dar solución al uso que pueda surgir en el mundo real, a saber: reconocimiento, movimiento, reconstrucción 3D, entre otros. [6]

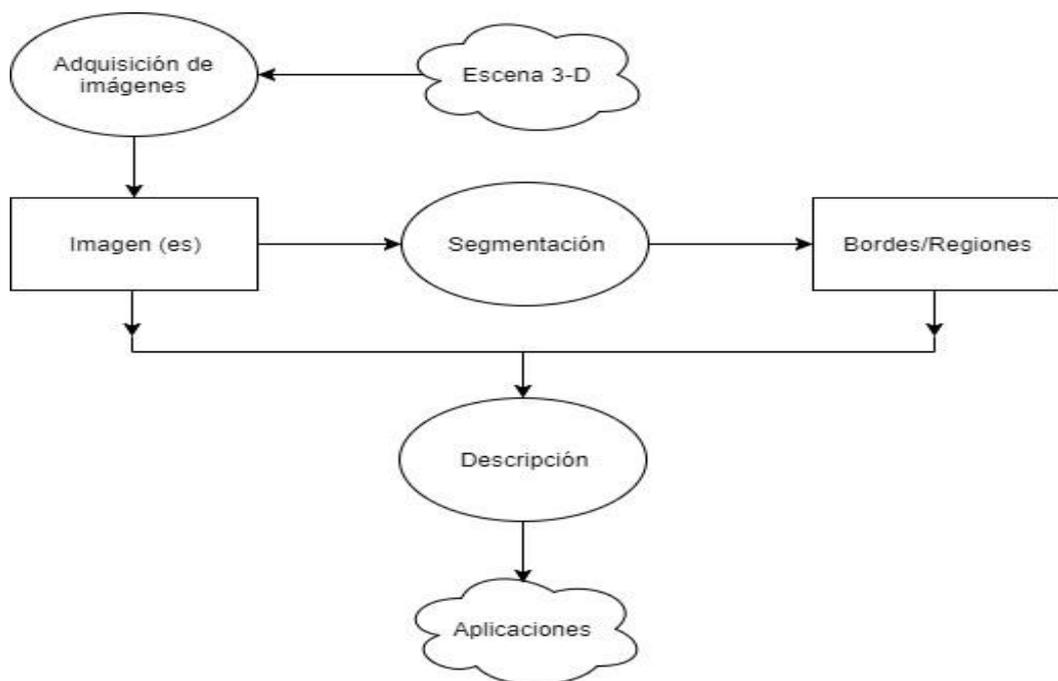


Figura 4: Diagrama de bloques Visión Artificial

Fuente: Shapiro & Stockman, 2000 [8]

1.2.3.3 OPENCV

OpenCV es una colección de tipos de datos, funciones y clases para el procesamiento de imágenes con algoritmos de visión por computadora proviene de las siglas en inglés Open Source Computer Vision Library, una biblioteca que hasta la primera versión fue desarrollada en el Centro de Desarrollo de Software Intel y por un equipo ruso en Nizhny Novgorod, esta biblioteca es muy popular debido a su apertura y la capacidad de usarla de forma gratuita con fines educativos y comerciales. OpenCV está escrito en un lenguaje de alto nivel C / C ++ y contiene algoritmos para realizar funciones como:

- Interpretar imágenes
- Calibrar una cámara contra una referencia
- Eliminar las distorsiones ópticas
- Determinar la similitud
- Analizar el movimiento del objeto
- Determinar la forma de un objeto y rastrear un objeto
- Reconstrucción 3d
- Segmentación de objetos
- Reconocimiento de gestos

En la versión 3, la estructura de la biblioteca se ha reorganizado, ahora en lugar de grandes módulos universales como cxcore (estructuras y algoritmos de datos básicos, operaciones en matrices numéricas multidimensionales, álgebra matricial, funciones matemáticas, generadores de números aleatorios, estructuras de datos de XML), Cvaux (búsqueda y descripción de características faciales, calibración estéreo, autocalibración, búsqueda de coincidencia estéreo y clics en gráficos) y Highgui (creación de una interfaz de usuario, funciones para organizar una interfaz de usuario simple, captura de video de cámaras y archivos de video), OpenCV tiene pequeños módulos de acuerdo con el uso funcional en la tabla 1 se observa su clasificación.[18]

Tabla 1: Clasificación de Open CV

Fuente: OpenCV Computer Vision with Python [18]

Módulos OpenCV	Descripción
opencv_core	Núcleo: estructuras básicas, cálculos como funciones matemáticas, generación de números pseudoaleatorios, DFT, DCT, entrada / salida XML, etc.
opencv_imgproc	Procesamiento de imágenes como filtros, transformaciones, etc.
opencv_highgui	Interfaz de usuario simple, cargando / guardando imágenes y videos.
opencv_ml	Métodos y modelos de aprendizaje automático como SVM, árboles de decisión, etc.
opencv_features2d	Clase base abstracta para detectores de características de imágenes 2D y extractores de descriptores.
opencv_video	Análisis de movimiento y seguimiento de objetos como flujo óptico, patrones de movimiento, eliminación de fondo.
opencv_objdetect	Detección de objetos en la imagen, wavelets de Haar, HOG, etc.
opencv_calib3d	Calibración de la cámara, búsqueda de coincidencia estéreo y elementos de procesamiento de datos 3D.
opencv_flann	Biblioteca de búsqueda rápida de vecinos más cercanos
opencv_contrib	Módulos extras de OpenCV para CUDA
opencv_legacy	Código heredado retenido para compatibilidad con versiones anteriores
opencv_gpu	Aceleración de algunas funciones de OpenCV debido a CUDA (NVIDIA)

1.2.3.4 Preprocesamiento de la imagen Filtro espacial

Preprocesamiento de la imagen filtro espacial es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella, este tipo de filtros realiza el filtrado a los bordes de la imagen en todas direcciones (los resultados que se obtienen pueden considerarse como una “suma” de los obtenidos tras aplicar todos los modelos de filtrado). En esta ocasión se trabaja con la segunda derivada, que permite obtener unos mejores resultados de filtrado, a pesar del aumento del ruido que se produce en la imagen empleada para destacar y resaltar con mayor precisión los bordes que se localizan en una dirección determinada. Trabaja con los cambios de intensidad existentes entre píxeles contiguos. [17]

1.2.3.5 Procesamiento

El procesamiento es el tratamiento de los datos, la detección de bordes, segmentación, la búsqueda de ciertos patrones como los de color, además de ser procesamientos digitales de la imagen es la manipulación matemática de una señal de información para modificarla o mejorarla. [17]

Pasos del procesamiento

A continuación se detalla cada paso para el procesamiento de imágenes dando a conocer su principal funcionalidad y características.

Adquisición de imágenes:

Adquisición de imágenes es la extracción de información avanzada de diferentes entornos y procesos, en forma de características o parámetros relacionados con la imagen, se necesita un sensor de imágenes y la posibilidad de digitalizar la señal producida por el sensor.

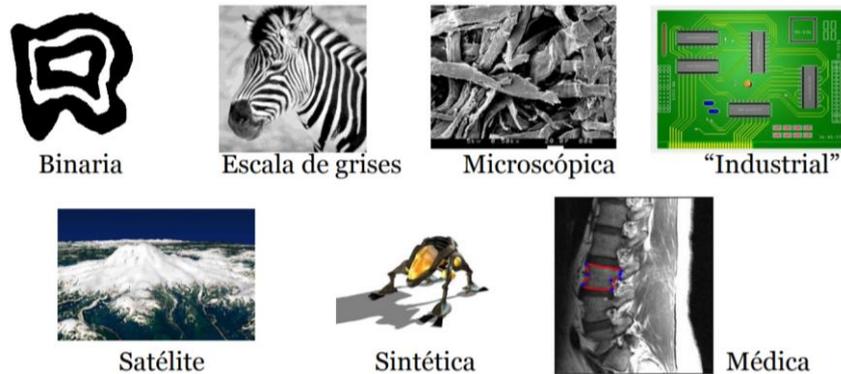


Figura 5: Imágenes con posibilidad de ser digitalizadas

Fuente: Profesorado.us.es/Imagen Digital

Preprocesado: El preprocesado es una fase que comparten algunos lenguajes de programación, anterior a la fase de compilación propiamente dicha. Se aplica generalmente sobre el archivo principal que contiene el código fuente su principal función es hacer inteligible para el compilador el código fuente, cambiando las directivas de preprocesamiento por valores para el compilador.

Mejorar la imagen de forma que aumenten las probabilidades de éxito en los procesos posteriores como eliminación de ruido, suavizado, realce y mejorar el contraste en este caso el realce se logra mediante iluminación.

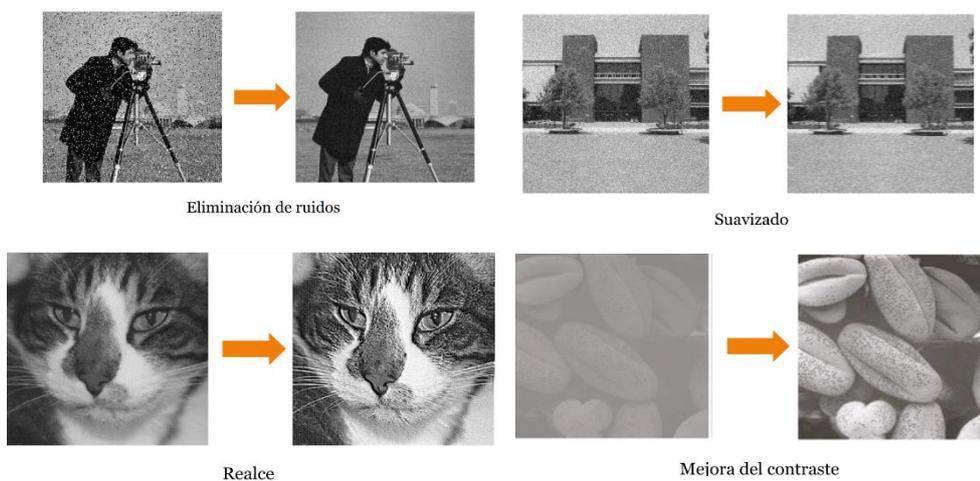


Figura 6: Ejemplo de imágenes con alta probabilidad de éxito

Fuente: Profesorado.us.es/Imagen Digital

Segmentación: Método para la división de una imagen en dos o más regiones es decir dividir la imagen en sus partes sustituyentes u objetos para el análisis de imágenes digitales. Los resultados de los mismos serán utilizadas en etapas sucesivas para su descripción, reconocimiento e interpretación su principal función es identificar regiones que correspondan a objetos físicos, los métodos de segmentación asumen que las regiones a extraer poseen algunas características homogéneas distintivas, pueden existir límites físicos entre regiones similares que no aparecen como tales en la imagen

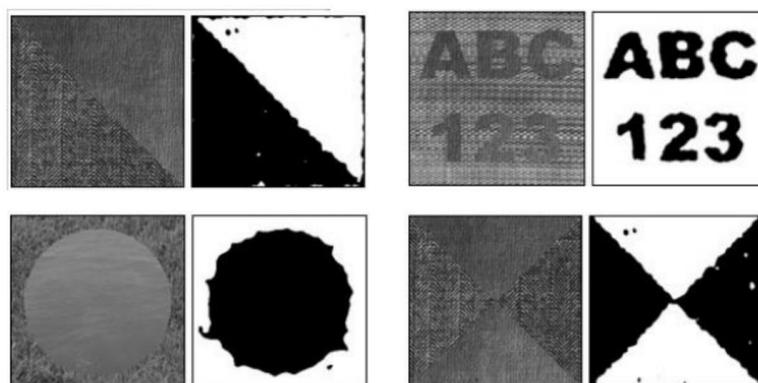


Figura 7: Imágenes divididas en partes sustituyentes

Fuente: Profesorado.us.es/Imagen Digital

Representación: Convertir los datos de los píxeles (de un contorno o una región), obtenidos tras la segmentación, en una forma adecuada para el procesamiento por computadora. En la figura 8 la representación como un contorno es la adecuada cuando el interés radica en las características de la forma exterior, esquinas e inflexiones, la representación es adecuada cuando el interés se centra en propiedades internas, como la textura o la estructura de un objeto

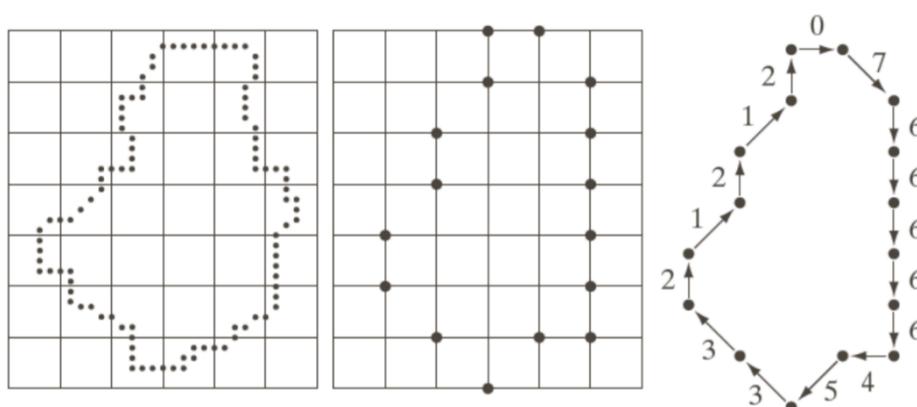


Figura 8: Imagen segmentada para el procesamiento

Fuente: Profesorado.us.es/Imagen Digital

La última etapa del procesamiento digital de imágenes visto en la Figura 9 incluye el reconocimiento e interpretación este paso puede ser tan simple como detallar las regiones de una imagen donde se sabe que se ubican información de interés, limitando así la búsqueda que ha de realizarse para hallar tal información

Reconocimiento: Es el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por los descriptores obtenidos en el apartado anterior.

Interpretación: Implica asignar significado a un conjunto de objetos reconocidos.

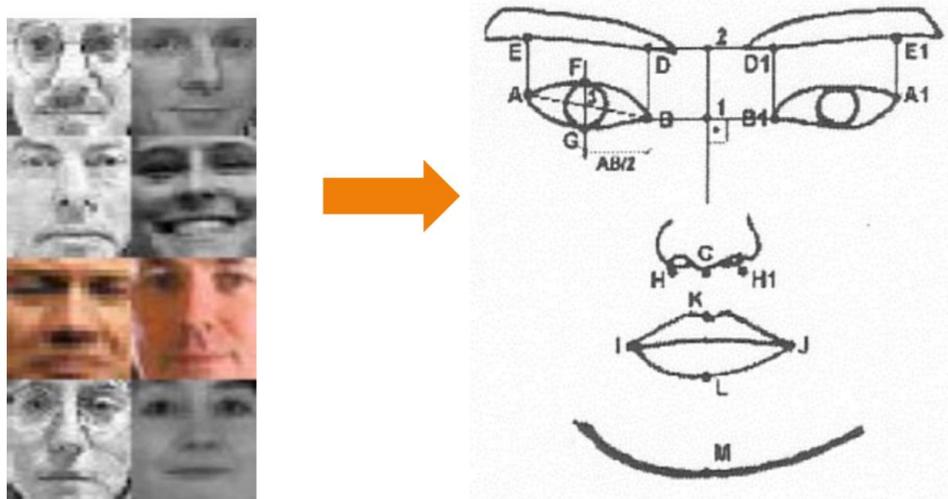


Figura 9: Imagen asignada para el reconocimiento y la interpretación de información

Fuente: Profesorado.us.es/Imagen Digital

1.2.3.6 Software de aplicaciones de visión por computadora

La Visión por computadora consiste en la extracción automatizada de información de las imágenes. Por información se entiende casi cualquier cosa; desde modelos 3D, posición de la cámara, reconocimiento de objetos, y agrupación y búsqueda de contenido. También puede incluir la deformación de las imágenes, la eliminación de ruidos y la realidad aumentada. Muchas veces la visión por computadora trata de imitar a la visión humana, pero otras veces la geometría o un enfoque más estadístico puede ser fundamental para resolver un problema. La Visión por computadora contiene una mezcla de programación, modelado y matemática que lo convierte en un campo de estudio sumamente atractivo. El software es un elemento muy importante en una aplicación de visión por computadora, pues es el encargado de analizar, procesar e identificar características de las imágenes. [23]

1.2.3.7 Microcontroladores

Un microcontrolador es un circuito integrado(CI) programable o circuito electrónico, en cuyo chip se ensambla un microordenador real encargado de realizar diferentes instrucciones de forma secuencial con el objetivo de controlar o automatizar cualquier proceso, se puede configurar la cantidad específica de memorias y dispositivo auxiliar, estos dispositivos tienen los mismos bloques de funcionamiento que una computadora básica, por eso “Algirdas Baskys” lo denomino como una computadora de un solo chip de bajo costo.[34]

Los microcontroladores están presentes en la mayoría de los bienes de consumo, se puede encontrar en los microcontroladores en la mayoría de los dispositivos que controlan, miden, calculan o muestran información, son diseñados para reducir costos y consumo de energía, se utilizan como soluciones simples con respecto a la lógica transistor-transistor, se las abrevia con las siguientes letras MCU provenientes de las siglas en inglés “microcontroller unit”. A pesar del complejo dispositivo, el principio de funcionamiento del microcontrolador es muy simple porque se basa en un principio de funcionamiento analógico, es decir que pueden tomar cualquier valor de la señal de entrada dentro de cierto rango. [35]

Un microcontrolador está diseñado para ser dependientes del control y recolección de datos, está integrado por varios componentes electrónicos para que puedan lograr muchos procesos diferentes por sí mismo, contiene varios componentes adicionales como microprocesador, temporizadores, conversores A/D(Análogo/Digital) y D/A(Digital/Análogo), referencias de voltaje, generadores PWM, interfaces seriales UART y USB. Se clasifican por el ancho del bus de datos de la CPU (4, 8, 16 y 32 bits), por sus funciones (universal y especializado) y por arquitectura del sistema informático como: CISC (computadoras con un conjunto de instrucciones complejas) y RISC (computadoras con un conjunto de instrucciones reducidas). El rendimiento y el consumo de energía del microcontrolador aumentan al aumentar la frecuencia del reloj. La frecuencia del reloj o velocidad del bus, determina cuántos cálculos se pueden realizar por unidad de tiempo. [34], [35]

Los dispositivos integrados más comunes en microcontroladores son: dispositivos de memoria, puertos de Entrada/Salida (E/S), interfaces de comunicación,

temporizadores y relojes del sistema. Los dispositivos de memoria incluyen RAM (memoria de acceso aleatorio), ROM (memoria solo de lectura), EPROM (ROM flash), EEPROM (ROM flash eléctrica). Los temporizadores incluyen tanto relojes en tiempo real como temporizadores de interrupción. Las herramientas de entrada/salida incluyen puertos de comunicación serie, puertos paralelos, convertidor analógico/digital y digital/analógico, LCD (controladores de pantalla de cristal líquido) o VFD (controladores de pantalla fluorescente de vacío). Los dispositivos integrados son altamente confiables ya que no requieren ningún circuito eléctrico externo. Los elementos básicos de un microcontrolador se pueden observar en la ilustración 10 y son las siguientes:[34]

- Unidad de proceso.
- Memoria de datos.
- Puertos de Entradas y Salidas.

En un microcomputador, estos elementos están separados como circuitos integrados, están conectados mediante un bus de datos en un circuito impreso y son controlados por un microprocesador (CPU proveniente de las siglas en inglés Central Processing Unit), que es considerado como el elemento más importante porque se encarga de almacenar, procesar y controlar todas las instrucciones o información recibida. El funcionamiento del microcontrolador y su periferia se lleva a cabo de acuerdo con un programa que se graba en la memoria interna y puede almacenarse en dicha memoria durante un período suficientemente largo. El sistema puede ser diseñado para adaptarse a aplicaciones particulares, según el tipo de CPU, capacidad de memoria y la selección de unidades de entrada/salida para el control de elementos externos, lectura de señales y comunicación entre otros microcontroladores

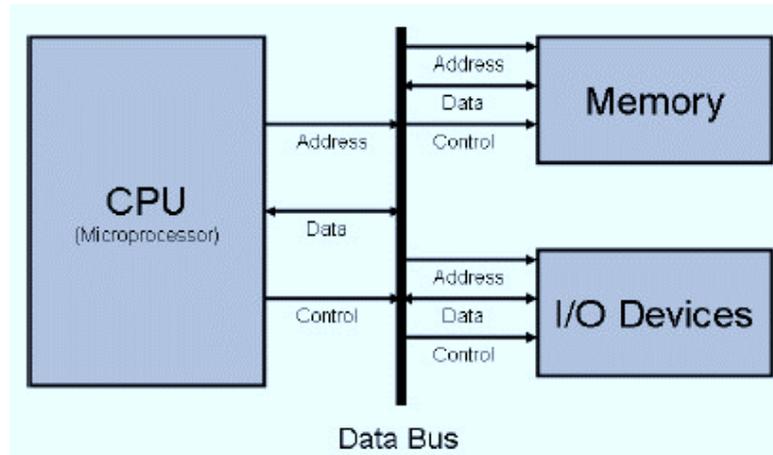


Figura 10: Elementos Básicos de un microcontrolador

Fuente: Interfacing PIC

Los microcontroladores actualmente tienen muchas aplicaciones, desde automatización industrial hasta electrodomésticos, desde la gestión de estaciones nucleares hasta juguetes para niños, desde sistemas militares secretos hasta cambio de canales en su radio. La invención y el desarrollo posterior de los microcontroladores han revolucionado la electrónica digital. No solo los circuitos y la base del elemento han cambiado, sino también los principios de los sistemas de construcción. El ciclo de desarrollo ha sufrido cambios significativos. Han aparecido clases enteras de dispositivos, cuya existencia hubiera sido imposible sin controladores.[35]

La separación de software y hardware permitió unificar la base del elemento. Se puede usar el mismo controlador para crear muchos dispositivos diferentes. La unificación conduce a menores costos de producción. Hay una gran variedad de dispositivos en esta clase. Todos estos dispositivos se pueden dividir en los siguientes tipos principales: [35]

- Microcontroladores integrados de 8 bits
- Microcontroladores de 16 y 32 bits
- Procesadores de señal digital

1.2.3.9 SENSORES

Los sensores son dispositivos, módulos, máquinas o subsistemas cuyo propósito es detectar eventos o cambios en su entorno y enviar la información a otros dispositivos

electrónicos. Los sensores están en todas partes, uno de los primeros sensores conocido por la humanidad es la brújula, es la herramienta de navegación más antigua, este fue creado entre los tiempos romanos, la composición básica es de material magnético, una aguja fina que siempre rota, es considerada como un sensor porque detecta campos magnéticos muy leves, en la figura 9 muestra de manera gráfica cómo funciona la brújula.

Los sensores varían en precisión, linealidad, resolución, respuesta de frecuencia, respuesta de ruido, impedancia de entrada y salida, y muchos otros parámetros. Según la estructura de construcción y según el método de conexión de los elementos, los sensores se pueden convertir secuencialmente, diferencial y compensación. Juegan un papel importante en la sociedad por eso deben cumplir con ciertas propiedades que son indispensables para una correcta medición las cuales son: repetitividad, estabilidad, sensibilidad y resolución, estas propiedades son descritas a continuación.[36]

1.2.3.10 Clasificación de sensores

Los sensores utilizados son muy diversos en la actualidad y se pueden clasificar según varios criterios que son: [37]

- Según el principio de funcionamiento.
- Según el tipo de señal de salida.
- Según el tipo de parámetro.
- Según el tipo de variable física.

Existe una variedad de sensores en la figura 11 se muestra algunos de los más conocidos que son: de temperatura, luz, distancia, color, humedad, velocidad, sonido, ópticos y magnéticos a continuación se muestra cada clasificación. [37]

Tabla 2: Ventajas y Desventajas de Cámaras

Fuente: Digital Video. Alameda. United States. [38]. Elaborado por el Autor

Ventajas	Desventajas
<ul style="list-style-type: none">• Rango dinámico más alto, lo que lo hace mejor para capturar detalles en blancos y negros.• Es ligera.• Tiene una resolución alta.• Su almacenamiento.• Filtros incorporados.	<ul style="list-style-type: none">• Costo.• Enfoque.• Duración de batería.

1.2.3.12 Entorno De Desarrollo Integrado (IDE)

Bibliotecas y Suite de Desarrollo Python

Este es un potente entorno de desarrollo interactivo para el lenguaje Python, posee funciones avanzadas de edición, pruebas interactivas, depuración e introspección y un entorno informático numérico. Gracias al soporte de IPython (intérprete interactivo mejorado de Python) y bibliotecas populares de Python como NumPy, Spyder, SciPy o matplotlib (trazado interactivo 2D / 3D), también se puede usar como una biblioteca que proporciona widgets potentes relacionados con la consola para nuestras aplicaciones basadas en PyQt. Se puede utilizar para integrar una consola de depuración directamente en el diseño de su interfaz gráfica de usuario.

Este IDE podrá ser instalado en diferentes sistemas Gnu/Linux y Windows, como muestran en la documentación oficial del proyecto. En este proyecto la instalación se lo realizará sobre Ubuntu 17.10. Para un correcto funcionamiento será necesario que cumplamos ciertos requisitos necesarios. Se pueden consultar estas dependencias en la sección de requisitos, en la que se detallan que otros paquetes son necesarios para realizar una instalación correcta. Una vez cumplidas las dependencias la instalación se iniciará abriendo la terminal. [27]

1.2.3.13 Comparación sistema de visión artificial y humano

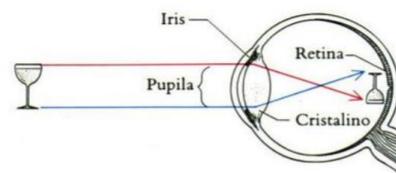
A continuación, se realiza una comparación entre los sistemas de visión Artificial y un sistema de visión humano.

Fuente de Luz: El ser humano necesita una cantidad mínima de iluminación para identificar objetos a su alrededor al igual que los sistemas de visión por computadora.

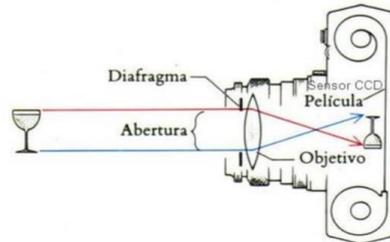
Sensores de imagen: El sensor de imagen es el elemento de una cámara electrónica, tanto de vídeo como de fotografía estática, que detecta y captura la información que compone la imagen, el trabajo realizado por el iris, las pupilas, la retina, y demás en el caso de las personas son emulados mediante la utilización de una cámara que tiene lente, sensores, y demás componentes en el caso de las cámaras como muestra en la Figura 12.

Sistemas de Adquisición: El sistema de adquisición consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otros dispositivos electrónicos (sistema digital). En el caso del ser humano posee un nervio óptico, que se encarga de llevar la información brindada por los conos y los bastones hacia el cerebro en forma de pulsos eléctricos, en el caso de los sistemas de visión artificial se encargan de transformar la información eléctrica de naturaleza analógica en una señal digital para su posterior procesamiento.

Procesado de la información: Es la acumulación y manipulación de elementos de datos para producir información significativa en el caso del ser humano el cerebro es el encargado del manejo de la información y su percepción, a diferencia de los SVA (Sistema de Visión Artificial) las computadoras con sus potentes procesadores emulan la tarea realizada por el cerebro humano. [10]



Ojo Humano



Cámara Fotográfica

Figura 12: Comparativa entre el ojo humano y una cámara fotográfica

Fuente: E. Hannula a.l. 2005

1.2.3.14 Aplicaciones de la visión por computadora

Son varias las áreas en las cuales se puede obtener grandes beneficios de los SVA, los cuales se aplican en la industria, el automovilismo, análisis biomédico, farmacéutico, electrónico, alimenticio, seguridad, robótica, etc. Además, varias son las tareas que pueden realizarse desde una inspección visual a ciertos procesos hasta complejos manejos de brazos robóticos en un sistema de control automatizado. [11]

1.2.3.15 Ventajas de los sistemas de visión artificial

Los Sistemas de Visión Artificial (SVA) resultan mejores al momento de realizar evaluaciones en magnitudes físicas, y no hay ningún inconveniente que se pudiera presentar debido a las denominadas ilusiones ópticas o en el entorno que rodea al objeto.

No necesita un contacto físico con el producto para efectuar sus mediciones, esto es importante en el manejo de objetos delicados o potencialmente nocivos para el ser humano.

Los SVA permiten una verificación de lugares inaccesibles para los operarios por ejemplo la detección de objetos mediante una cámara infrarroja, lo cual es imperceptible a la vista del ojo humano.

Un Sistema de Visión Artificial (SVA) puede mantenerse activo sin ningún inconveniente las veinticuatro horas del día en tareas de control de calidad, sin afectar la velocidad de la producción como es el caso de un ser humano en el que hay desgaste físico o agotamiento. [12]

Desventajas de los sistemas de visión artificial

Una de las principales desventajas es que los SVA no tienen la capacidad de un ser humano en la toma de decisiones de forma aleatoria de acuerdo con su experiencia, este tipo de situaciones se podrían prevenir si se toma en cuenta todas las situaciones posibles de ocurrencia.

Los SVA a diferencia del ser humano no pueden ser capaces de entender imágenes muy complejas y basados en una experiencia previa tomar la decisión más adecuada. [12]

Adquisición de la imagen Cámara

Adquisición de la imagen Cámara es la extracción de información avanzada de diferentes entornos y procesos, en forma de características o parámetros relacionados con la imagen se logra con un sensor el cual es el corazón de la cámara digital, el cual se clasifican de acuerdo con el tipo de sensor que se emplea, y esta se divide en dos grupos: cámara de dispositivos de carga acoplada y cámaras de semiconductor. [13]

1.2.3.16 Tipos de software de Inteligencia Artificial (IA)

El software de inteligencia artificial es un espacio muy general, con varias subcategorías diferentes, que incluyen plataformas de IA, chatbots, aprendizaje profundo y aprendizaje automático. [25]

Plataformas IA: Para desarrolladores que intentan construir sus propias aplicaciones inteligentes, las plataformas Inteligencia Artificial (IA) son la solución ideal. Al igual que una plataforma de aplicación estándar, estas herramientas a menudo proporcionan la funcionalidad de arrastrar y soltar con algoritmos

precompilados y marcos de código para ayudar a construir la aplicación desde cero. [25]

Chatbots: Un chatbot es una tecnología que permite que un usuario pueda mantener una conversación con un programa informático, habitualmente dentro de una app de mensajería. Los chatbots se utilizan a menudo como la primera línea de defensa para los centros de atención telefónica o los agentes de servicio al cliente de chat en vivo. Al utilizar un chatbot para determinar la gravedad de una solicitud o el motivo de la interacción, las empresas pueden dirigir mejor a los clientes o posibles clientes. [25]

Aprendizaje profundo: El aprendizaje profundo trata del uso de redes neuronales para mejorar cosas tales como el reconocimiento de voz, la visión por ordenador y el procesamiento del lenguaje natural, los algoritmos de aprendizaje profundo difieren de los algoritmos de aprendizaje automático específicamente porque usan redes neuronales artificiales para hacer sus predicciones y decisiones, y no necesariamente requieren entrenamiento humano. Con las redes neuronales artificiales, los algoritmos elaborados pueden tomar decisiones de una manera similar a la del cerebro humano. Sin embargo, las decisiones se toman en una escala más pequeña porque actualmente es imposible replicar la cantidad de conexiones neuronales en el cerebro humano. El aprendizaje profundo se puede dividir en las subcategorías de reconocimiento de imágenes (visión artificial), procesamiento de lenguaje natural (PNL) y reconocimiento de voz. [25]

Aprendizaje automático: La categoría de algoritmo de aprendizaje automático consiste en una amplia gama de bibliotecas y marcos que pueden realizar una variedad de tareas de aprendizaje automático cuando se implementan correctamente. Cuando se integran en el software, estos algoritmos predominantemente de código abierto permiten que las aplicaciones tomen decisiones y predicciones basadas completamente en datos. Estos algoritmos aprenden, a menudo utilizando aprendizaje supervisado o reforzado, en función de los conjuntos de datos que se les presentan para el consumo. [25]

Actuadores externos

Actuadores externos son los equipos o dispositivos que reciben la información del computador y cumplen la función de actuadores dentro del proceso o aplicación. Por

lo general los sistemas de visión artificial emplean microcontroladores en sus interfaces siendo estos un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. [23]

Detección de Bordes

Detección de Bordes es una herramienta fundamental en el procesamiento de imágenes y en visión por computadora, particularmente en las áreas de detección y extracción de características, que tiene como objetivo la identificación de puntos en una imagen digital en la que el brillo de la imagen cambia drásticamente o, más formalmente, tiene discontinuidades, el mismo problema de encontrar discontinuidades en señales 1D se conoce como paso. Los bordes extraídos de imágenes 2D a partir de escenas 3D se pueden clasificar según el punto de vista en: dependientes o independientes. El primero refleja propiedades inherentes de objetos tridimensionales, tales como marcas y formas en las superficies de estos. El segundo puede cambiar si se cambia el punto de vista, y refleja la geometría de la escena, tales como objetos que se tapan unos a otros. [18]

Segmentación

Segmentación es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos, el objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen, la segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. El resultado de la segmentación de una imagen es un conjunto de segmentos que cubren en conjunto a toda la imagen, o un conjunto de las curvas de nivel extraídas de dicha imagen, cada uno de los píxeles de una región son similares en alguna característica, como el color, la intensidad o la textura.[19]

Extracción de Caracteres

Extracción de caracteres es comúnmente conocido como Carácter de Reconocimiento Óptico (OCR), es texto simultáneamente leído de manera artificial, las aplicaciones e industrias comunes incluyen fecha/rastreo de lote en diversos tipos de empaques, lectura de números de serie en aplicaciones automotrices o electrónicas, etc. Este tipo de tecnología ha sido utilizada de manera extensa en aplicaciones comerciales desde los años 70, y actualmente es utilizada en la automatización de tareas tales como procesamiento de diversos tipos de documentación, empaques consumibles y aplicaciones en la medicina, los lectores y el software pueden ser utilizados, así como las cámaras inteligentes y sistemas de visión, los cuales tienen capacidades adicionales como lectura de código de barras e inspección de producto. [20]

1.2.3.17 Ejercicio Físico con peso muerto

El ejercicio físico de peso muerto es un ejercicio compuesto en él que están implicados diferentes grupos musculares de forma simultánea: glúteos, isquiotibiales y espalda baja. Dicho ejercicio es uno de los entrenamientos más exigentes que se dan en las salas de pesas, pero altamente efectivo si se realiza adecuadamente, es uno de los ejercicios más practicados en los gimnasios que trabaja todo el cuerpo desde el tren inferior pasando por la zona central hasta el tren superior viéndolo así se habla de un ejercicio muy completo, pero al mismo tiempo de un ejercicio complejo, consiste en levantar la barra del suelo hasta que los hombros queden detrás de la barra. [28]

1.2.3.18 Posición de los pies, la barra y el agarre.

La posición de los pies, la barra y el agarre consiste en la posición y el peso de la barra que van a actuar de manera vertical contra el suelo con el agarre firme. Por lo tanto, el primer punto para tener en cuenta es que el recorrido de la barra sea vertical en todo momento, evitando desviaciones. Si la barra se mueve hacia delante o hacia atrás (de manera horizontal), el deportista o levantador va a tener que gastar energía en volver a encontrar el equilibrio, posicionar su centro de masas y buscar el trayecto correcto de la barra, el equilibrio se encuentra al colocar gran parte del cuerpo detrás

de la barra, y con el medio pie como punto de referencia. De esta manera, posicionando la barra en el medio pie y dibujando una línea vertical desde ese punto, daremos con el trayecto óptimo. Sabiendo esto, la barra se situará a 2-3cm de las espinillas, que coincidirá con el medio pie. La separación entre las piernas dependerá, a su vez, de la anchura de la cadera, así como de la altura del deportista, pero será cercana a 25-30 cm entre los talones (una posición cómoda, como si fuéramos a realizar un salto vertical). Las puntas de los pies apuntarán hacia afuera, 10° más o menos, y las rodillas estarán ligeramente rotadas externamente, para así reclutar la musculatura aductora y rotadora externa en el movimiento. Esta posición también ayudará a que mejore la participación de la musculatura extensora.

Una vez se han situado los pies en la correcta posición, entra en juego el agarre, cuanto más abiertas se tenga las manos, el ROM (rango de recorrido) será mayor, por tanto, se deberá buscar una posición cómoda y cerca de las piernas. El agarre será doble prono, para así evitar descompensaciones que puede acarrear el agarre mixto, pudiendo llegar a provocar problemas en el tendón bicipital del brazo en posición de supinación (palmas de las manos mirando hacia el techo) (Beggs, 2011). Por tanto, el agarre mixto sería interesante cuando se vayan a realizar levantamientos muy pesados. Siguiendo con esto, el agarre es un punto muy importante en el peso muerto, ya que la espalda no va a levantar lo que las manos no pueden agarrar. [22]

1.2.3.19 Posición de la espalda.

Posición de la espalda es la posición anatómica, sin forzar situaciones de extensión o flexión, por una parte está el problema de redondear la zona lumbar, pero la hiperextensión de la zona puede ser igual de peligrosa, es muy importante que la mirada se dirija al suelo en diagonal, en dicha posición las fuerzas se generan principalmente en las extensiones de rodilla y cadera, logrando que la fuerza se transmita por la columna hasta las escápulas, los brazos, las manos y así, finalmente, llegar a la barra. De este modo, habrá muchísima musculatura implicada para mantener rígida la columna y para situar las escápulas y hombros en su sitio.

En la Figura 13 muestra la posición de los trapecios y romboides, dos músculos importantes que actuarán de forma isométrica manteniendo las escápulas abiertas y los hombros en una posición neutral. [22]



Figura 13: Correcta posición en levantamiento de potencia

Fuente: Halterofilia y levantamiento de pesas

En la Figura 14 el ejercicio compuesto en los cuales se involucran casi todos los músculos del cuerpo como son los de las piernas, dependiendo de la posición de los pies será en el área que se trabajará cuádriceps, femoral, espalda y brazos.

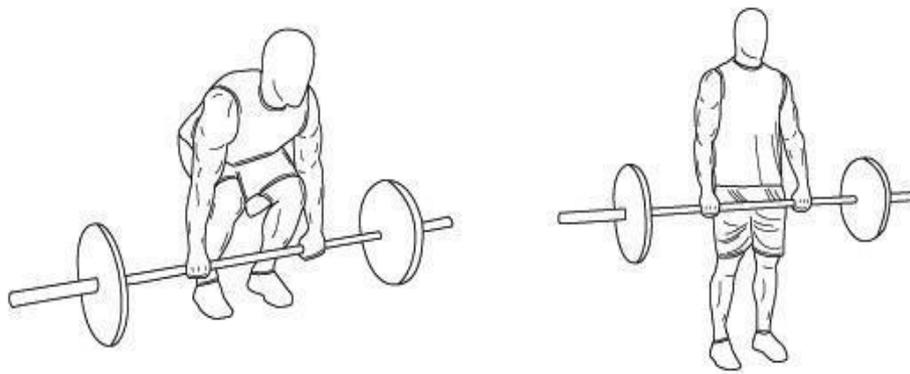


Figura 14: Ejecución de levantamiento de potencia

Fuente: Halterofilia y levantamiento de pesas

Para la correcta ejecución de esta rutina, las recomendaciones se describen a continuación

- De pie, separar los pies de acuerdo con la medida de los hombros de cada individuo
- Colocar una pesa en la parte exterior de cada extremo de la barra
- Inclinarsse y agarrar firmemente la barra con pesas

- Mantener las piernas tiasas, espalda recta, y la cabeza erguida con la vista diagonal
- Incorporarse con los codos tiosos
- Bajar la espalda al suelo con las piernas tiasas
- Inspirar hacia arriba, espirar hacia abajo



Figura 15: Pasos para Ejecución de levantamiento de potencia

Fuente: Halterofilia y levantamiento de pesas. [29]

Pasos para realizar correctamente el Peso Muerto

- La barra tiene que estar pegada a las tibias.
- El peso debe recaer sobre los talones.



- Tiene que seguir este orden: cabeza, cadera, rodilla y tobillo. Es frecuente cometer errores como poner la cadera por debajo de las rodillas o la cabeza por debajo de las caderas.



- Activar los glúteos y estirar dinámicamente el piriforme y flexores de cadera antes de realizar el movimiento.



- La espalda debe estar completamente recta con el glúteo hacia atrás y la cadera encajada, manteniendo tensión en la columna vertebral.



- Sacar pecho y meter las escápulas hacia atrás. [28]



Figura 16: Ejecución de levantamiento de potencia, posición de cabeza

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Durante la ejecución aumentar la presión intraabdominal durante todo el recorrido. Para ello debemos de coger aire fuertemente y endurecer el abdomen hacia fuera durante todo el recorrido sin soltarlo.

Mantener la columna en posición neutral. Pensar en llevar las escápulas a los bolsillos de atrás del pantalón. Mantener la proximidad de la barra durante todo el recorrido para que se pueda manejar mejor el peso (nos mantenemos más compactos) y finalmente empujar el suelo con las piernas (no intentar levantar la barra con la espalda). [28]

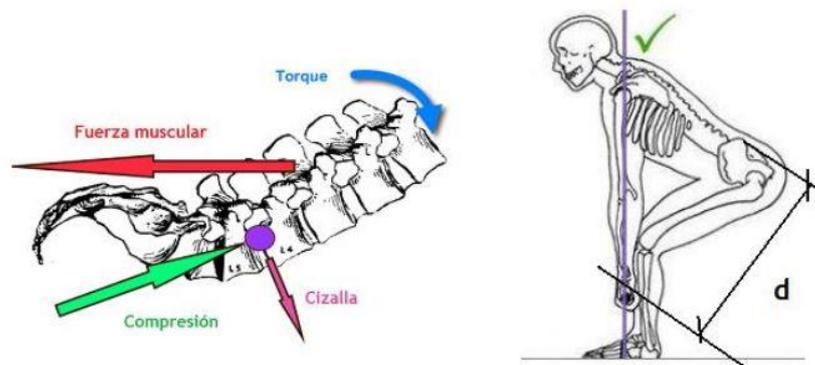


Figura 17: Ejecución de levantamiento de potencia, posición de columna

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Aplicación deportiva

Se respalda la idea de que en los programas eficaces de entrenamiento de fuerza para deportes en los que los movimientos sagitales en la articulación de la rodilla y/o salto vertical sean dominantes (fútbol, balonmano, vóley, carreras de medio-fondo, además de los dominantes de cadera) como muestra en la Figura 18 dicho ejercicio es clave para el fortalecimiento de los músculos que implican en la mayoría de los deportes, el peso muerto es seguramente uno de los ejercicios más eficientes, menos tiempo de entrenamiento, mayores resultados, el peso muerto figura como un ejercicio en la rutina de entrenamiento de numerosos deportistas y/o atletas. [28]

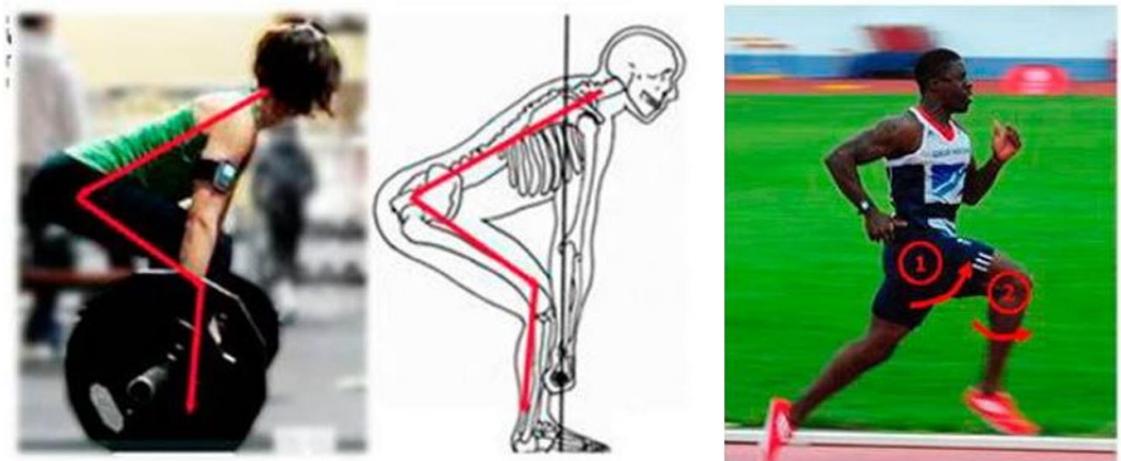


Figura 18: Aplicación del levantamiento de potencia en el deporte

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Frecuencia Cardíaca

Como se ve en la Figura 19, los esfuerzos percibidos y medidos de la Frecuencia Cardíaca, se demostró una diferencia significativa de aumento entre realizarlo con una postura incorrecta frente a no hacerlo sólo después de corregir la postura. Probablemente, se relacione con aumento de la demanda cardiovascular como consecuencia de la mayor carga aplicada al eliminar la limitación de la fuerza de agarre. Consecuentemente, esto permitió generar más trabajo a través de grandes grupos musculares y aumentar más el gasto cardíaco.

La escala de esfuerzo general percibido fue mayor al realizarlas con postura correcta sólo después de las dos primeras series, observándose un 16% más de fatiga

percibida en la región lumbar en estos casos. Por otro lado, la correcta postura permitió que la fatiga percibida en los músculos del antebrazo fuera un 45% más bajo. [28]

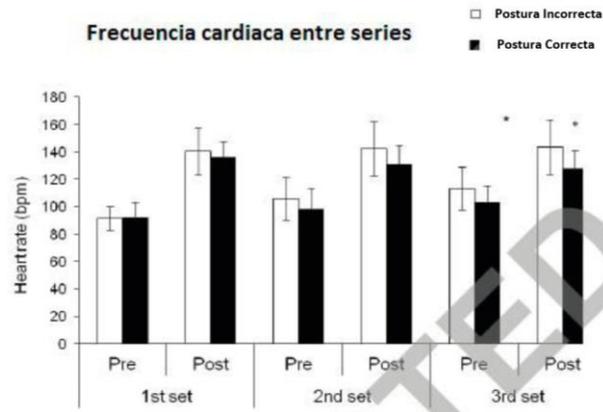


Figura 19: Frecuencia Cardiaca en Levantamiento de Potencia

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Recomendaciones para realizar el peso muerto

Las proporciones de longitud de brazos y tronco en relación con la altura de cada persona también son buenas indicadoras de qué tipo de peso muerto puede ser más ventajoso para cada uno. Las siguientes dos tablas se muestran estas relaciones de manera detallada. [28]

Tabla 3: Proporción del torso y brazos en relación con la altura de la persona

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

	Corto(s)	Media	Largo(s)
Torso	<47%	47%	>47%
Brazos	<38%	38%	>38%

Tabla 4: Recomendaciones de peso muerto sumo o convencional en relación con la tabla anterior

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

		Torso		
		Corto	Media	Largo
Brazos	Cortos	Cualquiera	Sumo	Sumo
	Media	Convencional	Cualquiera	Sumo
	Largos	Convencional	Convencional	Cualquiera

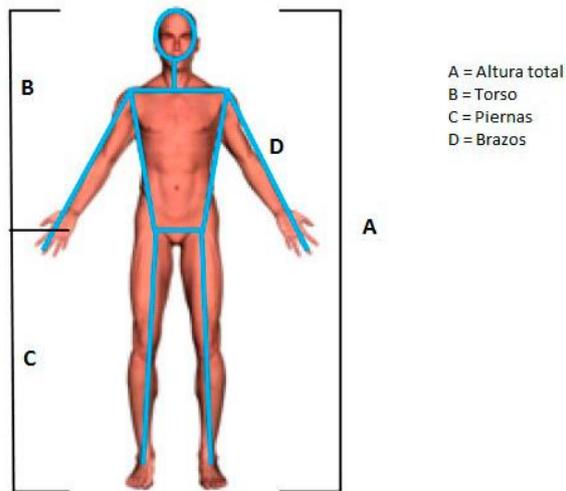


Figura 20: Referencia para personas que ejecutan Levantamiento de Potencia

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Las personas con brazos cortos en relación con la altura total tienen que inclinarse más para agarrar la barra lo que genera un mayor brazo de palanca en el levantamiento y una tendencia hacia la curvatura lumbar a medida que ascienden en el movimiento. [28]

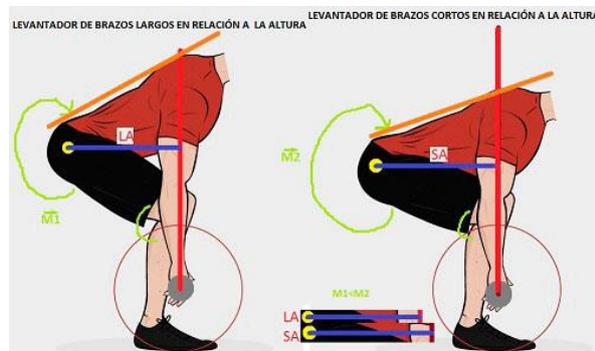


Figura 21: Referencia para personas que ejecutan Levantamiento de Potencia

Fuente: HSN Blog Nutrición, Salud y Deporte. [29]

Funcionamiento del Algoritmo explicado Matemáticamente

El funcionamiento general predice lo que está mal con las estimaciones actuales y corrige de forma iterativa en lugar de predecir directamente los resultados de una sola vez, utilizando un modelo de autocorrección que cambia progresivamente una

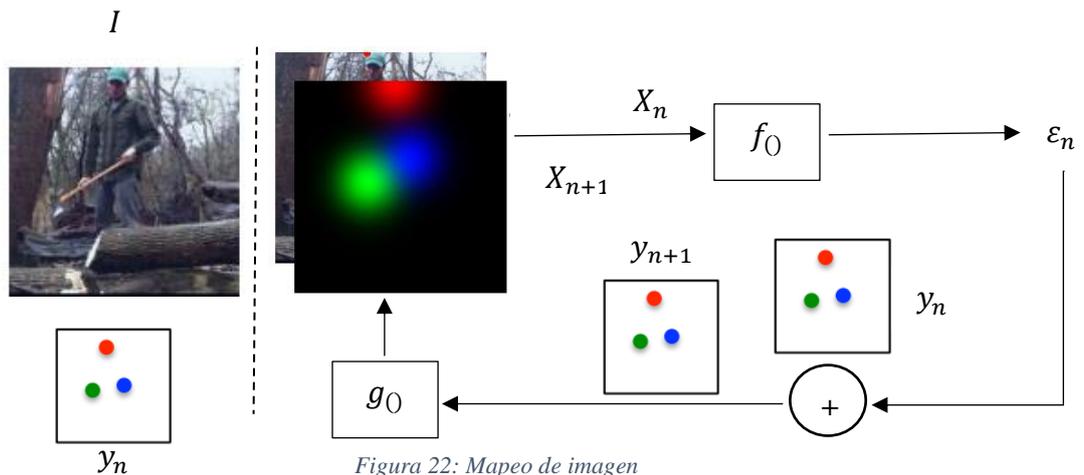
solución inicial al retroalimentar las predicciones de errores, y este proceso se denomina retroalimentación de errores iterativos (IEF)

Canalización del modelo

La entrada consta de la imagen I y una representación de la salida anterior y_{n-1} . Hay que tener en cuenta que este es un proceso iterativo y el mismo resultado se repite en los pasos del procesamiento de imagen.

Entrada, $(X_n) = y_0 + G(y_{n-1})$ donde y_0 es la imagen de y_{n-1} es la salida anterior $F(y_n)$ emite la corrección ε_n y esto sumado a la salida actual y_n para generar y_{n+1} , que tiene en cuenta las correcciones

$g(y_{n-1})$ convierte cada punto clave en y_{n+1} en un canal de mapa de calor para que se puedan apilar en la imagen y_0 , para formar la entrada para la próxima iteración, este proceso se repite (n) veces hasta que se consigue un refinado y_{n+1} y se acerca a la verdad fundamental mediante la adición de ε_n . [40]



Fuente: Convolutional Neuronal Networks. [40]

MAPEO DE IMAGEN MATEMÁTICAMENTE REPRESENTADO

$$\varepsilon_n = F(X_n)$$

$$y_{n+1} = y_n + \varepsilon_n$$

$$X_{n+1} = y_0 + g(y_{n+1})$$

- $F()$ y $g()$ son aprendibles y $F()$ es una CNN (Red Neuronal Convolutacional).

- Un punto importante para tener en cuenta es que como $F()$ es una Red Convolutiva toma $(y_0) + g(y_n)$ como entradas, tiene la capacidad de aprender características sobre el espacio conjunto de entrada y salida, lo cual es bastante bueno.
- Los parámetros Θ_g y Θ_F aprenden optimizando la siguiente ecuación

$$\min_{\Theta_g, \Theta_F} \sum_{n=1}^N h(\varepsilon_n, mi(y, y_n))$$

Donde ε_n y $mi(y, y_n)$ son todas las correcciones previstas y objetivos respectivamente. La función h es una medida de distancia, como una pérdida cuadrática N es el número de pasos de corrección realizados por el modelo.

Para la comprensión de la arquitectura de la CNN se describe a continuación la capa convolutiva del proceso. [40]

Este proceso es la forma de filtrar características necesarias para el procesamiento de la imagen, para lo cual se utiliza la matriz de convolución llamada matriz de Kernell, esta matriz (Tabla 5) es el filtro que se va a utilizar en la imagen, hay un sinnúmero de filtros ya sea para bordes, contraste y sombras. Cuando se tiene una imagen de un solo color es decir en escala de grises de 16 x 16 de ancho y alto, significa que utilizara 256 neuronas, pero si fuese una imagen a color necesita 3 canales RGB (Red, Green, Blue) es decir se utilizaría 16 x 16 x 3 lo que son 768 neuronas, estas neuronas forman nuestra capa de entrada como se observa en el ejemplo de la tabla 6, los colores varían de 0 a 255, pero en CNN (Convolutiva Neuronal Network) se debe transformar de 0 a 1 para que cada color deba realizar la convolución con la matriz de Kernel ya que nuestro modelo es en 3D. [40]

$G[3,3] =$	-1	0	-1
	-1	0	1
	-1	0	1

Tabla 5: Matriz de Kernell

Fuente: Realizado por el Autor

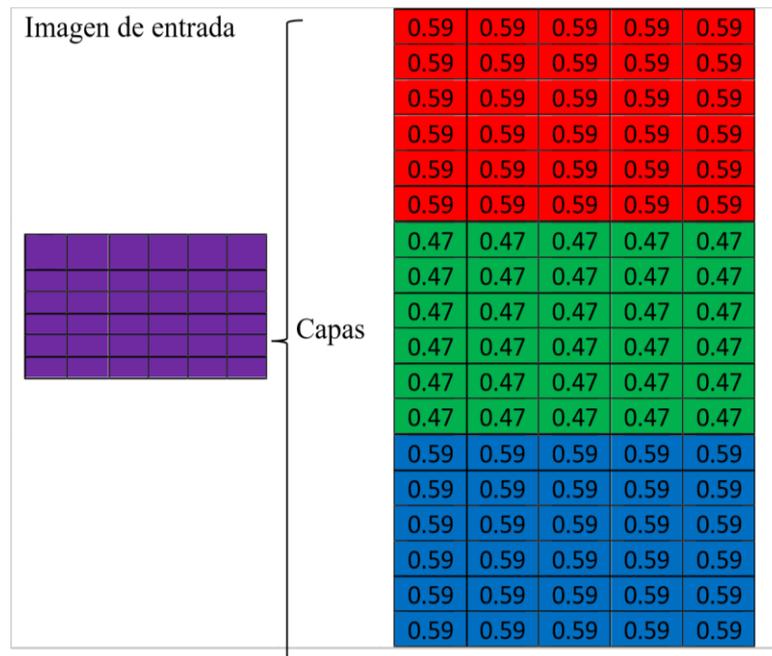


Tabla 6: Matriz de entrada

Fuente: Convolutional Neural Network.[40]

Realizado por el Autor

Cálculo de la convolución en señales discretas (SD) la solución es la sumatoria de la convolución entre la imagen de entrada y la matriz de Kernell.

Para una explicación de una forma rápida a medida que se desliza el Kernel y se obtiene una nueva imagen filtrada por el kernel. En esta primera convolución y siguiendo con el ejemplo anterior, es como si obtuviera 32 imágenes filtradas nuevas. Estas imágenes nuevas lo que están dibujando son ciertas características de la imagen original. Esto ayudará en el futuro a poder distinguir un objeto de otro en la Figura 23 muestra el proceso del Kernel realizando el producto matricial con la imagen de entrada y desplazando a 1 píxel de izquierda a derecha y de arriba abajo va generando una nueva matriz para finalmente obtener un mapa de detección de características

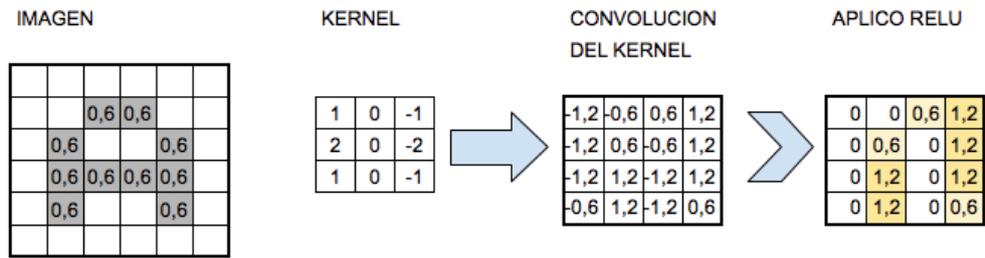


Figura 23: Pasos para obtener el mapa de detección de características

Fuente: Convolutional Neural Network [40]

$$f(n)*g(n) = \sum_{k=-\infty}^{\infty} f(k)g(n - k) \quad (1)$$

$$F[6,6] * G[3,3]$$

			0.59	0.59	0.59
		0.47	0.47	0.47	0.47
0.59	0.59	0.59	0.59	0.59	0.59
0.59	0.59	0.59	0.59	0.59	0.59
0.59	0.59	0.59	0.59	0.59	0.59
0.59	0.59	0.59	0.59	0.59	0.59
0.59	0.59	0.59	0.59	0.59	0.59

-1	0	-1
-1	0	1
-1	0	1

Tabla 7: Convolución del Kernel

Fuente: Convolutional Neural Network [40]

Elaborado por el Autor

$$\begin{aligned}
H[1,1] &= F[1,1] * H[1,1] + F[1,2] * H[1,2] + F[1,3] * H[1,3] + F[2,1] \\
&\quad * H[2,1] + F[2,2] * H[2,2] + F[2,3] * H[2,3] + F[3,1] \\
&\quad * H[3,1] + F[3,2] * H[3,2] + F[3,3] * H[3,3]
\end{aligned}$$

$$\begin{aligned}
H[1,1] &= (0.59 * -1) + (0.59 * 0) + (0.59 * -1) + (0.59 * -1) + (0.59 \\
&\quad * 0) + (0.59 * 1) + (0.59 * -1) + (0.59 * 0) + (0.59 * 1)
\end{aligned}$$

$$H[1,1] = -0.59 + 0 - 0.59 - 0.59 + 0 + 0.59 - 0.59 + 0 + 0.59$$

$$\mathbf{H[1, 1] = -1, 18 K}$$

$$\begin{aligned}
[1,1] &= F[1,1] * H[1,1] + F[1,2] * H[1,2] + F[1,3] * H[1,3] + F[2,1] \\
&\quad * H[2,1] + F[2,2] * H[2,2] + F[2,3] * H[2,3] + F[3,1] \\
&\quad * H[3,1] + F[3,2] * H[3,2] + F[3,3] * H[3,3]
\end{aligned}$$

$$\begin{aligned}
K[1,1] &= (0.47 * -1) + (0.47 * 0) + (0.47 * -1) + (0.47 * -1) + (0.47 \\
&\quad * 0) + (0.47 * 1) + (0.47 * -1) + (0.47 * 0) + (0.47 * 1)
\end{aligned}$$

$$K[1,1] = -0.47 + 0 - 0.47 - 0.47 + 0 + 0.47 - 0.47 + 0 + 0.47$$

$$\mathbf{K[1, 1] = -0, 94}$$

$$\begin{aligned}
M[1,1] &= F[1,1] * H[1,1] + F[1,2] * H[1,2] + F[1,3] * H[1,3] + F[2,1] \\
&\quad * H[2,1] + F[2,2] * H[2,2] + F[2,3] * H[2,3] + F[3,1] \\
&\quad * H[3,1] + F[3,2] * H[3,2] + F[3,3] * H[3,3]
\end{aligned}$$

$$\begin{aligned}
M[1,1] &= (0.59 * -1) + (0.59 * 0) + (0.59 * -1) + (0.59 * -1) \\
&\quad + (0.59 * 0) + (0.59 * 1) + (0.59 * -1) + (0.59 * 0) \\
&\quad + (0.59 * 1)
\end{aligned}$$

$$M[1,1] = -0.59 + 0 - 0.59 - 0.59 + 0 + 0.59 - 0.59 + 0 + 0.59$$

$$\mathbf{M[1,1] = -1, 18}$$

H[4,4]=	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18
K[4,4]=	-0,94	-0,94	-0,94	-0,94
	-0,94	-0,94	-0,94	-0,94
	-0,94	-0,94	-0,94	-0,94
	-0,94	-0,94	-0,94	-0,94
M[4,4]=	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18
	-1,18	-1,18	-1,18	-1,18

Tabla 8: Resultado de convolución

Fuente: Covolutional Neural Network [40]

Realizado por el Autor

A cada resultado de la convolución se la conoce como mapa de características, la imagen realiza una convolución con un Kernel y aplica la función de activación, en este caso ReLU (Unidad Lineal Rectificada).

En la capa convolucional se debe aplicar la función rectificadora ReLU (Unidad Lineal Rectificada) conocida como la función de activación el cual sirve para aumentar la no linealidad de la red neuronal

$$f(x) = x^+ = \max(0, x) \quad (3)$$

El ReLU iguala todos los valores negativos a 0 y conserva los valores mayores a cero de cada píxel. Al aplicar ReLU a 2 mapas de características de la Tabla 9 se obtiene los resultados de la Tabla 10.

H[4,4]				M[4,4]			
0	0	0	7	0	0	0	0
2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tabla 9: Matriz de convolución

Fuente: Convolutional Neural Network [40]

Realizado por el Autor

0	0	0	7	→	2	0	7
2	0	0	0		2	0	0
0	0	0	0		0	0	0

Tabla 10: Resultado ReLU

Fuente: Convolutional Neural Network [40]

Realizado por el Autor

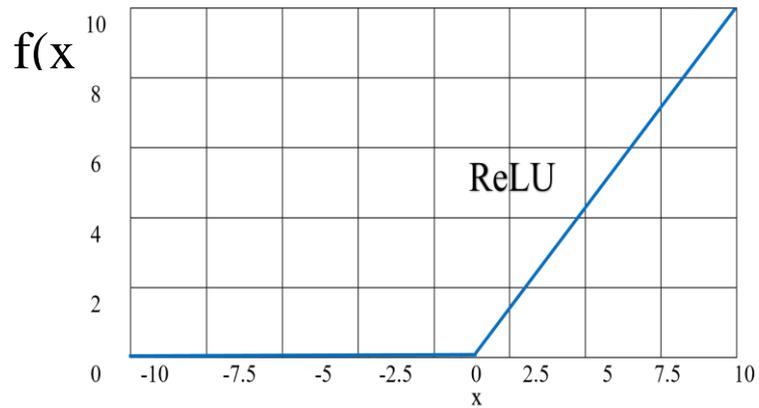


Figura 22: Comportamiento de Linealidad ReLU

Fuente: Convolutional Neural Networks [40]

El ReLU iguala todos los valores negativos a 0 y conserva los valores mayores a cero de cada píxel. Al aplicar ReLU a 2 mapas de características de la Tabla 8 y 9 se obtiene los resultados de la Tabla 10. La explicación de la convolución lleva a la definición de la capa convolucional que es el conjunto de todos los mapas de características. En la capa convolucional se expresa de la siguiente manera con su respectiva representación gráfica.[40]

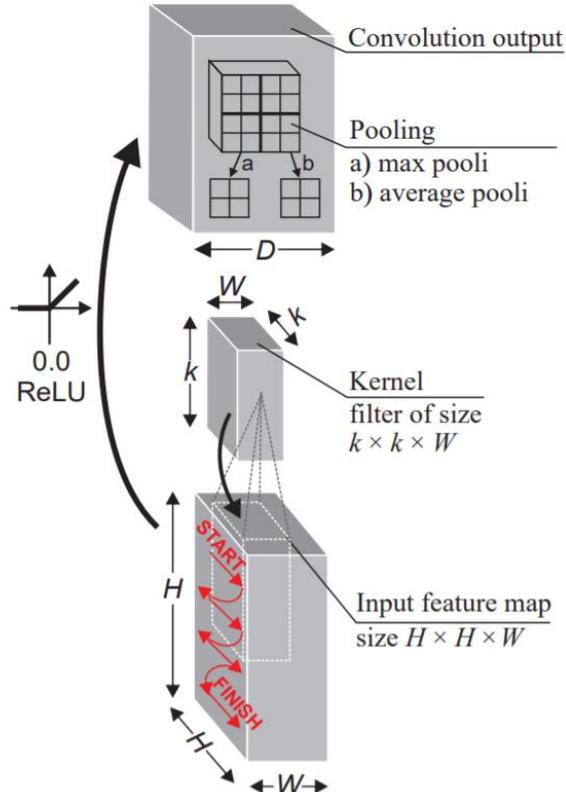


Figura 23: Etapa convolucional neuronal.

Fuente: Deep Convolutional Neural Networks Structure Feature [40]

1.3 OBJETIVOS

OBJETIVO GENERAL

Implementar un sistema electrónico que permita el monitoreo para la detección y corrección de posturas en la realización de ejercicios de levantamiento de potencia mediante la implementación de visión artificial con software libre.

OBJETIVOS ESPECÍFICOS

- Investigar sobre los métodos y técnicas de entrenamiento que permita el buen desempeño de ejercitación física con peso muerto.
- Desarrollar un método basado en visión artificial que permita identificar los patrones de posiciones de extremidades en la ejecución de levantamiento de potencia.
- Diseñar un sistema asistido por computadora que permita realizar la tarea de adquisición de datos mediante el monitoreo de imágenes de personas realizando ejercicio de levantamiento de potencia en gimnasios.
- Realizar pruebas de funcionamiento y validación del sistema electrónico de monitoreo del cuerpo humano durante la práctica del ejercicio físico para la correcta postura presentada y la preestablecida en el sistema.
- Disminuir el porcentaje de lesiones en los deportistas al momento de realizar el levantamiento de peso muerto

CAPITULO II.- METODOLOGIA

2.1. MATERIALES

El trabajo de investigación empezó con la definición del software y con el análisis de todos los antecedentes, requerimientos y documentación. Un análisis pertinente y detallado del problema a resolver, en el caso concreto, el desarrollo de un sistema que sea capaz de detectar la figura humana, procesar la imagen para corregir la postura de la ejecución de ejercicio realizado por los deportistas en los gimnasios en

tiempo real. Así como de las variables que puedan afectar el funcionamiento del dispositivo, y el entorno de desarrollo a utilizar, así como el análisis del dispositivo como el más eficaz y conveniente para la aplicación con ingeniería de Software libre y el uso de hardware apropiado para el procesamiento de la información y poder realizar las pruebas de funcionamiento se empleará un prototipo en el cual estará conectada una cámara la cual tenga todas las características necesarias para poder monitorear al individuo en el entrenamiento. Para el desarrollo del proyecto de investigación, se tendrá en consideración las siguientes fases.

1. Recolección de información acerca de las características de comunicación entre las librerías y Lenguaje libre de programación
2. Análisis la estructura de conexión del hardware y su compatibilidad con el software.
3. Análisis de los métodos y tipos de señales además del filtrado de imágenes que permiten el desarrollo del sistema
4. Análisis del funcionamiento y comparación del desempeño de la tarjeta hardware para el procesamiento de imágenes
5. Especificaciones para el sistema de control del posicionamiento en el ejercicio de levantamiento de potencia.
6. Diseño del sistema de control de monitoreo del individuo mediante visión artificial utilizando software libre
7. Diseño y elaboración del prototipo
8. Realización de pruebas de funcionamiento empleando el control visual para verificar la eficacia del sistema
9. Interpretación de los datos obtenidos en el equipo de medición e identificación para determinar la calidad del sistema
10. Interpretación de los datos obtenidos en el equipo de medición e identificación para determinar la calidad del sistema
11. Realizar pruebas de funcionamiento del prototipo de sistema

Estas fases de desarrollo del sistema electrónico de monitoreo están detalladas en el cronograma de actividades dando fechas para su eventual culminación.

2.2 Métodos

2.2.1 MODALIDAD DE INVESTIGACIÓN

A. Modalidad Aplicada

Debido a que el presente Proyecto es de Investigación Aplicada, las modalidades de la investigación se desarrollaron en gran parte mediante pruebas del sistema, obteniendo diferentes datos y parámetros del funcionamiento lo que permitió implementar la comunicación en forma física, resolviendo así la problemática de la transmisión de datos a través de imágenes.

B. Modalidad Bibliográfica

La investigación tuvo modalidad bibliográfica en la búsqueda de las alternativas para desarrollar el algoritmo para el monitoreo en el ejercicio físico utilizando además software libre, mediante el uso de artículos científicos, libros técnicos, fichas bibliográficas y nemotécnicas.

C. Modalidad de Campo

Se tendrá una investigación experimental debido a que se realizó pruebas necesarias para analizar el correcto funcionamiento del prototipo.

D. Modalidad Experimental

El proyecto de investigación tuvo lugar a ser implementado y posteriormente experimental para recolectar datos que validaran la funcionalidad, será de gran aporte al desempeño deportivo

2.2.2 RECOPIACIÓN DE LA INFORMACIÓN

Para la investigación de campo se utilizó guías de ejercicio normativas del rendimiento físico, observación de ejecución de ejercicios de peso muerto, se efectuó una revisión de varios textos y documentos, revistas, paper's, tesis de grado, folletos y páginas web especializados para el control y monitoreo de objetos utilizando algoritmos para la visión artificial con el lenguaje de programación en

Python para de esta forma analizar los factores posibles que inciden en problemas debido a lesiones corporales por el mal posicionamiento.

Como siguiente paso se realizó un estudio de las ventajas y desventajas que se presentan con respecto al desarrollo de este proyecto, tomando en cuenta cada uno de los datos técnicos obtenidos de libros, revistas y artículos científicos, de este modo se propuso la mejor solución para el proyecto.

CAPITULO III.- RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

El desarrollo del sistema se enfoca en la implementación de un algoritmo basado en programación libre, dicho algoritmo podrá monitorear constantemente al individuo cuando haga ejercicio de levantamiento de peso muerto es decir levantamiento de potencia monitoreando la postura al momento de ejecutar dichos ejercicios con la lectura de patrones que la cámara detecta y se ubica estratégicamente en el cuerpo de la persona monitoreada gracias a la programación del algoritmo y al libre acceso a las herramientas tecnológicas como los elementos electrónicos, la Visión Artificial y el uso de lenguaje de programación son claves para conseguir el propósito de este proyecto de investigación que se basa en definir zonas de interés sobre las que se pueden aplicar el sensor de pulso cardiaco fabricado con el único propósito de aumentar la eficiencia del sistema el cual lanzará un aviso que advierta al individuo que la ejecución del ejercicio realizado es de forma incorrecta si es el caso, cumpliendo con estándares que rigen en estudios de la salud y la actividad física, el resultado es una herramienta electrónica computacional que apoya a los deportistas a estimular de forma correcta los músculos involucrados en el levantamiento de peso muerto, evitando las posibles consecuencias negativas que puedan surgir de una mala práctica.

3.2 Desarrollo de la propuesta

3.2.1 Estudio de Factibilidad

Los detalles del análisis de la factibilidad del presente proyecto de investigación se las describen a continuación, tomando en consideración todo lo necesario para la implementación del prototipo y software.

3.2.1.1 Factibilidad Técnica

El presente proyecto de investigación es factible técnicamente en razón a que los materiales para la construcción de la estructura y dispositivos electrónicos requeridos están disponibles en el mercado nacional e internacional.

3.2.1.2 Factibilidad Económica

Se analizó la factibilidad económica del presente proyecto de investigación y se determinó su accesibilidad debido que los gastos fueron financiados por el investigador.

3.2.1.3 Factibilidad Bibliográfica

La información requerida para este proyecto de investigación se la puede encontrar en variedad de libros, artículos científicos, repositorios, tesis, etc. Por tanto, tiene factibilidad bibliográfica

3.2.2 Requerimientos o especificaciones para el desarrollo del sistema

para dicho proceso se tiene los siguientes requerimientos.

Definición del Software: Corresponde a la visión del producto, a sus aspectos desde el punto de vista final.

Análisis de requerimientos: Implica el entendimiento del dominio del producto a ser desarrollado; esto es funciones, comportamiento y relación con sistemas externos.

Diseño: Es la forma en que la solución se implementará, esto se basa en análisis previos, esto incluye diagramas, algoritmos.

Codificación: Corresponde a la implementación de la solución de acuerdo a cómo se ha estipulado durante el diseño.

Pruebas: Se debe asegurar que el producto satisfaga los requerimientos, es decir

que cumplan con el comportamiento esperado. Esto se hace con el fin de asegurar la calidad del sistema electrónico y reducir los riesgos de falla de la aplicación en el entorno final de la implementación.

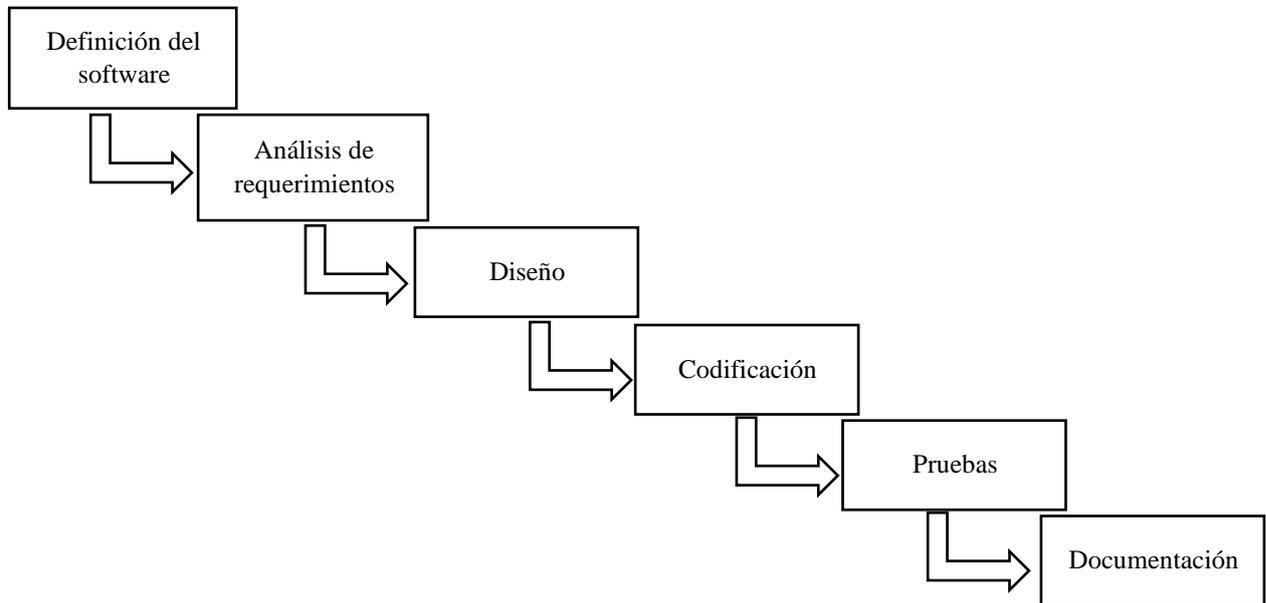


Figura 24: Diagrama de bloques de los requerimientos del desarrollo del sistema

Elaborado por el Autor

El desarrollo de trabajo empezó la definición del software y con el análisis de todos los antecedentes, requerimientos y documentación del diverso estado del arte. Un análisis pertinente y detallado del problema a resolver, en el caso concreto, el desarrollo de un sistema que sea capaz de detectar la forma del esqueleto humano, procesar la imagen para corregir la postura de la ejecución de ejercicios realizados por los deportistas en espacios de actividad física en tiempo real. Así como de las variables que afecten el funcionamiento del dispositivo, y el entorno de desarrollo a utilizar.

El proceso de diseño, tanto de la Interfaz de Usuario como del sistema en general que contenga apartados de información respecto a los ejercicios, y las respectivas ventanas de corrección de posturas, la generación y diseño de un algoritmo que sea capaz de corregir las mismas en base a la postura establecida por las normativas de deporte. Seguido de lo anterior, el tercer aspecto consistente en la conexión del dispositivo electrónico, además de la programación de los algoritmos; por último, se considera el testeo del producto final en distintos ambientes, como gimnasios,

espacios deportivos y centros de actividad física de la Universidad Técnica de Ambato, así como en ambientes reales, concretamente un gimnasio de pesas convencional. Por último, se puntualiza la documentación general del proyecto y los resultados generados durante el proceso de investigación y experimentación, así como la comprobación de éstos, en comparación a la hipótesis en la Figura 25 se presenta el diagrama de flujo del desarrollo general del sistema siguiendo la metodología usada para su elaboración. [29]

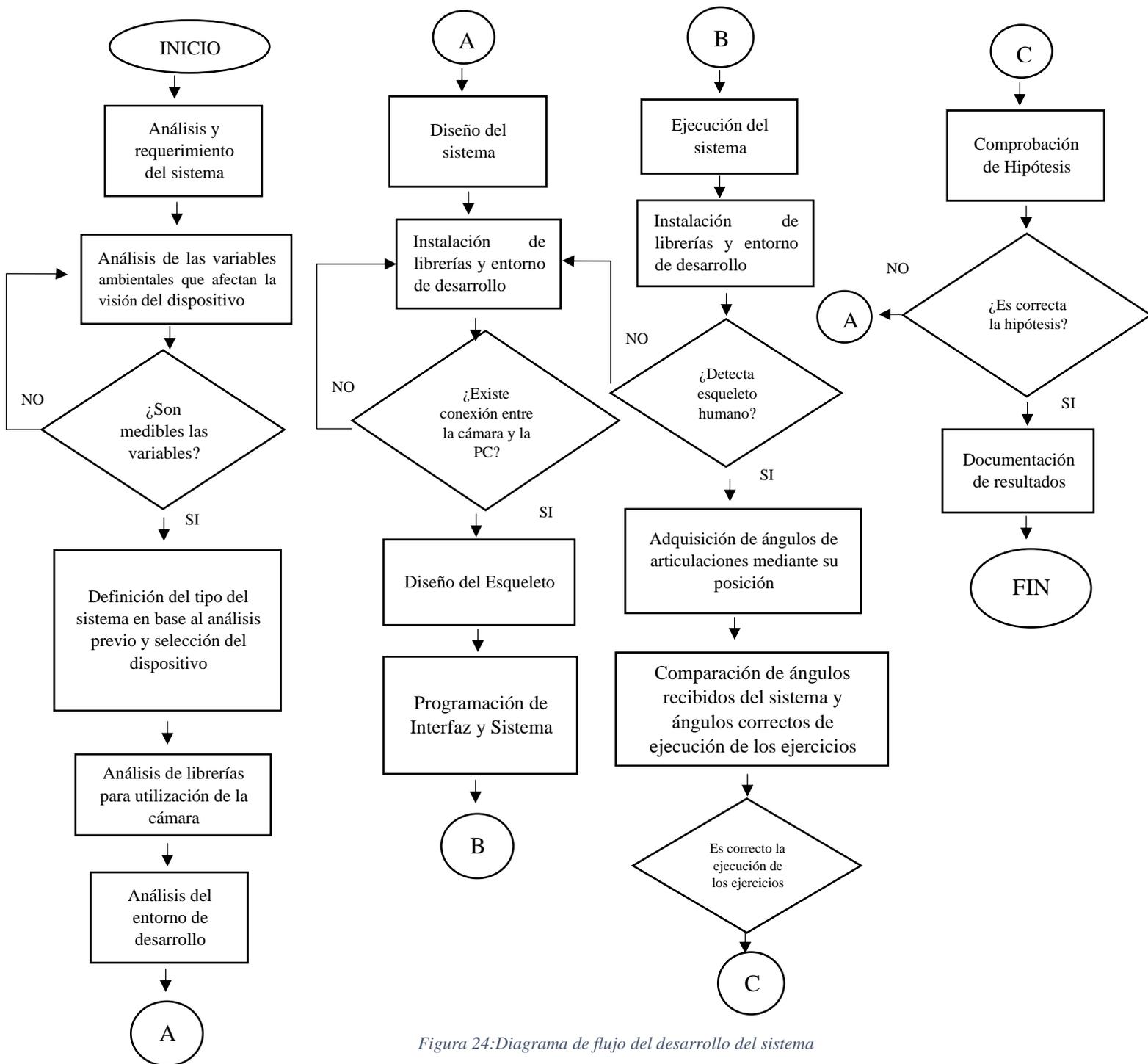


Figura 24: Diagrama de flujo del desarrollo del sistema

Realizado por el Autor

2.2.2 Selección de elementos para la elaboración del sistema

3.2.2.3 Cámara

La cámara web es la recolección de imágenes en la implementación del sistema electrónico, para el análisis y recolección de datos en los modelos reentrenados que posee las bibliotecas de Python en la tabla 3 muestra algunas de ellas.

Cámaras Web				
Descripción gráfica				
Marca	Mecatech	ELP	Altek	Genius
Modelo	001	X000QSOMLP	Xmk-82183	FACECAM 1000X
Sensor de imagen	CCD 10,0 Mp	GC2053	-----	CMOS hd720p
Interfaces	USB2.0	USB2.0	USB 2.0	USB 2.0
Drivers de instalación	Incluye en CD	Incluye en CD	Incluye en CD	Incluye en CD

Resolución de imagen	5 Mpx	720 Mpx	12 Mpx	10Mpx
Resolución de video	640x480.	Full HD 1080p	VGA 480p	HD 720p
Peso	55g aprox	70g aprox	45g aprox	50g aprox
Compatibilidad	Windows, Linux, Mac	Windows, Linux	Windows, Linux, Mac	Windows, Mac OS, Linux 2.6.21
Costo	\$ 16.48	\$ 40	\$ 25	\$ 23.99

Tabla 11: Comparación de cámaras web.

Fuente: Tienda Digital Santiago de Chile [26]

Realizado por el Autor

Para el Hardware se optó por hacer uso de la Cámara ELP-X000QSOMLP Mini USB para la detección del movimiento del cuerpo humano ya que cuenta con lo esencial para llevar a cabo la programación de una aplicación de éste tipo; cuenta con una cámara RGB (Rojo, Verde, Azul) que permite la captura del entorno tratando la señal de vídeo por separado de los tres colores primarios, proporcionando mayor calidad y una reproducción más fiel al color, y sensores de profundidad que ayudan a la detección de imágenes, ésta funciona midiendo las distancias entre la cámara y el objeto que se detecte. [29]

La cámara ELP con la que cuenta es una cámara digital, la luz atraviesa una lente que la dirige a un filtro encargado de separarla en los colores primarios (rojo, verde y azul), proyectados sobre un sensor fotosensible. El sensor genera una señal

eléctrica en función de la intensidad de la señal que incide sobre él. Esta señal es convertida a digital mediante un ADC (Analog Digital Converter) que es analizada y reconstruida para su almacenamiento.

Cámara ELP Mini USB con reconocimiento facial

ELP HD 1.3Megapixel USB cámara (modelo: ELP-X000QSOMLP serial) adopta el sensor AR0130 HD, puede funcionar con IR Cut(Corte Infrarrojo), es buena para usar por la noche, y ha sido ampliamente utilizada en Linux, o aquellos Sistema equipados que vienen con Linux para prototipo de Android. [26]

Algunas características adicionales

- * Cámara web HD USB 2.0 de 1.3 megapíxeles
 - * Baja iluminación Min 0.01lux
 - * Sensor CMOS Aptina AR0130 HD de 1/3 pulgadas.
 - * Lente de enfoque fijo HD de megapíxeles de 3,6 mm
 - * Formato de compresión MJPEG y YUY2 opcional
 - * Puerto USB 2.0 de alta velocidad
 - * Soporte OTG protocol: puede conectar su móvil a través de un cable OTG
 - * Balance de blancos ajustable: resuelve problemas de procesamiento de tono y reducción de color
 - * Controlador gratuito, plug & play para usar en cualquier dispositivo
- Cámara con UVC, uso compatible en Linux, Windows XP, WIN CE, MAC, SP2 o Android 4.0 arriba



Figura 25: ELP HD 1.3Megapixel USB cámara (modelo: ELP-X000QSOMLP)

Fuente: ELP AILIPU TECHNOLOGY CO.LTD Fabricante de cámaras para Videos Vigilancia

3.2.2.4 Sensor de monitoreo de pulso Cardíaco

El sensor de monitoreo de pulso cardíaco es el que se va a encargar de leer toda señal que envíe el individuo al momento de ejecutar el levantamiento de potencia, existen una variedad de estos sensores cada uno con características y funcionamientos diferentes, se puede decir que uno mejor que otro. En la tabla 12 se describe algunos sensores de proximidad con sus respectivas características.

Sensor de proximidad	Fotoeléctrico Infrarrojo	Infrarrojo	Ultrasónico	Infrarrojo
Descripción gráfica				
Nombre/Referencia	E18-D80NK	TCRT5000L	HC-SR04	IR FC-51
Voltaje de funcionamiento [V]	+5V DC	6VDC	+5V DC	3.3~5VDC
Corriente de trabajo	>25mA(min) ~100mA(máx)	300mA	15mA	18mA~20mA (5V)
Ángulo para detección	15°	-----	30°	35°

Temperatura ambiente	-25°C~55°C	-25°C~70°C	-15°C~70°C	-----
Rango de detección	3cm ~ 80 cm	1mm – 8mm	2cm~4m	2~30cm
Dimensión	1.7 x 4.5cm	18 x 75mm	45x20x15mm	3.2 x 1.4cm
Costo	\$ 25	\$ 66,24	\$ 4	\$4,25

Tabla 12: Comparación de las características de sensores de proximidad.

Fuente: Realizado por el Autor

Para la selección del sensor de monitoreo el cual pueda hacer la lectura del flujo sanguíneo que cumpla con los requerimientos técnicos y electrónicos como son la ubicación de las terminaciones nerviosas y la posición de enfoque de los diodos infrarrojos para el desarrollo del sistema, se realizó una comparación entre los sensores descritos en la tabla 12, se puede observar mediante las características que los sensores son de mucha utilidad para la medición y detección en ambientes de muy poca distancia, pero se debe tomar en cuenta parámetros importantes para el desarrollo y necesidad del sistema los cuales se describen a continuación:

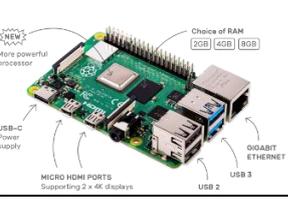
- Superficie de detección se considera porque los deportistas tienen diferente punto de detección.
- Rango de detección para monitorear al deportista
- Dimensión debido a que el dispositivo ira en el cuerpo

El sensor infrarrojo TCRT5000L se acopla bien a los parámetros de la implementación del sistema, la distancia para la detección es correcto y no habrá problema para el funcionamiento del circuito electrónico, su dimensión es perfecta para el movimiento al momento de la ejecución del ejercicio, y su costo es accesible para el investigador, pero la desventaja está en el rango de detección que es muy corta o la transmisión de luz puede fallar ya que ocurre una reacción química en el

cuerpo y puedo lanzar pulsos eléctricos irrelevantes o peor aún la negativa activación del sistema.

3.2.2.5 Microcontrolador

El objetivo principal del microcontrolador es la adquisición y procesamiento de los datos obtenidos por el sensor y el actuador que es la cámara, ejecutara todas las instrucciones que requiere el sistema electrónico, se hizo uso de un equipo de computación con los requerimientos mínimos para soportar la tecnología Raspberry pi y los entornos de desarrollo, en la tabla 13 se realizó un análisis comparativo con las características principales de los microcontroladores más conocidos y utilizados. Con la descripción de las características de los microcontroladores y los requerimientos del sistema electrónico se decidió utilizar el microcontrolador RaspBerryPI 4, que es una plataforma de hardware libre y entorno fácil de programación, el cual dispone para la utilización 40 pines digitales de I/O, en el cual se alojara el algoritmo desarrollado y dará paso a la convergencia de todos los dispositivos electrónicos del sistema, y para evitar vibraciones, temperaturas o humedad antes de su instalación se colocó una protección.

Microcontrolador	Raspberry PI	Raspberry PI 4
Descripción gráfica		
Voltaje Operativo	5V	5V
Voltaje de entrada recomendado	7~12V	7~12V

Voltaje límite máximo de entrada	6~20V	6~20V
Pines digitales I/O	40 Pines GPIO	40 Pines GPIO
Puertos USB	4	4
Corriente por pines I/O	40mA	40mA
Memoria Expandible	8GB	16GB
Velocidad de reloj	1.2GHz	1.5 GHz
Memoria Flash	2GB SDRAM	8 GB SDRAM
Puerto LAN	SI	SI
Memoria EEPROM	4KB	4KB
Costo	\$30	\$35
Procesador	QUAD-CORE ARMV8	ARM CORTEX -A72

Tabla 13: Clasificación de Microcontroladores

Fuente: Hardzone.es

Realizado por el Autor

La Placa Raspberry Pi 4 trae incorporado en el hardware cualidades que si se desea implementar con cualquier otro microcontrolador adicional sus funciones no se verán afectadas, la tabla 13 arroja como resultado que la placa Raspberry Pi 4 es la

mejor opción debido a su gran capacidad de memoria la cual es muy importante debido a la gran cantidad de datos que maneja dicho algoritmo así no será lento al momento de procesar y por tener 2 opciones de salida de video, ya que en algunos circuitos es necesario instalar permanentemente un televisor con conector RCA.

El microprocesador es un circuito integrado que contiene la unidad central de proceso (CPU). También llamado procesador de un computador. El CPU está formado por la Unidad de control, que interpreta las instrucciones, y el BUS de datos, que los ejecuta. Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control para remitir conectarle con la memoria y los módulos de (Entrada y Salida) E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine.

Un microcontrolador es un sistema cerrado, todas las partes del computador están contenidos en su interior y solo salen las líneas que gobiernan los periféricos. Usted podría pensar que las características de un sistema cerrado representan una desventaja con relación a los microprocesadores, pero en la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es difícil no encontrar uno que se adapte a nuestros requerimientos al momento. Concluimos que la diferencia fundamental entre un Microprocesador y un Microcontrolador es que el Microprocesador es un sistema abierto con el que se puede construir un computador con las características que se desee, acoplándole los módulos necesarios. Un Microcontrolador es un sistema cerrado que contiene un computador completo y de prestaciones limitadas que no se pueden modificar. [41]

3.2.2.6 Procesador

En el caso concreto de la computadora que se utilizó para la programación y pruebas del sistema, ésta cuenta con las siguientes características, para el análisis de video es esencial que la laptop tenga un procesador o CPU de gama alta por las funciones

que debe realizar al momento de ejecución la visión artificial, para el procesamiento de información utilizando operaciones aritméticas y lógicas, también para controlar el funcionamiento de todo el hardware de la computadora, las operaciones individuales pueden tomar desde varias fracciones de una medida hasta decenas de medidas. Medido en megahercios o gigahercios cuanto mayor sea la velocidad del reloj, más rápido procesará la CPU la información entrante y es lo que se necesita para el funcionamiento de esta etapa.

Cabe recalcar que la elección de un procesador siempre va a depender de cómo y para qué van a utilizarse, a continuación, en la tabla 14 se describe los procesadores de gama alta más conocidos y utilizados en el campo de la informática en la actualidad con sus respectivas características enfocando en las principales y necesarias para el funcionamiento del prototipo del sistema electrónico.

Procesador	AMD	INTEL
Descripción gráfica		
Consumo de energía	Alto	Bajo
Precio	Bajo	Alto
Refrigeración	Picos altos y bajos	Larga duración

Velocidad	4.6 Ghz	5Ghz
Núcleos	16/32	8/16
Overclocking	Si	No
TDP	65W	127W
Caché	L3: 32MB	12 MB Intel® Smart Cache
PCI Express	PCIe 4.0 x16	PCIe 3.0 x16
Memoria	DDR4	DDR4-2933
Temperatura máxima	95°C	100°C
Aplicaciones	Ingeniería Juegos por PC	Ingeniería Juegos por PC

Tabla 14: Comparación de procesadores

Fuente: Computer la.

Realizado por el Autor

El procesador Intel Core I7 (I7-8550U) que cumple con todos los parámetros necesarios para esta implementación del sistema, el número de núcleos es correcto para para el software a utilizar, la frecuencia es perfecta porque el sistema requiere que la velocidad de la CPU en el procesamiento sea rápida, es recomendable para el

software, y el costo es accesible para el desarrollo del algoritmo que una vez implementado se pasará al microcontrolador para que el sistema sea estético.

3.2.2.7 Lenguaje de programación

Los lenguajes de programación que se empleara en el trascurso de la implementación son los descritos en la tabla 15, se hace énfasis por ser la herramienta primordial para crear varias instrucciones que permite acciones consecutivas para el control físico y lógico del sistema

Lenguaje de Programación	C++	Python
Origen	Bjarne Stroustrup	Guido Van Rossum
Significado	Incremento de C	Lenguaje interpretado
Características	<ul style="list-style-type: none"> • Compatibilidad con diferentes bibliotecas • Orientado a objetos • Compilación de bajo nivel para ejecución • Didáctico 	<ul style="list-style-type: none"> • Código definido como scripts. • No importa el sistema operativo. • Orientado a objetos. • Escritura dinámica.
Ventajas	<ul style="list-style-type: none"> • Alto rendimiento • Multiplataforma • Extendido • Lenguaje actualizado 	<ul style="list-style-type: none"> • Rápido y simplificado • Interpretable • Productividad • Legible y organizado • Portable

Aplicaciones y usos	<ul style="list-style-type: none"> • Base de datos • Navegadores web • Videojuegos • Compiladores • Sistemas Operativos 	<ul style="list-style-type: none"> • IA • Análisis de datos • Desarrollo web
---------------------	--	---

Tabla 15: Comparación de lenguajes de programación

Fuente: Universia.net

Elaborado por el autor

Microcontrolador PIC

El PIC 12F683 por su alta capacidad de memoria, su pequeño tamaño y gran funcionalidad que es lo suficiente para dicha aplicación este dispositivo responde a microcontroladores de arquitectura abierta pudiendose expansionar en el exterior para poder sacar los datos direcciones y control. Estas MCU proporcionan hasta 5 MIPS, 3,5 Kbytes de memoria de programa con una memoria de datos que tiene 128 bytes de RAM y EEPROM de 256 bytes. En la placa hay un oscilador RC configurable con una precisión de ± 1 %.

Características

- 35 instrucciones
- Pila de hardware de 8 niveles
- Oscilador interno de 8 MHz – Rango seleccionable de 8 MHz a 125 kHz
- 6 contactos de E/S
- 1 comparador
- Convertidor analógico a digital (ADC) de 4 canales y 10 bits
- Módulo de captura, comparación y PWM (CCP)
- Dos temporizadores de 8 bits
- Un temporizador de 16 bits
- Programación serie en circuito (ICSP)

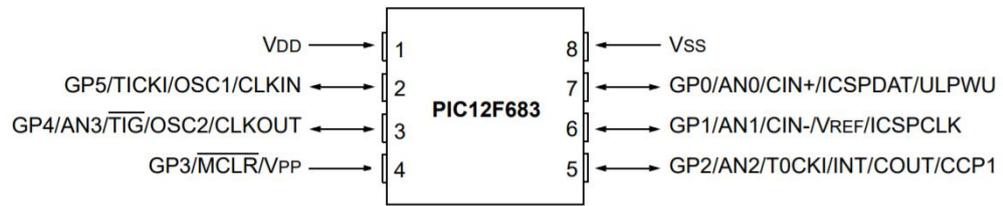


Figura 26: Diagrama PIC 12F683

Fuente: MICROCHP PIC12F683 8 pines CMOS Microcontroller

PIC12F683

TABLE 1-1: PIC12F683 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
VDD	VDD	Power	—	Positive supply
GP5/TICKI/OSC1/CLKIN	GP5	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	TICKI	ST	—	Timer1 clock
	OSC1	XTAL	—	Crystal/Resonator
	CLKIN	ST	—	External clock input/RC oscillator connection
GP4/AN3/TIG/OSC2/CLKOUT	GP4	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN3	AN	—	A/D Channel 3 input
	TIG	ST	—	Timer1 gate
	OSC2	—	XTAL	Crystal/Resonator
	CLKOUT	—	CMOS	Fosc/4 output
GP3/MCLR/VPP	GP3	TTL	—	GPIO input with interrupt-on-change
	MCLR	ST	—	Master Clear with internal pull-up
	VPP	HV	—	Programming voltage
GP2/AN2/T0CKI/INT/COU/CCP1	GP2	ST	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External Interrupt
	COU	—	CMOS	Comparator 1 output
	CCP1	ST	CMOS	Capture input/Compare output/PWM output
GP1/AN1/CIN-/VREF/ICSPCLK	GP1	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN1	AN	—	A/D Channel 1 input
	CIN-	AN	—	Comparator 1 input
	VREF	AN	—	External Voltage Reference for A/D
	ICSPCLK	ST	—	Serial Programming Clock
GP0/AN0/CIN+/ICSPDAT/ULPWU	GP0	TTL	CMOS	GPIO I/O with prog. pull-up and interrupt-on-change
	AN0	AN	—	A/D Channel 0 input
	CIN+	AN	—	Comparator 1 input
	ICSPDAT	ST	CMOS	Serial Programming Data I/O
	ULPWU	AN	—	Ultra Low-Power Wake-up input
VSS	VSS	Power	—	Ground reference

Legend: AN = Analog input or output
TTL = TTL compatible input
HV = High Voltage
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
XTAL = Crystal

Tabla 16: Tabla de descripción de pines PIC 12F683

Fuente: MICROCHP PIC12F683 8 pines CMOS Microcontroller

3.2.2.8 Software IDE Proton

El compilador Proton Basic es un lenguaje de programación de nueva generación que hace más fácil y rápido el manejo de microcontroladores Pic micro de Microchip. El lenguaje Basic es mucho más fácil de leer y escribir que el lenguaje ensamblador Microchip. El Proton Basic produce un código que puede ser programado para una variedad de microcontroladores PIC que tengan de 8 a 68 pins y varias opciones en el chip incluyendo convertidores A/D, temporizadores y puertos seriales en la Figura 28 el entorno de desarrollo con el encabezado por defecto, además de esta estructura de base, algunos compiladores también permiten programación orientada a objetos, así como procedimientos y funciones. Sin embargo, protón IDE no permite los procedimientos y funciones con los objetos. Tiene enfoque orientado hacia el simple y sencillo llamado de instrucciones. [39]

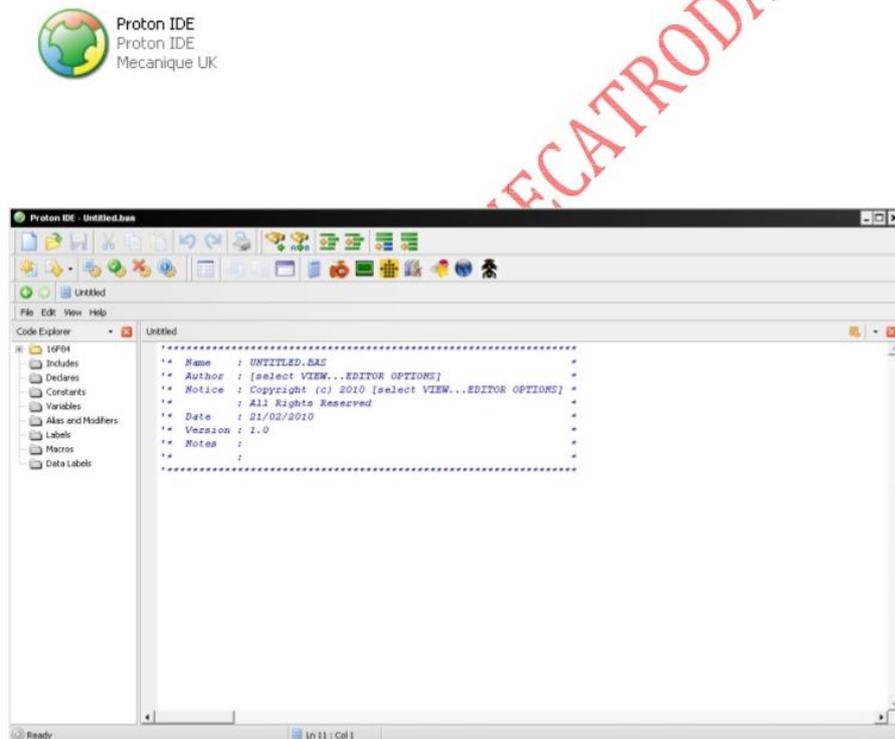


Figura 27: Ventanilla lista para crear programas en Proton IDE

Fuente: Software Proton Slideshare

3.2.3 Diseño del prototipo

El prototipo del sistema electrónico está basado en etapas complementarios que requieren de algunos elementos, con su respectivo análisis y procesos para la implementación, los cuales se describen a detalle a continuación en la Figura 29. La implementación de un sistema electrónico de corrección de posturas en el levantamiento de potencia en deportistas permite mejorar la ejecución de dicho ejercicio, modernizando de esta manera el entrenamiento, así se evitará lesiones o sintomatologías de dolor sin la necesidad de que un profesional en el levantamiento de pesas este siempre presente para supervisarlo esto se logra mediante la recolección de datos, procesamiento y análisis de imágenes de manera profunda, que permite detectar las articulaciones más importantes en el cuerpo para monitorear de igual forma el circuito electrónico del sensor de pulso cardiaco aumentará la efectividad del sistema dándonos como respuesta una herramienta tecnológica de gran ayuda para el deporte

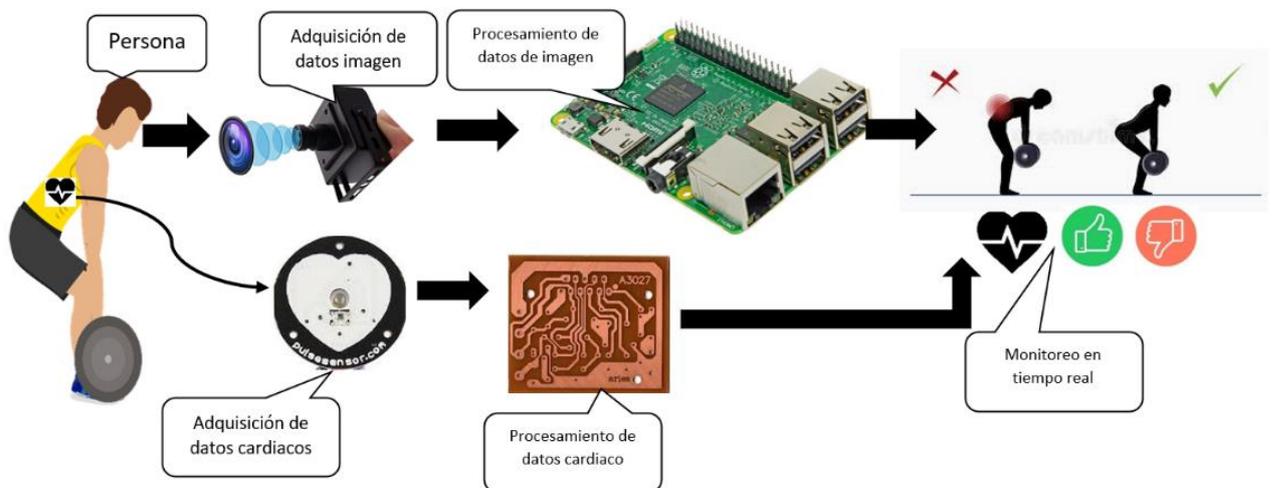


Figura 28: Propuesta de solución

Fuente: Realizada por el autor

3.2.4 Implementación

Instalar Python Anaconda

1.- Se realizó la descarga de la distribución Python Anaconda inicialmente se descargó los paquetes de instalación en la Figura 30 muestra las áreas que se encuentran en el sitio web.



Figura 29: Botón de descarga Anaconda.

Fuente: Python.org. [28]

2.- Los diferentes sistemas operativos Windows, Mac y Linux tienen diferentes versiones. Descargar y seguir las instrucciones para instalar en Windows de 64-Bits la versión 2.7 ya que es la más completa, inicialmente para desarrollar el algoritmo.

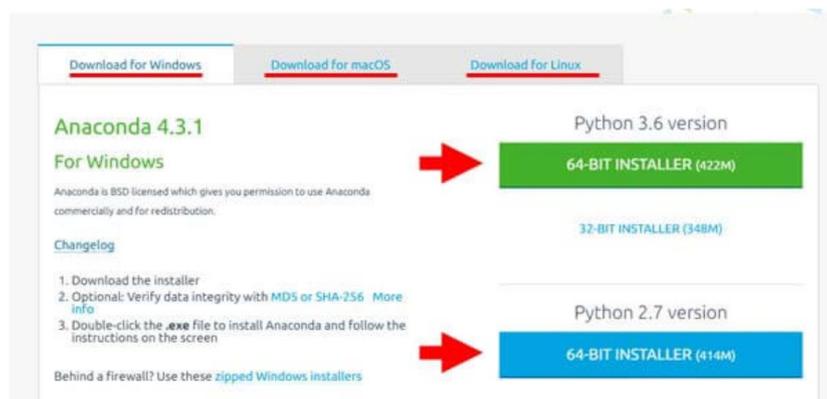


Figura 30: Elegir la versión de descarga Anaconda.

Fuente: Python.org. [28]

3.- Una vez descargado ejecutar el instalador de 64-Bits versión 2.7. Las siguientes ventanas dependerán del sistema operativo.

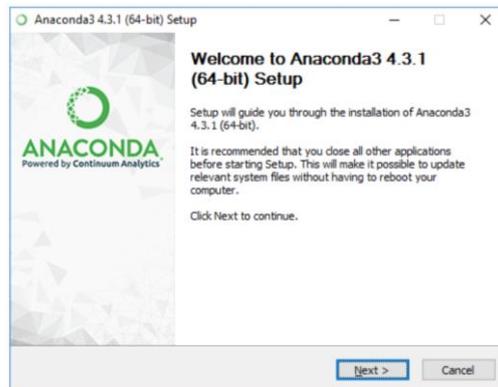


Figura 31: Ejecutable/ instalador automático Anaconda.

Fuente: Python.org. [28]

4.- En la Figura 33 verifica los Términos y condiciones y hacer clic en el botón [I Agree]

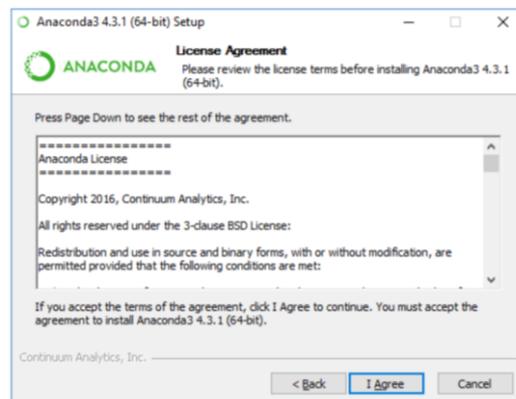


Figura 32: Instalador automático Anaconda.

Fuente: Python.org. [28]

5.- Una vez iniciado el instalador, en la ventana Install Python 2.7 (64 bit) activar las casillas de las opciones: Elegir All users (requieres admin privileges).

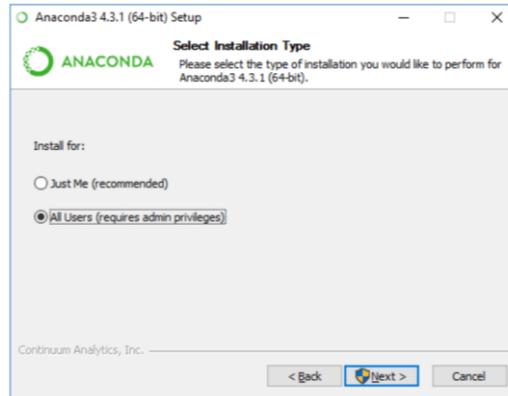


Figura 33: Instalador para todos los usuarios.

Fuente: Python.org. [28]

6.- En la ventana Chose Install Location verificar que están activas las opciones de la imagen, escribir la ruta del directorio de instalación "C:\Python2.7 " (o aceptar la ruta por defecto) y comenzar la instalación haciendo clic en el botón [Install]

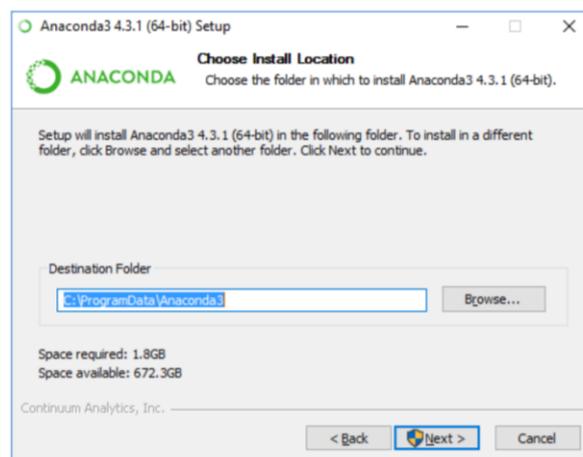


Figura 34: Ruta de instalación de Anaconda

Fuente: Python.org. [28]

7.- La ventana Advance Installation Option, este paso es importante ya que hay que seleccionar las 2 opciones que aparecen en la ventana ya que determina la compatibilidad con los ambientes de desarrollo externos y el proceso de instalación de Python inicia.

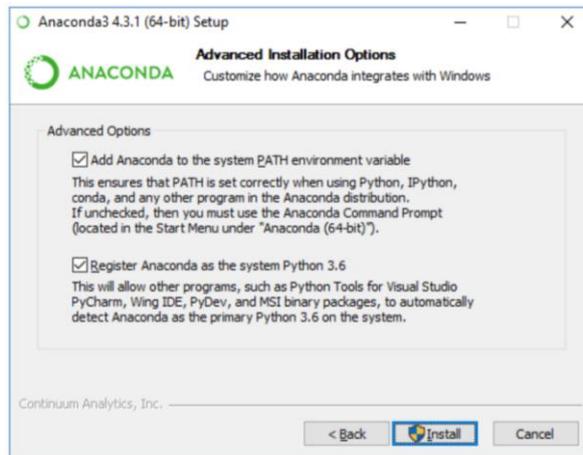


Figura 35: Paso Importante para la partición y compatibilidad de Anaconda.

Fuente: Python.org. [28]

8.- En la ventana Intallation Complete después de unos minutos la instalación se completa y hacer clic en el botón [Next]

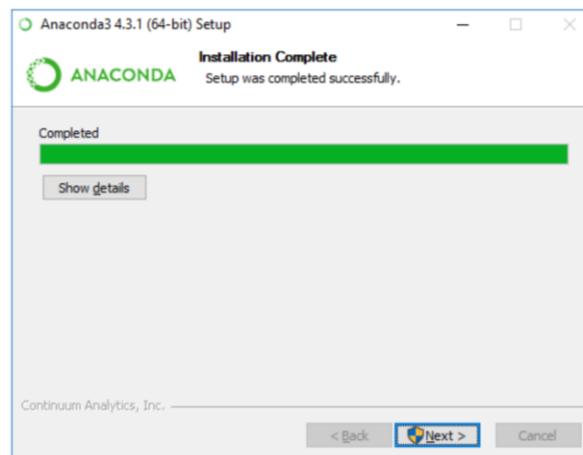


Figura 36: Configuración del sistema.

Fuente: Python.org. [28]

9.- En la ventana Setup was successful, una vez que ha concluido el proceso de instalación hacer clic en el botón [Finish]. Desde esta ventana es posible acceder a un tutorial online de Python, a la documentación oficial del lenguaje y a información con las novedades de la presente versión.

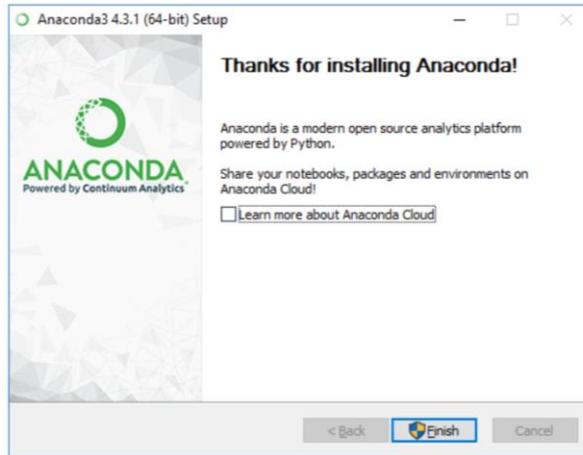


Figura 37: Instalación Completa de Anaconda con entorno de Python.

Fuente: Python.org. [28]

Con estos pasos está instalado Python y los paquetes que son necesarios para instalar OpenCV.

1 python -V

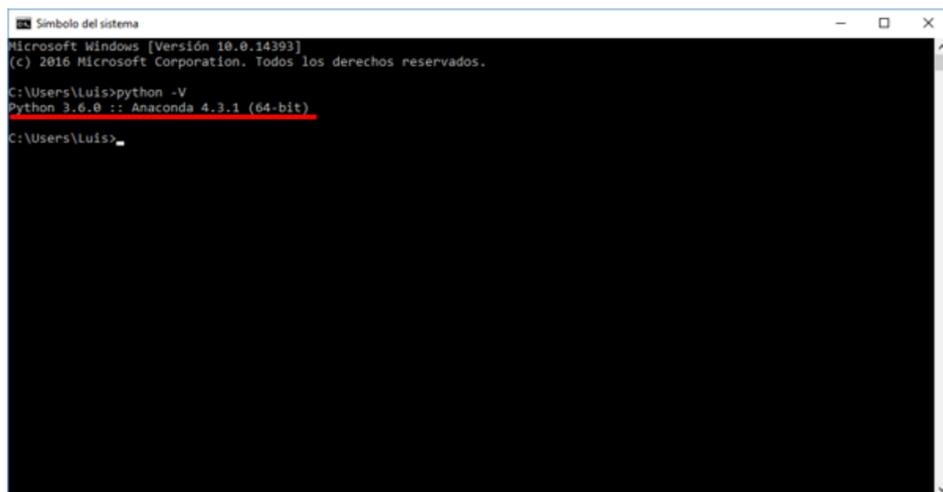


Figura 38: Versión de anaconda en el entorno Python.

Fuente: Realizado por el Autor

Instalar OpenCV puede ser una tarea bastante compleja, pero gracias a la investigación desarrollada se logró la instalación correcta y su posterior utilización. Para verificar que todo está correctamente instalado escribir el siguiente código.

```
import cv2
# Cargamos la imagen del disco duro
imagen = cv2.imread("logo.jpg")
cv2.imshow("prueba", imagen)
cv2.waitKey(0)
```

Abrir una línea de comandos e ir a la carpeta donde están los archivos prueba-opencv.py y la imagen. Ejecutar el siguiente comando

```
1 python prueba-opencv.py
```

Si todo ha ido bien aparecerá la imagen en una ventana como muestra en la Figura 40. Esto indica que OpenCV se ha instalado correctamente.

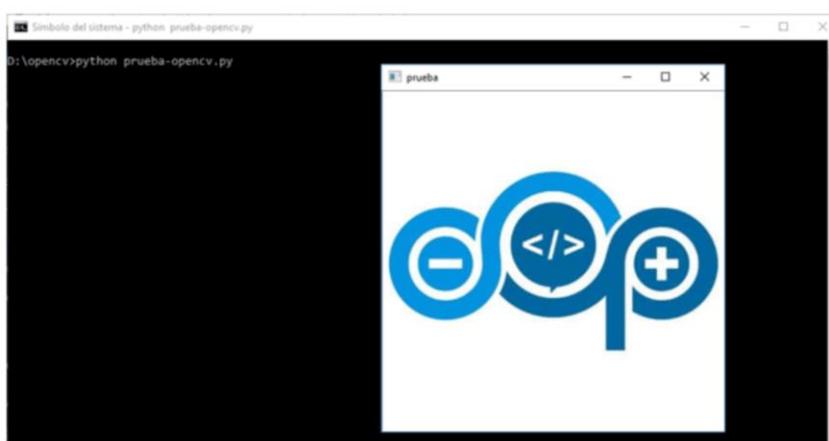


Figura 39: Prueba de verificación de OpenCV correctamente instalado.

Fuente: Realizado por el Autor

Tensor Flow

Tensor Flow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático, cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que les permite a los investigadores impulsar un aprendizaje automático innovador y, a los desarrolladores, compilar e implementar con facilidad aplicaciones con tecnología de Inteligencia Artificial.[30]

Instala el paquete pip de TensorFlow

```
pip install --upgrade tensorflow
```

Verificamos la instalación del paquete

```
python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

Operación Exitosa se instaló con éxito Tensor Flow

Open Pose

'Openpose', es el algoritmo de estimación de pose humana, que se ha implementado utilizando Tensorflow. También proporciona varias variantes que tienen algunos cambios en la estructura de la red para el procesamiento en tiempo real en la CPU o dispositivos integrados de bajo consumo.

Instalación en el PC

```
git clone https://www.github.com/iloonet/tf-pose-estimation
cd tf-estimación-pose
pip3 install -r requirements.txt
cd tf_pose/pafprocess
swig -python -c++ pafprocess.i && python3 setup.py build_ext --inplace
```

Instalación de Paquete

```
git clone https://www.github.com/iloonet/tf-pose-estimation
cd tf-estimación-pose
python setup.py install # 0, `pip install -e .`
```

Instalar Open Pose desde la Fuente

Instalar CMake GUI

```
sudo apt-get install cmake-qt-gui
```

Abrir GUI de CMake

```
cmake-gui
```

Colocar la ubicación del código fuente y la ubicación del directorio de compilación como

```
/path/to/openposey /path/to/openpose/build
```



Figura 34: Dirección de los archivos a instalar en código.

Fuente: Realizado por el Autor

Esto podría generar una ventana preguntando si se debe crear el directorio de compilación, presione sí.

Marque la BUILD_PYTHON casilla de verificación.

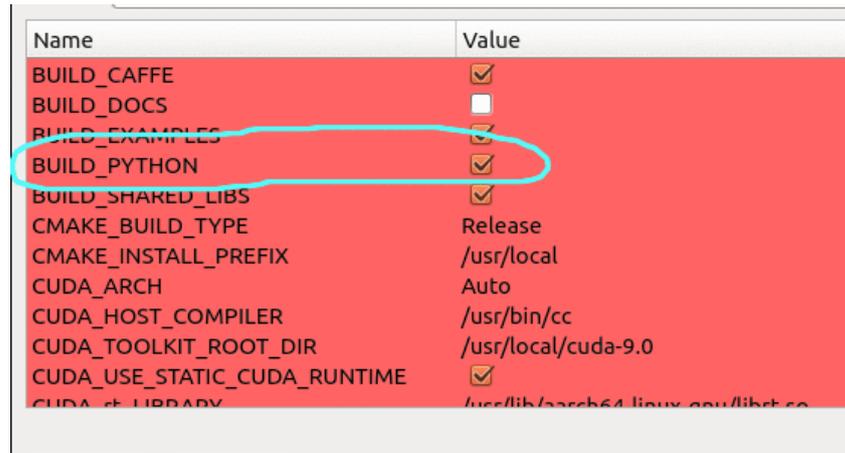


Figura 40: Elegir construir en Python.

Fuente: Realizado por el Autor

Presione Configurar, ahora debería recibir un mensaje que diga Configuring Done. Presione Generatey luego cierre la GUI.

Ir al directorio de compilación y realizar

```
cd build
make -j`nproc`
```

Instalación de OpenPose Python:

```
cd build / python
sudo make install
```

Agregue la ruta de construcción a bash:

```
vim ~ / .bashrc
# (Agregue esta línea al final del archivo)
export PYTHONPATH = / path / to / openpose / build / python:
```

Instalación Exitosa

```
Python 2.7.12 (predeterminado, 4 de diciembre de 2017, 14:50:18)
[GCC 5.4.0 20160609] en Linux
Escriba "ayuda", "derechos de autor", "créditos" o "licencia" para obtener
más información.
```

```
>>> importar openpose
Python 3.5.2 (predeterminado, 23 de noviembre de 2017, 16:37:01)
[GCC 5.4.0 20160609] en Linux
>>> importar openpose
```

Instalación de tf-pose.assessment

```
mkdir Pose_Estimation
cd Pose_Estimation
```

Crear un entorno Virtual

```
conda crear --name Tf2_Py37 python = 3.7.6 -y
conda activar Tf2_Py37
```

Instalar TF2

```
pip install --upgrade pip
pip install tensorflow
```

Instalar los paquetes basicos que se va a necesitar eb el desarrollo

```
conda install -c anaconda numpy
conda install -c conda-forge matplotlib
conda install -c conda-forge opencv
```

Clonar el repositorio de estimacion de pose

```
clon de git https://github.com/gsethi2409/tf-pose-estimation.git
```

Ir a la carpeta pose-assesment e intalar los requisitos

```
cd tf-pose-assessment /
pip install -r requirements.txt
```

Instalar SWIG compilador de interface que conecta programas escritos en C y C++ con lenguajes de Scrpthing como Python

SWIG

SWIG es una herramienta de desarrollo de software que conecta programas escritos en C y C ++ con una variedad de lenguajes de programación de alto nivel. SWIG se utiliza con diferentes tipos de lenguajes de destino, incluidos los lenguajes de programación comunes como Javascript, Perl, PHP, Python, Tcl y Ruby.[31]

Con Swig, cree la biblioteca C ++ para el posprocesamiento

```
conda instalar swig
cd tf_pose / pafprocess swig
-python -c ++ pafprocess.i && python3 setup.py build_ext -inplace
```

Instalar la biblioteca tf-slim , una biblioteca liviana que se usa para definir, entrenar y evaluar modelos complejos en TensorFlow.

```
pip install git + https://github.com/adrianc-a/tf-slim.git@remove_contrib
```

Dentro del directorio tf-pose- assesment hay un script de Python run.py , vamos a ejecutarlo, teniendo como parámetros

```
python run.py --model = mobilenet_thin --resize = 432x368 --image =. / images / ski.jpg
```

Aparecerá una ventana de Open cv independiente en la cual dibuje el modelo del algoritmo como se muestra en la Figura 42.

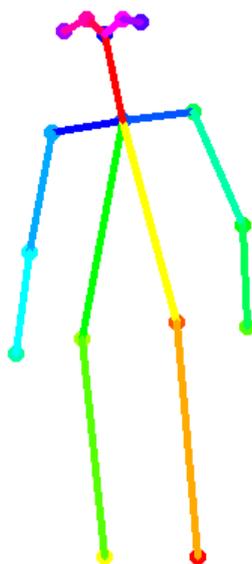


Figura 41: Figura que proporciona el algoritmo desarrollado del cuerpo humano

Fuente: Realizado por el Autor

Para Instalación más detallada ver sección de anexos y líneas de código utilizados en esta investigación

Detección Estimación Pose

La estimación de pose se refiere a las técnicas de visión por computadora que detectan figuras humanas en imágenes y videos, de modo que uno podría determinar, por ejemplo, dónde aparece el codo de alguien o un brazo en una imagen. Es importante tener en cuenta el hecho de que la estimación de pose simplemente estima dónde están las articulaciones clave del cuerpo y no reconoce quién está en una imagen o video.

El modelo Pose toma una imagen de cámara procesada como entrada y emite información sobre puntos clave. Los puntos clave detectados están indexados por una Identificación (ID) de pieza, con una puntuación de confianza entre 0.0 y 1.0.

la puntuación de confianza indica la probabilidad de que exista un punto clave en esa posición. Las diversas articulaciones corporales detectadas por el modelo Pose se tabulan a continuación:

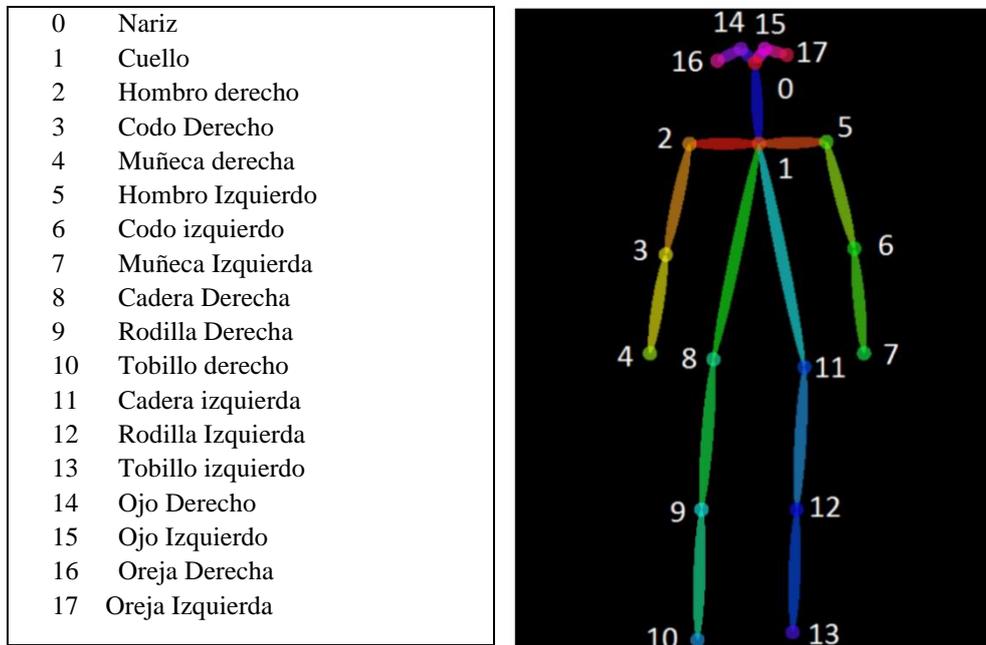


Figura 42: Figura que muestra los puntos de principales en el cuerpo humano

Fuente: Realizado por el autor

El rendimiento varía según el dispositivo y el ritmo de salida (mapas de calor y vectores de compensación). El modelo Pose es invariante en el tamaño de la imagen, lo que significa que puede predecir posiciones de pose en la misma escala que la imagen original, independientemente de si la imagen tiene una escala reducida. Esto significa que configura el modelo para tener una mayor precisión a expensas del rendimiento.

Cuanto mayor sea el paso de salida, menor será la resolución de las capas en la red y las salidas y, en consecuencia, su precisión. En esta implementación, el paso de salida puede tener valores de 8, 16 o 32. En otras palabras, un paso de salida de 32 dará como resultado el rendimiento más rápido pero la precisión más baja, mientras que 8 dará como resultado la precisión más alta pero el rendimiento más lento. El valor inicial recomendado es 16.

En la ilustración 44 muestra cómo el paso de salida determina cuánto se reduce la salida en relación con el tamaño de la imagen de entrada. Un paso de salida más alto es más rápido, pero da como resultado una menor precisión.

El paso de salida determina cuánto se reduce la salida en relación con el tamaño de la imagen de entrada. Afecta el tamaño de las capas y los resultados del modelo.[30]

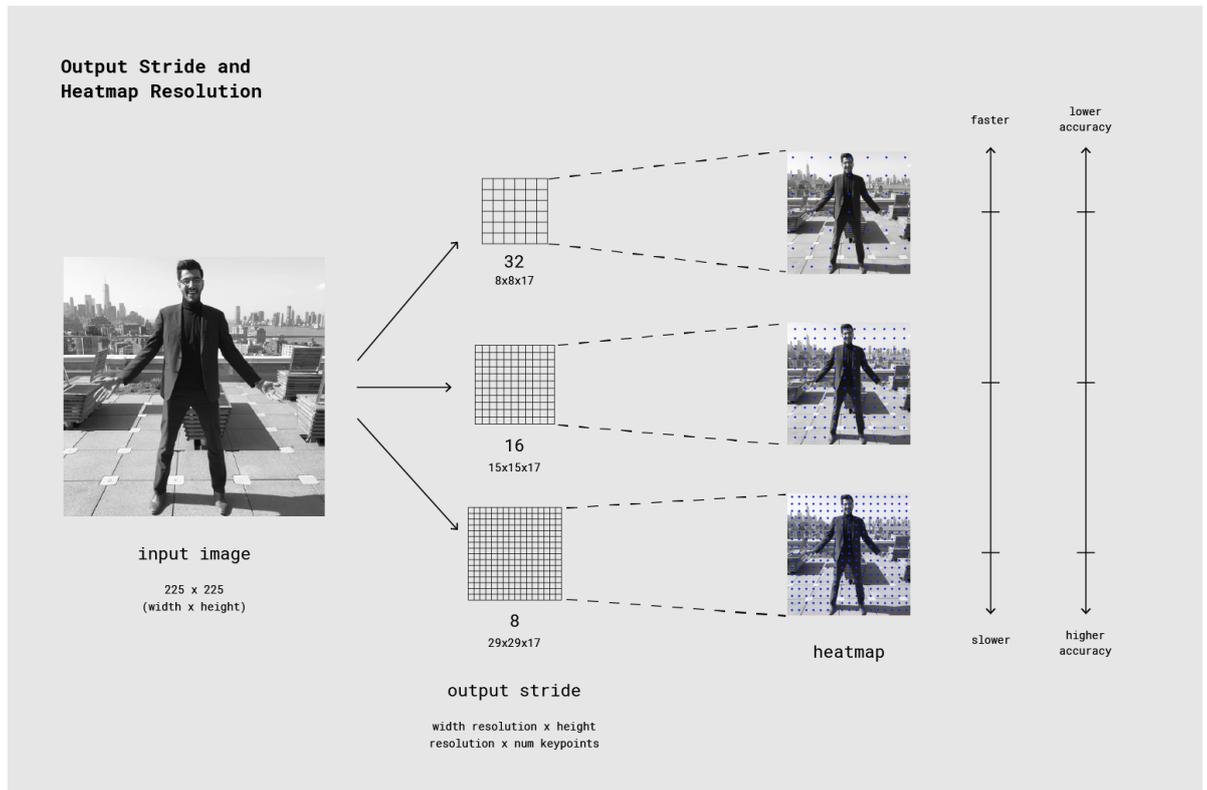


Figura 43: Figura que muestra la resolución para la salida de muestreo de imágenes

Fuente: www.tensorflow.org/pose-Models[30]

Estimación de pose usando TENSORFLOW Y OPENCV por medio SWIG

En este proceso utilizamos las herramientas de desarrollo para estimar la pose de una persona a partir de una imagen o un video mediante la estimación de las ubicaciones espaciales de las articulaciones corporales clave como es en la ejecución de las posiciones inicial, intermedia y final del levantamiento de potencia por medio de la herramienta Swig la imagen de ejecución de dicho ejercicio mapea las posiciones y el resultado final es la corrección de posturas.

Pasos para seguir.

Swig establece la ruta en la variable de entorno.

"tf-pose-assessment"

Instalar pycocotools usando este commando

```
pip install "git + https://github.com/philferriere/cocoapi.git#egg=pycocotools&subdirectory=PythonAPI"
```

Después de eso, elimine "pycocotools" de "requirements.txt"

Ahora instale otras librerías usando

```
pip install -r requirements.txt
```

Ejecute

```
"python3 run_webcam.py --camera video1.mp4"
```

Como resultado obtendremos la estimación de pose a partir de un video en las Figuras 45 obtenemos la estimación de pose de una persona.

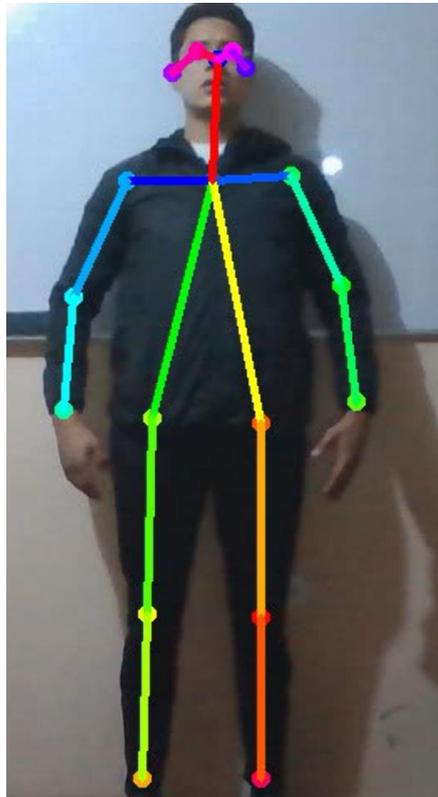


Figura 44: Estimación de pose usando Tensorflow y OPENCV

Fuente: Realizado por el Autor

En las Figuras 45 y 46 la estimación de pose se calibra gracias al mapeo de la imagen de ejecución del ejercicio de Levantamiento de potencia

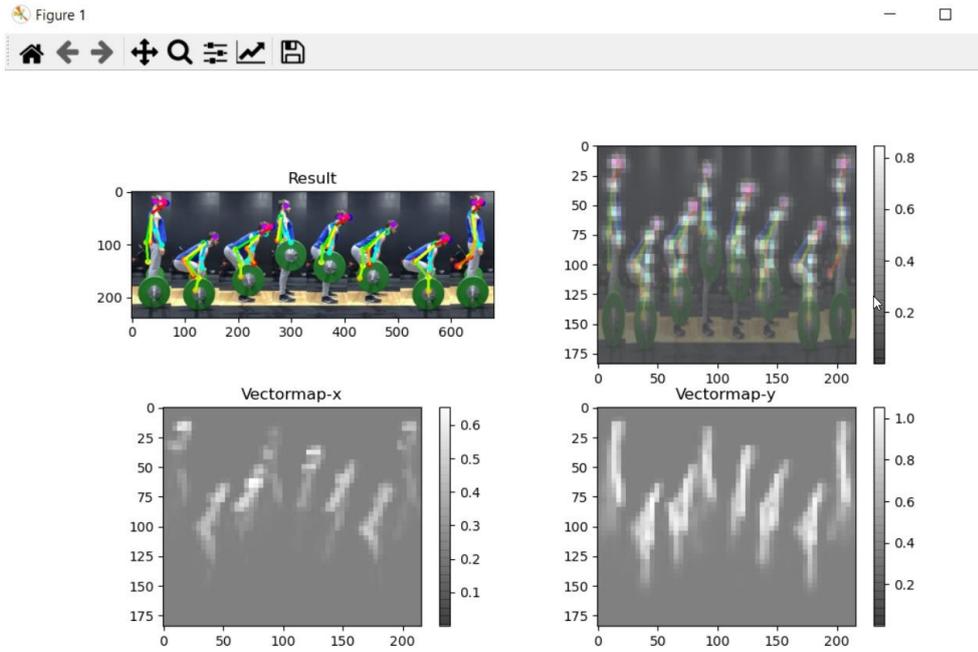


Figura 45: Mapeo de imagen

Realizado por el Autor

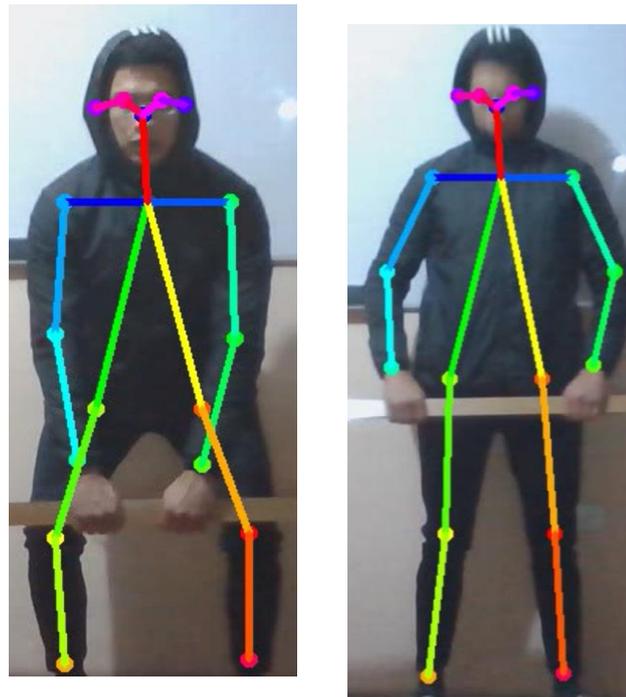


Figura 46: Ejemplo de funcionalidad del algoritmo

Fuente: Realizado por el Autor

Análisis del circuito sensor del pulso cardiaco

El bombeo de la sangre hacia los tejidos se incrementa en la medida en que aumenta la actividad muscular, por supuesto, que es aún mayor cuando se realizan los ejercicios físicos con un carácter competitivo pero en el levantamiento de potencia en malas posiciones es aún más, dado que los ejercicios se realizan en estado de apnea y en muchos de ellos con grandes pesos esto incrementa la presión arterial en la parte interna del tórax, es decir a la función del corazón de bombear más sangre para cubrir las exigencias energéticas de los ejercicios de potencia, se le adicionan las condiciones adversas en las cuales debe trabajar ese órgano, esto justifica y exige que en el proceso de entrenamiento de los levantadores de pesas estudiemos las respuestas cardiacas a las cargas de entrenamiento.

Procedimiento y Diseño de la placa

En el cuerpo se generan gran variedad de señales Electricas, provocadas por la actividad química que tienen lugar en los nervios y musculos. El corazón por ejemplo produce un patrón característico de variaciones de voltaje, este tipo de señales eléctricas permiten conocer su comportamiento y es posible conocer e identificar cuando el corazón funciona correctamente con la ayuda de un chip integrado programable que ayudara a la fácil interpretación de dichos pulsos. El circuito realiza el cálculo de las pulsaciones del corazón midiendo los cambios de absorción de luz que resultan de las pulsaciones del flujo de la sangre arterial. El sistema de detección consta de diodos emisores de luz (LED) de una sola longitud de onda, fotodetectores y un dispositivo microcontrolador, su funcionamiento se basa en el principio de la absorción diferencial (Bronzino, 2000). [42] La señal de entrada son los pulsos de frecuencia de que emiten los diodos de luz LED con una amplitud de 5VDC la primera etapa del circuito calcula la ganancia del amplificador operacional para la adecuación de la señal con ayuda del Filtro de retroalimentación que es un circuito que ayuda a reducir el ruido de alta frecuencia y la resistencia conectada a tierra para posteriormente enviar a la etapa de amplificación, en esta etapa el amplificador LM358 que es la encargada de tomar la salida y enviar a la entrada del microprocesador 12F683, este toma dicha señal la procesa gracias a las

En este apartado se diseña el circuito con el que va a acondicionar la señal. Con este circuito se busca filtrar la señal para eliminar el posible ruido y la componente continua del pulso sanguíneo y además amplificar la señal.

Amplificador de Transimpedancia

Tras la elección del sensor la primera etapa por la que pasará la corriente emitida por el fotodiodo será un amplificador de transimpedancia. Es un conversor de corriente a voltaje. Cuando le llega una corriente muy pequeña la transforma en un voltaje que le es proporcional. La impedancia de entrada es muy pequeña, ya que está diseñado para trabajar con una fuente de corriente.

Este tipo de circuitos es muy utilizado cuando se trabaja con sensores ya que la señal que estos emiten suele ser muy débil

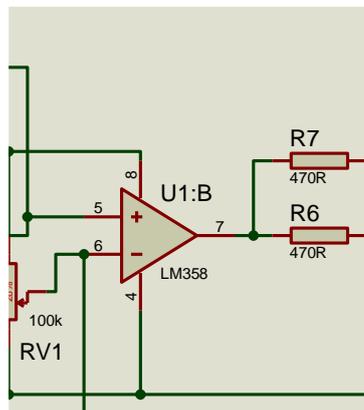


Figura 48: Etapa de amplificación

Realizado por el autor

Filtro pasa alto

La siguiente etapa tiene como objetivo eliminar la componente continua de la señal recibida tras pasar por el amplificador de transimpedancia. Para ello se utiliza un filtro de paso bajo con una frecuencia de corte de 0,7 Hz. En este trabajo de investigación y para el objetivo que aquí se plantea, se considera que el espectro de la señal por encima de los 0,7 Hz, por lo que todo lo que quede por debajo de esta frecuencia se considera parte de la componente continua.

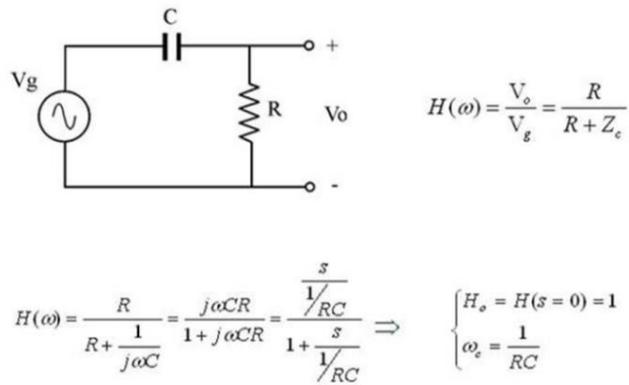


Figura 49: Características de un filtro pasa alto

Fuente Escuela Técnica de Ingenieros Industriales de Valencia

Amplificador operacional con filtro de paso bajo

La siguiente etapa consistirá en un filtro activo de paso bajo, es decir, un amplificador operacional en este caso con un LM358 con un filtro de paso bajo. Se puede observar que este circuito permite el paso de bajas frecuencias mientras que las altas frecuencias son atenuadas. Esta etapa se encarga de eliminar el ruido existente incluyendo los 50 Hz de la red de alimentación que representa una se amplifica la débil señal que nos llega del sensor.

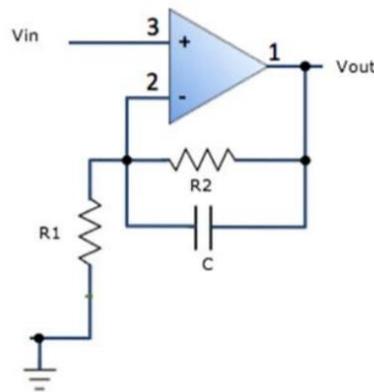


Figura 50: Amplificador operacional con filtro de paso bajo.

Fuente Escuela Técnica de Ingenieros Industriales de Valencia

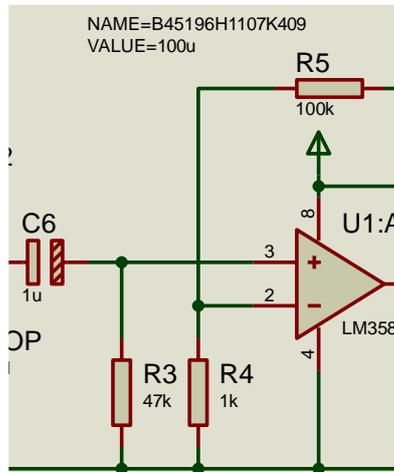


Figura 51: Etapa de Filtrado con un Amplificador

Realizado por el Autor

Amplificador seguidor no inversor

En último lugar se coloca un amplificador configurado como seguidor no inversor, también llamado buffer, se utiliza para eliminar los efectos de carga a la salida del amplificador, es decir disminuir la impedancia en su salida. Esta etapa es de gran ayuda si después se pretende digitalizar la señal con un conversor analógico-digital como es el caso en el presente proyecto. También se añade un seguidor entre el potenciómetro y el segundo filtro de paso alto para que la frecuencia de corte calculada en el primer filtro de paso alto se mantenga. Esta secuencia de etapas se repite de nuevo con el objetivo de obtener una ganancia igual ya que la señal pasa por dos de estas últimas etapas, pero en la salida de la primera secuencia, es decir a la salida del amplificador, se coloca un potenciómetro por si se quisiese una ganancia inferior a los 10201.

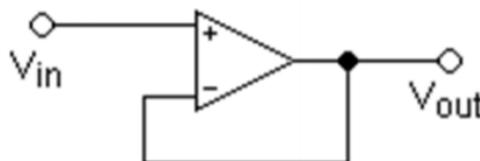


Figura 52: Amplificados seguidor no inversor

Fuente Escuela Técnica de Ingenieros Industriales de Valencia

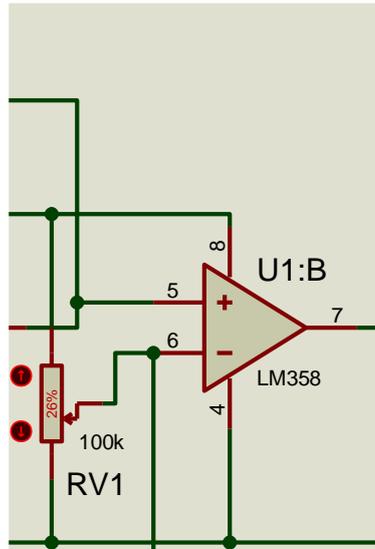


Figura 53: Amplificadores seguidor no inversor en el Circuito

Realizado por el Autor

Programación del PIC 12F683

La familia PIC12F683 de microcontroladores se basa en el núcleo de arquitectura media de Microchip con una pila de hardware de 8 niveles de profundidad y 35 instrucciones.

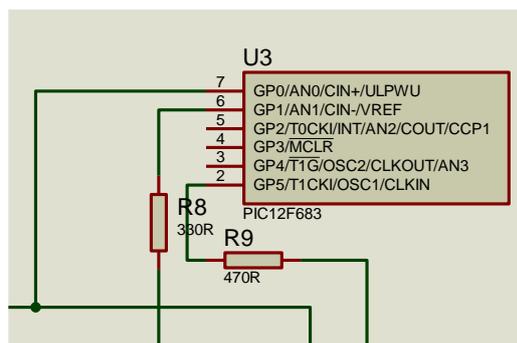


Figura 54: Conexión PIC 12F683 en el circuito

Realizado por el Autor

Programación del Microcontrolador PIC en Proton Basic es un simple y fácil lenguaje de programación. Tiene solo unas pocas normas y el control de estructuras que definen su gramática.

Estructura del programa en PIC Basic

- Programa de cabecera
- Declaraciones
- Explicación de los signos e identificadores
- Declaraciones y Comandos

La programación comienza en la parte superior, y la continua hacía. Sin embargo, permite las repeticiones. las primeras líneas del programa difieren en la memoria de los microcontroladores, EPROM, numero de puertos y registros, es necesario informar al compilador sobre el microcontrolador que se utilizara. En segundo lugar, la velocidad de procesamiento depende de la frecuencia del cristal. Por lo tanto, a fin de calcular con precisión del pulso cardiaco también es necesario informar al compilador de la frecuencia del cristal

Declaraciones de variables

Son instrucciones especiales sobre los diversos dispositivos a utilizar, esto ayuda al compilador a generar instrucciones específicas.

Identificadores

Los identificadores son símbolos de texto especial que se utiliza para representar algo, pueden ser utilizados como etiquetas para marcar ciertos lugares en el programa, modo que el programa se puede hacer para ir a la etiqueta y luego continuar con el programa a partir de entonces

Variables

Las variables se utilizan para almacenar datos temporalmente o para mantener números que se utilizarán en los cálculos. El número de variables que serán utilizadas en el programa dependen de la memoria RAM del microcontrolador, la arquitectura del microcontrolador, la parte de la memoria RAM, la memoria del programa.

Matrices

Para declarar una matriz, tenemos que mencionar el concepto de utilizar múltiples variables, con el mismo nombre, pero que tengan un número de índice que los refiera. Dado que un número de índice en sí mismo puede ser una variable, es más fácil caminar a través de un arsenal enorme de variables, cambiando simplemente el índice.

La Figura 47 muestra la programación del PIC 12F683 en las cuales se detallan las librerías utilizadas, los puertos necesarios para su conexión y posterior funcionamiento la parte importante es la línea de código que compara el valor de entrada con el valor establecido.

```

'* Name      : UNTITLED.BAS *
'* Author    : [select VIEW...EDITOR OPTIONS] *
'* Notice    : Copyright (c) 2020 [select VIEW...EDITOR OPTIONS] *
'*           : All Rights Reserved *
'* Date      : 03/12/2020 *
'* Version   : 1.0 *
'* Notes     : *
'*           : *
'*****
;-----
;**** Added by Fuse Configurator ****
; Use the Fuse Configurator plug-in to change these settings

Device = 12F683

Config FOSC_INTOSCIO, WDTE_OFF, PWRTE_ON, MCLRE_OFF, CP_OFF, CPD_OFF, BOREN_OFF, IESO_ON, FCMEN_ON

;**** End of Fuse Configurator Settings ****
;-----
;Velocidad de reloj
Xtal 4
;Configuracion Principal
OSCCON = %01110111
All_Digital 1
;Variables
Dim cuenta As Word

;configuracion de puertos
TRISIO.0 = 1
TRISIO.1 = 0
TRISIO.5 = 0

High GPIO.1
DelayMS 50
Low GPIO.1
DelayMS 100
High GPIO.1

DelayMS 50
Low GPIO.1

Main:
  cuenta = Counter GPIO.0, 5000
  If cuenta >= 20 Then
    High GPIO.1
    High GPIO.5
  Else
    Low GPIO.1
    Low GPIO.5
  EndIf
  GoTo Main

```

Figura 55: Estructura de código en PROTON IDE, PIC 12F683

Fuente: Realizado por el Autor

Ver Anexo B de esta investigación para más detalles de las líneas de código utilizadas en la programación del PIC 12F683 en el entorno de programación Proton IDE.

El lugar más apropiado para tomar las señales del pulso es en los dedos gracias a la comparación que tenemos en la Figura 51 haciendo un análisis entre las orejas, los dedos y lóbulos para esto en las Figuras 49, 50 y 52 detallan la correcta medición óptica aplicada al lugar establecido de acuerdo con la investigación.

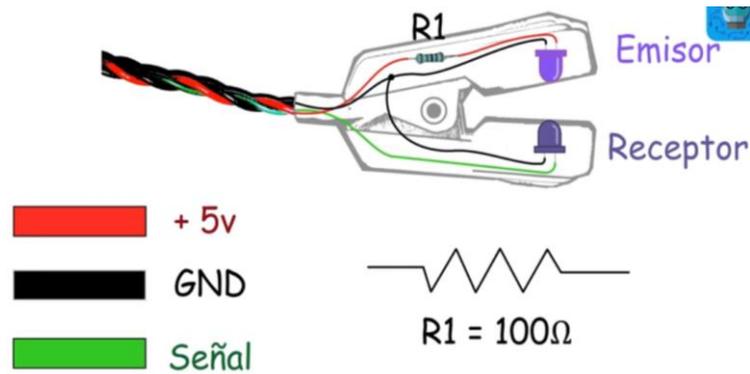


Figura 56: Detector sensor de pulso

Fuente: Realizado por el autor

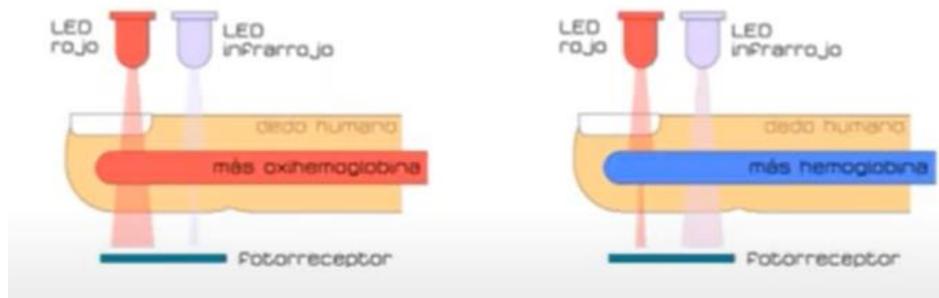


Figura 57: Medición óptica de Diodos LED

Fuente: Wilfrido Vargas Celi Pulsímetro

El corazón bombea la sangre varias decenas de veces al minuto, en cada segundo el nivel de sangre de la zona concreta que estamos midiendo varia. Y así, latido tras latido, el procesador de nuestro pulsímetro es capaz de determinar los latidos por minuto de nuestro cuerpo, el propio bombeo de sangre de nuestro corazón hace que la luz reflejada casi en cada instante. Estos cambios permiten al dispositivo realizar una medida estimada de las pulsaciones por minuto, todo ello con un nivel de precisión bastante aceptable. [32]

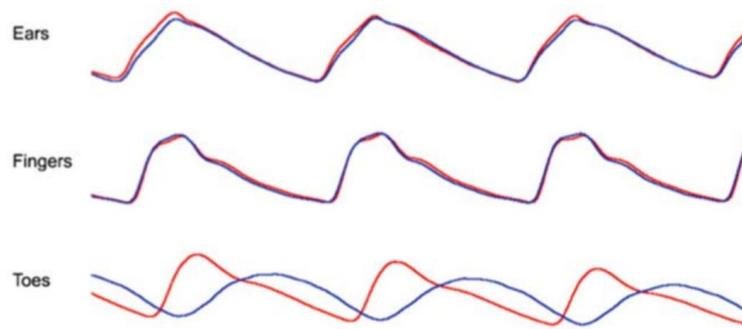


Figura 58: Señal de pulso cardiaco obtenida de diferentes partes del cuerpo

Fuente: Diseño e implementación de un pulsómetro de la Escuela Técnica Superior de Ingenieros Industriales de Valencia

En este circuito electrónico se escogió la configuración como presenta en la Figura 57 de transmisión ya que la mayoría de los sensores que existen tienen esta configuración debido a que es más fácil obtener una señal coherente con el pulso, pero debido al agarre de la barra de peso los dedos no dan lugar sin presión, en el comportamiento del ritmo, presión y frecuencia cardiaca de los levantadores de pesas de alto nivel competitivo se observó la tendencia saludable. En reposo las pulsaciones por minuto de los levantadores de pesas con años de entrenamiento y rendimiento deportivos fue como promedio de 62 latidos por minuto (62BPM) lo que conlleva a una configuración estimada de 70 pulsaciones por minuto en el lapso de 25000 milisegundos para una respuesta oportuna y evitar que el individuo sufra daños.[32]

Ver Anexo C para para ver programación del sensor y adquisición de datos.

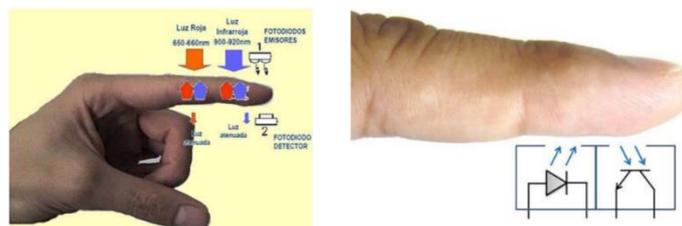


Figura 59: Configuración para obtener la señal de pulso.

Fuente: Diseño e implementación de un pulsómetro de la Escuela Técnica Superior de Ingenieros Industriales de Valencia

En las siguientes figuras se muestra la colocación de cada uno de los elementos del sistema electrónico de monitoreo con el objetivo de dar comodidad al momento de

utilizar dicho sistema ya que es un ejercicio que requiere de concentración. En las Figuras 53 y 54 muestran al prototipo del circuito colocado en la muñeca del deportista, lugar donde ofrece comodidad y correcta operabilidad del sistema



Figura 60: Placa electrónica del sistema electrónico para el monitoreo de pulso cardiaco

Fuente: Realizado por el autor

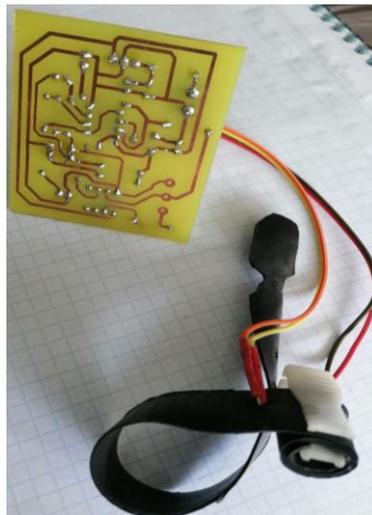


Figura 61: Pistas de la placa electrónica prototipo 1 del sistema electrónico para el monitoreo de pulso cardiaco

Fuente: Realizado por el Autor

En las Figuras 55 muestra al sistema de monitoreo de pulsos cardiacos se adapta a un guante deportivo para una mejor maniobrabilidad del sistema mejorando así la colocación de los diodos LED en el dedo disminuyendo la probabilidad de existencia de error en la toma de datos.



Figura 62: Prototipo mejorado del sistema Electrónico Implementado en guante de entrenamiento deportivo parte posterior con una caja de protección

Realizado por el autor



Figura 63: Prototipo Mejorado del sistema Electrónico Implementado en guante de entrenamiento deportivo parte Interior

Realizado por el autor

En la Figura 56 permite observar el circuito en funcionamiento una vez implementado en el guante deportivo permitiendo el agarre de la barra con normalidad.



Figura 64: Prueba para el agarre de levantamiento de potencia

Fuente Realizado por el Autor

En la Figura 57 muestra la etapa de pruebas del circuito de monitoreo de pulso cardiaco y el algoritmo monitoreando en tiempo real logrando que el sistema se adapte al entrenamiento que requiere el deportista el ambiente en el que se tomó esta fotografía es el lugar donde se desarrolló el sistema lugar en el que existe iluminación favorable.



Figura 65: Prueba del Sistema en Ambiente 2

Fuente: Realizado por el Autor

La Figura 59-A muestra que el esqueleto virtual se mantiene en una correcta posición la cual permite diferenciar la mala ejecución del ejercicio de levantamiento de potencia por parte del deportista esta acción permitió que el deportista tome una posición correcta evitando futuras lesiones y aprovechando al máximo el ejercicio como muestra la Figura 59-B.



A



B

Figura 66: Mala postura monitoreada en tiempo real

Realizado por el Autor

La Figura 60 muestra el ambiente adaptado para el deporte y el levantamiento de potencia el sistema electrónico de monitoreo funciona en una forma correcta.



Figura 67: Prueba real del circuito en entrenamiento de potencia

Realizado por el Autor

La ventana de visualización muestra el óptimo funcionamiento del algoritmo, la Figura 61 permite observar cuando el deportista realiza la ejecución del levantamiento de potencia desde su posición inicial y la posición final.



Figura 68: Prueba real del sistema Electrónico

Fuente: Realizado por el autor

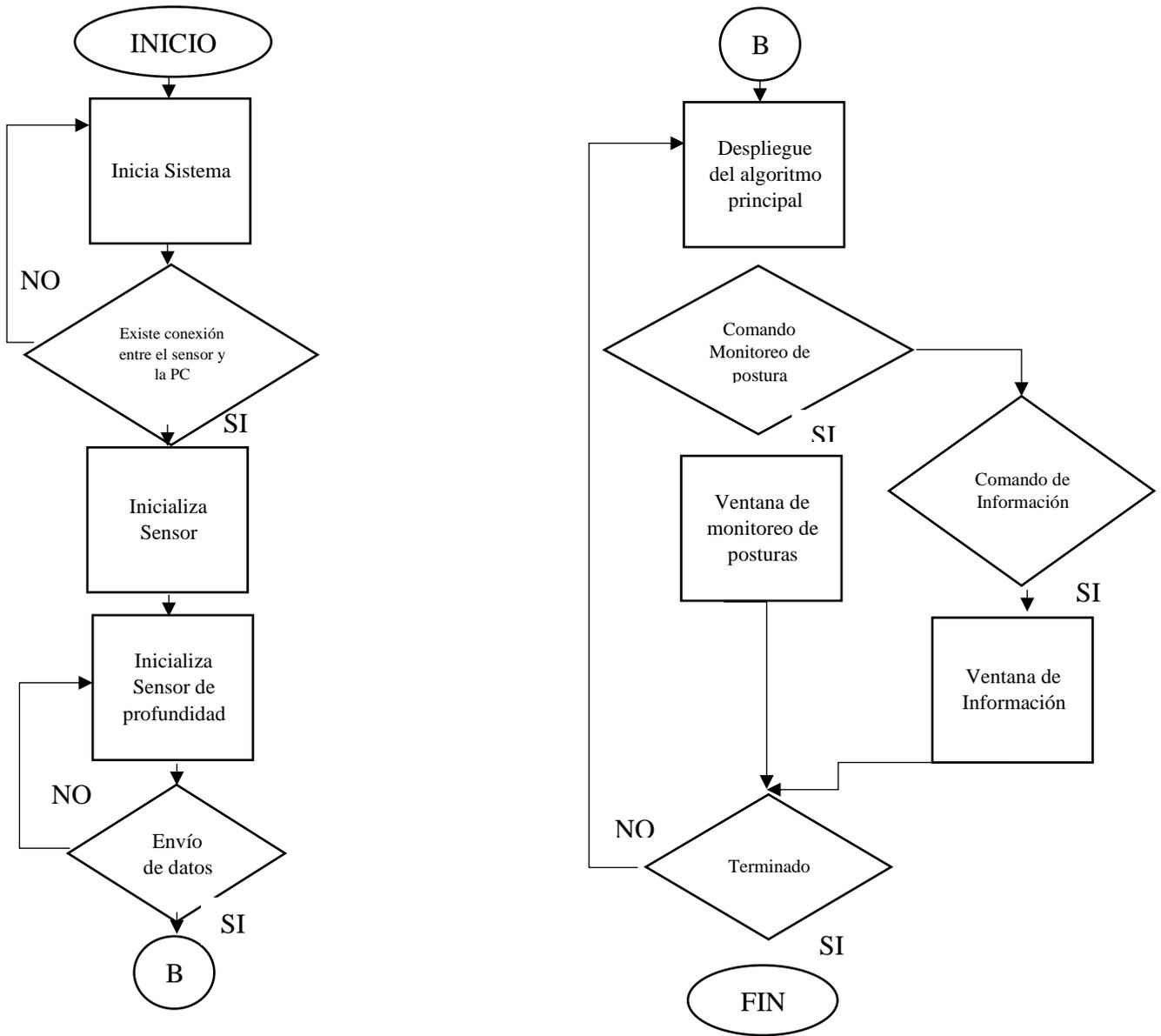


Figura 69: Diagrama de interacción del sistema

Fuente: Realizado por el Autor

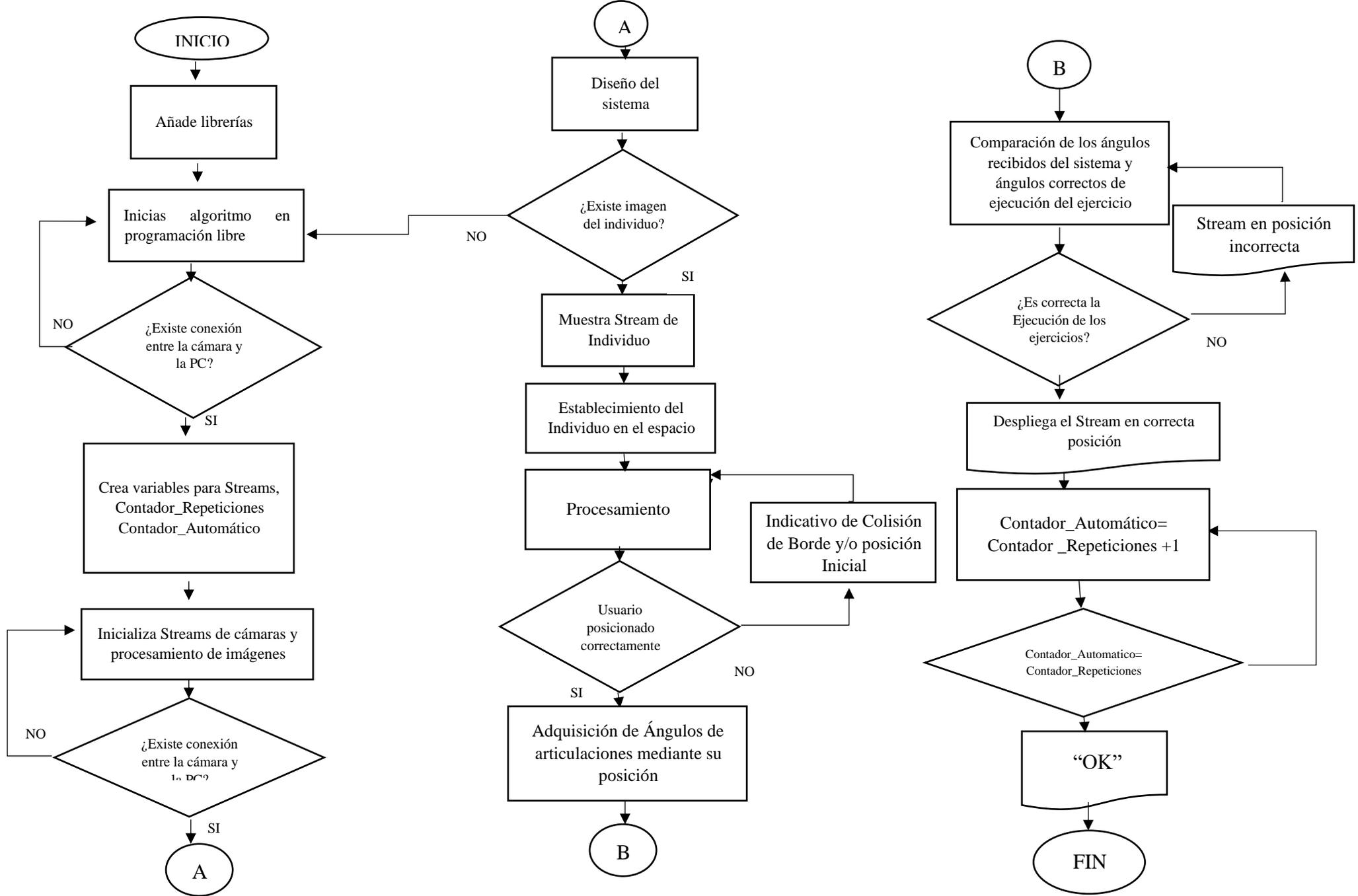


Figura 70: Diagrama de Flujo: Sistema para corrección de posturas

Fuente: Realizado por el Autor

3.3 Pruebas y Resultados

Etapa de pruebas del monitoreo en tiempo real, en la Ilustración 61 observamos como el avatar o esqueleto digital creado por el algoritmo tiene una clara eficiencia de monitoreo al no mantener la posición correcta el individuo tiene un ligero desnivel en el punto del cuello, movimiento que podría causar una lesión al momento de realizar el levantamiento de potencia.

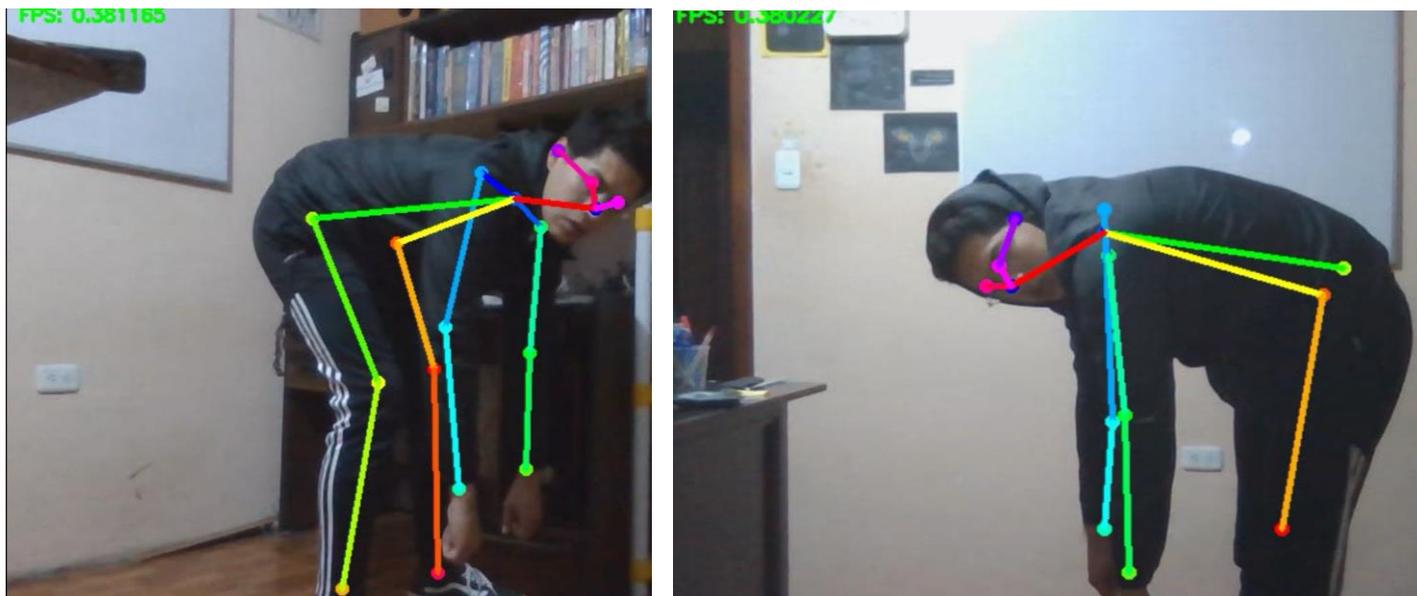


Figura 71: Estimación de pose usando Tensorflow y OPENCV con incorrecta posición

Fuente: Realizado por el Autor

Etapa de pruebas del monitoreo en tiempo real, en la Figura 65 observamos como el avatar o esqueleto digital creado por el algoritmo tiene una clara eficiencia de monitoreo al mantener la posición correcta el individuo tiene recta la columna y en el punto del cuello se alinea con el tren inferior del cuerpo al momento de realizar el levantamiento de potencia.

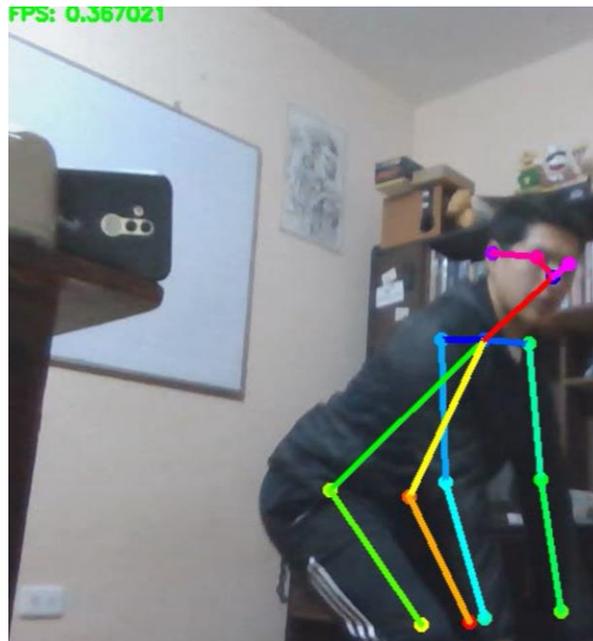


Figura 72: Estimación de pose usando Tensorflow y OPENCV con correcta posición

Fuente: Realizado por el Autor

Repetición en cada serie	Porcentaje de peso Máximo	Pulso
3	60%	149
3	70%	168
3	70%	166
2	75%	168
2	80%	170
2	85%	162
1	90%	157
1	90%	153
1	90%	152
Promedio de pulsaciones por minuto		160

Tabla 17: Respuesta promedio a la carga de entrenamiento con pesas

Fuente: Realizado por el Autor

Con la finalidad de evaluar el sistema electrónico de monitoreo en distintos escenarios se presenta la Tabla 17 donde se indican los porcentajes de peso máximo realizadas en 3 series con repeticiones de 10, actividad que certifica la eficiencia en la detección de posturas correctas para el ejercicio de levantamiento de potencia a lo largo de las 5 pruebas que se realizaron para la comprobación de dicho sistema.

Como ya se mencionó se realizaron pruebas en 4 ambientes diferentes a distintos tipos de iluminación los cuales se contienen en las gráficas; abarcando uno de los objetivos específicos del proyecto de investigación, el cual precisa realizar pruebas de funcionamiento y validación del sistema electrónico de monitoreo en la ejecución de ejercicios físicos para la correcta postura presentada y la preestablecida en el sistema.

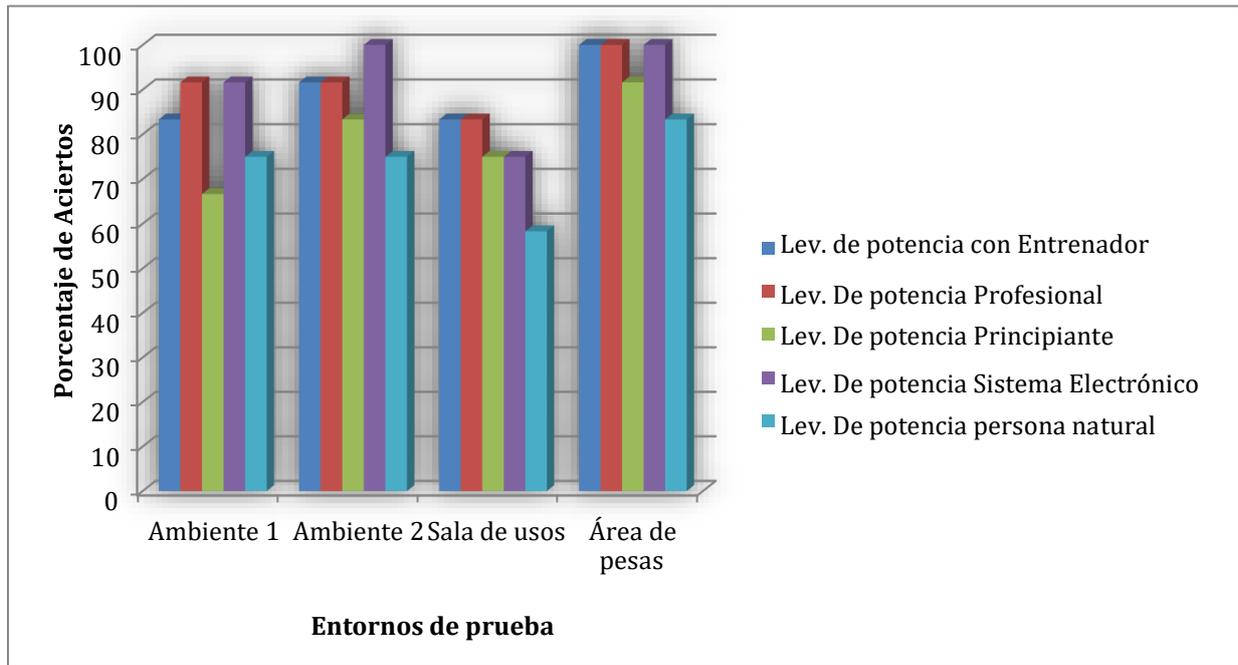


Figura 73: Porcentaje de aciertos en las pruebas del sistema

Fuente: Realizado por el Autor

Se llevaron a cabo pruebas de cuatro series por cada ejercicio comprendido por 10 repeticiones, tomando como muestra principal dos series de las cuatro que se efectuaron en total.

Cabe mencionar que las pruebas realizadas fueron realizadas dentro de los espacios en la casa del autor y espacio deportivo adaptado con fines de investigación para este proyecto debido a que no puede haber aglomeración de personas o asistir a lugares públicos por el estado de pandemia por COVID-19 dichas pruebas estuvieron supervisadas por un profesional del entrenamiento físico.

	Lev. De potencia Profesional	Lev. De potencia Sistema Electrónico	Lev. De potencia Principiante	Lev. de potencia con Entrenador	Lev. De potencia persona natural
Ambiente 1	20 (83.33%)	22 (91.63%)	16 (66.64%)	22 (91.63%)	18 (74.97%)
Ambiente2	22 (91.63%)	22 (91.63%)	20 (83.33%)	24 (100%)	18 (74.97%)
Sala de usos	20 (83.33%)	20 (83.33%)	18 (74.97%)	18 (74.97%)	14 (58.31%)
Área de pesas	24 (100%)	24 (100%)	22 (91.63%)	24 (100%)	20 (83.33%)

Tabla 18: Datos (Repeticiones y porcentaje) de repeticiones correctas en pruebas.

Realizado por el Autor

3.3.1 Análisis de Errores

Técnica del Agarre: Durante la observación en la ejecución del levantamiento se evidencio una mala medición del agarre en la barra como los brazos mucho más extendidos a la anchura normal de las caderas lo cual generaba tensión en los hombros provocando dolor. Otro error fue que los pesistas no se guiaban en las marcas de agarre que se encuentran en el centro de la barra separado cada una a 810mm generando una descompensación a la hora del levantamiento corriendo el riesgo de fuertes lesiones en la columna.

Calificación	Satisfactorio	Algo Satisfactorio	Nada satisfactorio
Deportista 1		X	
Deportista 2		X	
Deportista 3		X	
Deportista 4	X		
Deportista 5			X

Tabla 19: Tabla de evaluación para la técnica del Agarre

Realizado por el autor

Respiración antes de realizar el movimiento: Uno de los grandes problemas evidenciados fue una mala técnica de la respiración, en un peso muerto debe realizarse la respiración diafragmática que consiste en enviar el aire inhalado al abdomen para formar un cinturón fisiológico de protección que es generado por una contracción del recto abdominal para que a la hora de usar un cinturón de levantamiento logre generar la resistencia a la fuerza isométrica generada por el músculo recto anterior y generar una compresión en el compartimiento extensor de la zona lumbar de la columna. Este desconocimiento de los pesistas sobre los beneficios de una correcta respiración ocasionaba una fuerte tensión en la musculatura lumbar y dorsal dando resultado una fatiga en cada serie y contracturas musculares.

Calificación	Satisfactorio	Algo Satisfactorio	Nada satisfactorio
Deportista 1			X
Deportista 2			X
Deportista 3			X
Deportista 4		X	
Deportista 5		X	

Tabla 20: Tabla de evaluación en la respiración antes de realizar el movimiento

Realizado por el autor

Posición de la espalda durante el movimiento: Al ejecutar el levantamiento los deportistas presentaban una exagerada curvatura de la región lumbar otros en cambio presentaban curvaturas en la región torácica en la posición inicial y estas curvaturas se volvían más pronunciadas en la fase concéntrica del ejercicio que inhibe la fuerza y potencia generada por los músculos de tracción debido a una compresión de la columna.

Calificación	Satisfactorio	Algo Satisfactorio	Nada satisfactorio
Deportista 1			X
Deportista 2		X	
Deportista 3		X	
Deportista 4	X		
Deportista 5	X		

Tabla 21: Tabla de evaluación de la posición de la espalda durante el movimiento

Realizado por el autor

Donde solo 1 deportista que representan el 20% está evaluados de satisfactorios, con evaluación de Algo satisfactorio 40% de la muestra escogida que son 2 deportistas y los 2 deportistas restantes evaluados de nada satisfactorio que representan el 40%.

Teniendo la información anterior en cuenta se pudo observar que los resultados variaron acorde al tipo de entorno en el que se realizaron las pruebas en este caso es el área de pesas en donde es satisfactorio realizar este ejercicio, así mismo el ambiente 2 donde se desarrolló el sistema electrónico tuvo un correcto funcionamiento debido a que hubo la correcta iluminación para la toma de datos y monitoreo cardíaco óptimo con una variación de error entre un 17% a 25% respecto al ambiente óptimo, para dicho estudio se tuvo la participación de 5 deportistas con experiencia en el levantamiento de potencia.

Estos resultados son explicados debido a las especificaciones tanto de uso, como de Software del algoritmo conectado a una cámara externa marca ELP el cual indica que debe existir un ambiente iluminado adecuadamente, así como el hardware que indica que el individuo debe estar en con el sensor de pulso bien colocado, también el individuo debe encontrarse de pie frente a la cámara ya que puede haber ocultamiento o sobre posicionamiento en la detección de articulaciones por el algoritmo el cual trabaja obviando estas fallas

Finalmente se observó que gracias al desarrollo de este algoritmo el sistema hace una grabación para respaldar el entrenamiento realizado por el deportista el cual después de una sesión de ejercicio puede acudir y ver el progreso si es el caso con la ventaja de mejorar ya que es su propio cuerpo el cual está observando y tendrá respaldo al momento de realizar la postura inicial en el entrenamiento logrando grandes avances en la salud y superación en el deporte, comprobando así otro de los objetivos específicos que es disminuir el porcentaje de lesiones en los deportistas al momento de realizar el levantamiento de peso muerto.

3.4 RECURSOS

3.4.1 Institucionales

- Bibliotecas online
- Repositorios Digitales

3.4.2 Humanos

- Investigador.
- Docente Tutor.

3.4.3 Materiales

- Internet
- Computadora Personal
- Publicaciones
- Libros
- Revistas
- Impresora
- Software de Código Abierto
- Fuente Regulada de Voltaje
- Cámara
- Tarjeta con tecnologías programable
- Monitor
- Sensor cardiaco
- Placa Virgen
- Estaño
- Taladro
- Cautín
- Tarjeta microSD
- Cable Ethernet

SOFTWARE

- Software libre

- Librería OpenCV Anaconda Python

3.5 Económicos

Se muestra los materiales que se utilizarán para el prototipo, así como sus precios y el total de gastos que se requiera para el desarrollo de este, existen gastos que surgirán el momento de su implementación por tanto en la Tabla 22 se describe el gasto de imprevistos.

Dicho prototipo será financiado por el Investigador.

Item	Descripción	Unidad	Precio Unitario	Precio Total
1	Tarjeta Programable	1	\$ 70	\$ 70
2	PIC 12F683	2	\$ 6	\$ 12
3	Memoria USB	1	\$ 15	\$ 15
4	Cables de conexión	1	\$ 3	\$ 3
5	Memoria SD	1	\$ 15	\$ 15
6	Sensor Cardíaco	3	\$ 5	\$ 15
7	Monitor	1	\$ 330	\$ 330
8	Internet	1	\$ 40	\$ 40
9	Resistencias	12	\$0.05	\$0.60
10	Placa Virgen	1	\$ 2	\$ 2
11	Amplificador LM358	4	\$ 1.80	\$ 7.20
12	Led Infrarrojos	6	\$0.10	\$0.60

13	Potenciómetro	2	\$0.60	\$1.20
14	Capacitor	2	\$0.40	\$0.80
15	Imprevistos (30%)			\$ 45
	TOTAL			\$ 557.40

Tabla 22: Presupuesto del proyecto

Fuente: Realizado por el Autor

CAPITULO IV

4.1 Conclusiones

- El objetivo principal del proyecto de investigación de un sistema de visión artificial para realizar la corrección de posturas en el ejercicio de levantamiento de potencia mediante el desarrollo de un algoritmo, la utilización de una cámara y el monitoreo cardiaco se puede concluir que se ha cumplido con este objetivo de manera satisfactoria, manteniendo los grados de flexión en los rangos correctos de dicho ejercicio y establecidos en los estándares de los organismos de salud y deporte.
- Además, las herramientas adicionales como el sensor de monitoreo cardiaco y el estudio del comportamiento del corazón bajo actividad física sirvieron para la correcta realización del levantamiento de potencia en un ambiente real de un gimnasio, ya que proporcionó de manera concisa una explicación de las posturas de dicho ejercicio y permitió el principal enfoque a la ejecución de levantamiento de potencia.
- Con la ayuda este proyecto de investigación puede ser el apoyo para investigaciones futuras en la aplicación de las Ingeniería Electrónica al deporte y la actividad física por ejemplo suplir de manera definitiva a un instructor profesional que pueda además de corregir las posturas del levantamiento de potencia, corregir posturas en máquinas de ejercitación o en las distintas poses que la actividad física requiere, también, se puede dotar al sistema con un apartado que pueda incluir dietas de nutrición, rutinas para incrementar diferentes necesidades llevando un seguimiento de avance en el rendimiento físico, con información y estadísticas de porcentaje de grasa corporal, estatura, índice de masa muscular y cualquier otra necesidad en el deporte. Además, se aplicaría en innumerables campos de avance tecnológico que ayude al desarrollo del país como en la industria de la seguridad y protección social aplicando esta investigación en herramientas tecnológicas que aporten al avance Tecnológico.
- Finalmente, se encontraron factores de riesgo en la ejecución de esta técnica corroborando que el 100% de la población de deportistas que realizan el

levantamiento de potencia sufren sintomatologías dolorosas y lesiones, gracias al sistemas de monitoreo en tiempo real se logró una disminución considerable de un 95% de este porcentaje y los daños en dichos deportistas fueron menores y aumentó su rendimiento en sus actividades físicas.

4.2 RECOMENDACIONES

- Durante la realización de este proyecto se tuvo problemas con el desarrollo del algoritmo ya que los entornos de programación no tenían compatibilidad en algunos aspectos en manejo de archivos, ya que este tipo de programación requiere de una conexión a través de instrucciones, en este caso lo que se realizó fue administrar todos los recursos que adquieran archivos compartidos para dar paso a la ventana de resultado final así evitar también la saturación de memoria y limpia datos innecesarios.
- Se determino al inicio del proyecto que tipo de entorno de programación se utilizaría para el desarrollo del algoritmo, pero al momento de elegir nuevas herramientas para avanzar con nuestro proyecto hubo conflicto ya que se alojaban con versiones menores a las que requerían para su comunicación es por eso por lo que recomiendo instalar versiones compatibles entre sí para un ambiente de desarrollo sin complicaciones.
- El lento procesamiento de imágenes especialmente cuando se utilizan variables de forma básica, puede solucionarse eventualmente con un dispositivo de mayor capacidad y con un mejor procesador. Esto debido a que cualquier reconocimiento de más de un individuo puede disminuir la precisión del algoritmo.
- La ubicación del sensor para la adquisición de la señal del pulso cardiaco se debe instalar en un lugar estratégico donde obtenga menor probabilidad de error ya que la presión que ejerce las manos con el agarre de la barra conlleva a que el flujo sanguíneo puede hacer fallar al sensor y arrojar datos erróneos.

BIBLIOGRAFÍA

- [1] Ramírez de Arellano. "Industrial como ejemplo evolutivo de la industria 4.0". *Visión Artificial*. Madrid. (2017). Disponible en: <https://blog.infaimon.com/vision-artificial-industrial-industria-4-0/>
- [2] Gottau, G. "Ejercicios de peso". *Tendencias Hombres*. (2014). Disponible en: <https://www.tendenciashombre.com/fitness/pesomuertoun-ejercicio-basico-keno-debe-faltar-en-tu-rutina>
- [3] S, Amaya, D.P. Velásquez, I. D. Torres. "Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema". *Seleccionador de Productos de un Centro Integrado de Manufactura (CIM)*. Medellín Colombia. (2016).
- [4] N. López Pérez y. J. Toro, "Técnicas de Biometría Basadas en Patrones Faciales del ser Humano", Pereira: Universidad Tecnológica de Pereira, (2012). Disponible en: <http://repositorio.utp.edu.co/dspace/handle/11059/2738>
- [5] M.T. Calderón, D.J. Maldonado "Control por visión de un cuadricóptero utilizando ROS", Tesis, Ing. Electrónica y Eléctrica, Escuela Politécnica Nacional, Quito, Ecuador, 2014.
- [6] J. M. Martínez, "Sistema de Visión Artificial para la Detección y Corrección de Posturas en Ejercicios realizados por Fisicoculturistas". UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO, Centro Universitario UAEM Atlacomulco (2018). E
- [7] D. Martínez Mejía. "Interceptor de Trayectorias Basado en visión Artificial". Pontificia Universidad Javeriana Carrera de Ingeniería Electrónica de Bogotá (2005)
- [8] N. Aguirre Dobernack, "Implementación de un sistema de detección de señales de tráfico mediante visión artificial basado en FPGA". Procesamiento de Imágenes Capítulo 3, Ingeniería Electrónica y Eléctrica, Escuela Politécnica Nacional, Quito, Ecuador, 2014. Disponible en: http://bibing.us.es/proyectos/abreproy/12112/fichero/Documento_por_capitulos%252F3_Cap%C3%ADtulo_3.pdf
- [9] U. d. Vigo, "Sistema Visual Humano". Universidad de Vigo. 18 06 2010. [En línea]. Disponible en: xuliofh.webs.uvigo.es/Web-Tv/SVH.pdf

- [9] E. García, “Visión Artificial”. «etitudela,» 18 08 2011. [En línea]. Disponible en: www.etitudela.com/celula/downloads/visionartificial.pdf.
- [10] J. Ibane, “Visión Artificial por computadora”. «Instituto de Navarra,» 25 06 2013. Disponible en: www.iessanjuanibane.educacion.navarra.es/web/.../visionArtificial.pdf.
- [11] U. A. d. Barcelona. “Visión por Computador”. «Universidad Autónoma de Barcelona,» 15 07 2009. [En línea]. Disponible en: <http://www.uab.cat/web/entidadesagregadas/institutosycentrosdeinvestigacion/instituto-centro-deinvestigacion/centro-devisionporcomputadorcvc1345467957854.html>
- [12] J. Narváez, “Sistemas de visión artificial en la industria”. «infaomon,» 10 09 2012. [En línea]. Disponible en: <http://www.infaimon.com/es/sistemas-vision-integradosindustria.pdf>.
- [13] J. Vinuesa, “Calidad de la Industria Actual”. «Jasvisio,» 06 11 2010. [En línea]. Disponible en: <http://www.jasvisio.com/control-de-calidad-industrial-convisionartificial.html>.
- [14] S. Fiorelli, “Visión Artificial Aplicación Práctica”. «Optoelectronics,» 21 02 2009. [En línea]. Disponible en: http://visionartificial.fpcat.cat/wpcontent/uploads/UD_1_didac_Conceptos_previos.pdf.
- [15] M. Marchena, “Sensor Digital”. «digital fotored,» 05 06 2013. [En línea]. Disponible en: www.digitalfotored.com/imagendigital/sensorimagen.htm.
- [16] axis, “Cámara y sensores”. «axis,» 19 07 2012. [En línea]. Disponible en: www.axis.com/es/products/video/about.../sensors_cmos.htm.
- [17] U. d. Barcelona, “Lenguaje de programación en la actualidad”. «Universidad de Barcelona,» 18 04 2011. [En línea]. Disponible en: <http://www.ub.edu/javaoptics/teoria/castella/node3.html>.
- [18] J. Howse, *OpenCV Computer Vision with Python*. Olton, UNITED KINGDOM: Packt Publishing, Limited, 2013

- [19] U. A. d. Madrid, "Procesamiento de imágenes". «Universidad Autónoma de Madrid,» 22 06 2013.
[En línea]. Disponible en: www.ii.uam.es/~siguenza/procesamiento.pdf.
- [20] U. JAEN, "Detección de una imagen" «Universidad JAEN,» 10 07 2008. [En línea]. Disponible en: www4.ujaen.es/~satorres/practicass/practica3_vc.pdf.
- [21] L. Fabiani, "Alojamientos tecnológicos" 07 05 2009. [En línea].
Diponible en: <http://www.alojamientos.us.es/gtocomap/pid/tema4.pdf>.
- [22] H. Anderson, "Ingeniería y Programación" «softingenio,» 14 08 2009. [En línea].
Disponible en: <http://www.softingenio.com/tecnología/hardware/51.../56visionartificial.htm>
- [23] Viva Gym. "Todo lo que debes saber sobre ejercicios de peso muerto". 2010.
Disponible en: <https://blog.vivagym.es/2017/08/23/todo-loquetienes-quesaber-sobre-la-tecnica-del-peso-muerto/>
- [24] J.D.Cruz, "Clasificador automático de tapas rosca de plástico para procesos de producción continua, basados en la inspección de defectos superficiales", Tesis, Universidad Técnica del Norte, Ibarra, Ecuador, 2010
- [25] L.Gomez, M.Lenis, A.Villalba. "Factores de riesgo posturales dinámicos y de la ejecución técnica para lesiones osteomusculares en levantadores de pesas" Educación Física y Deporte Universidad de Antioquia. 26 enero 2017
- [26] F.Velasco. "El Mejor Software de Inteligencia Artificial (IA)". Tienda Digital. 28 dic. Santiago de Chile. Disponible en: <https://tienda.digital/2018/12/28/el-mejor-software-de-inteligencia-artificial/>
- [27] ELP (AILIPU TECHNOLOGY CO.LTD) Fabricante de cámaras para Videos Vigilancia. Disponible en: <http://www.webcamerausb.com/>
- [28] Python Software Foundation License. "Guía Ubuntu Entorno de Desarrollo".Disponible en: <https://ubunlog.com/spyder-entorno-desarrollo-python/>
- [29] C. Sanchez, J. Colomer, J. M. Olivencia, A. Creativa, M. Ramos, A. Rincón, A. Romero, A. Valdez, A. Ávila, A. Real, A. Romero, B. Bandera, B. Ruibal, C. Gallarda, D. D. Gil, P. Ferreira, F. H. Sisters, G. Castro, I. Del Rio, I. Montalbán, I. Sotelo, Joan, J. J. Pérez, J. Gonzales, J. Pozo, L. Barrios, M. J. Garcia, P. Sala, M. Muñoz, N. FitCook, P. Sanchez, P. Quintana, P. Garcia, S. Craviotto, S. Espinar, Tandem, F. Miron, M A. Chaves, N. Camps, P. Escalón, P. Explosive, R. Mora,

- Science4Fitness. "Peso Muerto: Todo lo que debes Saber". 2018. Disponible en: <https://www.hsnstore.com/blog/deportes/fitness/peso-muerto-todo-lo-que-debes-saber/>
- [30] G. Panataleo, L. L. Rinaudo. Ingeniería de Software. AlfaOmega. Buenos Aires. Argentina. 2015 disponible en: <https://catalogo.cecar.edu.co/cgi-bin/koha/opac-detail.pl?biblionumber=28212>
- [31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, Greg S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Yuan Yu, X Zheng. 2015. Software available from [tensorflow.org](https://www.tensorflow.org). Tensor Flow. Disponible en: https://www.tensorflow.org/lite/models/pose_estimation/overview
- [32] J. Ousterhout. TCL Developer Xchange. Mountain View California. 1998. Disponible en: <http://www.swig.org/exec.html>
- [33] A. Herrera Corzo. Respuesta Cardíaca a la carga de entrenamiento. EFDeportes.com, Revista Digital. Buenos Aires. Argentina. 2019
- [34] M. J. Epstein, J. F. Manzoni, and M. P. Bates, *Interfacing PIC Microcontrollers : Embedded Design by Interactive Simulation*. Jordan Hill, UNITED KINGDOM: Elsevier Science & Technology, 2006
- [35] S. Dawoud and R. Peplow, *Digital System Design : Use of Microcontroller*. Aalborg, DENMARK: River Publishers, 2010
- [36] J. Vetelino and A. Reghu, *Introduction to Sensors*. Bosa Roca, UNITED STATES: Taylor & Francis Group, 2010
- [37] Academic Resource Center, "SENSORS: Types and Characteristics Academic Resource Center."
- [38] E. Sadun, *Digital Video Essentials : Shoot, Transfer, Edit, Share*. Alameda, UNITED STATES: Wiley, 2003.
- [39] C. Maker. Proton IDE entorno programación. 2018.Disponible en: <https://mecaelectronics.weebly.com/proton-ide.html>
- [40] J. Wu, "Convolutional Neural Networks," 2020, doi:10.5220/0006972204340439.
- [41] Dar85112 Programación en Proton IDE plus. 2012. México. Disponible en: <https://es.slideshare.net/dar85112/tutorial-proton-ide-plus-part-1>
- [42] J. D. Bronzino *The Biomedical Engineering handbook*. 2018. California. Disponible en: https://brainmaster.com/software/pubs/brain/The_Biomedical_Engineering_Handbook_.pdf

ANEXOS

Anexo A

LINEAS DE CÓDIGO DEL LOGARITMO DESARROLLADO PARA VISUALIZAR EL ESQUELETO VIRTUAL (POSE ESTIMATION) A PARTIR DE UNA IMAGEN O VIDEO

```
#=====
import argparse
import logging
import time
from pprint import pprint
import cv2
import numpy as np
import sys
from tf_pose.estimator import TfPoseEstimator
from tf_pose.networks import get_graph_path, model_wh
import math

logger = logging.getLogger('TfPoseEstimator-WebCam')
logger.setLevel(logging.DEBUG)
ch = logging.StreamHandler()
ch.setLevel(logging.DEBUG)
formatter = logging.Formatter('[%(asctime)s] [%(name)s] [%(levelname)s] %(message)s')
ch.setFormatter(formatter)
logger.addHandler(ch)

fps_time = 0
def find_point(pose, p):
    for point in pose:
        try:
            body_part = point.body_parts[p]
            return (int(body_part.x * width + 0.5), int(body_part.y * height + 0.5))
        except:
            return (0,0)
    return (0,0)
def euclidian( point1, point2):
    return math.sqrt((point1[0]-point2[0])**2 + (point1[1]-point2[1])**2 )
def angle_calc(p0, p1, p2 ):
    """
    p1 is center point from where we measured angle between p0 and
    """
    try:
```

```

    a = (p1[0]-p0[0])**2 + (p1[1]-p0[1])**2
    b = (p1[0]-p2[0])**2 + (p1[1]-p2[1])**2
    c = (p2[0]-p0[0])**2 + (p2[1]-p0[1])**2
    angle = math.acos( (a+b-c) / math.sqrt(4*a*b) ) * 180/math.pi
except:
    return 0
return int(angle)
def plank( a, b, c, d, e, f):
    #There are ranges of angle and distance to for plank.
    """
        a and b are angles of hands
        c and d are angle of legs
        e and f are distance between head to ankle because in plank distace will be maximum.
    """
    if (a in range(50,100) or b in range(50,100)) and (c in range(135,175) or d in
range(135,175)) and (e in range(50,250) or f in range(50,250)):
        return True
    return False
def mountain_pose( a, b, c, d, e):
    """
        a is distance between two wrists
        b and c are angle between neck,shoulder and wrist
        e and f are distance between head to ankle because in plank distace will be maximum.
    """
    if a in range(20,160) and b in range(60,140) and c in range(60,140) and d in
range(100,145) and e in range(100,145):
        return True
    return False
def draw_str(dst, xxx_todo_changeme, s, color, scale):

    (x, y) = xxx_todo_changeme
    if (color[0]+color[1]+color[2]==255*3):
        cv2.putText(dst, s, (x+1, y+1), cv2.FONT_HERSHEY_PLAIN, scale, (0, 0, 0),
thickness = 4, lineType=10)
    else:
        cv2.putText(dst, s, (x+1, y+1), cv2.FONT_HERSHEY_PLAIN, scale, color, thickness
= 4, lineType=10)
    #cv2.line
    cv2.putText(dst, s, (x, y), cv2.FONT_HERSHEY_PLAIN, scale, (255, 255, 255),
lineType=11)
if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='tf-pose-estimation realtime webcam')

```

```

parser.add_argument('--camera', type=str, default=1)
parser.add_argument('--resize', type=str, default='432x368',
                    help='if provided, resize images before they are processed. default=432x368,
Recommend : 432x368 or 656x368 or 1312x736 ')
parser.add_argument('--resize-out-ratio', type=float, default=4.0,
                    help='if provided, resize heatmaps before they are post-processed.
default=1.0')

```

```

parser.add_argument('--model', type=str, default='cmu', help='cmu / mobilenet_thin')
parser.add_argument('--show-process', type=bool, default=False,
                    help='for debug purpose, if enabled, speed for inference is dropped.')
args = parser.parse_args()

```

```

print("mode 0: Only Pose Estimation \nmode 1: People Counter \nmode 2: Fall Detection
\nmode 3: Yoga pose angle Corrector \nmode 4: Planking/Push up Detection \nmode 5:
Hourglass ratio")
mode = int(input("Enter a mode : "))

```

```

logger.debug('initialization %s : %s' % (args.model, get_graph_path(args.model)))
w, h = model_wh(args.resize)
if w > 0 and h > 0:
    e = TfPoseEstimator(get_graph_path(args.model), target_size=(w, h))
else:
    e = TfPoseEstimator(get_graph_path(args.model), target_size=(368, 368))
logger.debug('cam read+')
cam = cv2.VideoCapture(args.camera)
ret_val, image = cam.read()
logger.info('cam image=%dx%d' % (image.shape[1], image.shape[0]))
count = 0
i = 0
frm = 0
y1 = [0,0]
global height,width
orange_color = (0,140,255)
while True:
    ret_val, image = cam.read()
    i = 1
    humans = e.inference(image, resize_to_default=(w > 0 and h > 0),
upsample_size=args.resize_out_ratio)
    pose = humans
    image = TfPoseEstimator.draw_humans(image, humans, imgcopy=False)
    height,width = image.shape[0],image.shape[1]

```

```

if mode == 1:
    hu = len(humans)
    # print("Total no. of People : ", hu)
elif mode == 2:
    for human in humans:
        for i in range(len(humans)):
            try:
                a = human.body_parts[0] #Head point
                x = a.x*image.shape[1]
                y = a.y*image.shape[0]
                y1.append(y)
            except:
                pass
            if ((y - y1[-2]) > 30):
                print("fall detected.",i+1, count)#You can set count for get that your detection
is working
elif (mode == 3):
    if len(pose) > 0:
        # distance calculations
        head_hand_dst_l = int(euclidian(find_point(pose, 0), find_point(pose, 7)))
        head_hand_dst_r = int(euclidian(find_point(pose, 0), find_point(pose, 4)))
        m_pose = int( euclidian(find_point(pose, 7), find_point(pose, 4)))
        # angle calcucations
        angle1 = angle_calc(find_point(pose, 6), find_point(pose, 5), find_point(pose, 1))
        angle5 = angle_calc(find_point(pose,3), find_point(pose,2), find_point(pose,1))

        if (mode == 3) and mountain_pose(m_pose, angle1, angle5, head_hand_dst_r,
head_hand_dst_l):
            # draw_str(frame, (20, 220), " Mountain Pose", orange_color, 1.5)
            action = "Mountain Pose"
            is_yoga = True
            #if prev_action == 'Unknown' or prev_action == "Unknown_First":
            # yoga_duration = time.time()
            draw_str(image, (20, 50), action, orange_color, 2)
            logger.debug("*** Mountain Pose ***")
elif mode == 4:
    if len(pose) > 0:
        # distance calculations
        head_hand_dst_l = int(euclidian(find_point(pose, 0), find_point(pose, 7)))
        head_hand_dst_r = int(euclidian(find_point(pose, 0), find_point(pose, 4)))
        # angle calcucations
        angle2 = angle_calc(find_point(pose,7), find_point(pose,6), find_point(pose,5))

```

```

        angle4      =      angle_calc(find_point(pose,11),      find_point(pose,12),
find_point(pose,13))
        angle6 = angle_calc(find_point(pose,4), find_point(pose,3), find_point(pose,2))
        angle8 = angle_calc(find_point(pose,8), find_point(pose,9), find_point(pose,10))

        if (mode == 4) and plank(angle2, angle6, angle4, angle8,head_hand_dst_r,
head_hand_dst_l):
            action = "Plank"
            is_yoga = True
            #if prev_action == 'Unknown' or prev_action == "Unknown_First":
            #    yoga_duration = time.time()
            #logger.debug("*** Plank ***")
            draw_str(image, (20, 50), " Plank", orange_color, 2)
            logger.debug("*** Plank ***")
        elif mode == 5:
            shoulder_hand_dst_l = int( euclidian(find_point(pose, 5), find_point(pose, 7)))
            shoulder_hand_dst_r = int( euclidian(find_point(pose, 2), find_point(pose, 4)))
            body_part_dst_l      = int( euclidian(find_point(pose, 11), find_point(pose, 13)))
            body_part_dst_r      = int( euclidian(find_point(pose, 8), find_point(pose, 10)))
            body_ratio_l         = round(shoulder_hand_dst_l/body_part_dst_l,2)
            body_ratio_r         = shoulder_hand_dst_r/body_part_dst_r
            head_ankle           = int( euclidian(find_point(pose, 15), find_point(pose,
13)))
            hand_hand             = shoulder_hand_dst_l + shoulder_hand_dst_l + int(
euclidian(find_point(pose, 2), find_point(pose, 5)))

            total_ratio          = round(head_ankle/hand_hand,2)
            draw_str(image, (20, 80), "hand_to_leg_ratio = "+str(body_ratio_l), orange_color,
1.5)
            draw_str(image, (20, 50), "height_to_width_ratio = "+str(total_ratio), orange_color,
1.5)

        cv2.putText(image,
            "FPS: %f" % (1.0 / (time.time() - fps_time)),
            (10, 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
            (0, 255, 0), 2)
        #image = cv2.resize(image, (720,720))
        if(frm==0):
            out = cv2.VideoWriter('outpy.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 30,
(image.shape[1],image.shape[0]))
            print("Initializing")
            frm+=1

```

```

cv2.imshow('tf-pose-estimation result', image)
if i != 0:
    out.write(image)
    fps_time = time.time()
    if cv2.waitKey(1) == 27:
        break

cv2.destroyAllWindows()

```

ANEXO B PROGRAMACION DE PROTON BASIC PARA CONTROLAR PULSO

```

*****
'* Name   : UNTITLED.BAS *
'* Author : [select VIEW...EDITOR OPTIONS] *
'* Notice : Copyright (c) 2020 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved *
'* Date   : 03/12/2020 *
'* Version : 1.0 *
'* Notes  : *
'*       : *
*****
;-----
;**** Added by Fuse Configurator ****
; Use the Fuse Configurator plug-in to change these settings

Device = 12F683

Config FOSC_INTOSCIO, WDTE_OFF, PWRTE_ON, MCLRE_OFF, CP_OFF,
CPD_OFF, BOREN_OFF, IESO_ON, FCMEN_ON

;**** End of Fuse Configurator Settings ****
;-----
;Velocidad de reloj
Xtal 4
;Configuracion Principal
OSCCON = %01100111
All_Digital 1
Declare Serial_Data 8
;Variables
Dim cuenta As Word

```

;configuracion de puertos

TRISIO.0 = 1

TRISIO.1 = 0

TRISIO.5 = 0

Low GPIO.5

High GPIO.1

DelayMS 50

Low GPIO.1

DelayMS 100

High GPIO.1

DelayMS 50

Low GPIO.1

DelayMS 100

High GPIO.1

DelayMS 50

Low GPIO.1

Main:

cuenta = Counter GPIO.0, 10000

cuenta = cuenta

SerOut GPIO.5,84,[Dec cuenta,13]

If cuenta >= 60 Then

High GPIO.1

HPWM 1,125,1000

Else

Low GPIO.1

HPWM 1,0,0

EndIf

cuenta = 0

GoTo Main

ANEXO C

PROGRAMA EN PYTHON PARA LEER PUERTO SERIAL DE LOS DATOS DEL SENSOR DE PULSO CARDÍACO

```
import time
import serial
import numpy as np

N = 5

data = np.zeros((N, ), dtype=int )

data2=[1]

tx = serial.Serial('COM5', baudrate=9600, timeout=1.0)
with tx:
    ii = 0
    while ii < N:
        try:
            line = tx.readline()
            if not line:
                # HACK: Descartamos líneas vacías porque fromstring produce
                # resultados erróneos, ver
                # https://github.com/numpy/numpy/issues/1714
                continue
            data[ii] = np.fromstring(line.decode('ascii', errors='replace'), sep=' ')
            ii += 1
            data2[0]=np.mean(data)
        except KeyboardInterrupt:
            print("Exiting")
            break
    print(data2[0])
```

ANEXO D

PROGRAMA EN PYTHON PARA MOSTRAR GUI (VENTANA DE APLICACIÓN GRÁFICA)

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'Visor_valor.ui'
#
# Created by: PyQt5 UI code generator 5.12.3
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QLCDNumber

class Visor_valor(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("PULSOS POR MINUTO")
        MainWindow.resize(633, 425)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.lcdNumber = QtWidgets.QLCDNumber(self.centralwidget)
        self.lcdNumber.setGeometry(QtCore.QRect(120, 40, 320, 220))
        self.lcdNumber.setObjectName("lcdNumber")

        self.lcdNumber.setMode(QLCDNumber.Dec)
        self.lcdNumber.display(65)
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(320, 280, 150, 91))
        self.label.setObjectName("label")
        MainWindow.setCentralWidget(self.centralwidget)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
```

```

        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("Pulsos", "PULSOS POR MINUTO"))
        self.label.setText(_translate("MainWindow",
                                     "<html><head/><body><p
align=\"center\"><span          style=\"          font-size:26pt;          font-
weight:600;\">BPS</span></p></body></html>"))

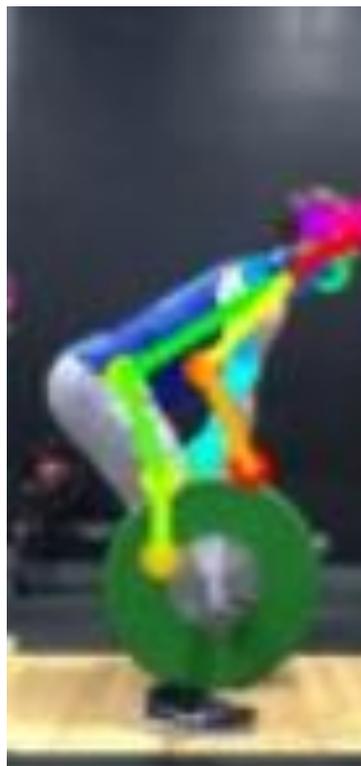
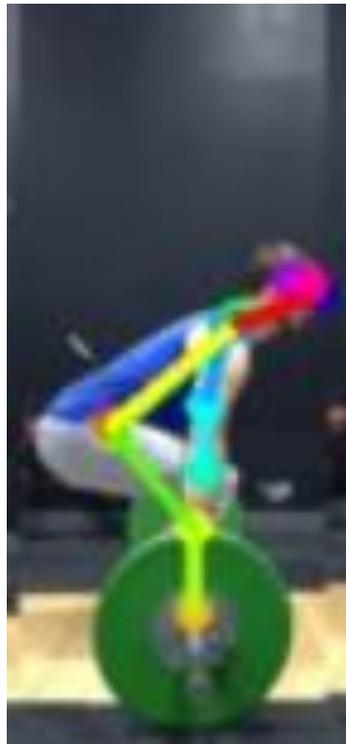
```

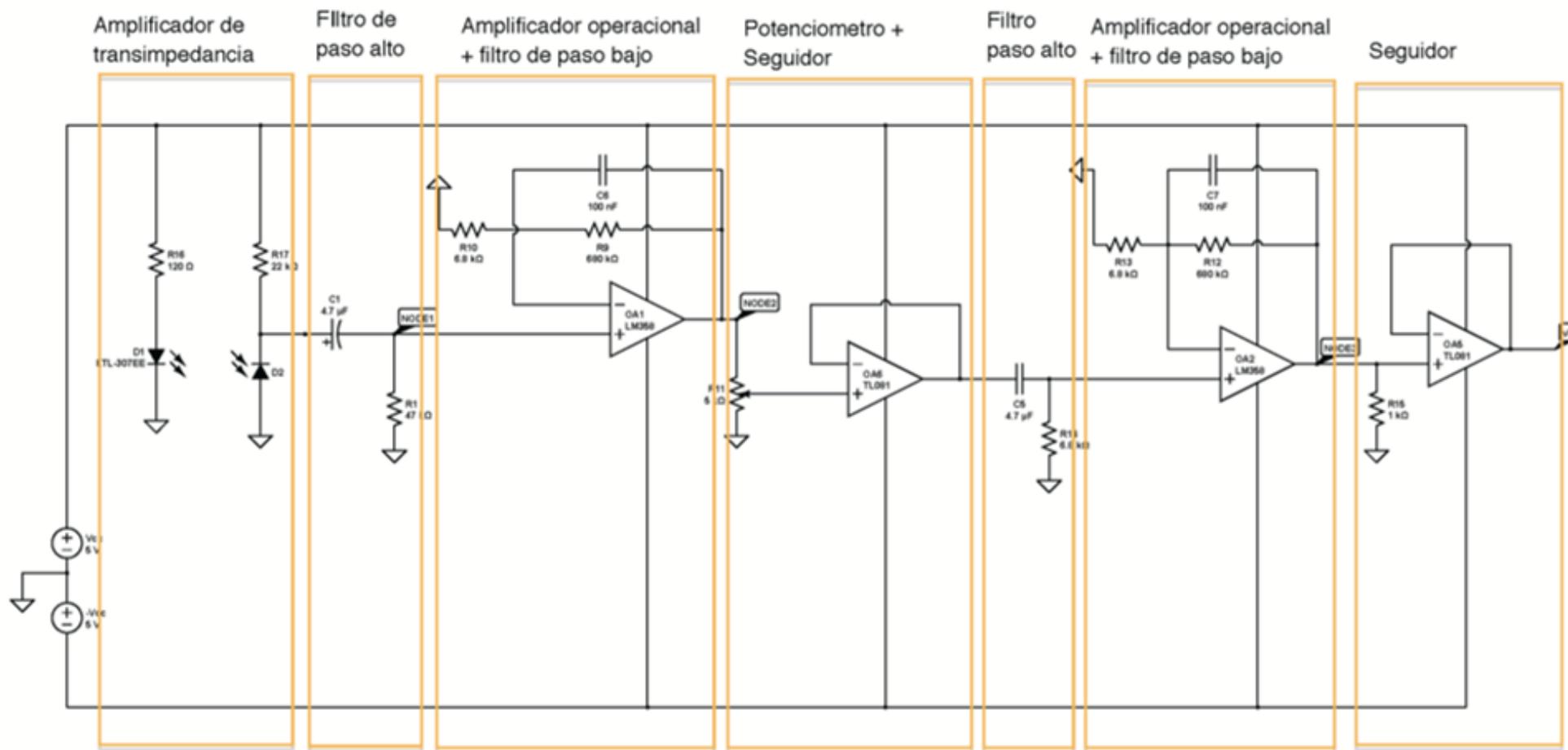
```

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Visor_valor()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

ANEXO E FISICOCULTURISTA PROBANDO EL SISTEMA





INSTITUCIÓN: UNIVERSIDAD TÉCNICA DE AMBATO
 FISEI
 INGENIERIA EN ELECTRÓNICA Y COMUNICACIONES

AUTOR: Byron Rodrigo Palacios Sánchez

TIPO: Trabajo Final de Grado en Ingeniería en
 Electrónica y Comunicaciones

TEMA: Diseño del Circuito Electrónico encargado de
 monitorear el pulso cardiaco

FECHA: febrero 2021

