



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO
ELECTRÓNICO DE ESTIMULACIÓN TEMPRANA MOTRIZ
GRUESA PARA NIÑOS DE 4 A 5 AÑOS EMPLEANDO TÉCNICAS
DE VISIÓN ARTIFICIAL”**

Trabajo de Integración Curricular

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES:

CRISTHIAN GEOVANNY FAJARDO ORTIZ

WILMER JAVIER YANCHAPANTA TAMAQUIZA

Riobamba – Ecuador

2022



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO
ELECTRÓNICO DE ESTIMULACIÓN TEMPRANA MOTRIZ
GRUESA PARA NIÑOS DE 4 A 5 AÑOS EMPLEANDO TÉCNICAS
DE VISIÓN ARTIFICIAL”**

Trabajo de Integración Curricular

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES:

CRISTHIAN GEOVANNY FAJARDO ORTIZ

WILMER JAVIER YANCHAPANTA TAMAQUIZA

DIRECTOR: ING. JOSÉ LUIS TINAJERO LEÓN

Riobamba – Ecuador

2022

©2022, Cristhian Geovanny Fajardo Ortiz y Wilmer Javier Yanchapanta Tamaquiza

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, Cristhian Geovanny Fajardo Ortiz y Wilmer Javier Yanchapanta Tamaquiza, declaramos que el presente Trabajo de Titulación es de nuestra autoría y que los resultados de este son auténticos y originales. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores, asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 16 de noviembre de 2022.



Cristhian Geovanny Fajardo Ortiz

085003382-0



Wilmer Javier Yanchapanta Tamaquiza

180487482-2

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular tipo: Dispositivo Tecnológico, **“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO ELECTRÓNICO DE ESTIMULACIÓN TEMPRANA MOTRIZ GRUESA PARA NIÑOS DE 4 A 5 AÑOS EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL”**, realizado por los señores: **CRISTHIAN GEOVANNY FAJARDO ORTIZ** y **WILMER JAVIER YANCHAPANTA TAMAQUIZA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

	FIRMA	FECHA
Ing. Nancy del Rocio Velasco Erazo PRESIDENTE DE TRIBUNAL		2022/11/16
Ing. José Luis Tinajero León DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR		2022/11/16
Ing. Pablo Eduardo Lozada Yáñez ASESOR DE TRABAJO DE INTEGRACIÓN CURRICULAR		2022/11/16

DEDICATORIA

A Dios quien ha sido mi guía, fortaleza y con su amor ha estado acompañándome e iluminándome durante todo el trayecto de mis años de estudio. A mis padres quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy una meta más, les dedico todo mi esfuerzo, en reconocimiento a todo el sacrificio puesto en mi para que yo pueda culminar mi carrera profesional. A mi hermana, por ser parte fundamental de mi vida y creer en mi durante todo el lapso de esta importante etapa. Y a todos los familiares y amigos que de una u otra forma contribuyeron para hacer este anhelo posible.

Cristhian

A Dios;

A mi tía, Luci;

A mis padres y hermanos;

A mi prima, Yessy;

A mi querida sobrina, Celeste Isabel;

A mis familiares y amigos; por estar siempre a mi lado cuando más los necesitaba, por ser los pilares a lo largo de mi carrera e impulsarme a continuar y no desmayar. Por ser y estar en el momento adecuado.

Wilmer

AGRADECIMIENTO

Quiero expresar mi gratitud a Dios porque sin él nada de esto hubiera sido posible. A mis padres, Malve y Geovanny, por depositar en mí su confianza, y darme su apoyo y motivación durante mi formación académica, sin ellos jamás hubiera podido conseguir lo que, hasta ahora, gracias por todo su amor. A mi hermana Cris, quien ha estado conmigo siempre que lo necesito, gracias por creer siempre en mí. Finalmente, a mis profesores, que marcaron con sus enseñanzas el futuro de todos nosotros, gracias por prepararnos no solo como los mejores profesionales sino también como mejores personas. A todos ustedes, ¡Gracias!

Cristhian

Agradezco a Dios por permitirme cumplir uno de mis sueños, por no abandonarme en los momentos más difíciles. A mi padre, por el apoyo incondicional y desinteresado que siempre me ha brindado. A la Escuela Superior Politécnica de Chimborazo por el valioso conocimiento brindado a lo largo de estos años. A mis profesores quienes me guiaron durante mi formación profesional. Al Ing. José Luis Tinajero, por haber dedicado su tiempo y conocimiento en la dirección de esta tesis. A mis compañeros y amigos, con quienes he compartido esta maravillosa etapa de mi vida.

Wilmer

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE ILUSTRACIONES.....	xi
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE ANEXOS	xiv
RESUMEN.....	xv
ABSTRACT	xvi
INTRODUCCIÓN	1

CAPÍTULO I

1.	DIAGNÓSTICO DEL PROBLEMA.....	2
1.1.	Antecedentes	2
1.2.	Formulación del problema	2
1.3.	Sistematización del problema.....	2
1.4.	Justificación	3
1.4.1.	<i>Justificación teórica</i>	3
1.4.2.	<i>Justificación aplicativa</i>	3
1.5.	Objetivos	4
1.5.1.	<i>Objetivo general</i>	4
1.5.2.	<i>Objetivos específicos</i>	4

CAPÍTULO II

2.	MARCO TEÓRICO	5
2.1.	Estimulación temprana.....	5
2.1.1.	<i>Importancia de la estimulación temprana</i>	5
2.1.2.	<i>Objetivos de la estimulación temprana</i>	7
2.1.3.	<i>Beneficios de la estimulación temprana</i>	8
2.1.4.	<i>Funciones de la estimulación temprana</i>	9
2.1.5.	<i>Estimulación temprana en niños de 4 a 5 años</i>	9
2.1.6.	<i>Áreas de desarrollo de estimulación temprana</i>	10
2.1.6.1.	Área de desarrollo cognitivo y de lenguaje	11
2.1.6.2.	Área de desarrollo espiritual y moral	12
2.1.6.3.	Área de desarrollo social y emocional	13
2.1.6.4.	Área de desarrollo físico y motor.....	14

2.1.7.	<i>Motricidad fina</i>	15
2.1.8.	<i>Motricidad gruesa</i>	16
2.2.	Herramientas de evaluación de habilidad motriz	17
2.3.	Prueba de desarrollo motriz grueso (TGMD-2)	18
2.4.	Visión artificial	19
2.4.1.	<i>Breve historia de la visión artificial</i>	19
2.4.2.	<i>Visión Artificial</i>	20
2.4.3.	<i>Importancia de la Visión artificial</i>	21
2.4.4.	<i>Etapas de la Visión Artificial</i>	22
2.4.4.1.	Adquisición de imagen.....	22
2.4.4.2.	Preprocesamiento	24
2.4.4.3.	Segmentación	25
2.4.4.4.	Extracción de características	26
2.4.4.5.	Reconocimiento	27
2.5.	Inteligencia Artificial	27
2.5.1.	<i>Aprendizaje automático (Machine learning)</i>	29
2.5.2.	<i>Técnicas computacionales</i>	29
2.5.2.1.	Hard Computing.....	29
2.5.2.2.	Soft Computing	29
2.5.2.3.	Lógica Difusa.....	30
2.5.2.4.	Redes Neuronales Artificiales.....	30
2.5.2.5.	Algoritmos genéticos	31
2.5.3.	<i>Redes Neuronales Artificiales Convolucionales</i>	32
2.6.	Segmentación, reconocimiento de posiciones y articulaciones del cuerpo humano	34
2.6.1.	<i>Modelo de Deep Learning para identificar postura del cuerpo humano</i>	35
2.7.	Modelos de Pose Estimation	36
2.7.1.	<i>PosNet Tensorflow</i>	37
2.7.2.	<i>Mediapipe</i>	38
2.7.2.1.	Mediapipe detección de objetos	39
2.7.2.2.	Detección puntos relevantes en rostros y segmentación	40
2.8.	Sistemas embebidos	42
2.9.	Componentes principales de un sistema embebido	43
2.9.1.	<i>Procesador</i>	43
2.9.2.	<i>Memoria</i>	44
2.9.3.	<i>Periféricos</i>	44
2.10.	Tarjetas controladoras para Visión Artificial	45

2.10.1.	<i>Raspberry pi</i>	46
---------	---------------------------	----

CAPÍTULO III

3.	MARCO METODOLÓGICO	47
3.1.	Análisis de requerimientos generales	47
3.1.1.	<i>Definición de variables</i>	48
3.2.	Selección del <i>Hardware</i>	49

CAPÍTULO IV

4.	PROPUESTA Y DISEÑO DE PROTOTIPO	52
4.1.	Preparación del Raspberry Pi	52
4.2.	Preparación entorno de programación	54
4.3.	Implementación programa configuración	55
4.3.1.	<i>Pruebas de la configuración inicial en adulto</i>	60
4.3.2.	<i>Medición de los ángulos entre articulaciones</i>	61

CAPÍTULO V

5.	MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	64
5.1.	Ejecución de ejercicios con el prototipo	64
5.1.1.	<i>Ejercicio 1</i>	64
5.1.2.	<i>Ejercicio 2</i>	66
5.1.3.	<i>Ejercicio 3</i>	67
5.1.4.	<i>Ejercicio 4</i>	69
5.1.5.	<i>Ejercicio 5</i>	71
5.1.6.	<i>Ejercicio 6</i>	73
5.2.	Resumen de la evaluación	74
5.3.	Ejecución de ejercicios sin el prototipo	76

CAPÍTULO VI

6.	VALIDACIÓN DEL PROTOTIPO	79
6.1.	Evaluación estadística	79

CAPÍTULO VII

7.	EVALUACIÓN ECONÓMICA	81
7.1.	Costos	81

7.1.1.	<i>Costos directos</i>	81
7.1.1.1.	Costos implementación del prototipo.....	81
7.1.1.2.	Costos implementación estructural	81
7.1.1.3.	Costos directos totales.....	81
7.1.2.	<i>Costos indirectos</i>	81
7.1.3.	<i>Costo total</i>	82
7.1.4.	<i>Comparativa costos</i>	82
CONCLUSIONES		84
RECOMENDACIONES		85
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-2:	Habilidades desarrolladas por los niños en la de 4 y 5 años.....	10
Tabla 2-2:	Principales diferencias entre IA Débil e IA Fuerte.....	28
Tabla 3-2:	Procesador vs Microcontrolador.....	43
Tabla 1-3:	Variables del sistema.....	49
Tabla 1-5:	Evaluación ejercicio 1 de los 15 niños	65
Tabla 2-5:	Evaluación de ejercicio 2 de los 15 niños.....	66
Tabla 3-5:	Evaluación ejercicio 3 de los 15 niños	68
Tabla 4-5:	Evaluación ejercicio 4 de los 15 niños	70
Tabla 5-5:	Evaluación ejercicio 5 de los 15 niños	72
Tabla 6-5:	Evaluación ejercicio 6 de los 15 niños	73
Tabla 7-5:	Tiempos de ejecución con el prototipo ejecutándose	75
Tabla 8-5:	Número de veces que se cumple o no cada ejercicio, con el prototipo en ejecución.....	76
Tabla 9-5:	Resumen de ejecución de ejercicios sin el prototipo	77
Tabla 10-5:	Número de veces que se cumple un ejercicio sin la utilización del prototipo	78
Tabla 1-6:	Promedios de los ejercicios con el prototipo	79
Tabla 2-6:	Promedios de los ejercicios sin el prototipo	79
Tabla 3-6:	Resumen de valores de la Prueba T de Student.....	80
Tabla 1-7:	Costos electrónicos.....	81
Tabla 2-7:	Costos estructurales	81
Tabla 3-7:	Costos directos totales	81
Tabla 4-7:	Costos indirectos	82
Tabla 5-7:	Costos totales.....	82
Tabla 6-7:	Costos de dispositivos relacionados inteligentes que utilizan visión artificial ...	82

ÍNDICE DE ILUSTRACIONES

Ilustración 1-2:	Desarrollo del cerebro humano.....	6
Ilustración 2-2:	Habilidades motores gruesas: (a) Correr propio sitio, (b) Rebote Pelota una mano, (c) Slide, (d) Kick, (e) Lanzamiento pelota una mano, (f) Salto largo	19
Ilustración 3-2:	Relación entre Visión Artificial e Inteligencia Artificial.....	21
Ilustración 4-2:	Etapas de la Visión Artificial.....	22
Ilustración 5-2:	Adquisición de imagen	23
Ilustración 6-2:	Procesamiento de imagen	24
Ilustración 7-2:	Segmentación de imagen	25
Ilustración 8-2:	Extracción de características de una imagen.....	26
Ilustración 9-2:	Reconocimiento de una imagen	27
Ilustración 10-2:	Fuzzy logic control temperatura	30
Ilustración 11-2:	Neurona biológica y neurona artificial	31
Ilustración 12-2:	Proceso CNN	33
Ilustración 13-2:	Neurona biológica y neurona artificial	33
Ilustración 14-2:	Ejemplo de la desfragmentación de una imagen RGB a matrices	34
Ilustración 15-2:	Modelos de segmentación cuerpo humano	35
Ilustración 16-2:	Segmentación cuerpo humano	35
Ilustración 17-2:	Proceso para la segmentación cuerpo humano	36
Ilustración 18-2:	Segmentación cuerpo humano	36
Ilustración 19-2:	Puntos clave generados por PoseNet	37
Ilustración 20-2:	Puntos clave PosNet, debilidad del modelo.....	38
Ilustración 21-2:	Detección de objetos utilizando MediaPipe.....	40
Ilustración 22-2:	Detección de puntos relevantes en rostro, diagrama de proceso.....	41
Ilustración 23-2:	Segmentación puntos del rostro utilizando MediaPipe.....	41
Ilustración 24-2:	Diagrama de proceso para la detección de puntos relevantes cuerpo humano	42
Ilustración 25-2:	Componentes de un sistema embebido.....	43
Ilustración 26-2:	Diagrama de un microprocesador	43
Ilustración 27-2:	Tipo de memorias en sistemas embebidos.....	44
Ilustración 28-2:	Diagrama general de periféricos del sistema embebido.....	45
Ilustración 29-2:	Raspberry pi.....	46
Ilustración 30-2:	Esquemático de Raspberry Pi	46
Ilustración 1-3:	Metodología para el desarrollo del proyecto.....	47
Ilustración 2-3:	Arquitectura general del prototipo	48

Ilustración 3-3:	Comparativa Raspberry pi varias versiones.....	50
Ilustración 4-3:	Raspberry pi 4 model b.....	50
Ilustración 5-3:	Cámara web COOAU1080 de 2Mpx.....	51
Ilustración 6-3:	Mouse y teclado USB marca UNNO.....	51
Ilustración 1-4:	Raspberry Pi Imager.....	52
Ilustración 2-4:	Configuración SSH y Wifi.....	52
Ilustración 3-4:	Raspberry sistema operativo.....	53
Ilustración 4-4:	VNC Viewer.....	53
Ilustración 5-4:	Advance IP Scanner.....	54
Ilustración 6-4:	PyCharm.....	54
Ilustración 7-4:	Instalación de paquetes.....	55
Ilustración 8-4:	Función findPose (en español encontrar pose).....	57
Ilustración 9-4:	Función findPosition que identifica puntos en articulaciones y ordena en una matriz.....	57
Ilustración 10-4:	Pose landmarks.....	58
Ilustración 11-4:	Resultado de la configuración inicial.....	58
Ilustración 12-4:	Impresión datos posición landmarks.....	59
Ilustración 13-4:	Resolución de la cámara web.....	59
Ilustración 14-4:	Tamaño por defecto de la ventana emitida por la cámara web.....	59
Ilustración 15-4:	Prueba 1 en un adulto.....	60
Ilustración 16-4:	Prueba 2 en un adulto.....	60
Ilustración 17-4:	Prueba 3 en un adulto.....	61
Ilustración 18-4:	Medida de los ángulos.....	61
Ilustración 19-4:	Resultado medición de los ángulos externos.....	62
Ilustración 1-5:	Ejercicio 1, correr en el mismo sitio, etapa evaluación.....	64
Ilustración 2-5:	Ejercicio 2, rebotar la pelota en el mismo sitio, etapa evaluación.....	66
Ilustración 3-5:	Ejercicio 3, Floss Dance o slide, etapa de evaluación.....	68
Ilustración 4-5:	Ejercicio 4, kick, etapa evaluación.....	70
Ilustración 5-5:	Ejercicio 5, Overhand Throw, etapa evaluación.....	71
Ilustración 6-5:	Ejercicio 6, Saltos largos horizontales, etapa evaluación.....	73

ÍNDICE DE GRÁFICOS

Gráfico 1-2:	Diagrama de flujo del algoritmo genético	32
Gráfico 2-2:	Diagrama de flujo detección de objetos utilizando MediaPipe.	39
Gráfico 1-4:	Diagrama de flujo del programa en su configuración inicial	56
Gráfico 2-4:	Diagrama de flujo medida de los ángulos	63
Gráfico 1-7:	Comparación prototipo vs diferentes dispositivos en el mercado	83

ÍNDICE DE ANEXOS

- ANEXO A:** DISPOSITIVO RASPBERRY CON LOS PERIFÉRICOS CONECTADOS Y EJECUTÁNDOSE
- ANEXO B:** PRUEBAS EJECUTADAS POR NIÑO DE 5 AÑOS
- ANEXO C:** PRUEBAS EJECUTADAS POR NIÑA DE 5 AÑOS
- ANEXO D:** PRUEBAS REALIZADAS A NIÑOS EN UNIDAD EDUCATIVA QUISAPINCHA
- ANEXO E:** PANTALLA INICIAL DE LA INTERFACE DEL PROGRAMA DE ESTIMULACIÓN MOTRIZ
- ANEXO F:** OFICIO UNIDAD EDUCATIVA QUISAPINCHA
- ANEXO G:** PROGRAMACIÓN EN PHYTON

RESUMEN

El presente trabajo de integración curricular tuvo como objetivo diseñar y construir un dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial, para lo cual mediante la consulta de fuentes bibliográficas y estudios que avalan lo favorable del desarrollo de habilidades motoras gruesas en etapas de desarrollo temprano, se identificaron los ejercicios que se tomaran en cuenta para la realización de las pruebas. Al establecer los requerimientos del dispositivo se seleccionó como controlador una Raspberry Pi modelo B y como entorno de programación Pycharm Community Edition. La programación se desarrolló en lenguaje Python, misma que utilizó segmentación y reconocimiento de posiciones y articulaciones a través de un modelo de estimación del cuerpo humano: PosNet basado en arquitectura de Tensorflow y Mediapipe para la detección del cuerpo humano. Se programaron las funciones encontrarpose, encontrarpuntos y findangle mismas que son la base de toda la estructura de programación. Se ejecutaron pruebas utilizando una cámara web conectada al dispositivo tanto en niños como en adultos para establecer la configuración inicial, se verificó que el sistema es capaz de identificar los puntos más relevantes de las articulaciones fuertes del cuerpo humano. En las pruebas realizadas con 15 infantes, entre 4 y 5 años de edad, el dispositivo fue capaz de reconocer las articulaciones principales del cuerpo a una distancia promedio de 241cm, entre los 6 ejercicios establecidos cada infante realizó satisfactoriamente cada actividad en un tiempo promedio de 3 minutos y 40 segundos. De los 5 intentos ejecutados para cada uno de los 6 ejercicios, los niños en promedio acertaron cuatro ejercicios poseyendo una efectividad del 89%. Para una mejor respuesta en cuanto a la escalabilidad del proyecto al agregar más ejercicios a la rutina, es necesario realizar un levantamiento sensorial en el plano espacial.

PALABRAS CLAVE: <INGENIERÍA Y TECNOLOGÍA ELECTRÓNICA>, <INTELIGENCIA ARTIFICIAL>, <VISIÓN ARTIFICIAL>, <DISPOSITIVO ELECTRÓNICO>, <ESTIMULACIÓN TEMPRANA MOTRIZ>, <PYTHON (SOFTWARE)>, <RASPBERRY (HARDWARE)>.



D.B.R.A.I.
Ing. Cristian Castillo



ABSTRACT

The objective of this curricular integration work was to design and build an electronic device for early gross motor stimulation for children from 4 to 5 years old using artificial vision techniques; by researching bibliographical sources and studies that support the favorable development of gross motor skills in early development stages, the exercises taken into account for the tests were identified. When establishing the requirements of the device, a Raspberry Pi model B was selected as the controller, and Pycharm Community Edition as the programming environment. The programming was developed in Python language, which used segmentation and recognition of positions and articulations through an estimation model of the human body; PosNet based on Tensorflow and Mediapipe architecture for the detection of the human body. The functions *encontrarpose*, *encontrarpuntos*, and *findangle*, which are the base of the entire programming structure, were programmed. Tests were run using a webcam Connected to the device in both children and adults to establish the initial setting, it was verified that the system is capable of identifying the most relevant points of the strong joints of the human body. In the tests carried out with 15 infants, aged between 4 and 5 years, the device was able to recognize the main joints of the body at an average distance of 241cm. Among the 6 established exercises, every kid satisfactorily performed each activity in an average time of 3 minutes and 40 seconds. Of the 5 attempts executed for each of the 6 exercises, the children on average succeeded in four exercises with an effectiveness of 89%. For a better answer regarding the scalability of the project when adding more exercises to the routine, it is necessary to carry out a sensory survey in the spatial plane.

Key words: <ENGINEERING AND ELECTRONIC TECHNOLOGY>, <ARTIFICIAL INTELIGENCE>, <ARTIFICIAL VISION>, <ELECTRONIC DEVICE>, <EARLY MOTOR STIMULATION>, <PYTHON (SOFTWARE)>, <RASPBERRY (HARDWARE)>.



Lic. Nelly Padilla P. Mgs
0603818717
DOCENTE FIE

INTRODUCCIÓN

En la actualidad vivimos en la era de la tecnología, inmersa en un mundo de equipos inteligentes que apuestan hacer la vida más placentera. Este gran avance de la humanidad promueve el adelanto e innovación. Introducir tecnología en niños a temprana edad es factible e importante para su formación. Es fundamental el desarrollo de dispositivos o aplicaciones que ayuden a la enseñanza de los infantes tanto en hogares como en centros educativos (SÁNCHEZ & GUZMÁN, 2012).

Una de las primeras tecnologías aplicadas al proceso de enseñanza fue la televisión, mediante programas educativos se impartía conocimiento de forma audiovisual diferente a la tradicional; luego aparecieron juguetes didácticos para aprender colores, figuras, sonidos, etc. (GÓMEZ, 2012, p. 3).

Existen diversos sistemas los cuales combinan software y hardware para la estimulación infantil, uno de estos es ENGKEY el cual enseña inglés a niños, diseñado por la empresa ABC. Robi es un juguete Fisher Price (2017) que reproduce más de 40 canciones, aprendizajes y bailes, activándole desde su barriga o pie.

Teebot (EL UNIVERSO, 2015) es un robot educativo que brinda la oportunidad a niños entre 4 y 12 años a aprender las bases de programación y electrónica. Pingüino robot asistencia (CAYAMBE YAMBAY & CASTAÑEDA CHATE, 2016) el cual interactúa con el niño en el típico juego de buscar y patear la pelota controlado por un sensor Kinect, para el seguimiento en tres dimensiones. Pushaq el juguete inteligente (SALINAS SALINAS & CHAMORRO CARRERA, 2017), tiene la capacidad de interactuar lúdica y cognitivamente con un niño a la vez en etapa pre-operacional, cuenta con interfaz gráfica y varias opciones como: cuentos, juegos, galería de imágenes y reconociendo facial. El reconocimiento facial se basa en clasificadores cascada y algoritmo AdamBoost para entrenar una red neuronal, utiliza visión artificial para el reconociendo de contorno y color de objetos 2D.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Antecedentes

En el mercado se encuentran varios robots educativos comerciales como: Dash&Dot, Lego Mindstroms, Zowi, Bee-Bot, Albert, Moway Scratch, etc., pero no son exclusivos para realizar una correcta estimulación en niños a temprana edad. El principal problema de todos estos sistemas de estimulación temprana es no verificar que el niño realice todas las actividades impuestas por el mismo ya que el niño se distrae con facilidad. Es por esto se plantea diseñar y construir un sistema de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial. Este sistema permitirá estimular tempranamente a un niño haciendo uso de una pantalla en la cual se visualizará a un niño realizando una actividad específica la cual deberá ser realizada por el niño en estudio una vez realizada la actividad se mostrará otra actividad así sucesivamente. Si el niño no realiza satisfactoriamente la actividad se hace uso del módulo de actuación para llamaran la atención y cumpla con la actividad planteada. El uso de una cámara y técnicas de visión artificial permitirá verificar si el niño cumple con las actividades planteadas por el sistema y al sistema activar los actuadores de ser necesario.

1.2. Formulación del problema

¿Cómo diseñar y construir un dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial?

1.3. Sistematización del problema

- ¿Qué técnicas de estimulación temprana motriz gruesa existen?
- ¿Qué tipos de dispositivos electrónicos de estimulación temprana motriz gruesa existen?
- ¿Cuáles son los requerimientos que debe satisfacer el dispositivo electrónico de estimulación temprana motriz gruesa?
- ¿Qué *hardware* y *software* permiten diseñar el dispositivo electrónico de estimulación temprana motriz gruesa?
- ¿Qué diseño cumple con los requerimientos propuestos para la construcción del dispositivo electrónico de estimulación temprana motriz gruesa?
- ¿Cumple el dispositivo electrónico implementado con todos los requerimientos planteados en esta investigación?

1.4. Justificación

1.4.1. Justificación teórica

La estimulación temprana es un método de estimulación oportuna de las capacidades de un niño, es decir, un aprendizaje temprano, pero no precoz. Según varios estudios científicos, los primeros años de vida son el mejor momento para que las neuronas sean activadas o estimuladas (REGIDOR, 2003, pp. 16-17).

La estimulación de la motricidad gruesa que posee cada ser humano para mover los músculos del cuerpo de forma coordinada y en equilibrio permitirá a los niños desarrollar mayor agilidad, fuerza y velocidad en el movimiento de sus articulaciones. (NAVARRO, 2009). A partir de los 4 años los niños cuentan con un equilibrio de movimientos, elimina los movimientos involuntarios y va marcando progresivamente la independencia segmentaria. Esto conlleva a una mayor precisión de dinamismo manual, a la aparición de gestos propios y el perfeccionamiento de la coordinación óculo-manual (SÁEZ & MATEO, 2004).

La guía para familias de niños y niñas entre cero y cinco años realizada por Unicef indica que a partir del 4to año comienza a tener iniciativas propias, se interesa mucho por todo lo que lo rodea y disfruta aprendiendo sobre distintos temas, en esta etapa el infante permanece más tiempo realizando alguna actividad, puede realizarla por etapas y se emociona cuando las finaliza. A los 5 años el infante comienza a sentir curiosidad e interesarse por todo lo que le rodea, les atrae conocer la utilidad y el origen de las cosas, comienza a aprender y aceptar las reglas de los juegos, maneja cada vez mejor los movimientos de las manos, lo que le permite realizar actividades con ellas (UNICEF, 2015).

Basados en el Plan Nacional de Desarrollo 2017 – 2021- Toda una Vida el cual plantea en su Eje No. 1: “*la primera infancia- es una etapa de especial relevancia: durante este periodo se sienta las bases para el futuro desarrollo cognitivo, afectivo y social de las personas*” (SECRETARÍA NACIONAL DE PLANIFICACIÓN Y DESARROLLO, 2017, P. 49), y en la inexistencia de prototipos que realicen un control en las actividades de estimulación temprana realizadas por un infante. Se plantea diseñar y construir un dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial.

1.4.2. Justificación aplicativa

El sistema que se pretende diseñar busca la estimulación temprana motriz gruesa de niños de 4 a 5 años, con la ayuda de una pantalla en la cual los niños verán los retos o actividades que tendrán que realizar. Cuando el niño logre realizar la actividad el sistema automáticamente

planteará otro reto que el niño deberá efectuar, hasta que termine una serie de actividades establecidas por el sistema. En el caso de que el niño no pueda resolver una actividad, el sistema buscará estimular al niño para que realice la acción, para esto se usara el módulo de actuación como actuadores lumínicos, sonoros, entre otros que ayuden al niño a terminar con el reto o actividad.

El sistema consta de una pantalla en la cual el niño podrá visualizar las actividades que tendrá que efectuar. Una cámara que captará los movimientos del niño y enviará las imágenes a un módulo de control. En el módulo de control guardarán las imágenes obtenidas por la cámara en una base de datos para su posterior procesamiento. El sistema comparará las imágenes obtenidas con las imágenes guardadas en la base de datos y determinará si el infante logró realizar la actividad. En el caso de que no se realicen los ejercicios el módulo de control enviará señales al módulo de actuación para llamar la atención del niño y realice la actividad planteada por el sistema.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar y construir un dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial.

1.5.2. Objetivos específicos

- Investigar las técnicas de estimulación temprana motriz gruesa y dispositivos electrónicos existentes.
- Establecer los requerimientos que debe satisfacer el dispositivo electrónico de estimulación temprana motriz gruesa.
- Seleccionar el hardware y software que permita diseñar el dispositivo electrónico de estimulación temprana motriz gruesa.
- Diseñar y construir el dispositivo electrónico que cumpla con los requerimientos establecidos.
- Evaluar si el dispositivo electrónico implementado cumple con todos los requerimientos planteados en esta investigación.

CAPÍTULO II

2. MARCO TEÓRICO

En esta sección se presenta los contenidos esenciales a comprender para la elaboración del trabajo de integración curricular.

2.1. Estimulación temprana

Es primordial en los primeros años de vida, porque en esta etapa se desarrollan y moldean capacidades esenciales como: el área cognitiva, social, física, emocional, espiritual, entre otras, este proceso puede iniciarse desde los 45 días después del nacimiento con un programa adecuado de estimulación. Las actividades que se desarrollan con los pequeños inician con una serie de juegos y ejercicios que se desarrollaran en un ambiente adecuado y preciso para cada edad (GÓMEZ, 2014),

Estimular significa "despertar a la acción, estimular, animar, aumentar la energía o la actividad de". La estimulación en la etapa de la primera infancia, por lo tanto, significa esfuerzos para activar el desarrollo temprano del niño (MURALIDHARAN, 1992).

La estimulación temprana intenta proporcionar experiencias de aprendizaje al niño para mejorar su desarrollo. Sin embargo, las experiencias de aprendizaje están tan planificadas que están de acuerdo con el nivel de maduración del niño. Estas experiencias deben darse desde el nacimiento e involucran actividades que exigen una interacción cercana entre el niño y el cuidador (MURALIDHARAN, 1992).

De acuerdo con (BARRENO Y MACIAS, 2015), los niños son considerados aprendices activos desde que nacen, por esto es de gran importancia que se encuentren en un ambiente adecuado y que la persona encargada esté debidamente preparada, dado que su papel es primordial al momento de interactuar con el infante. Durante los primeros años de vida es el mejor momento para estimular las neuronas, ya que estas se encuentran más receptivas a nuevas experiencias.

2.1.1. Importancia de la estimulación temprana

El desarrollo cerebral comienza desde antes del nacimiento hasta la edad adulta. Una arquitectura sólida del cerebro debe ser fomentada por la estimulación temprana, lo que mejora el aprendizaje, salud y conducta en el crecimiento del niño. De acuerdo con (CENTER ON THE

DEVELOPING CHILD, 2007) “En los primeros años de vida, más de un millón de conexiones neuronales se forman cada segundo. Luego de este periodo de rápida proliferación, las conexiones se reducen mediante un proceso llamado “poda”, que permite que los circuitos cerebrales se vuelvan más eficientes. Las vías sensoriales como las de la visión y audición básicas son las primeras en desarrollarse, seguidas por habilidades tempranas de lenguaje y funciones cognitivas superiores. Las conexiones proliferan y se podan en un orden establecido: los circuitos cerebrales más complejos van construyéndose sobre los circuitos anteriores más simples.”

En la ilustración 1-2, se muestra el desarrollo de las conexiones neuronales para las diferentes funciones que se desarrollan secuencialmente en el cerebro humano.

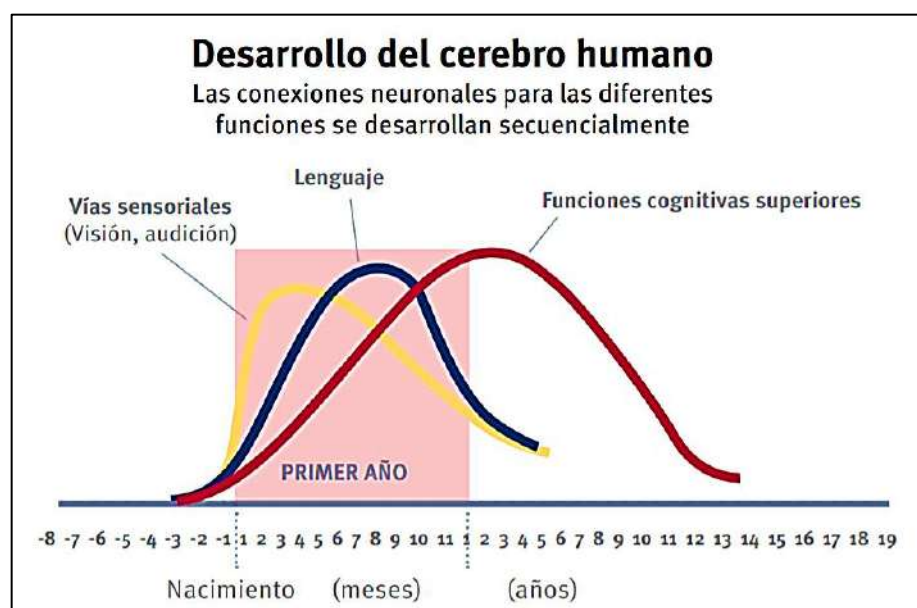


Ilustración 1-2: Desarrollo del cerebro humano

Fuente: (NELSON Y BOSQUET, 2000)

Los avances recientes en investigaciones del cerebro han demostrado que el entorno de un bebé tiene un efecto dramático en la construcción del cerebro y el desarrollo saludable. Es esta etapa temprana del desarrollo del cerebro la que da como resultado cómo y qué tan bien se piensa y se aprende, tanto de niños como de adultos. En los primeros años de vida de un bebé, el cerebro está ocupado construyendo su sistema de cableado. La actividad en el cerebro crea pequeñas conexiones eléctricas llamadas sinapsis. La cantidad de estimulación que recibe un bebé afecta directamente la cantidad de sinapsis que se forman. La estimulación repetitiva y constante fortalece estas conexiones y las hace permanentes. Es posible que las conexiones que no se utilicen se eliminen. Los primeros años son el "mejor momento" para un cerebro joven en desarrollo. Este intenso período de crecimiento cerebral y capacidad de construcción de redes ocurre solo una vez en la vida. Como cuidadores y padres, tenemos esta breve pero única

oportunidad de ayudar a fomentar la formación de circuitos cerebrales en nuestros bebés (LUCICH, 2011). Aquí hay algunos datos fascinantes que los investigadores han descubierto:

- Los bebés tienen una necesidad biológica y un deseo de aprender.
- La red fundamental de las sinapsis del cerebro está casi completa después del rápido desarrollo cerebral de los primeros tres años.
- Mientras más experiencias interesantes y apropiadas para su edad, tanto físicas como socioemocionales, en las que participe un bebé, más circuitos se construyen para mejorar el aprendizaje en el futuro.
- Los bebés tienen una preferencia definida por el rostro, la voz, el tacto y el olfato humanos sobre todo lo demás. Por lo tanto, el mejor juguete del bebé es usted, mientras habla, mueve, toca y habla con él.
- La estimulación interesante puede mejorar la curiosidad, la atención, la concentración y el amor por el aprendizaje en los bebés y niños pequeños en crecimiento.
- La estimulación del lenguaje es fundamental para todas las áreas del desarrollo cognitivo. Los bebés y los niños con quienes se conversa, se les lee y se involucran en muchas interacciones verbales muestran habilidades lingüísticas más avanzadas que los niños que no están tan involucrados verbalmente con sus cuidadores.

2.1.2. Objetivos de la estimulación temprana

Según (BARRENO Y MACIAS, 2015), uno de los objetivos principales de la estimulación temprana hacer que esta sea una práctica agradable, que vaya estableciendo lazos y fortaleciendo la relación madre-hijo o padre-hijo, para luego potenciar el desarrollo de las habilidades con calidad y calidez, por medio de las experiencias obtenidas de la relación con sus padres basadas en ejercicios placenteros sustentados en el progreso integral.

Continuando con (BARRENO Y MACIAS, 2015) otro de los objetivos primordiales es usar el intenso periodo de crecimiento cerebral y la capacidad de construcción de redes neuronales que suceden en el niño a temprana edad, para que estas potencien el desarrollo de las capacidades cognitivas, físicas, emocionales y sociales, a través de métodos lúdicos fundamentados en programas de estimulación temprana. No obstante, esto no significa que el desarrollo del niño se va a acelerar o a forzar, por el contrario, es potenciar su desarrollo y mejorar su autoestima.

Paralelamente (AYALA, 2018), sustenta que entre objetivos más importantes de la estimulación temprana en niños constan los siguientes:

- Permitir el máximo desarrollo del niño a escala general o en áreas específicas tales como la intelectual, social, del lenguaje, etc.

- Adaptar las actividades a la etapa de desarrollo por la cual está pasando el niño, a fin de que la viva plenamente y la supere.
- Evitar forzar al niño a realizar una actividad para la cual no está preparado, y así avanzar en su desarrollo.
- Servir de estrategia para evitar y atenuar riesgos a que están expuestos los niños y que puedan alterar su evolución normal.
- Favorecer un cambio de actitud de los padres y miembros de la comunidad en cuanto al manejo del ambiente, para que conviertan éste en un lugar sano, alegre y adecuado para el óptimo desarrollo del niño.
- Canalizar el deseo innato de aprender de cada niño para desarrollar su potencial creativo.
- Despertar la sensibilidad artística desde temprana edad a través de experiencias sensoriales enriquecedoras.
- Darle la oportunidad al niño de manipular diferentes materiales para favorecer el desarrollo satisfactorio de las destrezas que posee el bebé, aumentando su seguridad y confianza.
- Aprovechar el tiempo padre-hijo, enriqueciendo la relación interpersonal y el goce de estar juntos. A la vez, mientras se relacionan padres y niños, se obtiene una meta educativa a través de las actividades.

2.1.3. Beneficios de la estimulación temprana

Las investigaciones muestran que la educación de la primera infancia es una de las mejores formas de ayudar a su hijo a desarrollar las habilidades sociales, emocionales, cognitivas, físicas y espirituales que necesita para prepararse para la escuela primaria y más allá. Existen múltiples beneficios de una estimulación temprana en niños, entre los más importantes se tiene:

- Potencia el desarrollo de la capacidad de creación de nuevas neuronas del niño, beneficiando las diferentes áreas mencionadas anteriormente (RECALDE, 2018).
- Ayuda a mejorar la interpretación del ambiente que los rodea por medio de las experiencias que la estimulación temprana les brinda (RECALDE, 2018).
- Desarrollarán buenos hábitos, el cuidado infantil es ideal para que los niños adopten una buena rutina. Esto no solo los prepara para la escuela, también facilita la vida familiar (RECALDE, 2018).
- Se desarrollan habilidades de alfabetización y aritmética, las cuales tienen un gran impacto en el éxito académico del niño. Algunas investigaciones muestran que los niños estimulados a temprana edad se desempeñan mejor en las pruebas impuestas en la escuela y a lo largo de su vida (RECALDE, 2018).
- Fomenta la resiliencia emocional, esto ayuda a formar relaciones saludables con otras personas. A medida que crece el niño, este usará sus talentos para desarrollar amistades que

influirán en su sentido de identidad y futuro. Las investigaciones sugieren que los niños estimulados a edad temprana son más atentos y afrontan mejor sus emociones (GIALAMAS ET AL. 2014).

- Disfrutarán de un futuro exitoso, Una encuesta de 30 años reveló que los adultos que tenían una educación infantil de alta calidad de 0 a 5 años tenían más probabilidades de tener un empleo continuo y de graduarse de la universidad. Las habilidades que los niños aprenden en los primeros años de su vida son cruciales para su desempeño social, emocional y académico. De hecho, las investigaciones revelan que los niños que no asisten a la guardería tienen un 50% más de probabilidades de comenzar la escuela con vulnerabilidad del desarrollo (CAMPBELL et al. 2012).
- Desarrollarán un amor por el aprendizaje de por vida. El acceso a la estimulación en la primera infancia puede proporcionar a los niños experiencias sociales y cognitivas que les ayuden a ser independientes y desarrollar una actitud positiva hacia el aprendizaje (ELLIOTT, 2006).

2.1.4. Funciones de la estimulación temprana

(ANTOLÍN et al. 2005), sugiere la existencia de importantes funciones de la estimulación temprana para el desarrollo infantil, dentro de las cuales se tiene:

- **Hominización:** Según la antropología, la hominización, integra el principio que determina que no es suficiente nacer de un ser humano, por esa razón es primordial que se estimulen las áreas biológicas y psicológicas que cada infante tiene al nacer, para en el futuro ser personas capaces y competentes.
- **Socialización:** El proceso de la socialización se desarrolla con la interacción del infante con los demás niños, a través de normas, códigos de lenguaje, formas de vinculación, entre otras. Esto permite absorber conocimiento de las experiencias vividas del infante, lo que permitirá que este se desenvuelva armónicamente con los demás.
- **Culturización:** El proceso de culturizar se logra cuando el infante atiende, comprende y hace suya la cultura del grupo al que pertenece. Esto también permite al niño desenvolverse mejor en la sociedad.

2.1.5. Estimulación temprana en niños de 4 a 5 años

Los 5 primeros años de vida son un periodo dorado para el crecimiento del infante, se produce un desarrollo cerebral increíble, y la amplitud de las primeras experiencias de un niño influye en su capacidad de aprendizaje, así como el lugar donde vive. Es un momento crítico cuando sus cerebros en rápido desarrollo están listos para aprender, y una investigación reciente sugiere que hasta 90% del desarrollo cerebral ya ha tenido lugar antes de que un niño comience la escuela

primaria. La calidad de la estimulación y la crianza que reciben los niños en estos años marca la diferencia, y es por eso que la estimulación temprana de calidad tan importante.

En la de 4 a 5 años, la estimulación debe enfocarse en desarrollar las habilidades psicomotrices, fortalecer la actividad infantil, potenciar las actividades cognitivas como la atención, percepción e inteligencia, lo que conlleva a un proceso de socialización mejorado (LUDEÑA Y LISSETH, 2013).

(BURNHAM et al. 2018), argumenta que entre los 4 y 5 años los niños obtienen habilidades en las diferentes áreas de desarrollo, que se muestran en la tabla 1-1.

Tabla 1-2: Habilidades desarrolladas por los niños en la de 4 y 5 años

Área	Habilidades
Psicomotriz gruesa	<p>Puede correr y evitar obstáculos. Puede saltar con una cuerda Puede lanzar una pelota grande a su compañero y atraparla Tenga buen equilibrio y coordinación Puede subir y bajar escaleras un pie a la vez Empiece a andar en bicicleta</p>
Psicomotriz fina	<p>Puede formar letras Puede escribir su propio nombre Puede colorear en imágenes Puede enhebrar cuentas pequeñas</p>
Cognitivo	<p>Puede contar con precisión hasta 10 Puedo sumar dos conjuntos de objetos Pueden coincidir con conjuntos iguales Puede comprender la necesidad de reglas Puede nombrar la hora del día asociada con las actividades. Puede dar razones para resolver problemas.</p>
Lenguaje	<p>Puede contar en voz alta con precisión hasta 10 Utiliza oraciones complejas con palabras como "porque" Hablar sobre lo que sucedió y lo que podría suceder. Usar el lenguaje para discutir y responder Capaz de contar historias Disfruta de las bromas</p>
Emocional y social	<p>Puede averiguar lo que otras personas pueden estar pensando Capaz de comprender la necesidad de reglas Puede desarrollar amistades cercanas El comportamiento es principalmente cooperativo Capaz de separarse de los padres más fácilmente</p>

Fuente: (BURNHAM et al. 2018)

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.1.6. Áreas de desarrollo de estimulación temprana

Para favorecer al óptimo desarrollo de la primera infancia, la estimulación temprana se enfoca en cuatro áreas fundamentales: área de desarrollo cognitivo y de lenguaje, espiritual y moral, social y emocional, físico y motor (SNYDER, 2016).

Las áreas de desarrollo mencionadas están interrelacionadas y se afectan entre sí; por lo tanto, abordar al niño en su totalidad requiere la consideración de cada área de desarrollo.

Las habilidades que adquiere un niño están influenciadas por sus diferentes habilidades de desarrollo. Esto significa que la educación social influye en las relaciones entre el cuidador y el niño; los comportamientos cognitivos promueven o limitan las habilidades sociales y las actividades físicas impactan el lenguaje y los procesos de pensamiento (SNYDER, 2016).

Las capacidades cognitivas, emocionales y sociales están inextricablemente entrelazadas a lo largo del curso de la vida. El cerebro es un órgano altamente integrado y sus múltiples funciones operan de manera ricamente coordinada. El bienestar emocional y la competencia social proporcionan una base sólida para las habilidades cognitivas emergentes, y juntas son los ladrillos y el cemento que conforman la base del desarrollo humano. La salud emocional y física, las habilidades sociales y las capacidades cognitivo-lingüísticas que surgen en los primeros años son requisitos previos importantes para el éxito en la escuela y, más tarde, en el lugar de trabajo y la comunidad (SHONKOFF, 2009).

2.1.6.1. Área de desarrollo cognitivo y de lenguaje

Las teorías del desarrollo cognitivo buscan explicar los procesos dinámicos a través de los cuales la mente humana crece y cambia desde la infancia a lo largo de la vida. La cognición se refiere a capacidades que incluyen la memoria, el pensamiento y el razonamiento, el procesamiento espacial, la resolución de problemas, el lenguaje y la percepción. Es importante destacar que las teorías del desarrollo cognitivo tienen como objetivo explicar los mecanismos de cambio, y por lo tanto el desarrollo, en lugar de simplemente describir las capacidades de los niños de distintas edades o entre niños, adultos y poblaciones que envejecen. Esta entrada revisa el pensamiento histórico sobre si el desarrollo cognitivo es el resultado de características individuales o del entorno y analiza los modelos actuales de desarrollo cognitivo (RICHLAND, FRAUSEL Y BEGOLLI 2016).

La teoría de Piaget se centra en las teorías del desarrollo y el aprendizaje. El desarrollo se enfoca en las capacidades del más magro y el aprendizaje se enfoca en la realización de tales capacidades y la educación dentro de la teoría es extrínseca. La teoría cognitiva, el comportamiento refleja el surgimiento de diversas estructuras psicológicas, unidades organizadas o patrones de pensamiento que influyen en cómo los niños interpretan la información. Las teorías del desarrollo cognitivo explican el cambio en el nivel de razonamiento de un niño que adquiere nuevas formas de entender su mundo. La teoría de la implicación de Piaget supone que todos los niños pasan por la misma secuencia de desarrollo, pero lo hacen a ritmos diferentes. Los maestros deben hacer un esfuerzo especial para proporcionar actividades en el aula para individuos y grupos pequeños, en lugar de para el grupo total de la clase. La evaluación debe basarse en el progreso individual, más que en los estándares normales de

compañeros de la misma edad. Los individuos construyen su propio conocimiento durante el curso de la interacción con el entorno (LEFA, 2014).

El desarrollo del lenguaje en la primera infancia se refiere a la adquisición de habilidades que le permiten al niño relacionarse con el entorno que los rodea. Según Jean Piaget, el pensamiento es un requisito previo para hablar, este sigue creciendo progresivamente y ocurre en cada etapa del desarrollo como resultado de la experiencia y el razonamiento. El desarrollo de los niños en el lenguaje temprano está estrechamente relacionado con una variedad de actividades, atracciones y eventos que experimentan los niños; tocar, oír, ver, sentir y oler (INDRAYANI, 2016). Vygotsky argumentó en (VYGOTSKY 1980) que el desarrollo cognitivo y el lenguaje de los niños están estrechamente relacionados con la cultura y las comunidades en las que viven los niños.

2.1.6.2. Área de desarrollo espiritual y moral

La espiritualidad es una dimensión importante del desarrollo integral de los niños pequeños, al igual que la autonomía, la resiliencia y la responsabilidad. Desafortunadamente, a menudo permanece como un área olvidada en la educación de la primera infancia en muchas culturas (ZHANG, 2012).

El desarrollo espiritual es un elemento importante de la educación de un niño y fundamental para otras áreas de aprendizaje. Sin curiosidad, sin la inclinación a cuestionar y sin el ejercicio de la imaginación, el conocimiento y la intuición, los jóvenes carecerían de motivación para aprender y su desarrollo intelectual se vería afectado. Privados de autocomprensión y, potencialmente, de la capacidad de comprender a los demás, pueden experimentar dificultades para convivir con vecinos y colegas en detrimento de su desarrollo social. Si no pudieran ser conmovidos por sentimientos de asombro y admiración por la belleza del mundo en que vivimos, o por el poder de los artistas, músicos y escritores para manipular el espacio, el sonido y el lenguaje, vivirían en un desierto espiritual interior (GILLARD, 2011).

El desarrollo moral, como el desarrollo espiritual, no se puede definir con una simple declaración. Implica varios elementos:

- La voluntad de comportarse moralmente como un principio, esta actitud es fundamental para el desarrollo moral.
- Conocimiento de los códigos y convenciones de conducta acordados por la sociedad, tanto los no estatutarios como los prescritos por la ley.
- Conocimiento y comprensión de los criterios propuestos como base para emitir juicios responsables sobre cuestiones morales.
- La capacidad de emitir juicios sobre cuestiones morales, a medida que surgen mediante la aplicación de principios, conocimientos y razonamientos morales.

Los niños pequeños rara vez tienen la capacidad o la experiencia para tomar sus propias decisiones sobre lo que está bien y lo que está mal. Una cuestión moral es aquella que involucra a las personas a tomar una decisión sobre la base de lo que está bien y lo que está mal. La decisión a menudo requerirá acciones que, se espera, promoverán la bondad y minimizarán el mal. Los niños necesitan saber la diferencia entre el bien y el mal, aunque los niños muy pequeños a menudo no distinguen entre los contextos en los que se utilizan palabras como "correcto" e "incorrecto". A veces, por ejemplo, la palabra 'incorrecto' se referirá solo a un comportamiento socialmente inaceptable (está mal sacar la lengua), mientras que en otras ocasiones se trata de un absoluto moral. Sin embargo, los niños necesitan ser introducidos desde una edad temprana a los conceptos de lo correcto y lo incorrecto para que el comportamiento moral se convierta en un hábito instintivo (GILLARD, 2011).

La moral personal combina las creencias y valores de los individuos, los de los grupos sociales, culturales y religiosos a los que pertenecen, y las leyes y costumbres de la sociedad en general. Se debe esperar que las escuelas mantengan los valores que contienen absolutos morales (GILLARD 2011).

2.1.6.3. Área de desarrollo social y emocional

Durante sus primeros años de vida, el cerebro de los niños se desarrolla rápidamente, al igual que su capacidad para aprender habilidades sociales y emocionales esenciales (PALMER, 2019). El desarrollo social y emocional en los primeros años, también conocido como salud mental en la primera infancia, se refiere a la capacidad emergente de los niños para:

- Experimente, regule y exprese una variedad de emociones.
- Desarrolle relaciones cercanas y satisfactorias con otros niños y adultos.
- Explore activamente su entorno y aprenda.

Dicho esto, las competencias sociales y emocionales se reconocen cada vez más como críticas para el éxito de los niños, tanto en la escuela como en otros entornos, y en fases posteriores de la vida hasta la edad adulta. Los especialistas en desarrollo infantil en múltiples disciplinas (p. Ej., Educación, medicina, bienestar infantil) reconocen la importancia del desarrollo social y emocional positivo para el bienestar general del niño y el tema continúa ganando importancia en el discurso público (DARLING-CHURCHILL Y LIPPMAN, 2016).

Las experiencias sociales y emocionales con los cuidadores primarios, así como las interacciones con otros niños y adultos en una etapa temprana de la vida, preparan el escenario para futuros resultados académicos y personales, y sustentan otras áreas de desarrollo. A medida que los niños desarrollan habilidades sociales y emocionales, adquieren la confianza y la

competencia necesarias para construir relaciones, resolver problemas y hacer frente a las emociones. Las competencias sociales y emocionales relacionadas con la preparación para la escuela han recibido una enorme atención. Las investigaciones indican que las habilidades sociales y las habilidades del proceso que las acompañan (p. Ej., Atención y enfoques de aprendizaje) evidentes al ingresar a la escuela (es decir, alrededor de los 5 años) son los mejores predictores de competencias sociales y emocionales posteriores, como manejar el comportamiento, establecer conexiones sociales y tolerar la frustración con los compañeros. Las competencias sociales y emocionales también suelen predecir de manera única el rendimiento académico, incluso cuando se tienen en cuenta otros factores, como el éxito académico anterior. Además, los niños con mayor autocontrol (un aspecto de la autorregulación) tienen más probabilidades de convertirse en adultos con mejor salud (por ejemplo, mejor salud física, menos abuso de sustancias), tener ingresos más altos y menos luchas financieras, y menos delincuencia. convicciones que aquellos con habilidades de autorregulación más débiles (DARLING-CHURCHILL Y LIPPMAN, 2016).

El avance del conocimiento sobre la importancia del desarrollo social y emocional de la primera infancia ha traído consigo una variedad de herramientas de responsabilidad de programas informativos y de servicio al niño. El creciente cuerpo de investigación que vincula las evaluaciones de las competencias sociales y emocionales con los resultados del niño, y que involucra el uso de estas evaluaciones para informar la práctica y la intervención, proporciona evidencia acumulada que cuando los niños pequeños pueden desarrollar relaciones prosociales, sentirse seguros de sí mismos y expresar y manejar sus emociones, es más probable que estén preparados para aprender y tener éxito en la escuela. Sin embargo, a pesar de esta preponderancia de evidencia, el desarrollo de medidas psicométricamente válidas que están alineadas para su uso dentro de los sistemas de evaluación y rendición de cuentas se ha retrasado (DARLING-CHURCHILL Y LIPPMAN 2016).

2.1.6.4. Área de desarrollo físico y motor

La adquisición de competencia en las actividades motoras es una importante tarea de desarrollo de la niñez. Todos los niños, excepto algunos con discapacidades graves, tienen el potencial de desarrollar y aprender una variedad de patrones de movimiento fundamentales y habilidades motoras más especializadas. Estas actividades son una parte integral del repertorio conductual de los niños y proporcionan el medio a través del cual los niños experimentan muchas dimensiones de sus entornos (MALINA, 2004).

El desarrollo de la competencia motora durante la infancia depende y está influenciado por las características de crecimiento y madurez del niño (morfológica, fisiológica y neuromuscular). El entorno en el que se cría un niño también es importante. Las oportunidades ambientales y las

restricciones para el movimiento interactúan con los sustratos biológicos del crecimiento y la maduración para determinar el repertorio motor del niño (MALINA, 2004).

El desarrollo motor es el proceso a través del cual un niño adquiere patrones y habilidades de movimiento. Es un proceso continuo de modificación que involucra la interacción de varios factores: (1) maduración neuromuscular; (2) el crecimiento físico y las características de comportamiento del niño; (3) el ritmo del crecimiento físico, la maduración biológica y el desarrollo del comportamiento; (4) los efectos residuales de las experiencias de movimiento previas y (5) las nuevas experiencias de movimiento. Todo esto ocurre en el contexto de los entornos físicos y socioculturales en los que se cría un niño. Las experiencias ambientales interactúan con el crecimiento y la maduración para influir en el desarrollo motor y la competencia (MALINA, 2004).

Cuando un niño no recibe una estimulación temprana adecuada en la primera infancia, es posible que nunca alcancen la competencia total en determinadas actividades físicas en la adolescencia o la edad adulta. (SEEFELDT, 1980) argumentó que podría haber un "umbral crítico" de competencia en habilidades motoras. Por encima de este umbral, los niños podrán completar las tareas y actividades físicas necesarias en la vida adulta, mientras que por debajo del umbral los niños no podrán realizar algunas habilidades y tareas motoras y no podrán realizar determinadas actividades físicas. Por tanto, la competencia en habilidades motoras es un determinante importante de la actividad física. Además, se ha asumido que la percepción de un niño de su competencia en una tarea motora influye en la persistencia del niño en la tarea. Como resultado, las percepciones de la competencia y la dificultad de la tarea también influyen en la participación y el compromiso del niño en una variedad de tareas y juegos físicos.

2.1.7. Motricidad fina

Un niño en edad preescolar agudiza la percepción visual y las habilidades motoras finas necesarias para realizar actividades que requieren integración visomotora, como la escritura a mano. Las habilidades motoras finas también juegan un papel importante en el sustento de la vida diaria de las personas. La competencia en la motricidad fina es un componente esencial de las actividades de la vida diaria. Las habilidades motoras finas deficientes pueden provocar un aumento de la ansiedad, la angustia en el rendimiento académico y una baja autoestima (AKIN, 2019).

La mayor parte de la vida diaria de los niños en edad escolar se pasa en la escuela y el 30-60% de este tiempo se dedica a realizar tareas de motricidad fina. Casi el 85% del tiempo dedicado a las tareas de motricidad fina consiste en actividades basadas en papel y lápiz, que son quizás las habilidades más importantes para el rendimiento académico. Se encontró que los niños con

fuertes habilidades motoras finas mostraron un desarrollo más temprano en las habilidades de lectura y un mayor rendimiento académico (AKIN, 2019).

Debido al desarrollo de la tecnología, la cantidad de tiempo que los niños están expuestos a productos tecnológicos aumenta día a día, lo que limita las experiencias de movimiento necesarias para que los niños continúen su vida diaria. Sin embargo, algunos estudios han sugerido que las habilidades motoras finas no se ven afectadas negativamente porque se requieren habilidades motoras finas para utilizar productos tecnológicos (ADAMS, MARGARON Y KAPLAN, 2012). Sin embargo, estas habilidades se ven obligadas a una órbita diferente, lo que conduce a la pérdida de la motricidad fina que un individuo necesita en la vida diaria.

2.1.8. Motricidad gruesa

Históricamente, la investigación sobre el desarrollo infantil se ha centrado en medir la capacidad cognitiva temprana y posterior, con intervenciones destinadas a mejorar el aprendizaje y la capacidad de aprendizaje. Actualmente, existe un renovado interés en el papel que juega el desarrollo motor en el desarrollo cognitivo, social y emocional. En varias investigaciones se sugiere que el desarrollo motor puede actuar como un "parámetro de control" para un mayor desarrollo, en el sentido de que algunas habilidades motoras pueden ser un requisito previo para la adquisición o práctica de otras funciones del desarrollo como la capacidad perceptiva o cognitiva. La propuesta de que existe una relación entre el desarrollo motor y el desarrollo cognitivo no es nueva. Por ejemplo, Piaget en 1953 propuso que la actividad y la experiencia sensoriomotora son importantes para el surgimiento de la capacidad cognitiva (PIEK et al. 2008).

Las habilidades motoras gruesas involucran los músculos grandes del cuerpo que permiten movimientos corporales importantes, como caminar. Como un subconjunto de las habilidades motoras gruesas, las habilidades locomotoras en los niños incluyen aquellas que requieren movimientos fluidos del cuerpo a medida que el niño se mueve horizontal o verticalmente de un lugar a otro, como caminar, correr y saltar. Las habilidades de control de objetos son aquellas que demuestran movimientos eficientes de lanzar, golpear y atrapar. Por ejemplo, a la edad de 3 años, la mayoría de los niños pueden subir y bajar escaleras alternando los pies, saltar en su lugar y lanzar por encima de la cabeza. A los 4 años, la mayoría de los niños pueden atrapar una pelota que rebota, saltar con un comienzo de carrera y pedalear en un triciclo. A los 5 años, la mayoría de los niños pueden galopar con un pie, rodar como un tronco y propulsarse en un columpio. Se ha demostrado que la competencia motora está asociada con múltiples aspectos de la salud, como la actividad física, la aptitud cardiovascular y el estado de peso saludable, así como con el desarrollo del lenguaje y el rendimiento académico (KWON Y O'NEILL, 2020).

Comprender los factores que influyen en la competencia de las habilidades motoras es importante para educar a los niños pequeños para que tengan la mejor oportunidad de desarrollar habilidades motoras fundamentales adecuadas y para informar las pautas basadas en el movimiento que pueden ayudar a los padres y médicos. Según los estudios, la obesidad se asocia con deficientes habilidades motoras relacionadas con el peso corporal, como saltar y saltar, entre los niños de 4 años. Entre los factores conductuales, se ha demostrado que los niveles más altos de actividad física se asocian con una mayor competencia motora gruesa entre los niños en edad preescolar (KWON Y O'NEILL, 2020).

2.2. Herramientas de evaluación de habilidad motriz

El desempeño de una habilidad motora fundamental en la primera infancia se puede investigar utilizando una amplia gama de herramientas de evaluación. La herramienta de evaluación que se utilice depende del tipo de prueba, el contexto y la disponibilidad. Además, la mayoría de las pruebas están dirigidas a un grupo específico, por lo que es útil centrarse en las habilidades que son más relevantes para cada grupo. Hay dos clasificaciones principales para los instrumentos de prueba: referenciado a normas y referenciado a criterio. La escala con referencia a normas se utiliza para comparar una norma estándar, que se estableció previamente, con un grupo o persona. Por otro lado, la escala con referencia a criterios señala si una persona coincide con el nivel de habilidad que se sabe que se adquiere en secuencia. El individuo se compara con su propio desempeño anterior en lugar de con una norma grupal. La mayoría de las pruebas que se utilizan para acceder al desarrollo motor de un niño se basan en normas (BRAGANÇA MORAES, 2018).

Otro enfoque clasifica las herramientas de medición de habilidades motoras como evaluación orientada al proceso u orientada al producto. Las evaluaciones orientadas al producto “*se basan en el tiempo, la distancia o el número de intentos exitosos que resultan del desempeño de una habilidad*”. Básicamente, este enfoque se centra solo en el resultado final de la actuación. Por otro lado, las evaluaciones orientadas al proceso “*se refieren a cómo se realizó la habilidad o al proceso responsable del resultado del desempeño*” (BURTON Y MILLER, 1998). Las herramientas de evaluación más comunes que se encuentran en la literatura son: Motoriktest für Vier- bis Sechsjährige Kinder (MOT 4-6), Batería de evaluación del movimiento para niños (Movement-ABC), Escalas de desarrollo Peabody (PDMS), Prueba de desarrollo motor grueso (TGMD), la prueba de Maastrichtse Motoriek (MMT) y la prueba de Bruininks-Oseretsky de competencia motora (BOTMP) (BRAGANÇA MORAES, 2018).

2.3. Prueba de desarrollo motriz grueso (TGMD-2)

El objetivo del TGMD-2 (ULRICH, 2000), es medir la motricidad gruesa de los niños de 3 a 10 años, a partir de aspectos cualitativos de su desempeño. Según (ULRICH 2000; BURTON Y MILLER, 1998), la prueba puede describirse como orientada al proceso y al producto, así como como una herramienta de evaluación con referencia a normas y criterios. Se utiliza para identificar si el desarrollo de las habilidades motoras de un niño está significativamente detrás del desarrollo de sus compañeros. Además, la prueba es útil para planificar un programa de instrucción para ayudar en el desarrollo de habilidades motoras, para evaluar el progreso del niño, para evaluar el éxito del programa de habilidades motoras y para servir como un instrumento de medición para la investigación que involucra el desarrollo motor grueso (BRAGANÇA MORAES, 2018).

La prueba se divide en la subprueba de locomotora y la subprueba de habilidad con el balón. La subprueba de locomotora incluye correr, galopar, brincar, brincar horizontalmente y deslizarse. La subprueba de destreza con el balón incluye golpe de dos manos de una pelota estacionaria, golpe de derecha con una mano de una pelota autobotada, regate estacionario con una mano, atrapar una pelota estacionaria con las dos manos, patear una pelota estacionaria, lanzamiento por encima y por encima de la mano (ULRICH 2000). Los 12 ítems toman alrededor de 20 minutos para evaluar, y cada uno de ellos tiene una serie de criterios de desempeño que se deben demostrar y usar para evaluar si el desempeño de la habilidad se considera adecuado o deficiente. El niño tiene que realizar cada ítem dos veces y cada intento recibe una puntuación de '0' si se realiza incorrectamente y una puntuación de '1' si se realiza correctamente. Se obtiene una puntuación bruta para cada elemento mediante la suma de los dos ensayos. Los puntajes brutos luego se traducen a puntajes estándar considerando la edad del niño (BRAGANÇA MORAES, 2018), que para este trabajo de titulación se acordó que sea de 4 a 5 años.

En este trabajo de titulación sólo se va a utilizar la subprueba locomotora. Esta mide las siguientes habilidades motoras gruesas que requieren movimientos fluidos coordinados del cuerpo a medida que el niño se mueve en una dirección u otra.

- **Correr:** La capacidad de avanzar de manera constante dando pasos bruscos para que ambos pies abandonen el suelo por un instante con cada paso.
- **Rebote de pelota:** Capacidad de hacer rebotar una pelota contra el piso utilizando una sola mano.
- **Slide:** La capacidad de replicar un movimiento determinado de brazos y piernas horizontalmente.
- **Kick:** La capacidad de patear un balón a una corta distancia.

- Lanzamiento de pelota una mano: la capacidad de lanzar un pequeño objeto con la mano, teniendo un impulso inicial
- **Salto largo:** La capacidad de realizar todas las habilidades asociadas con saltar sobre un objeto.

En la ilustración 2-2, se muestran gráficamente las habilidades motoras gruesas explicadas anteriormente.

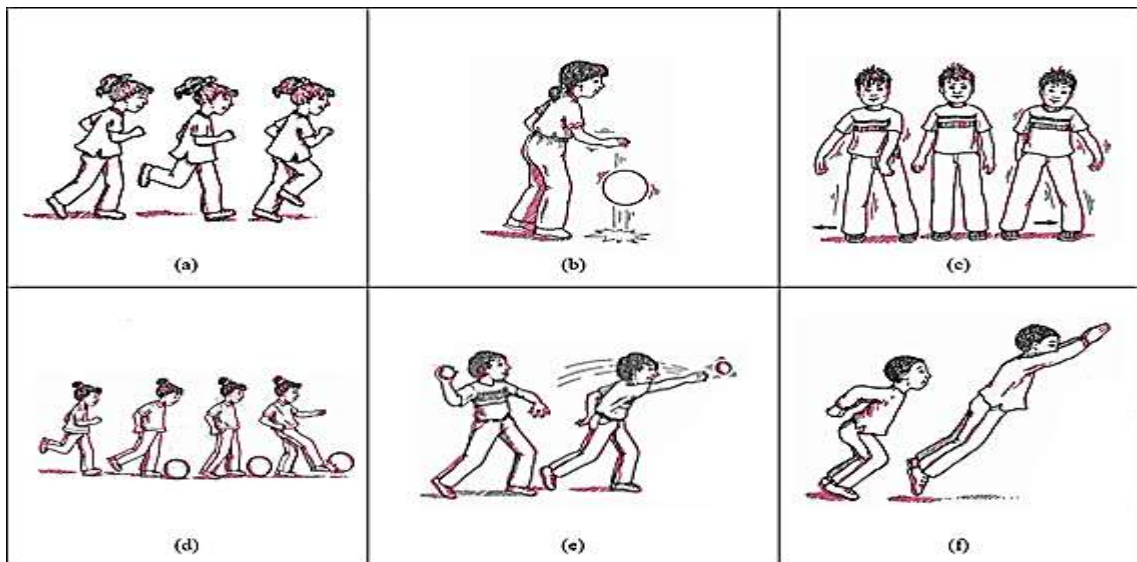


Ilustración 2-2: Habilidades motores gruesas: (a) Correr propio sitio, (b) Rebote Pelota una mano, (c) Slide, (d) Kick, (e) Lanzamiento pelota una mano, (f) Salto largo

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.4. Visión artificial

2.4.1. Breve historia de la visión artificial

Es comúnmente aceptado que el padre de la Visión Artificial es Larry Roberts, quien en su tesis de doctorado en el MIT discutió las posibilidades de extraer información geométrica 3D desde vistas en perspectiva 2D de bloques (poliedros) (ALOIMONOS, 1992). Muchos investigadores, en el MIT y en otros lugares, en Inteligencia Artificial, siguieron este trabajo y estudiaron la visión por computadora en el contexto del mundo de los bloques. Más tarde, los investigadores se dieron cuenta de que era necesario abordar imágenes del mundo real. Por lo tanto, se necesitaba mucha investigación en las llamadas tareas de visión de “bajo nivel”, como la detección de bordes y la segmentación.

Un hito importante fue el marco propuesto por David Marr en el MIT, quien adoptó un enfoque ascendente para la comprensión de la escena (MARR, 2010). Los algoritmos de procesamiento de imágenes de bajo nivel se aplican a las imágenes 2D para obtener el "boceto primario"

(segmentos de borde dirigido, etc.), a partir del cual se obtiene un boceto 2.5 D de la escena usando estéreo binocular. Finalmente, se utilizan técnicas de alto nivel (análisis estructural, conocimiento a priori) para obtener representaciones de modelos 3D de los objetos en la escena. Este es probablemente el trabajo más influyente en la visión por computadora. Muchos investigadores lloraron: "*Desde el paradigma creado para nosotros por Marr, nadie puede expulsarnos*".

Sin embargo, más recientemente, varios investigadores de visión por computadora se dieron cuenta de algunas de las limitaciones del paradigma de Marr y abogaron por un enfoque más heterogéneo y de arriba hacia abajo. Básicamente, el programa de Marr es extremadamente difícil de llevar a cabo, pero, lo más importante es que para muchas, si no para la mayoría de las aplicaciones de visión por computadora, no es necesario obtener modelos completos de objetos en 3D. Por ejemplo, en la navegación autónoma del vehículo usando la visión por computadora, puede ser necesario averiguar solo si un objeto se está alejando o acercándose a su vehículo, pero no el movimiento 3D exacto del objeto. Este nuevo paradigma a veces se llama "Visión de Propósito", lo que implica que los algoritmos deben ser dirigidos por objetivos y en muchos casos podrían ser cualitativos (ROBERTS, 1963). Uno de los principales defensores de este nuevo paradigma es Yiannis Aloimonos, de la Universidad de Maryland.

Al revisar la historia de la visión por computadora, es importante tener en cuenta que, debido al amplio espectro de aplicaciones potenciales, la tendencia ha sido la fusión de la visión por computadora con otros campos estrechamente relacionados. Estos incluyen: Procesamiento de imágenes (las imágenes sin procesar deben procesarse antes de un análisis posterior). Fotogrametría (las cámaras utilizadas para la obtención de imágenes deben calibrarse. Determinar las poses de los objetos en 3D es importante tanto en la visión por computadora como en la fotogrametría). Gráficos por computadora (el modelado 3D es fundamental tanto para la visión por computadora como para los gráficos por computadora. Muchas aplicaciones interesantes necesitan visión por computadora y gráficos por computadora.

2.4.2. Visión Artificial

Como algunos autores la llaman visión por computadora, se define como un campo de estudio que busca desarrollar técnicas para ayudar a las computadoras a ver y comprender la información dentro de imágenes digitales como fotografías y videos. La información puede significar cualquier cosa, desde modelos 3D, posición de la cámara, detección y reconocimiento de objetos hasta agrupar y buscar contenido de imágenes. Típicamente, esto implica desarrollar métodos que intenten reproducir la capacidad de la visión humana. El problema de la visión por computadora parece simple porque lo resuelven trivialmente las personas, incluso los niños muy

pequeños. Sin embargo, sigue siendo una sonda sin resolver basada en la comprensión limitada de la visión biológica y debido a la complejidad de la percepción de la visión en un mundo físico dinámico y casi infinitamente variable (BROWNLEE, 2019).

En un nivel abstracto, el objetivo de los problemas de visión por computadora es utilizar los datos de imagen observados para inferir algo sobre el mundo (PRINCE, 2012). Es un campo multidisciplinario que podría llamarse en términos generales un subcampo de inteligencia artificial y aprendizaje automático, que puede implicar el uso de métodos especializados y el uso de algoritmos generales de aprendizaje.

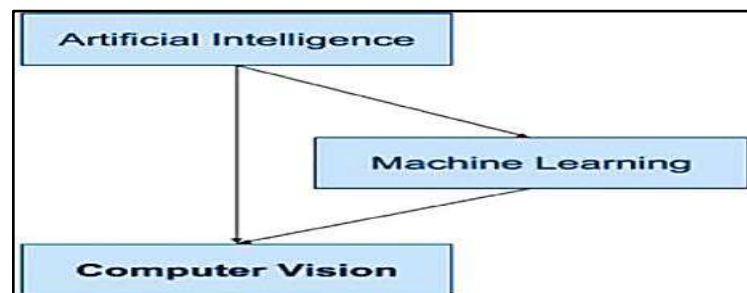


Ilustración 3-2: Relación entre Visión Artificial e Inteligencia Artificial

Fuente: (Brownlee 2019)

2.4.3. Importancia de la Visión artificial

La importancia de la visión por computadora radica en los problemas que puede resolver. Es una de las principales tecnologías que permite que el mundo digital interactúe con el mundo físico.

A medida que las cámaras se están convirtiendo en hardware de PC estándar y en una característica requerida de los dispositivos móviles, la visión por computadora pasa de ser una herramienta de nicho a una herramienta cada vez más común para una amplia gama de aplicaciones. Probablemente, algunas de estas aplicaciones saltan a la mente, como los programas de reconocimiento facial o las interfaces de juegos como Kinect. La visión por computadora también juega un papel muy importante en aplicaciones como los sistemas de seguridad automotriz, donde un automóvil detecta cuándo el conductor comienza a alejarse del carril o se está adormeciendo. Tiene aplicaciones en automatización industrial, biometría, medicina e incluso exploración planetaria. También se usa en algunos campos más sorprendentes, como la agricultura, donde se usa para inspeccionar y clasificar frutas y verduras. Es un campo diverso, con más y más aplicaciones interesantes que aparecen todos los días. En esencia, la visión por computadora se basa en los campos de las matemáticas, la física, la biología, la ingeniería y, por supuesto, la informática. Hay muchos campos relacionados con la visión por computadora, como el aprendizaje automático, el procesamiento de señales, la

robótica y la inteligencia artificial. Sin embargo, a pesar de que es un campo construido sobre conceptos avanzados, cada vez más herramientas lo hacen accesible para todos, desde aficionados a ingenieros de visión hasta investigadores académicos. Es un momento emocionante en este campo, y hay un sinnúmero de posibilidades para las aplicaciones. Una de las cosas que lo hace emocionante es que, en estos días, los requisitos de hardware son lo suficientemente económicos como para permitir que los desarrolladores más casuales ingresen al campo, abriendo la puerta a muchas nuevas aplicaciones e innovaciones (DEMAAGD; et al. 2012).

2.4.4. Etapas de la Visión Artificial

Según (GONZALEZ Y WOODS, 2007) las etapas fundamentales que comprenden un sistema Visión Artificial para el procesamiento de imágenes digitales son: adquisición de imágenes, preprocesamiento, segmentación, extracción de características (representación y descripción), reconocimiento (interpretación y clasificación), y base de conocimiento. En la ilustración 4-2, se muestra una representación gráfica de cada etapa y el modo de interacción de cada una con la otra. Además, en las siguientes subsecciones se presenta una breve explicación de cada etapa.

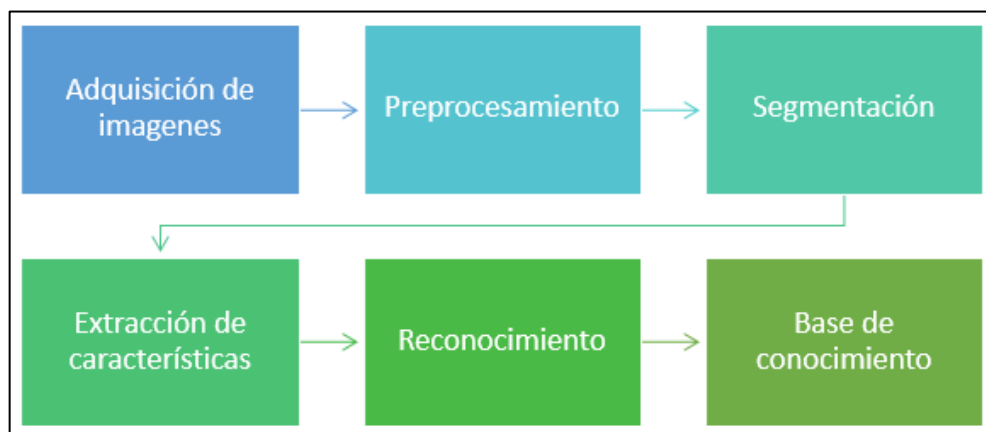


Ilustración 4-2: Etapas de la Visión Artificial

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.4.4.1. Adquisición de imagen

La mayoría de las imágenes que nos interesan son generadas por la combinación de una fuente de "iluminación" y la reflexión o absorción de energía de esa fuente por los elementos de la "escena" que se está representando. Incluimos iluminación y escena entre comillas para enfatizar el hecho de que son considerablemente más generales que la situación familiar en la que una fuente de luz visible ilumina una escena 3-D (tridimensional) común y cotidiana. Por ejemplo, la iluminación puede provenir de una fuente de energía electromagnética tal como un sistema de radar, infrarrojo o rayos X. Como se señaló anteriormente, podría originarse a partir de fuentes menos tradicionales, como el ultrasonido o incluso un patrón de iluminación

generado por computadora. Del mismo modo, los elementos de la escena pueden ser objetos familiares, pero también pueden ser moléculas, formaciones rocosas enterradas o un cerebro humano. Dependiendo de la naturaleza de la fuente, la energía de iluminación se refleja o se transmite a través de los objetos. Un ejemplo en la primera categoría es la luz reflejada desde una superficie plana. Un ejemplo de la segunda categoría es cuando los rayos X atraviesan el cuerpo de un paciente con el fin de generar una película de rayos X de diagnóstico (Gonzales & Woods, 2010).

Antes de que pueda comenzar cualquier procesamiento de imágenes, una cámara debe capturar una imagen y convertirla en una entidad manejable. Por lo tanto, para adquirir una imagen digital, se necesita un sensor de imagen y la capacidad de digitalizar la señal producida por ese sensor (WANDELL, EL GAMAL Y GIROD, 2002). El sensor puede ser una cámara de televisión, una cámara de exploración de línea, video, escáner, etc. Si la salida del sensor no es digital, entonces se necesita un convertidor analógico-digital para digitalizar la imagen. La imagen digital se obtiene como resultado del muestreo y la cuantización de una imagen analógica o creada ya en forma digital.

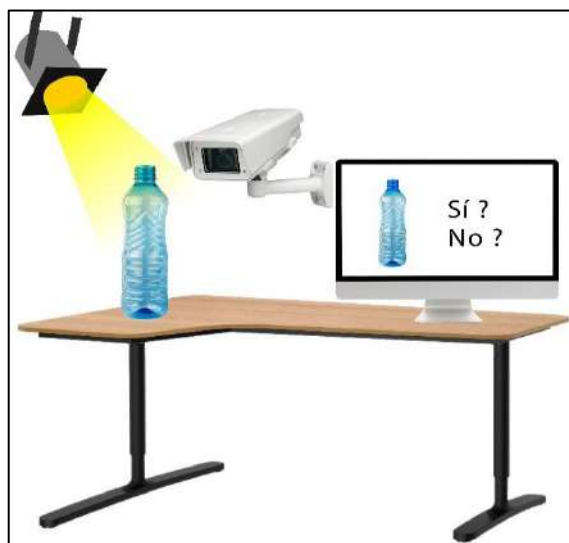


Ilustración 5-2: Adquisición de imagen

Realizado por: Fajardo C., Yanchapanta W., 2022.

Típicamente, una imagen digital se representa como una matriz bidimensional de números reales.

2.4.4.2. Preprocesamiento



Ilustración 6-2: Procesamiento de imagen

Realizado por: Fajardo C., Yanchapanta W., 2022.

Una vez que se ha adquirido una imagen digital, se pueden aplicar varios métodos de preprocesamiento para mejorar los datos de la imagen antes del procesamiento computacional. En esta etapa, la imagen se procesa y se convierte en una forma adecuada para su posterior análisis (CHOI, RO Y PLATANOTIS, 2011). La mayoría de las aplicaciones de visión por computadora requieren cuidado al diseñar la etapa de procesamiento para lograr resultados aceptables. Las operaciones de preprocesamiento también se denominan filtración.

Ejemplos de tales operaciones incluyen suavizado, corrección de exposición y balance de color, reducción de ruido (eliminación de ruido), aumento de nitidez, desdibujado de imagen, separación del plano de imagen, normalización, etc. La imagen obtenida después de esta etapa es la entrada al paso de segmentación (VERGARA; et al. 2014).

Los datos adquiridos suelen ser desordenados y provienen de diferentes fuentes. Para incluirlos a la red neuronal, deben estandarizarse y limpiarse. La mayoría de las veces, el preprocesamiento se utiliza para realizar pasos que reducen la complejidad y aumentan la precisión del algoritmo aplicado. No podemos escribir un algoritmo único para cada una de las condiciones en las que se toma una imagen, por lo tanto, cuando adquirimos una imagen, tendemos a convertirla en una forma que permita que un algoritmo general la resuelva (ELGENDY, 2020).

2.4.4.3. Segmentación

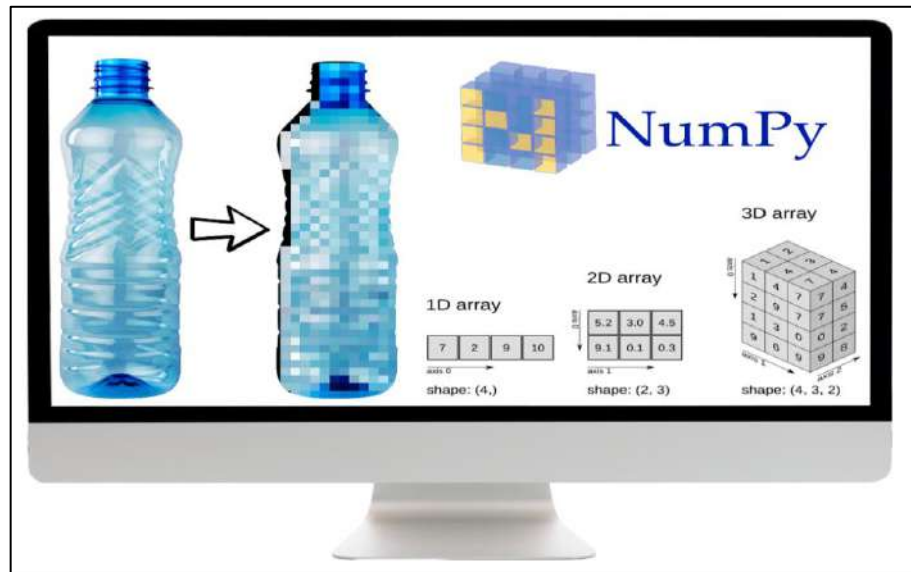


Ilustración 7-2: Segmentación de imagen

Realizado por: Fajardo C., Yanchapanta W., 2022.

La segmentación autónoma es una de las etapas más difíciles en Visión Artificial. La segmentación es el proceso de dividir una imagen digital en múltiples segmentos (conjuntos de píxeles, también conocidos como superpíxeles). El objetivo es simplificar y cambiar la representación de una imagen en algo que sea más significativo y fácil de analizar (NANTHAGOPAL Y SUKANESH, 2013). Un proceso de segmentación grueso retrasa el resultado de una solución satisfactoria. Por otro lado, un proceso de segmentación débil en la mayoría de los casos conducirá a errores.

Como resultado del proceso de segmentación, los datos de píxeles sin procesar que constituyen los límites entre las regiones o la información sobre qué píxel pertenecen a qué región que se obtiene. La segmentación de imagen es la operación que marca la transición entre el procesamiento de imágenes de nivel bajo a nivel medio. Entre los métodos de segmentación más utilizados se encuentran: umbrales, detección de contornos, cuencas morfológicas basadas en regiones y crecimiento regional (VERGARA, et al. 2014).

En (PÉREZ Y BROMBERG, 2012) se menciona que la segmentación, se define como un método para particionar una imagen I en R_k regiones, con $k = 1, \dots, K$, tal que cada región R_k es un objeto candidato. También indica que segmentación es el conjunto de píxeles en regiones, tal que las regiones R_k forman una partición, es decir, abarcan toda la imagen y no se superponen entre sí. Cada región R_k satisface un criterio de uniformidad (todos los píxeles de la región tienen una o más propiedades en común). Los píxeles pertenecientes a regiones adyacentes, cuando se toman en conjunto, no satisfacen el criterio de uniformidad.

2.4.4.4. Extracción de características

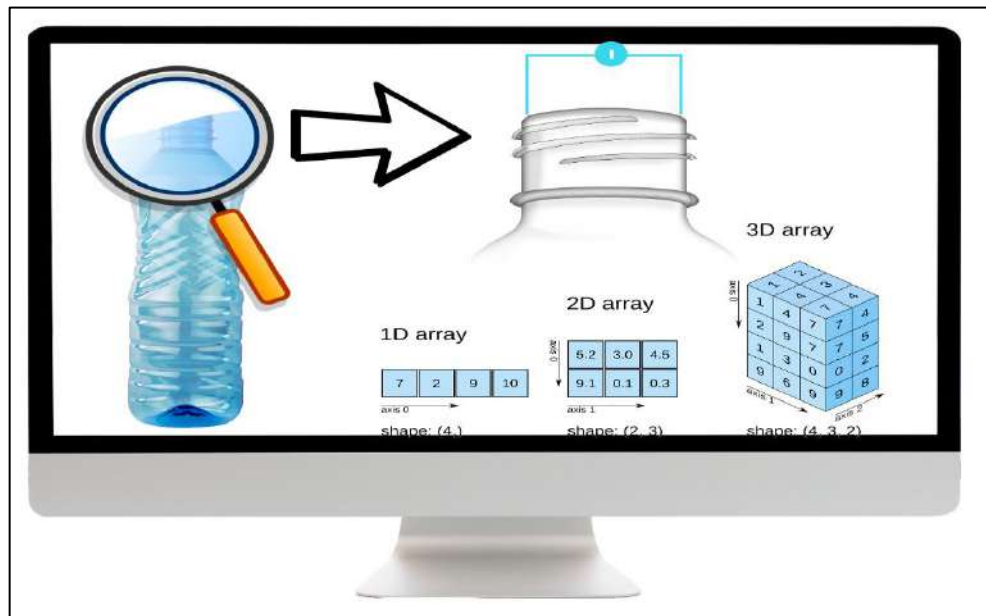


Ilustración 8-2: Extracción de características de una imagen

Realizado por: Fajardo C., Yanchapanta W., 2022.

Una vez que se ha segmentado una imagen, se pueden describir las regiones individuales resultantes. La extracción de características, también llamada representación y descripción de imágenes, es la operación realizada para extraer y resaltar características con cierta información cuantitativa que es esencial para distinguir una clase de objetos de otra. Es un paso crítico en la mayoría de las soluciones de visión por computadora porque marca la transición de la representación de datos pictóricos a los no pictóricos (VERGARA, et al. 2014).

Para almacenar las características extraídas de un objeto, se construye una matriz $n \times 1$ llamada vector de características. El vector de características es una representación compacta de una imagen y su contenido puede ser simbólico, numérico o ambos. El principal desafío en este paso es que las características extraídas deben ser invariables a los cambios en rotación, escala, traslación y contraste. La obtención de los invariantes asegura que el sistema de visión por computadora podrá reconocer objetos incluso cuando aparezcan con diferente contraste, tamaño, posición y ángulo dentro de la imagen (MULLEN, MONEKOSSO Y REMAGNINO 2013).

En la visión por computadora, una característica es un dato medible en su imagen que es exclusivo de este objeto específico. Puede ser un color distinto en una imagen o una forma específica, como una línea, un borde o un segmento de imagen. Una buena característica se utiliza para distinguir objetos unos de otros. Por ejemplo, si se da una característica como una rueda, y se pide que adivine si el objeto es una motocicleta o un perro, la respuesta sería una motocicleta. En este caso, la rueda es una característica fuerte que distingue claramente entre motocicletas y perros. Si se da la misma característica (una rueda) y se pide que adivine si el

objeto es una bicicleta o una motocicleta. En este caso, esta característica no es lo suficientemente fuerte como para distinguir entre ambos objetos. Se debe buscar más funciones como un espejo, matrícula, etc (ELGENDY 2020).

2.4.4.5. Reconocimiento

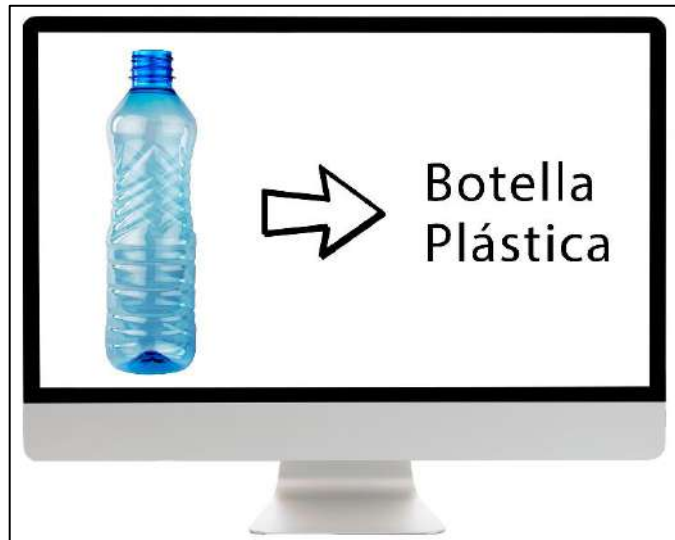


Ilustración 9-2: Reconocimiento de una imagen

Realizado por: Fajardo C., Yanchapanta W., 2022.

Esta etapa constituye el alto nivel de procesamiento de imágenes. El reconocimiento es el proceso para asignar una etiqueta a un objeto, en función de la información proporcionada por sus descriptores. Además, implica asignar significado a un conjunto de objetos reconocidos (XIAO Y WANG 2013). Los algoritmos de reconocimiento analizan las propiedades numéricas de varias características de imagen y clasifican los datos en categorías. Esta etapa se compone de dos fases: capacitación y evaluación.

Todos los algoritmos de clasificación se basan en el supuesto de que la imagen en cuestión representa una o más características y que cada una de estas características pertenece a una de varias clases distintas y exclusivas. Un analista especifica las clases a priori especificando el número de categorías deseadas (clasificación supervisada) o agrupadas automáticamente (clasificación no supervisada) en un conjunto de prototipos. Muy a menudo, los algoritmos utilizados para el reconocimiento incluyen redes neuronales artificiales (ANN), máquinas de vectores de soporte (SVM), distancia y medidas de similitud (k-Nearest Neighbours, Bayesian, Euclidean, Manhattan) y emparejamiento (VERGARA et al. 2014).

2.5. Inteligencia Artificial

En términos generales, la inteligencia artificial (o IA para abreviar) es, como su nombre indica, una simulación del proceso de inteligencia humana por máquinas como en un sistema

informático. Los sistemas IA generalmente tienen algunos de los comportamientos asociados con la inteligencia humana, como la planificación, el razonamiento, la resolución de problemas, la representación del conocimiento, el aprendizaje, el movimiento, la percepción, la manipulación y, en cierta medida, la inteligencia social y la creatividad. Hay dos tipos principales de IA, IA débil e IA fuerte (AHMET 2018).

- IA débil o estrecha existen en las computadoras hoy en día. Esta IA aprende ella misma cómo llevar a cabo tareas específicas sin tener que programarlos explícitamente. Un ejemplo perfecto es el reconocimiento de voz y lenguaje de Siri, el asistente virtual en iPhones, los sistemas de reconocimiento de visión de los automóviles sin conductor, o incluso la IA oculta que le muestra la publicidad o los sitios web que son relevantes para el espectador. Estas IA se denominan IA débil porque solo pueden aprender a hacer una tarea específica. La IA débil puede hacer muchas cosas, desde interpretar feeds de video desde drones que realizan inspecciones visuales de infraestructuras hasta tareas cotidianas, como organizar calendarios personales y comerciales. Pueden responder a consultas simples de servicio al cliente, coordinándose con otros sistemas de inteligencia para reservar una habitación de hotel al precio y ubicación correctos. Incluso puede ayudar a los radiólogos a detectar posibles tumores cancerosos en los rayos X, detectar el desgaste de los ascensores y marcar contenido inapropiado en línea para proteger a los espectadores (AHMET 2018).
- La IA fuerte, por otro lado, es una entidad completamente diferente. A diferencia de la IA débil que solo puede aprender una cosa, la IA fuerte tiene la misma inteligencia adaptable que se encuentra en los humanos. Esta flexibilidad les permite aprender cómo hacer muchas tareas diferentes, como construir hojas de cálculo, cortar el cabello o incluso conducir en función de su experiencia acumulada. Hasta ahora, todavía no existe, y los expertos en IA están divididos sobre cuándo será una realidad (AHMET 2018).

En la tabla 2-2 se pueden apreciar algunas de las diferencias que existen entre estos dos tipos de inteligencia artificial.

Tabla 2-2: Principales diferencias entre IA Débil e IA Fuerte

Inteligencia artificial débil	Inteligencia artificial fuerte
Existen en la vida real (AlphaGo, Watson, Sophia, Alexa, Cortana, Siri).	Sólo en la ciencia ficción (T-800, Sony, Wall-E, J.A.R.V.I.S.).
Orientado a problemas muy concretos (Juega muy bien al Go).	Resuelven problemas abiertos (Un poco de todo: viaje temporal, Eliminar a John Connor, etc.).
Reactivo (“Esperaré a que empieces a jugar”).	Proactivo (“Necesito tu ropa, tus botas y tu motocicleta”).
Sin flexibilidad (“Sólo al Go, no juegos más complejos”).	Son flexibles (“Correr es como caminar, pero más rápido”).
Programa un humano (“Dime que tengo que pensar”).	Se autoprograman (“Corriendo, aprendo sobre mis límites”).
Pocas redes neuronales ($p \rightarrow q$).	Muchas redes neuronales, a veces en conflicto (“Decido según mi programación”).

No razonan, sólo computan (“Juego bien, pero inconscientemente”).	Imitan el comportamiento humano (“Pienso, luego existo”).
Aprenden de ejemplos similares (“Presto atención a las fichas, pero no te escucho”).	Aprenden como las personas (“El ajedrez se parece a las damas”).
Tareas Repetitivas (“¿Otra partida?”).	Aprenden nuevas tareas (“Puedo programar”).
No pueden salirse de su marco de trabajo (“Del juego Go no me saques”).	Adaptabilidad a nuevos escenarios (“Connor eliminado, jefe, ¿qué toca?”).

Fuente: (WEEDIA BRAND SERVICE, 2018)

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.5.1. Aprendizaje automático (*Machine learning*)

El aprendizaje automático es un subcampo de la informática que evolucionó a partir del estudio del reconocimiento de patrones y la teoría del aprendizaje computacional en inteligencia artificial. El aprendizaje automático explora la construcción y el estudio de algoritmos que pueden aprender y hacer predicciones sobre los datos. Dichos algoritmos operan construyendo un modo a partir de entradas de ejemplo para hacer predicciones o decisiones basadas en datos, en lugar de seguir instrucciones estrictamente estáticas del programa (BISHOP 2006).

El aprendizaje automático se emplea en una variedad de tareas informáticas en las que no es factible diseñar y programar algoritmos explícitos. Las aplicaciones de ejemplo incluyen filtrado de spam, reconocimiento óptico de caracteres (OCR), motores de búsqueda y visión por computadora. El aprendizaje automático a veces se combina con la minería de datos, aunque eso se centra más en el análisis exploratorio de datos. El aprendizaje automático y el reconocimiento de patrones "pueden verse como dos facetas del mismo campo". Cuando se emplean en contextos industriales, los métodos de aprendizaje automático pueden denominarse análisis o modelos predictivos (STRICKLAND 2016).

2.5.2. Técnicas computacionales

2.5.2.1. *Hard Computing*

Conocida también como computación lógica binaria, es el enfoque tradicional utilizado en computación que necesita un modelo analítico establecido con precisión. El análisis objetivo de esta técnica es la estimación de resultados precisos, deterministas que definen acciones de control definidas utilizando un modelo o algoritmo matemático de conjuntos certeros. Sin embargo, la computación tradicional lógica, no es capaz de resolver los problemas del mundo real cuyo comportamiento es extremadamente impreciso donde la información cambia constantemente creando incertidumbre en la toma de decisiones (GUPTA Y KULKARNI 2013).

2.5.2.2. *Soft Computing*

Esta es un modelo de procesamiento de datos computacionales desarrollado para resolver los problemas no lineales que involucran soluciones inciertas, imprecisas y aproximadas a un

problema del entorno real multivariable. Este tipo de problemas requieren una intuición similar a la inteligencia humana basada en el conocimiento de la experiencia. El término *soft computing* fue descrita por primera vez por el Dr. Zadeh (Zadeh 2015), según él, esta técnica de computación es un enfoque que imita al razonamiento humano y aprende en un entorno de incertidumbre e imprecisión (DZITAC, FILIP Y MANOLESCU 2017).

Dentro del Soft Computing se exponen diferentes técnicas para la resolución de problemas de la vida real, que se encuentran bioinspirado al razonamiento de la mente humana, y son las siguientes:

2.5.2.3. Lógica Difusa

Se ocupa de la toma de decisiones y los problemas del sistema de control que no se pueden convertir en fórmulas matemáticas certeras. Básicamente, esto asigna las entradas a las salidas de una manera lógica y no lineal, ejemplo el control de un sistema de termostato, exactamente no se sabe que temperatura es la ideal, si es caliente, muy caliente o frio, está incertidumbre se puede tratar mediante esta técnica para poder tener el control adecuado como se esquematiza en la Ilustración 10-2.

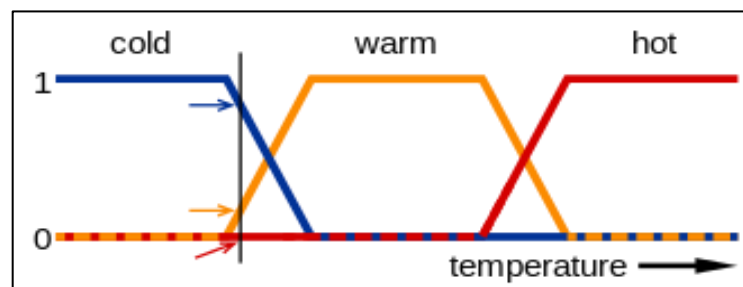


Ilustración 10-2: Fuzzy logic control temperatura

Fuente: (WIKIVERSITY, 2020)

2.5.2.4. Redes Neuronales Artificiales

Esta red está inspirada en el cerebro porque ambos tienen algunas similitudes. Al igual que en las redes neuronales biológicas, los sistemas de redes neuronales cuentan con una estructura análoga. Cuenta con elementos de entrada, parecido a las dendritas, una unidad de procesamiento de información similar al cuerpo celular de la neurona biológica, y el elemento de salida correspondiente al axón, como se observa en la ilustración 11-2.

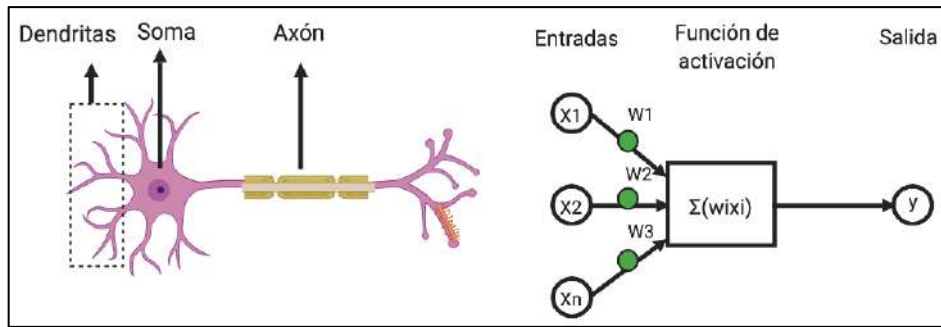


Ilustración 11-2: Neurona biológica y neurona artificial

Fuente: (GARCÍA 2020)

En la red neuronal, hay varias capas interconectadas de algoritmos llamados neuronas. Estos algoritmos se alimentan entre sí, y pueden ser entrenados para llevar a cabo tareas específicas al dar instrucciones simples que resaltan atributos importantes para ingresar datos cuando pasan entre capas. Puede sonar complicado, y lo es. Básicamente, la red neuronal funciona igual que un cerebro humano. Cada neurona hace una cosa específica, y está conectada a la neurona adyacente. Es posible estudiar lo que hace una neurona específica en un cerebro humano, y se puede entender el propósito general de un grupo de neuronas, pero todo el cerebro humano sigue siendo un misterio. Lo mismo podría decirse de la red de neuronas AI. Se puede estudiar un solo algoritmo y se puede entender el propósito de un clúster, pero no se puede descifrar todo el sistema detrás de la IA. Aun así, una cosa queda muy clara: todo el sistema simplemente funciona. También hay un subconjunto de aprendizaje automático conocido como aprendizaje profundo. Aquí, las redes neuronales se expanden en muchas redes en expansión en una gran cantidad de capas que están diseñadas para interpretar una gran cantidad de datos. Estas redes neuronales profundas impulsaron el salto hacia adelante en la capacidad de las computadoras para realizar tareas como reconocer el habla y la visión por computadora. También hay muchos tipos de redes neuronales, y todas ellas tienen sus propias fortalezas y debilidades. Las redes neuronales recurrentes, por ejemplo, son un tipo de red neuronal que se adapta bien al procesamiento del lenguaje y al reconocimiento del habla. Las redes neuronales convolucionales, por otro lado, están orientadas al reconocimiento de imágenes. El diseño de estas redes neuronales también está evolucionando (AHMET 2018).

Los investigadores están refinando una forma más efectiva de red neuronal profunda conocida como memoria a largo plazo, para permitir que las redes funcionen lo suficientemente rápido como para satisfacer la demanda de sistemas como Google Translate (AHMET 2018).

2.5.2.5. Algoritmos genéticos

El algoritmo genético es una técnica heurística de búsqueda y optimización inspirada en la evolución natural. Se han aplicado con éxito a una amplia gama de problemas del mundo real de gran complejidad. Esta técnica está fundamentada al principio darwiniano de evolución a través

de la selección (genética). Un algoritmo genético usa una versión muy abstracta de los procesos evolutivos para desarrollar soluciones a problemas dados. Cada Algoritmo Genético opera sobre una población de cromosomas artificiales. Estas son cadenas en un alfabeto finito (generalmente binario). Cada cromosoma representa una solución a un problema y tiene una idoneidad, un número real que es una media de cuán buena es la solución al problema en particular (MCCALL 2005).

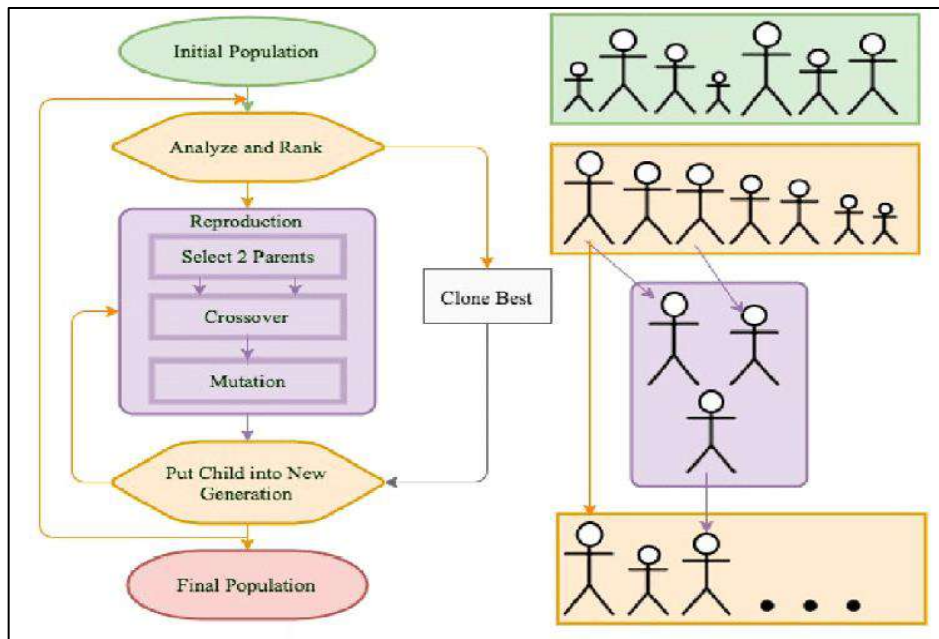


Gráfico 1-2: Diagrama de flujo del algoritmo genético

Fuente: (RESEARCHGEAT, 2020)

2.5.3. Redes Neuronales Artificiales Convolucionales

Una red neuronal convolucional es un algoritmo de aprendizaje automático que puede tomar una imagen de entrada, asignar pesos y sesgos aprendibles (características importantes) a varios aspectos y píxeles representativos de la imagen para poder diferenciar uno del otro. El proceso previo requerido en este algoritmo es mucho menor en comparación con otros algoritmos de clasificación (WALCZAK Y CERPA 2003). Mientras que en los métodos primitivos utilizando filtros adaptativos previos, prueba y error, las CNN tiene la capacidad de aprender diferentes filtros dependiendo de la situación y características para enfrentarse a una incertidumbre del entorno real (IJJINA Y CHALAVADI 2016).

La arquitectura funcional de las redes neuronales convolucionales es una analogía bio inspirada en el patrón de conectividad de las neuronas en el cerebro humano. Las neuronas individuales responden a los estímulos solo en una región restringida en el campo visual conocida como

campo receptivo, pero la conjugación de estos campos se puede llegar abarcar toda un área visual (WALCZAK Y CERPA 2003).

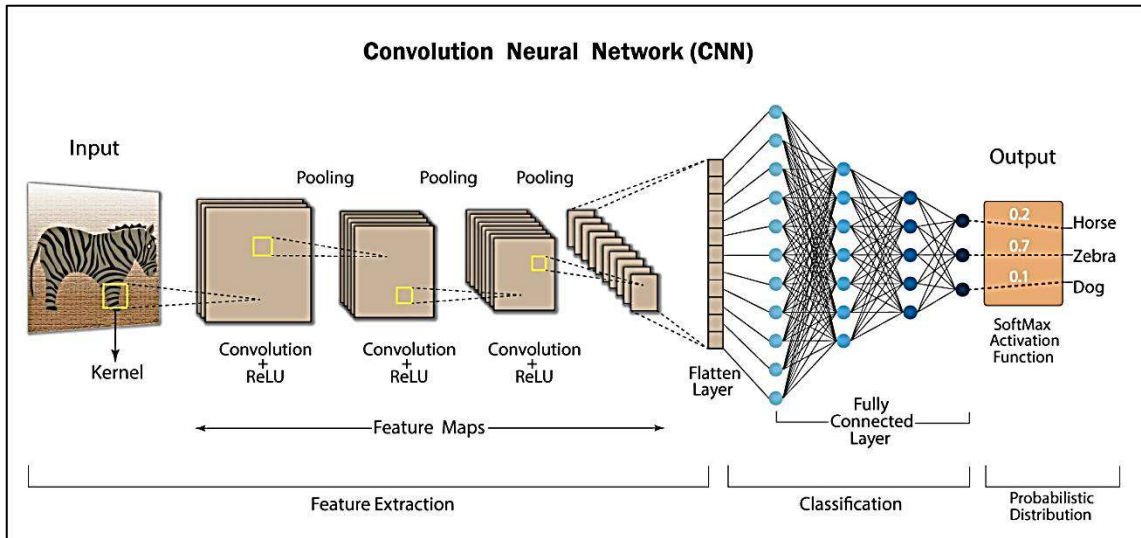


Ilustración 12-2: Proceso CNN

Fuente: (DEVELOPERSBREACH, 2020)

Una imagen no es nada más que una matriz de valores de pixeles, el proceso simplificado radica en la operar en una dimensión lineal, es decir una imagen con una matriz 3x3 a un vector 9x1 como se muestra en la ilustración 13-2, esto ingresa a la operatividad interna multicapa de la red neuronal (CHOI, RO Y PLATANOTIS 2011).

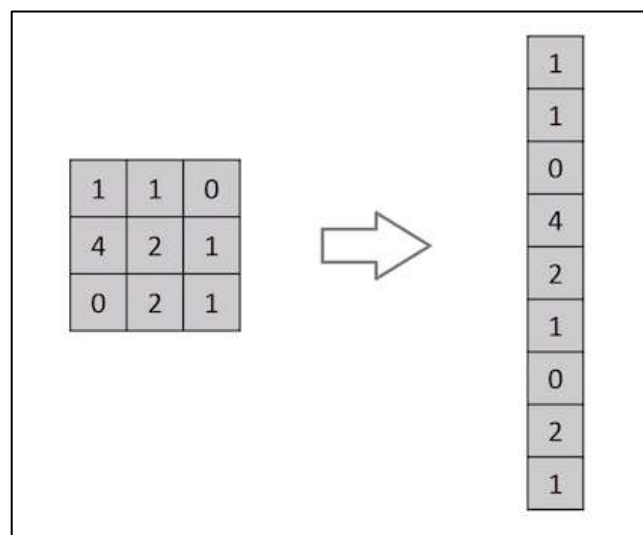


Ilustración 13-2: Neurona biológica y neurona artificial

Fuente:(CHOI, RO Y PLATANOTIS 2011)

En la ilustración 13-2, se ilustra la desfragmentación de una imagen en RGB en una matriz separada por sus tres planos de color, esto es un ejemplo para una imagen muy sencilla y fácil de tratar para un mejor entendimiento, pero en la hora de la practica estas matrices llegan a ser tan extensas que el procesamiento es muy tedioso y conlleva un mayor consumo de recursos en

memoria donde se realizan los cálculos. Es por eso, que las redes neuronales artificiales convolucionales reducen las imágenes a una forma que sea más fácil de procesar, sin perder características que son críticas para obtener una buena predicción. Esto es importante para diseñar una arquitectura, no solo ideal en reconocer patrones, si no procesarlos y operar con ellos a una escala grande (CHOI, RO Y PLATANOTIS 2011).

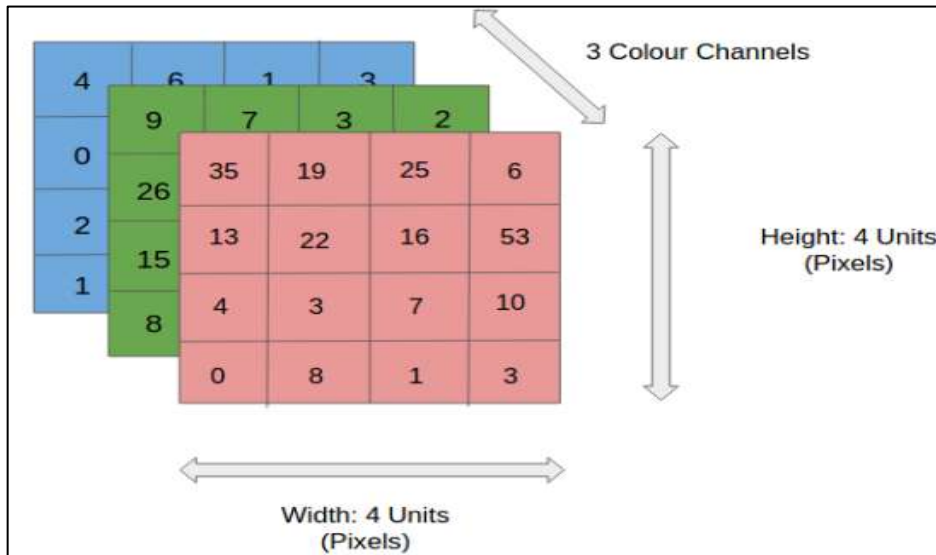


Ilustración 14-2: Ejemplo de la desfragmentación de una imagen RGB a matrices

Fuente:(CHOI, RO Y PLATANOTIS 2011)

Las Redes Neuronales Artificiales Convolucionales, en varios campos de reconocimiento de patrones han roto el molde en la esquematización del tratamiento de datos para emitir resultado bajo etiquetas previamente clasificadas, esta metodología llega a convertirse en la técnica de visión por computadora en la más usada en proyectos multidisciplinarios (SHIJIE et al. 2017).

2.6. Segmentación, reconocimiento de posiciones y articulaciones del cuerpo humano

Como se ha explicado anteriormente una imagen en RGB por sí sola no puede ser interpretado por el computador, es necesario acudir a modelos computacionales que desfragmente una imagen en unidades básicas de procesamiento matemático, esta tarea ayuda a determinar la ubicación precisa de los píxeles de puntos clave de la anatomía del cuerpo humano, como es nuestro caso de interés. Reconocer puntos característicos y relevantes de una postura humana y de las articulaciones de las extremidades es útil para tareas de nivel superior como reconocer acciones, movimientos y ejecutar tareas (NEWELL, YANG Y DENG 2016).

Los enfoques más recientes a esta línea de investigación se dividen en 2 categorías, 1) modelos de segmentación tradicionales ilustración 15-2, y 2) Modelos discriminativos basados en deeplearning ilustración 16-2 (TOMPSON et al. 2014).

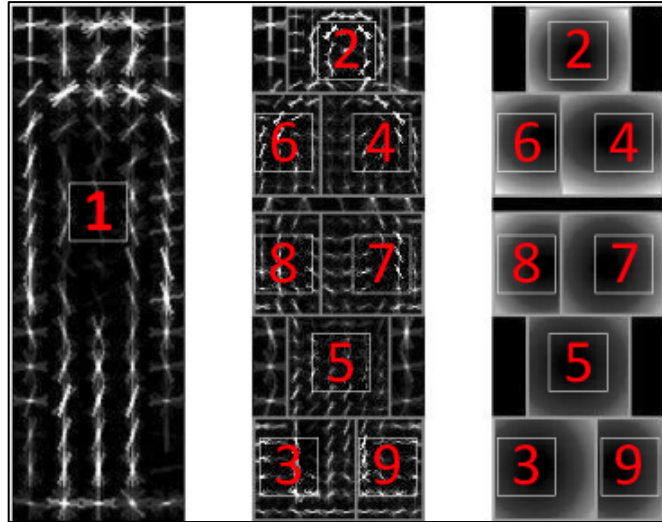


Ilustración 15-2: Modelos de segmentación cuerpo humano

Fuente:(TOMPSON et al. 2014)

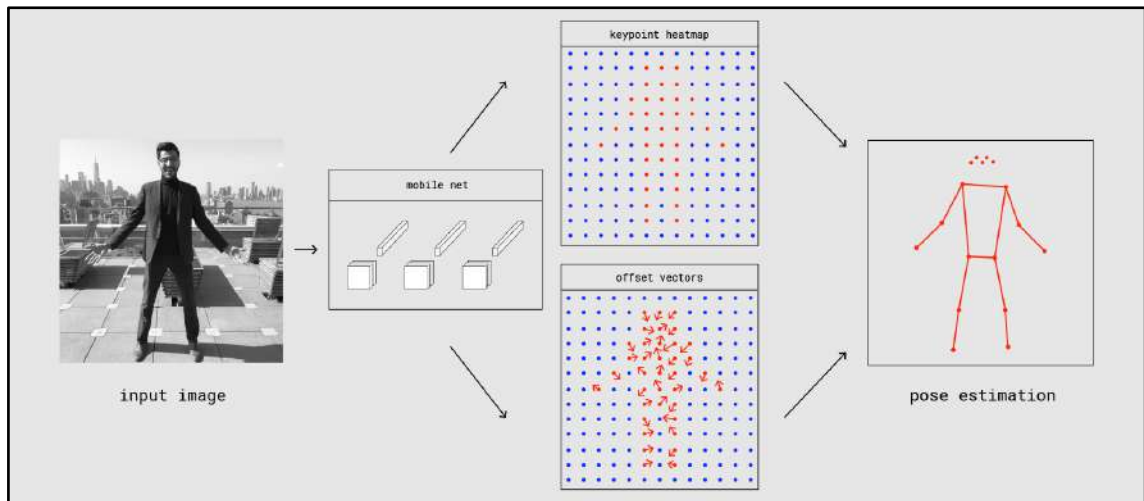


Ilustración 16-2: Segmentación cuerpo humano

Fuente: (THENEXTWEB, 2020)

2.6.1. Modelo de Deep Learning para identificar postura del cuerpo humano

Para la utilización de la terminología adecuada, a la postura del cuerpo humano lo identificaremos como pose, término en inglés que es comúnmente utilizado en artículos de interés científico. En la identificación de un pose según el autor (Toshev y Szegedy 2016), se debe considerar la codificación de las ubicaciones de todas las k articulaciones del cuerpo en un vector de pose definido como:

$$y = (\dots, y_i^T, \dots)^T, i \in \{1, \dots, k\} \quad \text{Ecuación 1-2}$$

Donde y_i contiene las coordenadas de X y Y, i^{th} es el punto de unión. Una imagen etiquetada se denota por (x, y) donde x representa los datos de la imagen y Y el vector que posee la representación actual de los puntos. Identificamos el cuerpo de la pose, y ese punto pertenece se

denota $b_c \in \mathbb{R}^2$, así como el ancho b_w y el alto b_h , $b = (b_c, b_w, b_h)$. Uniendo los puntos y_i se puede traducir por el centro del cuadro y la escala del cuadro al que nos referimos como normalización por b :

$$N(y_i; b) = \begin{pmatrix} 1/b_w & 0 \\ 0 & 1/b_h \end{pmatrix} \quad \text{Ecuación 2-2}$$

Además, se puede aplicar la misma normalización a los elementos del vector pose, $N(y; b) = (\dots, N(y_i; b)^T, \dots)^T$ dando como resultado un vector pose normalizado, para simplificarlo, se usará $N(x; b)$ para denotar un recorte de la imagen x por el cuadro delimitador b (TOSHEV Y SZEGEDY 2016).

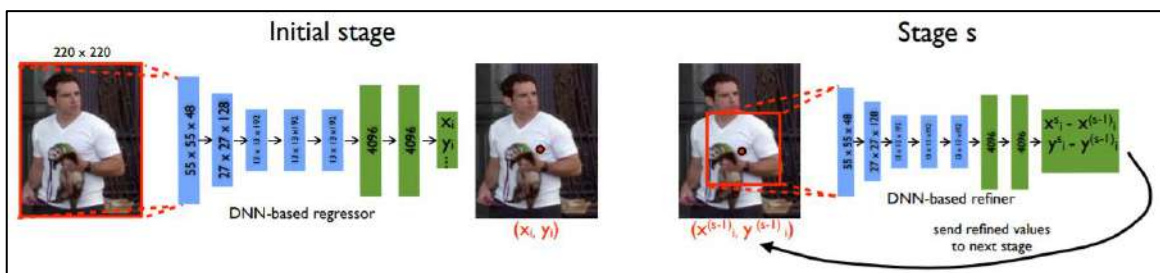


Ilustración 17-2: Proceso para la segmentación cuerpo humano

Fuente: (TOSHEV Y SZEGEDY 2016)

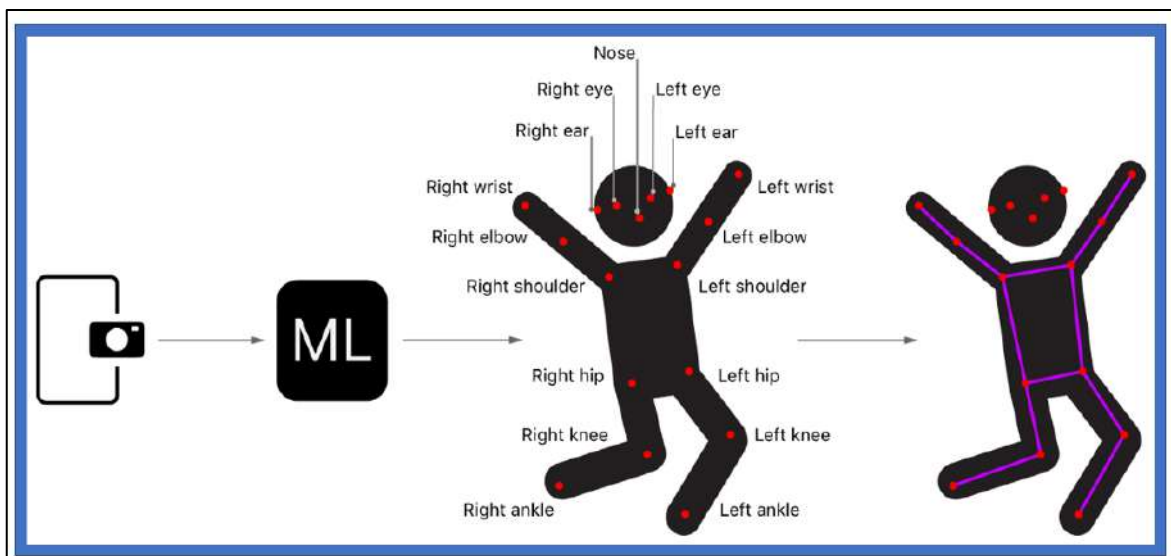


Ilustración 18-2: Segmentación cuerpo humano

Fuente: (DEVELOPER, 2020)

2.7. Modelos de Pose Estimation

La estimación de una postura se realiza en dos fases:

- **Primera fase:** a través de una red neuronal convolucional, se alimenta una imagen RGB como entrada.

- **Segunda fase:** se aplica un modelo de pose estimation única o múltiple para decodificar poses, posiciones de puntos clave, puntajes de confianza de poses y puntajes de confianza de puntos clave de los resultados del modelo.
- **Pose:** modelo computacional que identifica puntos característicos de las articulaciones principales del cuerpo humano.
- **Puntuaciones de confianza del Pose identificado:** este valor indica la confianza general en la estimación de la pose de una persona a partir de una imagen con valores entre 0.0 y 1.0. En función de este valor, el modelo decidirá mostrar las poses previstas u ocultarlas.
- **Puntos clave:** a partir de una imagen se determinan puntos relevantes en las uniones de las articulaciones de una persona, PoseNet es capaz de detectar 17 puntos clave (TENSORFLOW/TFJS-MODELS, 2019), y el modelo utilizado en esta investigación detecta 32 puntos (POSE, 2021).

2.7.1. PosNet Tensorflow

Es un modelo de estimación del cuerpo humano en tiempo real que en su motor de inteligencia artificial se basa en la arquitectura de Tensorflow.js ya que está enfocado a soluciones de producción basadas en la web.

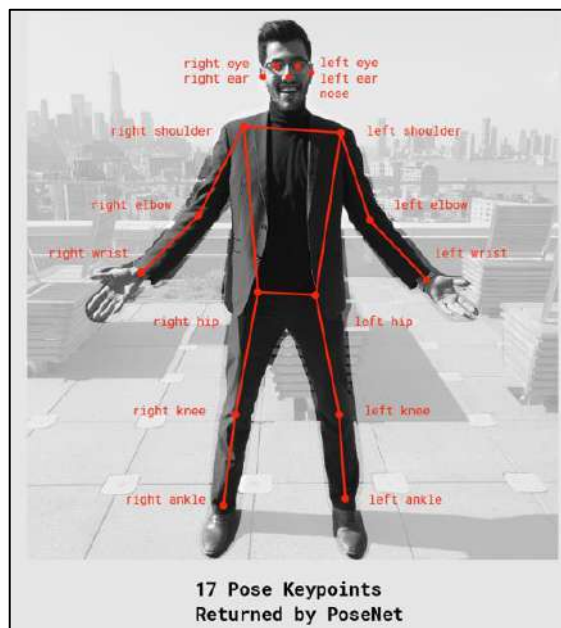


Ilustración 19-2: Puntos clave generados por PoseNet

Fuente: (TENSORFLOW/TFJS-MODELS, 2019)

La estimación que realiza PoseNet es la más simple y se puede ejecutar en tan solo 2 algoritmos. El caso de uso es ideal para cuando solo hay una persona en la imagen. La desventaja es que, si hay varias personas, los puntos clave de ambas personas probablemente se estimarán como parte de la misma pose, considerando a una sola persona, lo que significa, por

ejemplo, que el brazo izquierdo de la persona 1 y la rodilla derecha de la persona 2 podrían combinarse por el algoritmo como pertenecientes a la misma persona, como se observa en la ilustración 20-2.

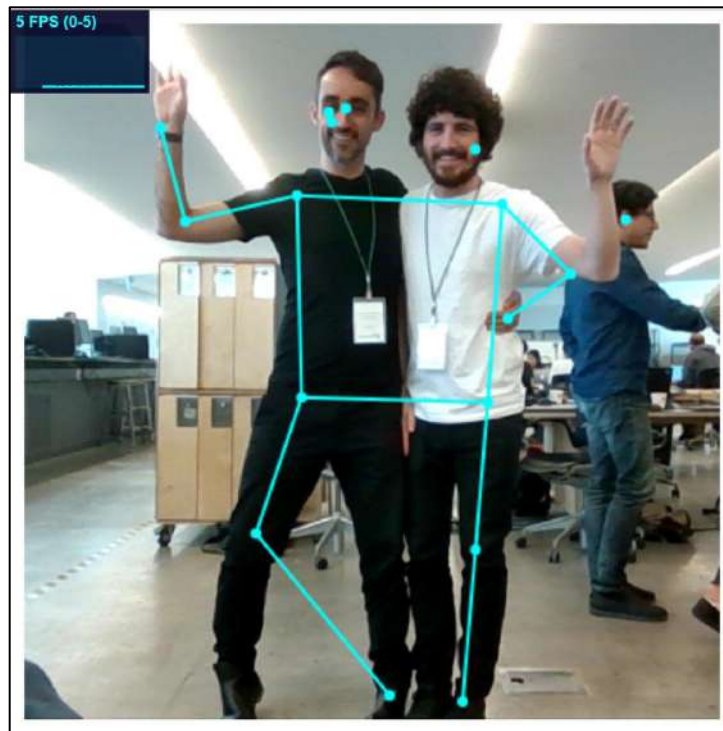


Ilustración 20-2: Puntos clave PosNet, debilidad del modelo

Fuente: (TWITTER, 2020)

2.7.2. *Mediapipe*

Para la creación de nuevas aplicaciones en diferentes áreas que procese entradas perceptivas como la identificación de articulaciones y partes del cuerpo humano, esta tarea implica más que ejecutar un modelo de aprendizaje automático. Los desarrolladores deben aprovechar las capacidades de una amplia gama de dispositivos y equilibrar los recursos para emitir resultados ideales a fin a lo esperado. Mediapipe nace con la necesidad de economizar recursos en el procesamiento de imágenes y estimaciones de aprendizaje automático en tiempo real de una manera que no se sacrifiquen los recursos de hardware. Mediapipe, administra los recursos de manera eficiente, tanto el CPU como el GPU, para un rendimiento de baja latencia, y así manejar la sincronización de datos en series de tiempo como fotogramas de audio y video para medir el rendimiento y consumo energético (LUGARESI et al. 2019).

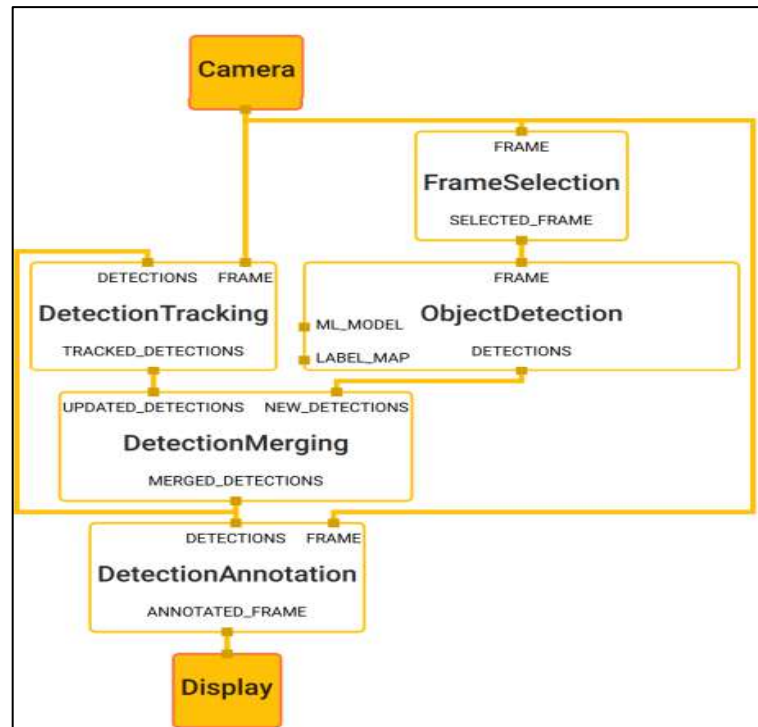


Gráfico 2-2: Diagrama de flujo de detección de objetos utilizando MediaPipe

Fuente: (LUGARESI et al. 2019)

En el gráfico 2-2, se muestra un ejemplo del diagrama de flujo que detecta objetos previamente entrenados, como puntos de referencia en rostros o cuerpos completos, los frameworks que se utilizan son los siguientes: 1) Tensorflow 2) Pytorch, 3) CNTK 4) MXnet que definen modelos de redes neuronales y MediaPipe asume el siguiente papel complementario de soporte nativo para la transmisión de datos en serie de tiempo con la ayuda de Open CV 4.0 (LUGARESI et al. 2019).

MediaPipe, consta de tres partes principales:

- Un Framework para la inferencia a partir de datos sensoriales.
- Un conjunto de herramientas para la evaluación del desempeño.
- Una colección de componentes de inferencia y procesamiento.

2.7.2.1. Mediapipe detección de objetos

Un requerimiento comúnmente utilizado en visión artificial y realidad aumentada es la detección de objetos en tiempo real. Dependiendo de la plataforma del dispositivo y la aplicación final, ejecutar la detección de objetos basada en machine Learning a una velocidad de fotogramas considerable ejemplo a 30FPS puede requerir un alto consumo de recursos o ser potencialmente inviable debido a los largos tiempos de emisión de resultados. Una alternativa es aplicar la detección de objetos a un flujo de fotogramas submuestreos temporalmente y propagar los resultados de la detección, es decir, cuadros limitadores y las etiquetas de clase

correspondientes como se ve en la ilustración 19-2, esto por cada fotograma, el seguimiento y la detección debe ejecutarse en paralelo, de modo que el tracking no sea bloqueado por el detector y pueda procesar cada fotograma. Este flujo de percepción se puede implementar fácilmente con Mediapipe, como se representa en el gráfico 2-2 (LUGARESI et al. 2019).

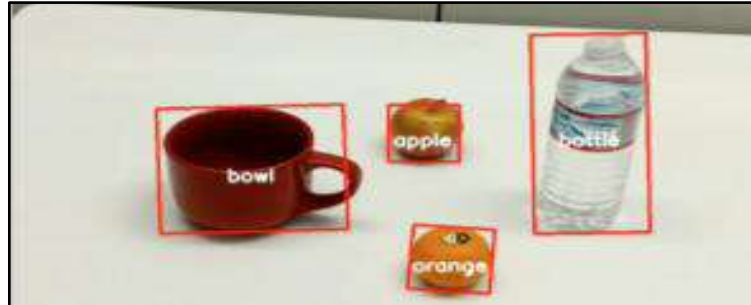


Ilustración 21-2: Detección de objetos utilizando MediaPipe

Fuente: (LUGARESI et al. 2019)

2.7.2.2. Detección puntos relevantes en rostros y segmentación

Este proceso se lo conoce como Face landmark detection and segmentation según (LUGARESI et al. 2019), como se muestra en la ilustración 22-2, Mediapipe permite detectar los puntos de referencia de rostros junto con la segmentación vertical de los mismos para conocer la variación de los puntos según el tamaño de entrada de la cámara. Para reducir la carga computacional necesaria, la estrategia radica en aplicar las tareas en dos subconjuntos disjuntos por cada frame. Esto se puede hacer utilizando un node demultiplexor que divide los paquetes en el flujo de entrada en subconjuntos entrelazados, con cada conjunto yendo a un flujo de salida separado que deriva los puntos de referencia detectados y las máscaras de segmentación en todos los frames interpolados, este proceso se ilustra en la ilustración 23-2 (LUGARESI et al. 2019).

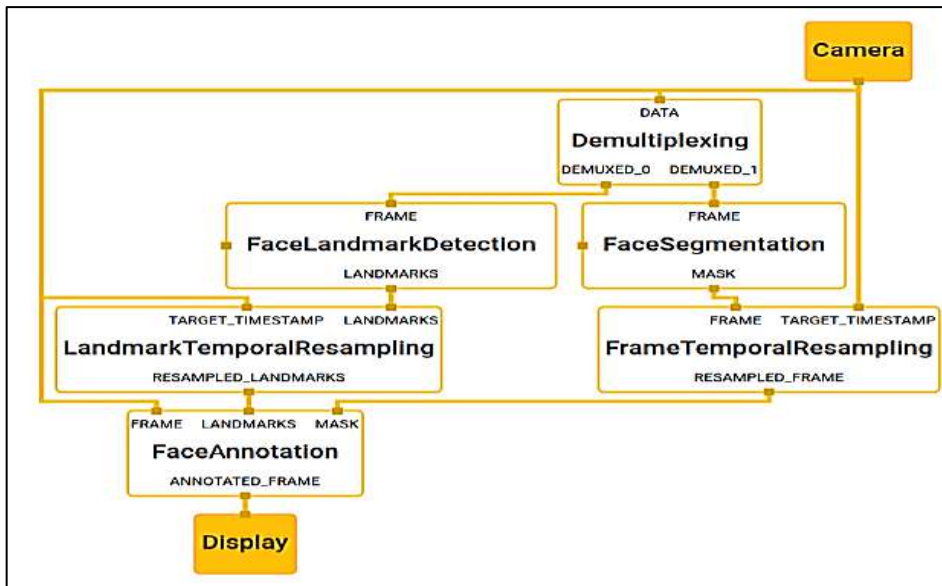


Ilustración 22-2: Detección de puntos relevantes en rostro, diagrama de proceso

Fuente: (LUGARESI et al. 2019)



Ilustración 23-2: Segmentación puntos del rostro utilizando MediaPipe.

Fuente: (LUGARESI et al. 2019)

El proceso no infiere mucho en el procesamiento de puntos relevantes (landmark) del cuerpo humano, en la ilustración 24-2, se esquematiza el diagrama de proceso de esta tarea.

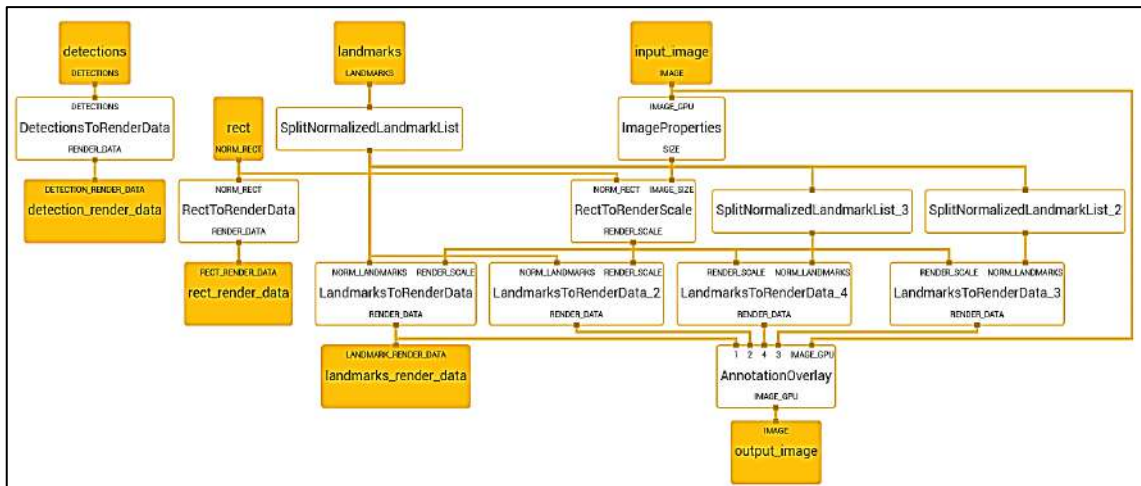


Ilustración 24-2: Diagrama de proceso para la detección de puntos relevantes cuerpo humano

Fuente: (MEDIAPIPE, 2020)

2.8. Sistemas embebidos

El término sistema embebido cubre una amplia gama de diseño de microprocesadores, y no solo engloba esta disciplina, el microprocesador debe poseer la capacidad de operar junto con un software que no sea Windows como sistema operativo o un procesador de texto. Puede ser un diseño de multiprocesador sofisticado que utilice varios procesadores integrados en una sola placa para ejecutar instrucciones mediante un SO (HEATH 2002).

En contexto, el diseño de un sistema embebido o integrado es la comprensión de la interacción que los diversos componentes dentro del sistema tienen entre sí. Es importante comprender cómo funciona el *hardware* y las restricciones que él puede tener con el uso de un determinado periférico, si existe alguna alteración con el funcionamiento de los integrantes del sistema embebido. Es fundamental saber cómo desarrollar el *software* para dichos sistemas y el efecto que los diferentes diseños de hardware pueden tener en el *software* y viceversa, de esta manera medir el rendimiento y la aplicación final para el objetivo que fue creada la implementación sea exitosa (HEATH 2002).

Un sistema embebido está basado en un microprocesador que está diseñado para controlar una función o un rango de funciones y se debe considerar que no está basado para la utilización de un usuario final como lo hace el computador tradicional (HEATH 2002).

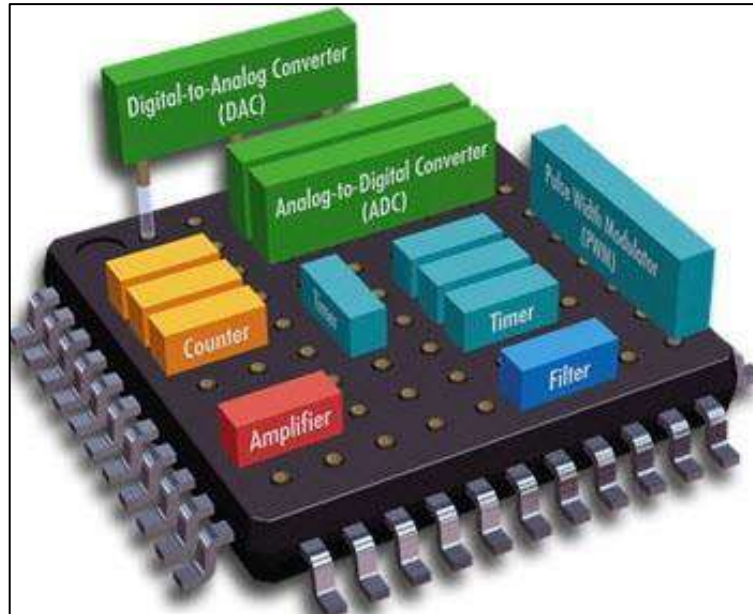


Ilustración 25-2: Componentes de un sistema embebido

Fuente: (WATELECTRONICS, 2020)

2.9. Componentes principales de un sistema embebido

2.9.1. Procesador

Un procesador integrado es un tipo de microprocesador diseñado en un sistema para controlar funciones eléctricas y mecánicas. Los procesadores integrados suelen ser de diseño simple, limitados en potencia computacional y capacidades de E/S, por su mínimo requisito de energía, los procesadores integrados son un chip con un CPU (HEATH 2002).

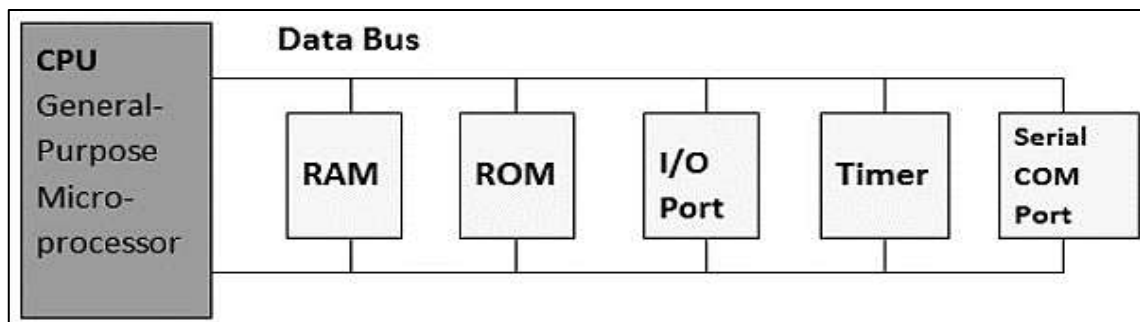


Ilustración 26-2: Diagrama de un microprocesador

Fuente: (TUTORIALSPPOINT, 2020)

Tabla 3-2: Procesador vs Microcontrolador

Procesador	Microcontrolador
Los procesadores son multitarea por naturaleza.	Orientado a una sola tarea.
RAM, ROM, puertos de E/S y temporizadores, pueden ser agregados externamente como periféricos.	RAM, ROM, puertos de E/S y temporizadores, No pueden ser agregados externamente.
Mas costosos	Baratos, y livianos
Los dispositivos externos requieren más espacio y su consumo de energía es mayor	Un sistema basado en microcontrolador consume menos energía y ocupa menos espacio

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.9.2. Memoria

Se utiliza para almacenar información. En el caso de los sistemas embebidos esta información corresponde a los diferentes algoritmos de un programa o varios, el registro de eventos, declaración de variables, etc. Existen muchos tipos de tecnologías que se utilizan al crear celdas de memoria y muchas formas de categorizarlas, pero podemos distinguir dos categorías principales basadas en la relación de memoria a la eliminación de la fuente de alimentación (Richter 2014).

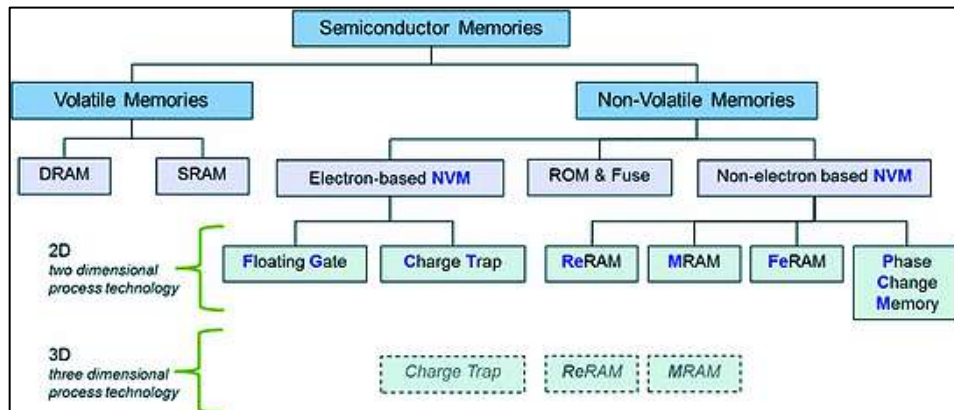


Ilustración 27-2: Tipo de memorias en sistemas embebidos

Fuente: (RICHTER 2014)

2.9.3. Periféricos

Un sistema integrado tiene que comunicarse con el mundo exterior y esto se hace mediante periféricos. Los periféricos de entrada suelen estar asociados con sensores que miden el entorno externo y, por tanto, controlan de forma eficaz las operaciones de salida que realiza el sistema integrado. De esta manera, un sistema integrado se puede modelar pipe line de tres etapas, donde los datos y la información ingresan a la primera etapa, en la segunda se procesa y la tercera generan datos para tomar decisiones (HEATH 2002).

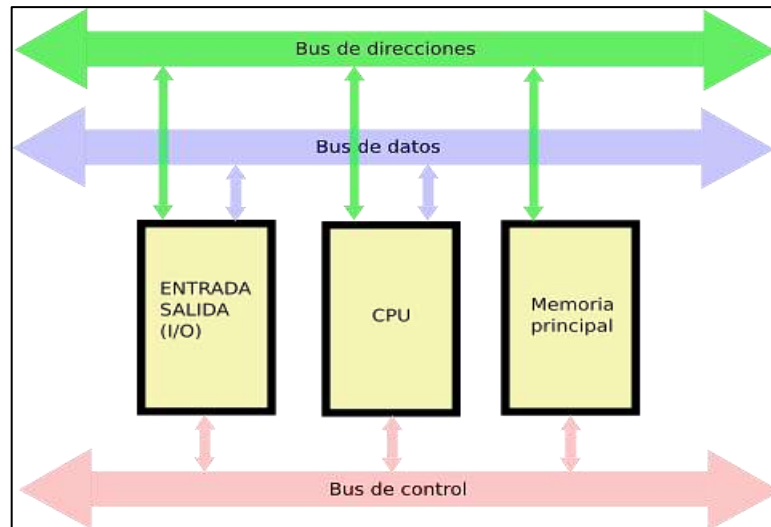


Ilustración 28-2: Diagrama general de periféricos del sistema embebido

Realizado por: Fajardo C., Yanchapanta W., 2022.

2.10. Tarjetas controladoras para Visión Artificial

Un sistema de visión artificial consta de una o más cámaras para la adquisición de imágenes combinadas con un controlador que ejecuta el software de adquisición y análisis. Estos sistemas pueden tomar algunas formas, pero por lo general son cámaras inteligentes en las que la cámara y el controlador están integrados en una sola unidad, cámaras conectadas a una computadora de escritorio o industrial por un puerto Ethernet o USB, o incluso conectadas a dispositivos de control industrial como controladores lógicos programables (PLC), donde un controlador administra la adquisición de imágenes, así como dispositivos de medición y E/S de segmento (NATIONAL INSTRUMENTS, 2019).

Se pueden utilizar cada uno de estos sistemas para resolver aplicaciones similares, pero todos tienen características que los hacen adecuados para cada aplicación a ejecutar. Es importante considerar las necesidades y características como capacidad de procesamiento, tipo de cámaras compatibles, capacidades de entrada y salida de periféricos (E/S), tipo de conexiones disponibles, integración de sistema (la compatibilidad entre hardware y software), etc. Compact Vision Systems, Compact RIO y controladores industriales de NI (National Instruments) son algunos de los productos que poseen alta potencia de procesamiento y por ende un rendimiento óptimo para aplicaciones de visión artificial con más de una cámara. También existen las tarjetas controladoras como lo son Raspberry, Arduino y otros microprocesadores, pero estos últimos están limitados a su poder de procesamiento, por ende, sus aplicaciones están limitadas a tareas básicas dentro de todo lo que ofrece la visión artificial (NATIONAL INSTRUMENTS, 2019).

2.10.1. Raspberry pi

Es un computador basado en una arquitectura embebida ilustración 29-2, fabricada por la fundación Raspberry pi org en reino unido,(Pi 2021) tiene como objetivo educar a personas con una metodología de informática de bajo costo y de alto nivel. Raspberry pi ejecuta como sistema operativo a Linux, provee un conjunto de pines GPIO (entrada – salida) para un propósito de electrónica general, lo que permite controlar componentes electrónicos para la computación física y explorar el Internet de las cosas (IoT) (PI 2021).

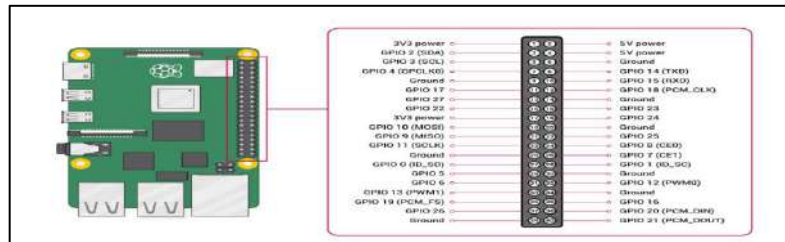


Ilustración 29-2: Raspberry pi.

Fuente: (APRENDIENDOARDUINO, 2020)

Raspberry pi opera en un ecosistema de código abierto, como antes se lo mencionó, ejecuta un kernel de Linux, pero su sistema conocido actualmente es Pi OS que se lo puede descargar en la página oficial de Raspberry Pi OS, esto permite a investigadores, alumnos, aficionados, etc, puedan crear un *software* propio como código abierto.

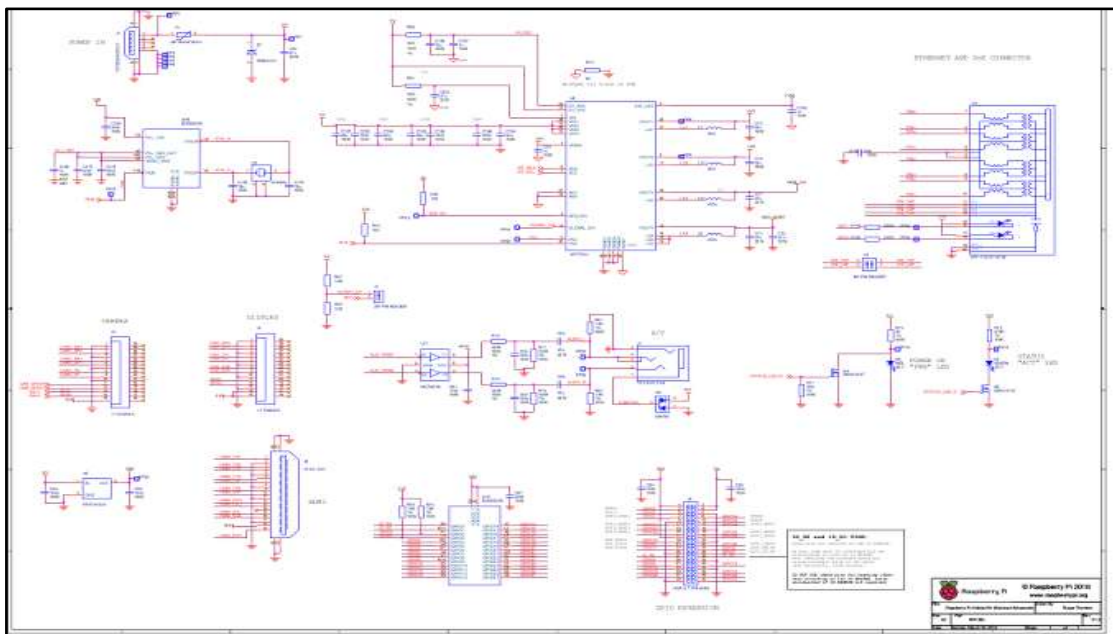


Ilustración 30-2: Esquemático de Raspberry Pi

Fuente: (OPENSOURCE, 2020)

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se detalla la metodología utilizada para el desarrollo del prototipo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial, por lo tanto, se adoptó la siguiente estrategia de trabajo mostrada en la ilustración 1-3.

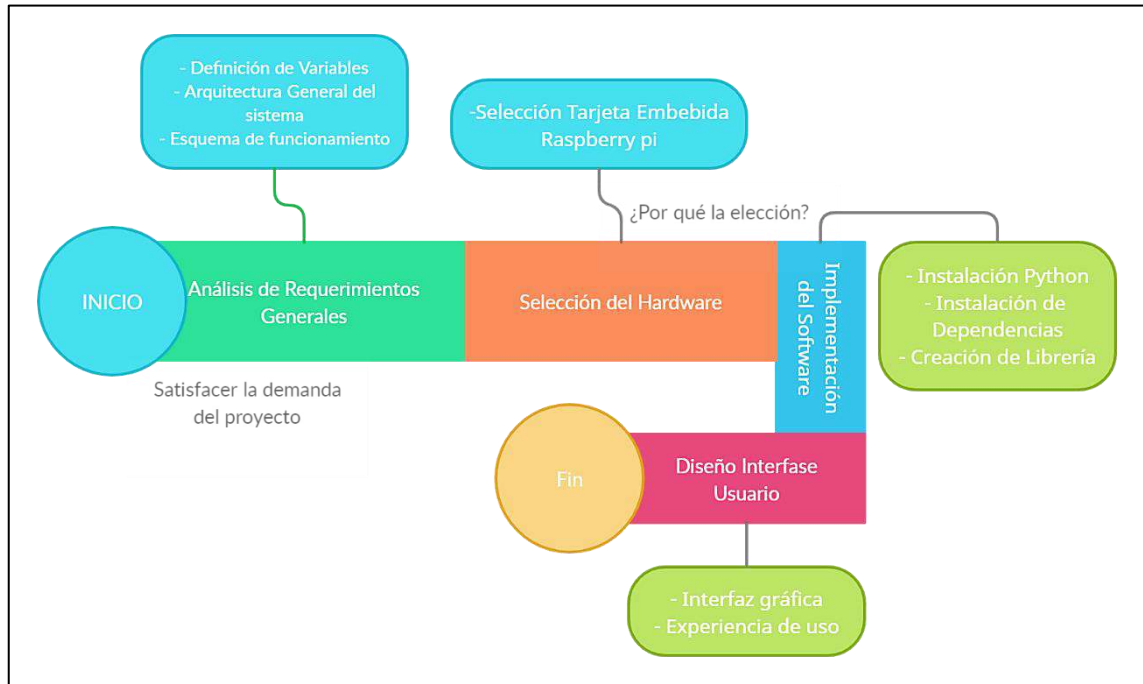


Ilustración 1-3: Metodología para el desarrollo del proyecto

Realizado por: Fajardo C., Yanchapanta W., 2022.

El desarrollo planteado está orientado al cumplimiento de metas y objetivos que determinan objetivamente el avance del proyecto para posteriormente plantear nuevos objetivos de cada etapa.

3.1. Análisis de requerimientos generales

En la presente investigación se pretende diseñar y construir un sistema capaz de identificar puntos relevantes del cuerpo humano, en especial de un niño de 4 a 5 años al cual va orientado las actividades físicas recomendadas por el profesional de la salud consultado y según la bibliografía proporcionada por el mismo, los requerimientos que se busca reflejar en dicho prototipo del sistema son:

- Capacidad de identificar la anatomía física, forma y contornos del cuerpo humano.
- Capacidad de identificar puntos relevantes en las articulaciones principales del cuerpo humano.
- Correcto funcionamiento a un promedio de 4 a 10 FPS.

- Versatilidad para ser ejecutado en un Raspberry Pi.
- Interfaz humano-máquina intuitivo, flexible, sencillo y fácil de interactuar.

En la ilustración 2-3, se ilustra el esquema del funcionamiento general según el grupo objetivo de la investigación. El dispositivo adquiere las imágenes mediante una webcam normal de 24 Mpx, los procesa para implementar en paralelo los puntos relevantes del cuerpo humano y operar con ellos, para posteriormente interactuar con el movimiento y evaluar el rendimiento de las actividades físicas del infante mediante la comparación de ángulos y aperturas de las articulaciones gruesas.

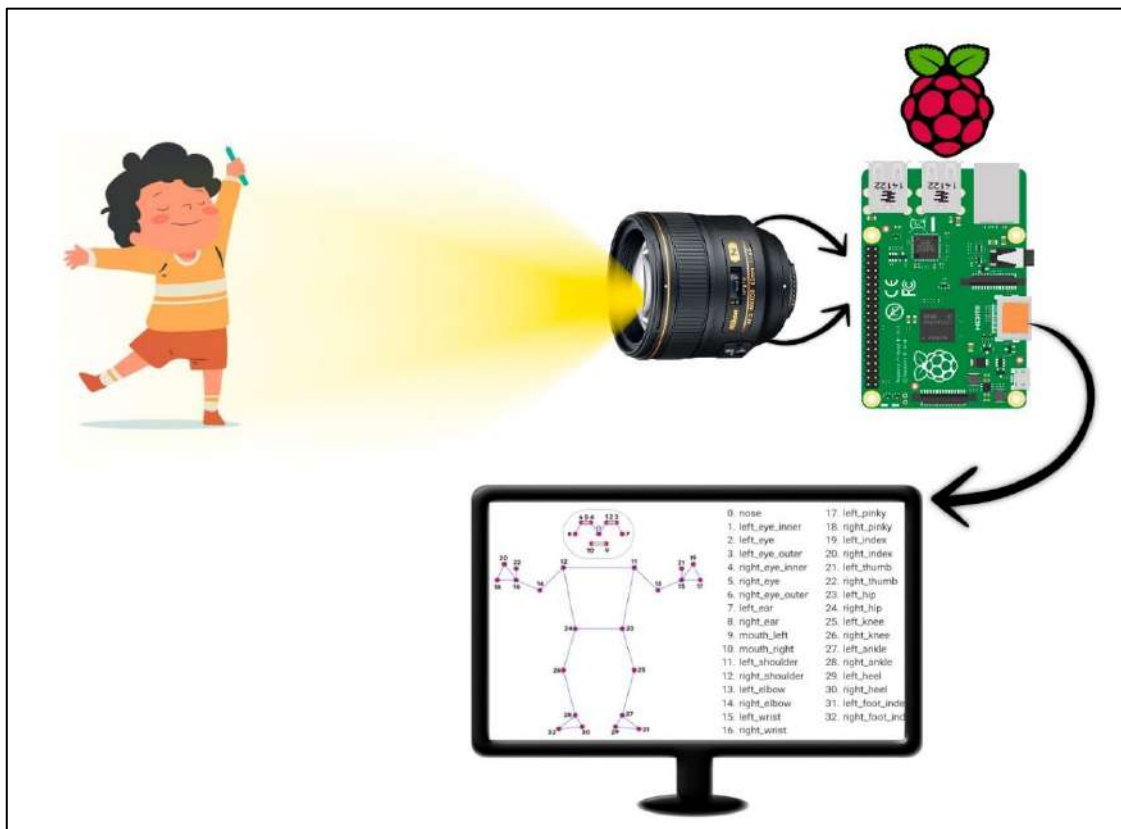


Ilustración 2-3: Arquitectura general del prototipo

Fuente: Fajardo C., Yanchapanta W., 2022

3.1.1. Definición de variables

Las acciones objetivas del sistema están limitadas según el ejercicio que cumpla el niño, como se expresó en el capítulo anterior, el infante debe cumplir con 6 ejercicios por separado, para la evaluación se tomarán estas 6 variables de manera binaria, cumple o no cumple, a continuación, en la tabla 1-3, se representan dichas variables.

Tabla 1-3: Variables del sistema

Variables de control	Variables controladas	
	Cumple	No cumple
Ejercicio 1		
Ejercicio 2		
Ejercicio 3		
Ejercicio 4		
Ejercicio 5		
Ejercicio 6		

Realizado por: Fajardo C., Yanchapanta W., 2022.

3.2. Selección del *Hardware*

Para el correcto funcionamiento del dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial, se evaluaron las siguientes consideraciones:

- Posibilidad de programar con Python.
- Cantidad de memoria RAM necesaria para trabajar con visión artificial entre 4 y 10 FPS.
- Capacidad de visualizar los resultados en tiempo real, mediante un monitor externo.
- Bajo consumo energético.
- Instalación y rápida operatividad.

Cabe recalcar, que la tendencia de operatividad de las plataformas electrónicas para el procesamiento de visión artificial requiere cierto grado de autonomía y encapsulamiento de hardware “Computer-On-Board” (HEATH, 2002).

Validando las consideraciones anteriores, se estableció un análisis de la variedad de productos existentes dentro del país, por su fácil adquisición, reposición y garantía, se establecen las siguientes opciones para trabajar:












	SoC	CPU	GPU	RAM	USB	V/A	Boot	Red	Alimentación	Tamaño	Fecha
 Model A	Broadcom BCM2835	700MHz ARM1176JZF-S	VideoCore IV	256MB	1	RCA Jack HDMI	SD	No	300mA 1,5w / 5v MicroUSB GPIO	85,6 x 53,98 mm	04/12
 Model A+	Broadcom BCM2835	700MHz ARM1176JZF-S	VideoCore IV	256MB	1	Jack HDMI	uSD	No	400mA 2w / 5v MicroUSB GPIO	65 x 56 mm	11/14
 3 Model A+	Broadcom BCM2837B0	1,4GHz QUAD ARM Cortex-A53	VideoCore IV	512MB	1	Jack HDMI	uSD	Dual-band WiFi, BT	2,5A 12,5w / 5v MicroUSB GPIO	65 x 56 mm	11/18
 Model B	Broadcom BCM2835	700MHz ARM1176JZF-S	VideoCore IV	512MB	2	RCA Jack HDMI	SD	ETH 10/100	700mA 3,5w / 5v MicroUSB GPIO	85,6 x 53,98 mm	04/12
 Model B+	Broadcom BCM2835	700MHz ARM1176JZF-S	VideoCore IV	512MB	4	Jack HDMI	uSD	ETH 10/100	500mA 2,5w / 5v MicroUSB GPIO	85 x 56 mm	07/14
 2 Model B	Broadcom BCM2836	900MHz QUAD ARM Cortex-A7	VideoCore IV	1GB	4	Jack HDMI	uSD	ETH 10/100	800mA 4w / 5v MicroUSB GPIO	85 x 56 mm	02/15
 3 Model B	Broadcom BCM2837	1,2GHz QUAD ARM Cortex-A53	VideoCore IV	1GB	4	Jack HDMI	uSD	ETH 10/100 WiFi, BT	2,5A 12,5w / 5v MicroUSB GPIO	85 x 56 mm	02/16
 3 Model B+	Broadcom BCM2837B0	1,4GHz QUAD ARM Cortex-A53	VideoCore IV	1GB	4	Jack HDMI	uSD	ETH 10/100/300 (USB) Dual-band WiFi BT	2,5A 12,5w / 5v MicroUSB GPIO PoE (HAT)	85 x 56 mm	03/18
 4 Model B	Broadcom BCM2711	1,5GHz QUAD ARM Cortex-A72	VideoCore IV	1, 2 o 4GB	2 (2.0) 2 (3.0)	Jack 2 micro HDMI	uSD	ETH 1000 Dual-band WiFi BT	2,5A 12,5w / 5v USB-C GPIO PoE (HAT)	85 x 56 mm	06/19
 Zero	Broadcom BCM2835	1GHz ARM1176JZF-S	VideoCore IV	512MB	1 Micro	Mini HDMI	uSD	No	160mA 0,8w / 5v MicroUSB GPIO	65 x 30 mm	11/15
 Zero W	Broadcom BCM2835	1GHz ARM1176JZF-S	VideoCore IV	512MB	1 Micro	Mini HDMI	uSD	Wifi, BT	160mA 0,8w / 5v MicroUSB GPIO	65 x 30 mm	02/17

Ilustración 3-3: Comparativa Raspberry pi varias versiones

Fuente: (COMOHACER, 2020)

Según la ilustración 3-3, el apartado más importante dentro de la comparativa anterior es la frecuencia de trabajo del CPU, y la memoria RAM, para no tener mayores complicaciones al momento de implementar el proyecto por precio y prestaciones se optó por trabajar con la Raspberry Pi 4 Model B de 8GB de RAM con una frecuencia del CPU a 1.5Ghz, que actualmente es la mejor tarjeta embebida del mercado para propósitos de visión artificial didáctica.

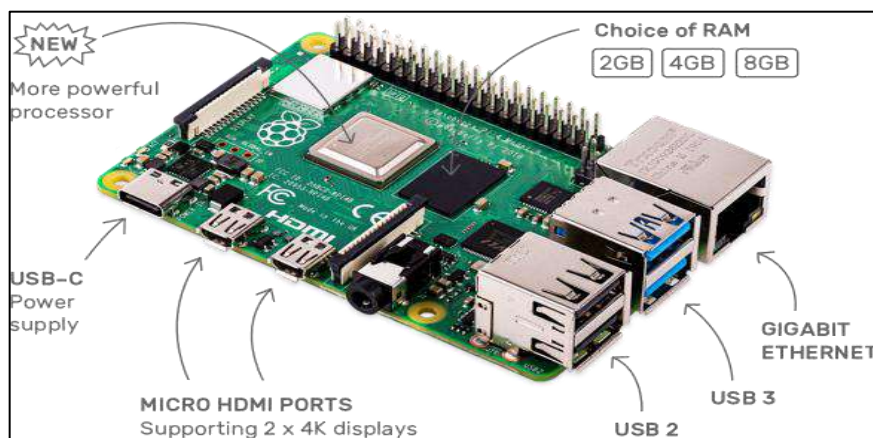


Ilustración 4-3: Raspberry pi 4 model b

Fuente: (RASPERRY, 2020)

Las primeras pruebas van a ejecutarse dentro de un computador portátil, el cual posee teclado, mouse y cámara web integradas. Hay que considerar la portabilidad del dispositivo y adquirir el *hardware* necesario para la siguiente etapa de las ejecuciones de las pruebas correspondientes.

Para esto es necesario adquirir una cámara web, un mouse y un teclado que posean una entrada USB misma que es compatible con una de las múltiples entradas USB que posee la Raspberry Pi 4 Model b anteriormente seleccionado.



Ilustración 5-3: Cámara web COOAU1080 de 2Mpx

Fuente: (TECNIT, 2020)



Ilustración 6-3: Mouse y teclado USB marca UNNO

Fuente: (UNNOTEKNO, 2020)

La ilustración 6-3 muestra el mouse y el teclado necesarios para interactuar con la interface implementada en la raspberry y conectada a una pantalla con entrada HDMI.

CAPÍTULO IV

4. PROPUESTA Y DISEÑO DE PROTOTIPO

En este capítulo se detalla el proceso para el diseño y el desarrollo del prototipo.

4.1. Preparación del Raspberry Pi

Para la preparación del Raspberry Pi 4 es necesario instalar el sistema operativo que administrará la arquitectura de la tarjeta electrónica, para ello es necesario utilizar un gestor implementado por los mismos desarrolladores de dicha tarjeta, *Raspberry pi imager* en la versión 1.6, es la forma más rápida y fácil de instalar Raspberry Pi OS y otros sistemas directamente en la MicroSD. Esta se indica en la ilustración 1-4.

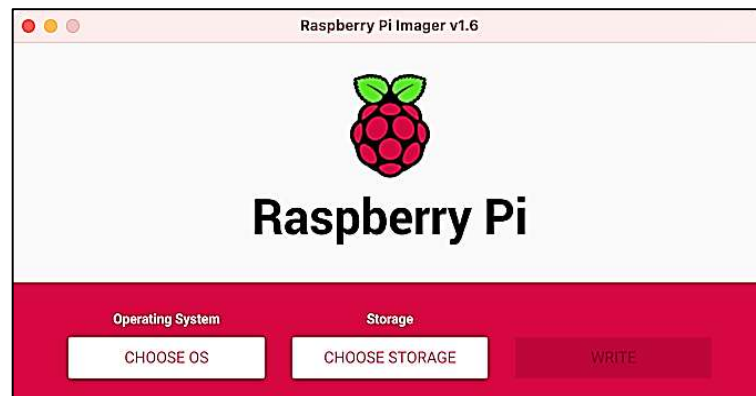


Ilustración 1-4: Raspberry Pi Imager

Realizado por: Fajardo C., Yanchapanta W., 2022.

Antes de empezar, como el raspberry pi 4 model B, utiliza drivers de video y se los configura desde su terminal es necesario activar el SSH y el Wifi previamente, para ello en el Raspberry Pi Imager presionamos la combinación de teclas **Ctrl + Mayus + X**, al hacer esto aparecerá una nueva ventana ilustración 2-4, se realiza esto con el objetivo de utilizar el raspberry pi remotamente desde el computador para su configuración.

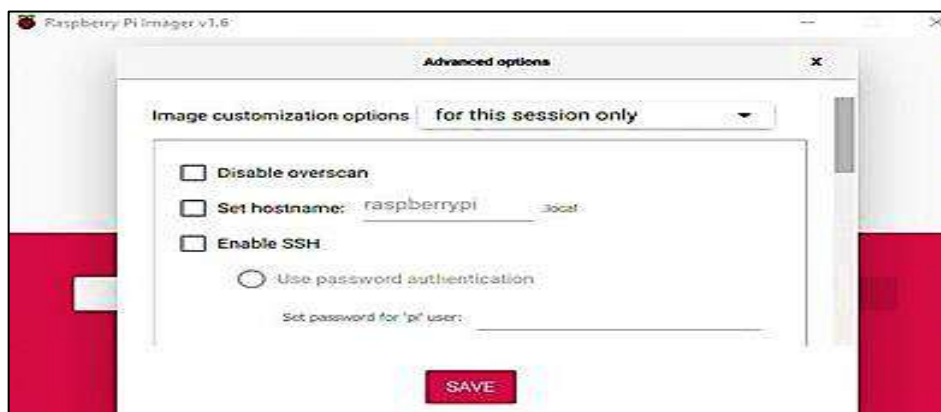


Ilustración 2-4: Configuración SSH y Wifi

Realizado por: Fajardo C., Yanchapanta W., 2022.

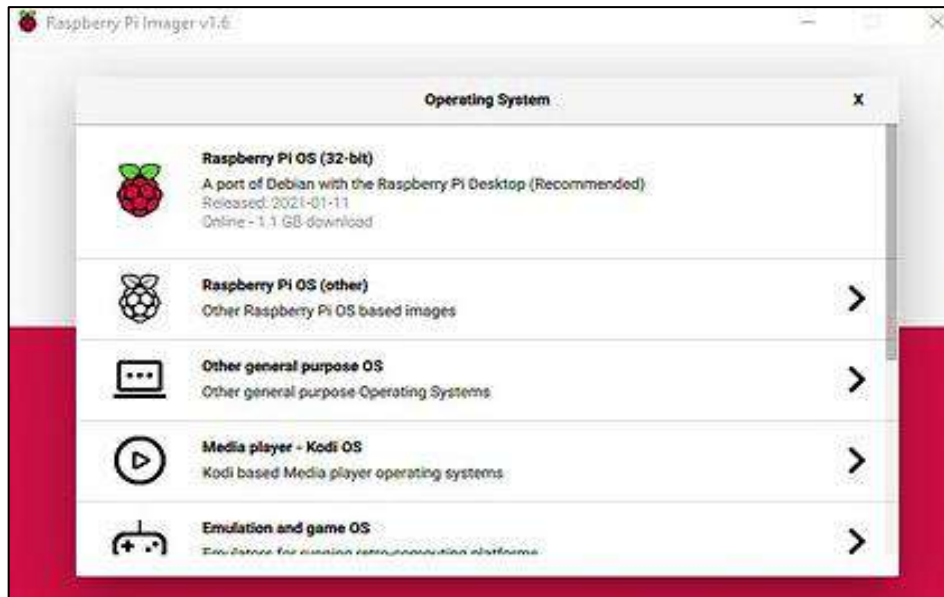


Ilustración 3-4: Raspberry sistema operativo

Realizado por: Fajardo C., Yanchapanta W., 2022.

Una vez instalado todo con éxito, insertamos la microSD al Raspberry Pi 4, conectamos a la fuente de alimentación 5V 3A (para el correcto funcionamiento se recomienda utilizar una fuente superior a los 2A). Esperamos 10 minutos hasta que termine la configuración.

Para conectarse remotamente desde el computador, es necesario instalar VNC Viewer como se puede ver en la ilustración 4-4, el mismo que permite hacer una configuración remota sin la necesidad de que el Raspberry Pi esté conectado a un monitor.

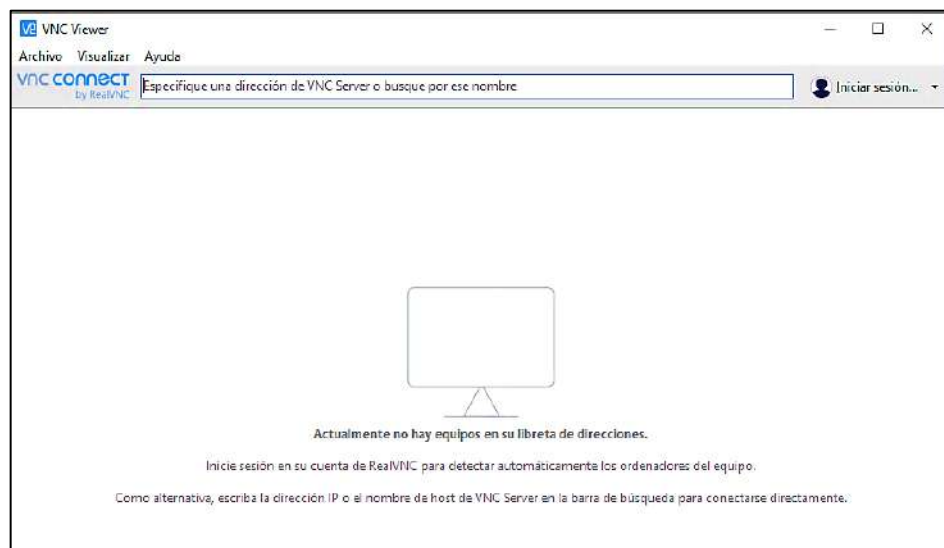


Ilustración 4-4: VNC Viewer

Realizado por: Fajardo C., Yanchapanta W., 2022.

Los requisitos para establecer una conexión es la habilitación del SSH explicado anteriormente desde el imager, y conocer la dirección IP de la tarjeta, para ello utilizamos Advance IP Scanner

ilustración 5-4, esta herramienta es un scanner de todos los dispositivos conectados a nuestra red local, exponiendo la dirección IP de cada dispositivo.

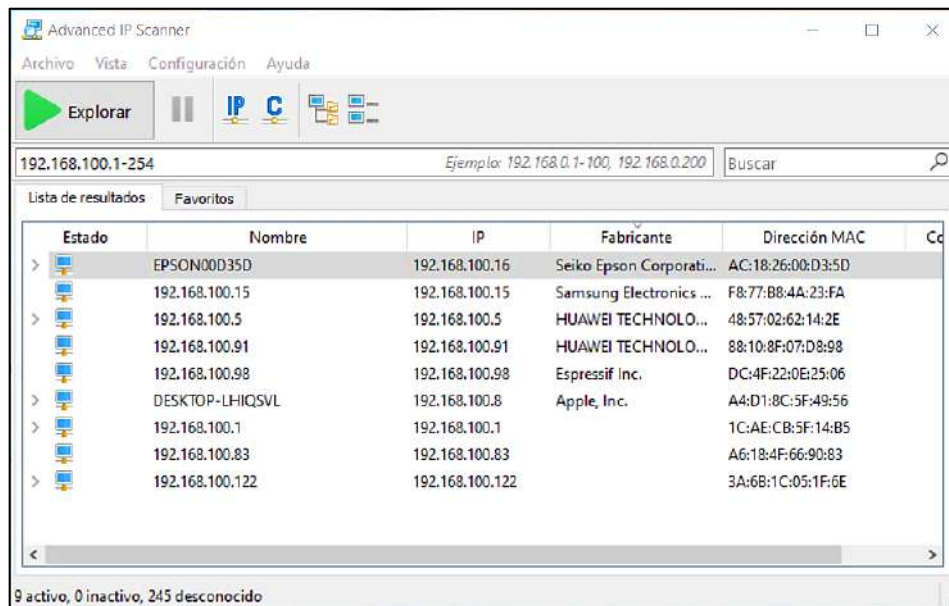


Ilustración 5-4: Advance IP Scanner

Realizado por: Fajardo C., Yanchapanta W., 2022.

4.2. Preparación entorno de programación

Una vez culminada la preparación del Raspberry pi 4, preparamos la aplicación del proyecto en un computador Windows, con el objetivo de una mayor flexibilidad en la programación, pruebas y consultas. Para esta tarea se decidió utilizar el entorno de desarrollo integrado de Python denominado Pycharm ilustración 6-4.

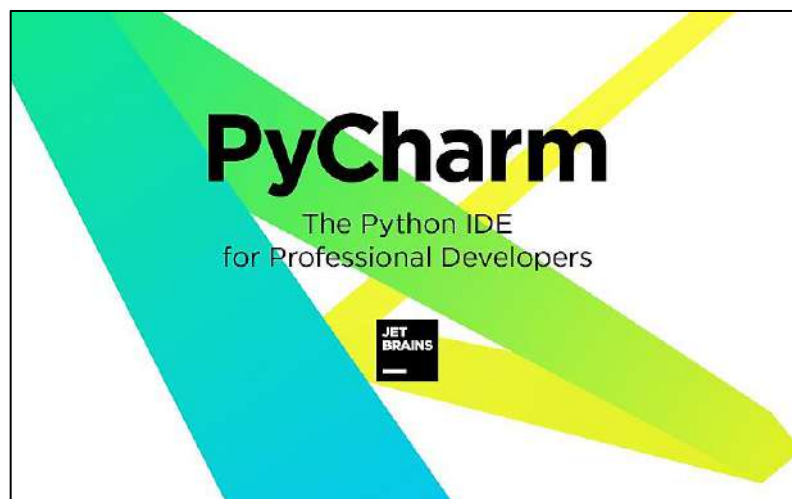


Ilustración 6-4: PyCharm

Fuente: (JETBRAINS, 2020)

Una vez instalado Python versión 3.8.x desde la plataforma oficial Python, abrimos el editor de texto, en el apartado de ajustes instalamos todas las dependencias necesarias para desarrollar el proyecto como se muestra en la ilustración 7-4.

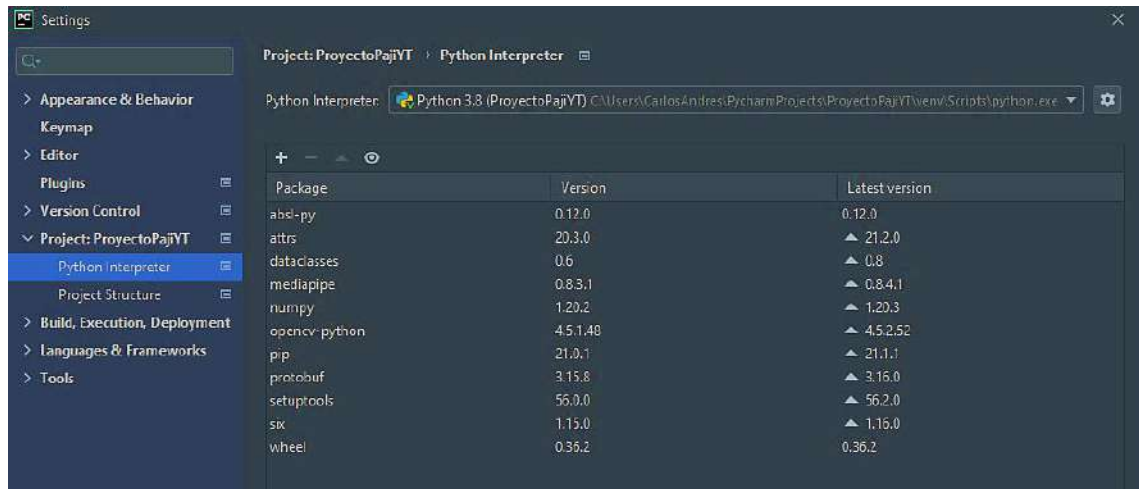


Ilustración 7-4: Instalación de paquetes

Realizado por: Fajardo C., Yanchapanta W., 2022.

- **NumPy:** Es un paquete fundamental en la operación computacional científica, eficaz para trabajar en el dominio del álgebra lineal, transformada de Fourier y en nuestro caso en el tratamiento operacional con matrices (NUMPY, 2021).
- **OpenCv-Python:** Es un paquete diseñado para resolver problemas de visión por computadora, aprendizaje automático y procesamiento digital de imágenes. Juntamente con NumPy se pueden realizar un tratamiento operacional de imágenes de una manera fácil y flexible en la solución de problemas de visión artificial (OPENCV-PYTHON, 2021).
- **Mediapipe:** Biblioteca basada en la construcción de canales de aprendizaje automático en tiempo real, dentro de las soluciones se encuentra detección de objetos, human landmarks, motion tracking, segmentación entre otras aplicaciones. En esta biblioteca nos basamos para la esquematización del cuerpo humano, fue ideal el uso por el bajo requerimiento de recursos que utiliza la CPU para trabajar entre 4 y 10 FPS, de esta manera no saturamos la unidad de procesamiento del Raspberry pi y el interfaz de interacción humano-máquina es fluida con una latencia muy baja. Ideal para trabajar en tiempo real y en el Raspberry Pi.

4.3. Implementación programa configuración

La elaboración del algoritmo para la configuración inicial permitirá reconocer todos los puntos relevantes del cuerpo humano, para ello es necesario la utilización de opencv para activar la cámara principal del computador, se activa la biblioteca de Mediapipe, tomando en consideración el control de los FPS, dicha actividad se encuentra reflejado en el diagrama de flujo en el gráfico 1-4.

Previo a la programación de los 6 ejercicios planteados para el prototipo, se requiere establecer las funciones principales las cuales serán la base principal para el correcto funcionamiento del programa.

Como primera instancia se define la clase PoseDetector misma que engloba las funciones generales del algoritmo las cuales son: findPose misma que detecta todos los puntos del cuerpo, findPosition función que dibuja todos los puntos relevantes del cuerpo visto en tiempo real, y por último findAngle obtiene los puntos relevantes del cuerpo visualizado a trabajar dentro de cada ejercicio para poder posteriormente identificarlos.

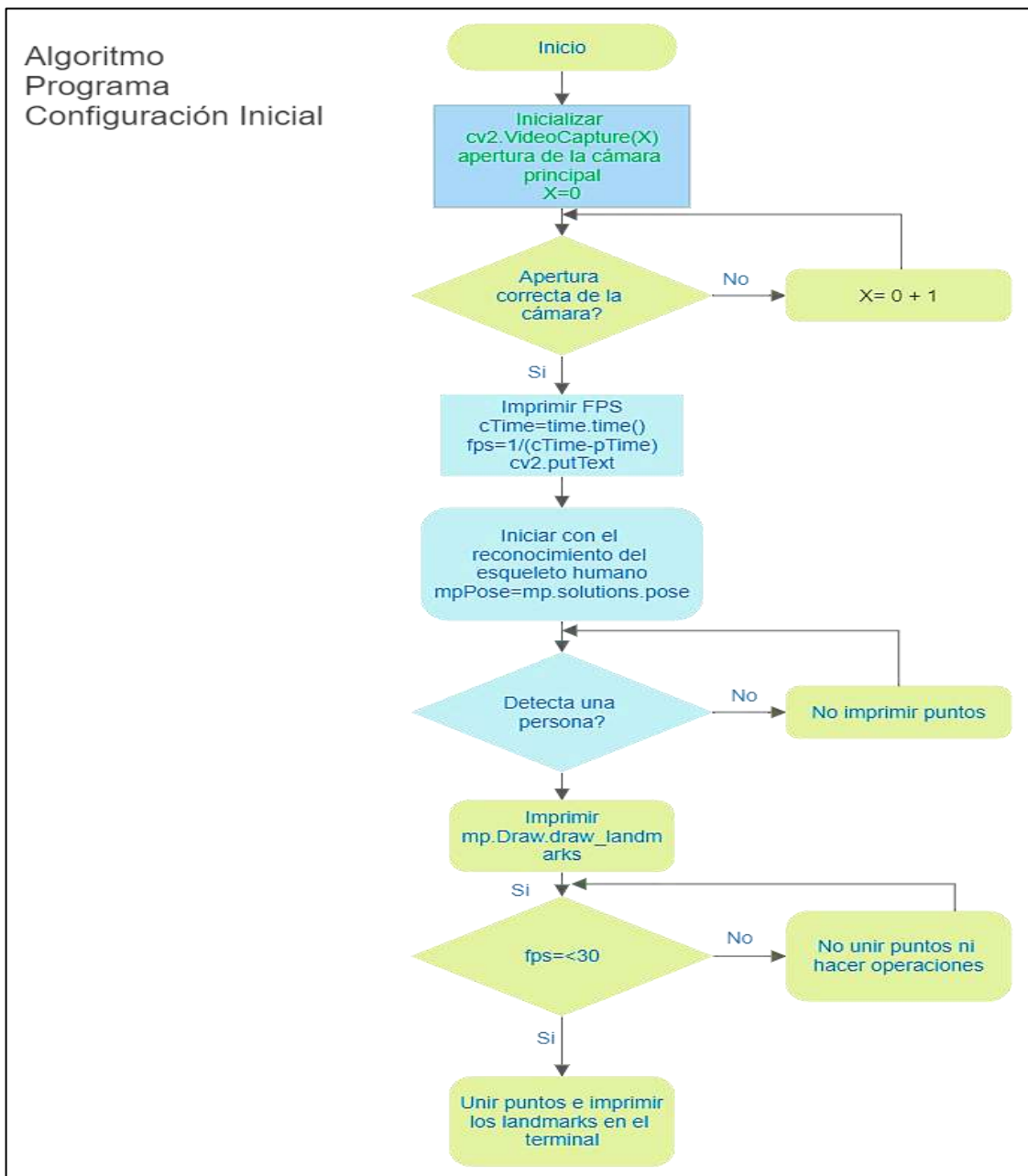


Gráfico 1-4: Diagrama de flujo del programa en su configuración inicial

Realizado por: Fajardo C., Yanchapanta W., 2022.

Dentro de la configuración inicial se considera implementar tres funciones para la posterior utilización en ejercicios específicos, con esto nos centramos en el modelamiento del algoritmo de cada actividad sin la necesidad de establecer parámetros iniciales nuevamente, esto ayuda a la flexibilidad del código para un mejor entendimiento y configuraciones a futuro.

```
def findPose(self, img, draw=False): #Detecta todo el cuerpo
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.pose.process(imgRGB)
    if self.results.pose_landmarks:
        if draw:
            self.mpDraw.draw_landmarks(img, self.results.pose_landmarks, self.mpPose.POSE_CONNECTIONS,
                                        self.mpDraw.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=4),
                                        self.mpDraw.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
            )
```

Ilustración 8-4: Función findPose (en español encontrar pose)

Realizado por: Fajardo C., Yanchapanta W., 2022.

La función *findPose* de la ilustración 8-4, resalta los puntos más relevantes de del cuerpo humano, esto gracias a la configuración *landmarks* de Mediapipe. Dentro de la instancia *pose.process* personalizamos la impresión de los puntos accediendo a los atributos de la clase *mpDraw.draw_landmarks* de la siguiente manera:

El método *results.pose_landmarks*, se encarga de imprimir los puntos de las articulaciones principales del cuerpo humano y *Pose_connections* dibuja la unión entre puntos, para poder identificarlos se ha cambiado de color cada elemento antes mencionado para poder reconocerlo.

En la ilustración 9-4, se ilustra la función *findPosition*, esto con la finalidad de imprimir en el terminal del editor de texto de manera ordenada una matriz con el id, identificador de la articulación juntamente con la posición en el plano cartesiano punto actual en el eje X e Y, de esta manera los puntos relevantes de las articulaciones establecidos por la librería Mediapipe ilustración 10-4, se muestran ordenados para su posterior manipulación ya que se guardan en las variables [id, cx, cy].

```
def findPosition(self, img, draw=True): #Dibuja en los puntos relevantes
    self.lmList = []
    if self.results.pose_landmarks:
        for id, lm in enumerate(self.results.pose_landmarks.landmark):
            h, w, c = img.shape
            #print(id, lm) #Imprime, id del punto relevante y su posición
            cx, cy = int(lm.x * w), int(lm.y * h) #Comprime en un arreglo, ID mas el punto en X
            self.lmList.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)
    return self.lmList
```

Ilustración 9-4: Función findPosition que identifica puntos en articulaciones y ordena en una matriz

Realizado por: Fajardo C., Yanchapanta W., 2022.

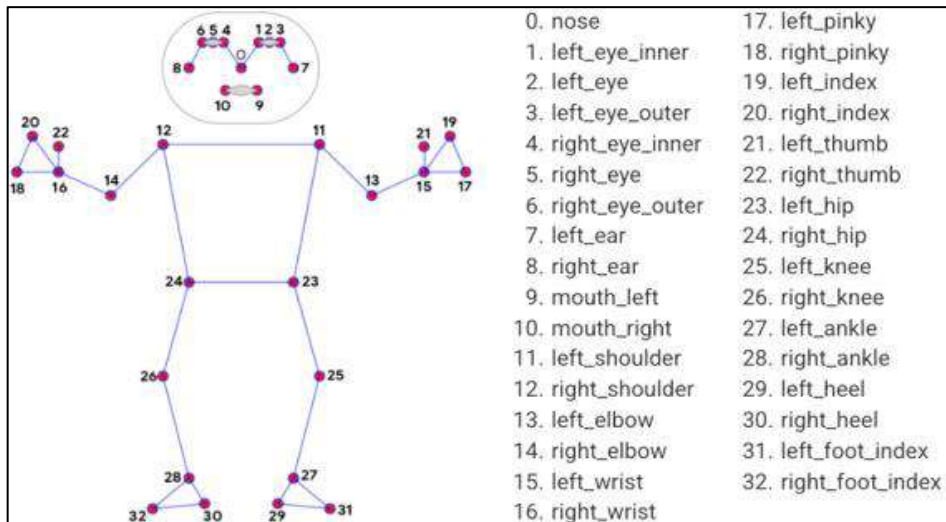


Ilustración 10-4: Pose landmarks

Fuente: (GITHUB, 2020)

Realizando las pruebas para la activación y depuración del primer algoritmo, se visualiza los resultados en la ilustración 11-4.

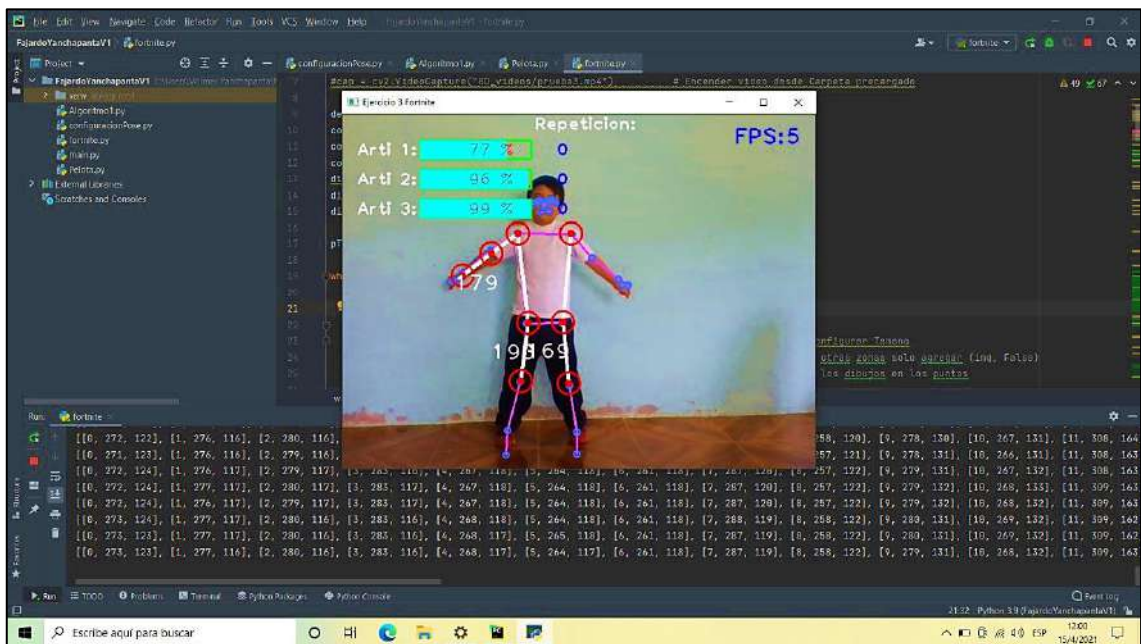


Ilustración 11-4: Resultado de la configuración inicial

Realizado por: Fajardo C., Yanchapanta W., 2022.

En la ilustración 12-4, se visualizan los datos ordenados de las matrices con el identificador de cada articulación junto con la posición en el eje X y, esto permite ver la dinámica de cada punto con respecto al tiempo, y operar la diferencia de un punto con respecto al otro.

```

Run: Algoritmo
[[0, 244, 219], [1, 269, 286], [2, 278, 269], [3, 287, 212], [4, 250, 197], [5, 244, 194], [6, 238, 190], [7, 309, 230], [8, 239, 193], [9, 254, 246]
[[0, 216, 225], [1, 240, 212], [2, 248, 217], [3, 257, 222], [4, 222, 199], [5, 218, 194], [6, 213, 189], [7, 286, 256], [8, 225, 189], [9, 230, 251]
[[0, 243, 234], [1, 265, 218], [2, 273, 221], [3, 281, 224], [4, 246, 218], [5, 241, 206], [6, 236, 202], [7, 302, 232], [8, 237, 201], [9, 254, 255]
[[0, 266, 238], [1, 294, 222], [2, 332, 226], [3, 315, 229], [4, 252, 211], [5, 242, 207], [6, 232, 203], [7, 309, 241], [8, 235, 204], [9, 290, 264]
[[0, 246, 256], [1, 268, 234], [2, 288, 237], [3, 292, 239], [4, 239, 226], [5, 230, 222], [6, 221, 218], [7, 318, 240], [8, 219, 216], [9, 266, 283]
[[0, 249, 259], [1, 274, 238], [2, 286, 240], [3, 298, 243], [4, 244, 230], [5, 234, 225], [6, 224, 221], [7, 328, 251], [8, 222, 219], [9, 270, 285]
[[0, 287, 280], [1, 386, 254], [2, 317, 250], [3, 329, 258], [4, 276, 249], [5, 265, 245], [6, 254, 241], [7, 333, 263], [8, 237, 233], [9, 302, 303]
[[0, 292, 296], [1, 313, 270], [2, 326, 271], [3, 339, 271], [4, 280, 264], [5, 267, 261], [6, 255, 257], [7, 357, 269], [8, 237, 249], [9, 307, 317]
[[0, 282, 270], [1, 394, 246], [2, 318, 248], [3, 331, 256], [4, 278, 237], [5, 257, 232], [6, 244, 228], [7, 352, 254], [8, 226, 223], [9, 298, 298]
[[0, 269, 270], [1, 286, 268], [2, 296, 250], [3, 305, 251], [4, 256, 241], [5, 246, 238], [6, 237, 234], [7, 322, 256], [8, 227, 233], [9, 281, 286]

```

Ilustración 12-4: Impresión datos posición landmarks

Realizado por: Fajardo C., Yanchapanta W., 2022.

A continuación, se muestra dos matrices con la unificación de todos los 32 puntos relevantes y su variación en el eje X y, la variación de estos puntos se encuentra limitados al tamaño máximo de la resolución por defecto de la cámara web, que es 640 x 480 px ilustración 13-4, dentro de este límite se encuentra las variaciones de cada punto.

```

[Punto relevante o Articulación, Posición en X, Posición en Y]
[[0, 246, 213], [1, 275, 206], [2, 284, 212], [3, 293, 218], [4, 255, 189], [5, 249, 182], [6, 244, 176], [7, 321, 233], [8, 251, 172], [9, 257, 239], [10, 235, 221], [11, 342, 320], [12, 238, 206], [13, 326, 401], [14, 215, 276], [15, 311, 381], [16, 222, 272], [17, 309, 380], [18, 224, 271], [19, 317, 367], [20, 228, 262], [21, 314, 362], [22, 235, 266], [23, 236, 387], [24, 187, 336], [25, 224, 385], [26, 171, 351], [27, 204, 391], [28, 151, 355], [29, 199, 387], [30, 152, 357], [31, 183, 411], [32, 135, 375]]
[[0, 244, 219], [1, 269, 206], [2, 278, 209], [3, 287, 212], [4, 250, 197], [5, 244, 194], [6, 238, 190], [7, 309, 230], [8, 239, 193], [9, 254, 246], [10, 232, 237], [11, 327, 287], [12, 205, 251], [13, 337, 369], [14, 195, 344], [15, 321, 397], [16, 255, 400], [17, 317, 406], [18, 266, 422], [19, 311, 400], [20, 277, 399], [21, 305, 394], [22, 279, 395], [23, 236, 366], [24, 178, 353], [25, 236, 371], [26, 181, 382], [27, 237, 397], [28, 169, 403], [29, 234, 394], [30, 171, 402], [31, 232, 427], [32, 167, 436]]

```

Ilustración 13-4: Resolución de la cámara web

Realizado por: Fajardo C., Yanchapanta W., 2022.



Ilustración 14-4: Tamaño por defecto de la ventana emitida por la cámara web

Realizado por: Fajardo C., Yanchapanta W., 2022.

4.3.1. Pruebas de la configuración inicial en adulto

Para la confirmación del funcionamiento, al sistema se le pone a prueba en un escenario diferente, esta vez con una persona adulta el cual comprobará la identificación de los puntos más relevantes del cuerpo humano con la finalidad de medir la escalabilidad del proyecto. Esta actividad se ve reflejada en las ilustraciones 15-4, 16-4 y 17-4.

Para no comprometer la extensión del presente capítulo, la programación completa se encuentra dentro del anexo E al final del documento. Allí se detallan todas líneas de código necesarias para el correcto funcionamiento del algoritmo, incluyendo los segmentos de cada uno de los seis ejercicios planteados para el prototipo.



Ilustración 15-4: Prueba 1 en un adulto

Realizado por: Fajardo C., Yanchapanta W., 2022.

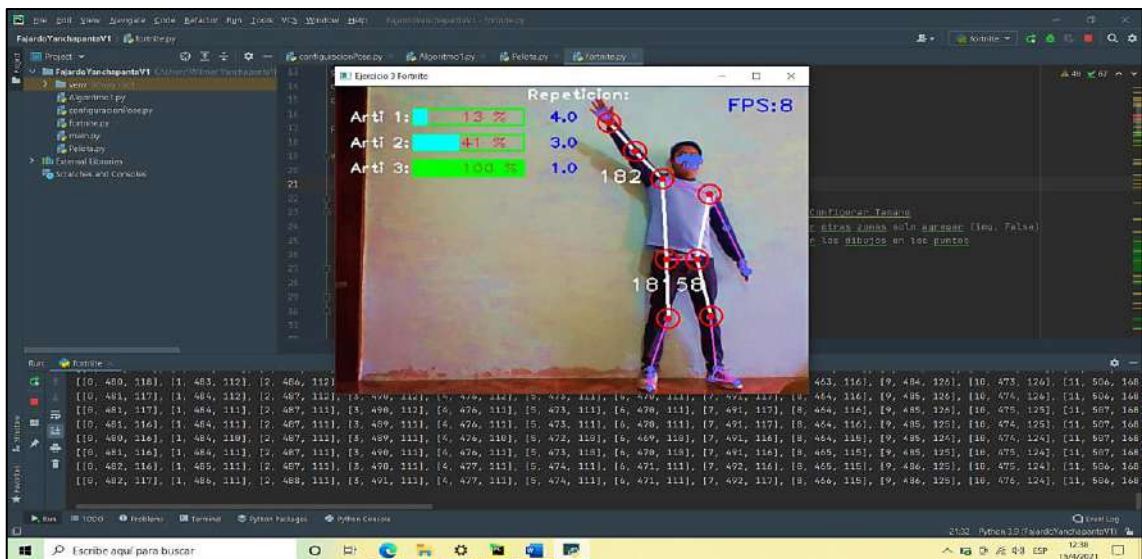


Ilustración 16-4: Prueba 2 en un adulto

Realizado por: Fajardo C., Yanchapanta W., 2022.



Ilustración 17-4: Prueba 3 en un adulto

Realizado por: Fajardo C., Yanchapanta W., 2022.

Mediante la comprobación del funcionamiento, se ha demostrado que el sistema no recae en un aprendizaje memorizado y es capaz de identificar los puntos más relevantes de las articulaciones fuertes del cuerpo humano sin importar la edad. Esto nos solventa las incertidumbres que se pueden presentar en las actividades, es decir, no importará que tan lejos o cerca esté el individuo, el proyecto funcionará, en el próximo capítulo se podrá a prueba este proceso para medir el rendimiento de una manera más objetiva.

4.3.2. Medición de los ángulos entre articulaciones

Para realizar un seguimiento adecuado de los ejercicios planteados y supervisar la correcta ejecución de estos se plantea la metodología de la medición de los ángulos de inclinación que tienen entre las dos semirrectas que forman las extremidades y el tronco con respecto a un vértice común, de esta manera se obtiene un valor cuantitativo para tener una orientación de como el infante está alcanzando los objetivos planteados en cada movimiento del ejercicio.

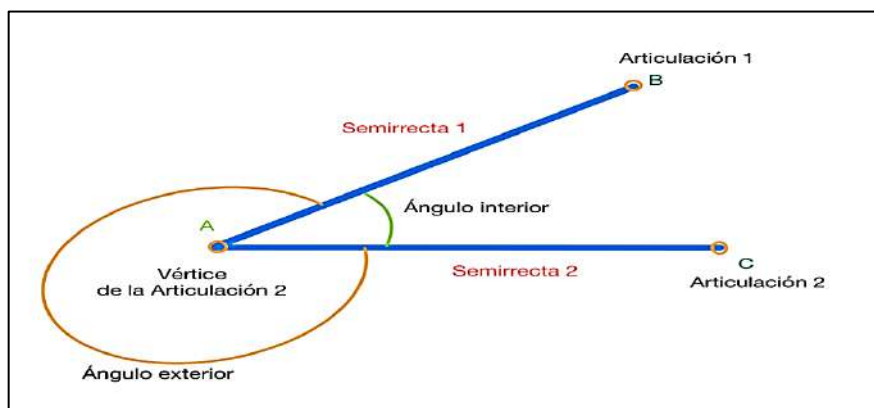


Ilustración 18-4: Medida de los ángulos

Realizado por: Fajardo C., Yanchapanta W., 2022.

Para cada ejercicio se identifica las articulaciones más relevantes de la actividad, estrictamente deben ser 3 puntos más relevantes para formar una triangulación como se observa en el diagrama de flujo del gráfico 2-4, y de esta manera medir los ángulos externos como se ve en la ilustración 19-4.



Ilustración 19-4: Resultado medición de los ángulos externos

Realizado por: Fajardo Cristhian y Yanchapanta Wilmer, 2022

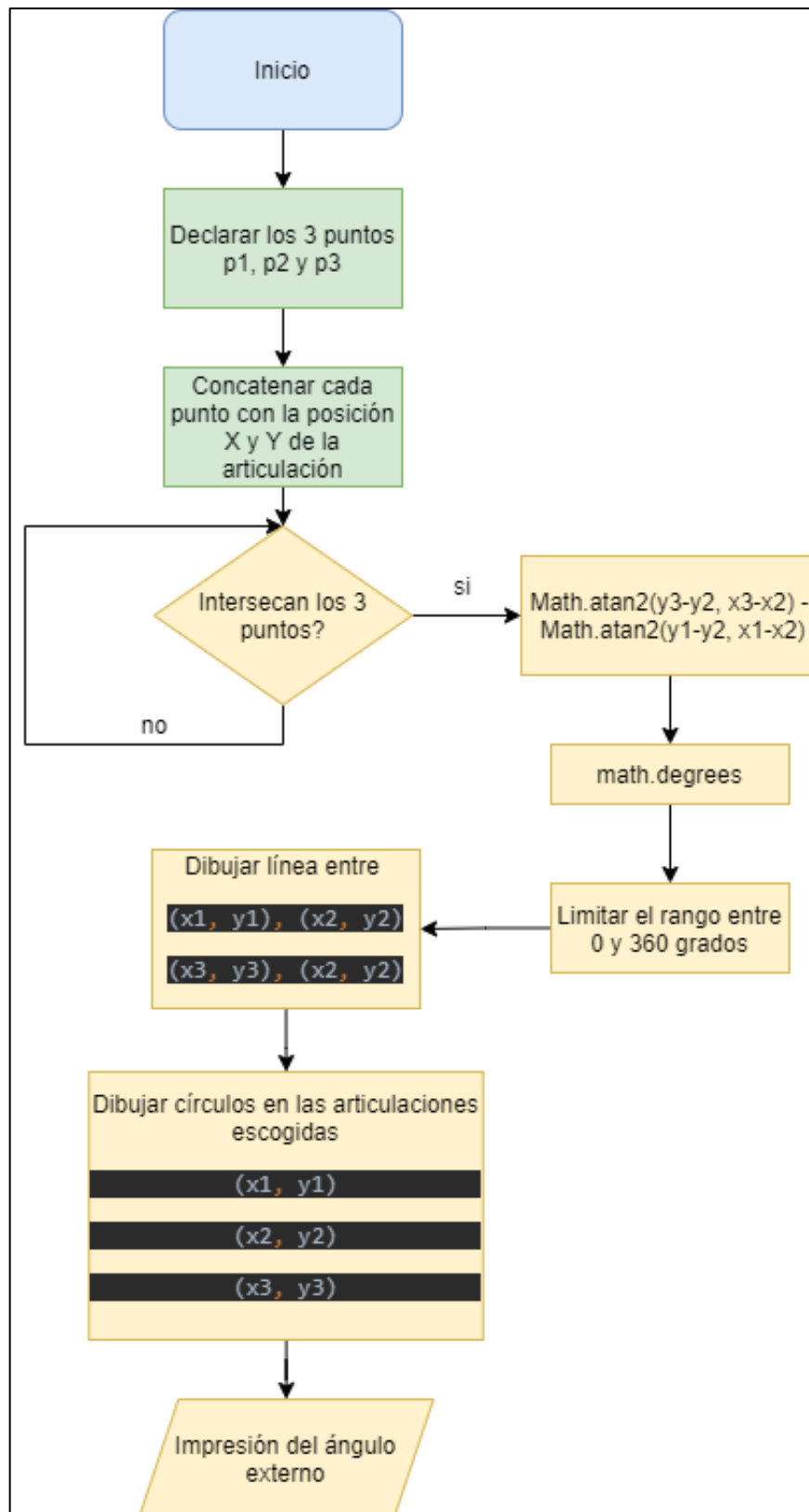


Gráfico 2-4: Diagrama de flujo medida de los ángulos

Realizado por: Fajardo C., Yanchapanta W., 2022.

CAPÍTULO V

5. MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En el presente capítulo se expresan las pruebas realizadas y los resultados obtenidos de todo el sistema en conjunto con la finalidad de comprobar el correcto funcionamiento y verificar el alcance de los objetivos planteados al iniciar esta investigación.

Para la evaluación objetiva del sistema se adoptó el criterio de la validación analítica, validación experta, validación observacional y validación experimental según (COBOS, 2017).

5.1. Ejecución de ejercicios con el prototipo

Para la puesta en marcha de los ejercicios, fue necesario previamente calibrar cada ángulo limitante en los algoritmos de cada actividad, de esta manera se tendrá un valor estándar para los niños de 4 y 5 años respectivamente. Las pruebas se realizaron con quince niños para la respectiva validación de los resultados, en total fueron 5 repeticiones por cada ejercicio dándonos un total de 75 repeticiones por cada actividad, y 450 escenarios de comprobación.

5.1.1. Ejercicio 1

El ejercicio 1 corresponde a correr en el propio sitio, levantando la rodilla hasta la altura de la cadera, el brazo extendido en la parte posterior y flexionado en la altura del pecho como se observa en la ilustración 1-5.



Ilustración 1-5: Ejercicio 1, correr en el mismo sitio, etapa evaluación

Fuente: (WIKIHOW, 2020)

Para poder cuantificar los resultados, se estableció una meta de 3 minutos en promedio para realizar 5 repeticiones por cada ejercicio, si el infante logra cumplir en menos de 3 minutos, el sistema y el niño se adaptaron correctamente al ejercicio, recalcando que el sistema debe identificar en tiempo real las articulaciones, si la postura es adecuada se cumple una repetición,

caso contrario no se suma al número de repeticiones, estos resultados se expresan a continuación en la siguiente tabla 1-5.

Tabla 1-5: Evaluación ejercicio 1 de los 15 niños

	Edad	Detalle	Repetición 1	Repetición 2	Repetición 3	Repetición 4	Repetición 5	Promedio
Niño 1	4	Tiempo de ejecución	03:10	02:40	02:39	02:43	02:41	02:47
		Si cumple		X	X	X	X	4
		No cumple	X					1
Niño 2	5	Tiempo de ejecución	02:46	03:02	03:12	02:55	02:43	02:56
		Si cumple				X	X	2
		No cumple	X	X	X			3
Niño 3	5	Tiempo de ejecución	02:57	02:23	02:24	02:31	02:18	02:31
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 4	4	Tiempo de ejecución	02:48	02:37	02:41	02:51	02:49	02:45
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	02:59	02:51	02:39	02:45	02:40	02:47
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de ejecución	02:50	03:07	02:57	02:55	03:02	02:58
		Si cumple	X		X	X		3
		No cumple		X			X	2
Niño 7	4	Tiempo de ejecución	03:12	03:03	02:52	02:57	03:01	03:01
		Si cumple			X	X		2
		No cumple	X	X			X	3
Niño 8	5	Tiempo de ejecución	02:56	02:45	02:39	02:52	02:47	02:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	02:39	02:32	02:41	02:45	02:29	02:37
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	02:48	02:54	02:58	02:50	02:42	02:50
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 11	5	Tiempo de ejecución	03:06	02:51	02:47	03:10	02:45	02:56
		Si cumple		X	X		X	3
		No cumple	X			X		2
Niño 12	5	Tiempo de ejecución	03:02	03:07	02:47	02:49	02:42	02:53
		Si cumple			X	X	X	3
		No cumple	X	X				2
Niño 13	5	Tiempo de ejecución	02:31	02:29	02:42	02:35	02:25	02:32
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	02:43	02:41	02:45	02:48	02:40	02:43
		Si cumple	X	X	X	X	X	5

		No cumple						0
Niño 15	5	Tiempo de ejecución	02:58	02:51	02:40	03:12	02:56	02:55
		Si cumple	X	X	X		X	4
		No cumple				X		1

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.1.2. Ejercicio 2

Siguiendo con la metodología anterior, para esta actividad se destinó ejecutar la simulación del juego de básquet, pero en el mismo sitio ilustración 2-5, para esta tarea fue necesario estimar un tiempo de 3 minutos para realizar 10 repeticiones, los resultados se reflejan en la siguiente tabla:



Ilustración 2-5: Ejercicio 2, rebotar la pelota en el mismo sitio, etapa evaluación

Fuente: (PINTEREST, 2020)

Tabla 2-5: Evaluación de ejercicio 2 de los 15 niños

	Edad	Detalle	Repeticion 1	Repeticion 2	Repeticion 3	Repeticion 4	Repeticion 5	Promedio
Niño 1	4	Tiempo de ejecución	03:20	03:10	02:58	02:55	02:56	03:04
		Si cumple			X	X	X	3
		No cumple	X	X				2
Niño 2	5	Tiempo de ejecución	02:51	02:55	02:45	02:47	02:42	02:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 3	5	Tiempo de ejecución	02:43	02:20	02:50	03:03	02:43	02:44
		Si cumple	X	X	X		X	4
		No cumple				X		1
Niño 4	4	Tiempo de ejecución	02:59	02:51	02:57	02:51	02:45	02:53
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	02:42	02:50	02:56	02:49	02:42	02:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de	03:12	03:02	03:05	02:54	02:58	03:02

		ejecución						
		Si cumple				X	X	2
		No cumple	X	X	X			3
Niño 7	4	Tiempo de ejecución	03:01	03:07	02:58	02:53	02:59	03:00
		Si cumple			X	X	X	3
		No cumple	X	X				2
Niño 8	5	Tiempo de ejecución	02:41	02:43	02:51	02:46	02:53	02:47
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	02:41	02:32	02:46	02:51	02:38	02:42
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	02:56	02:59	03:06	03:01	02:45	02:57
		Si cumple	X	X			X	3
		No cumple			X	X		2
Niño 11	5	Tiempo de ejecución	02:57	02:48	02:47	03:01	02:43	02:51
		Si cumple	X	X	X		X	4
		No cumple				X		1
Niño 12	5	Tiempo de ejecución	02:31	02:28	02:37	02:40	02:41	02:35
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 13	5	Tiempo de ejecución	02:58	02:46	02:41	02:52	02:50	02:49
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	02:45	02:50	02:57	03:12	03:05	02:58
		Si cumple	X	X	X			3
		No cumple				X	X	2
Niño 15	5	Tiempo de ejecución	02:47	02:56	02:40	02:51	02:45	02:48
		Si cumple	X	X	X	X	X	5
		No cumple						0

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.1.3. Ejercicio 3

Para esta actividad se optó por un baile trend denominado Floss de Fornite expresado en la ilustración 3-5, el cual se estimó un tiempo de 3 minutos para realizar 5 repeticiones, los resultados de las evaluaciones se expresan en la tabla 3-5.



Ilustración 3-5: Ejercicio 3, Floss Dance o slide, etapa de evaluación

Fuente: (WIKIHOW, 2020)

Tabla 3-5: Evaluación ejercicio 3 de los 15 niños

	Edad	Detalle	Repetición 1	Repetición 2	Repetición 3	Repetición 4	Repetición 5	Promedio
Niño 1	4	Tiempo de ejecución	02:04	02:12	02:09	02:02	02:11	02:08
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 2	5	Tiempo de ejecución	01:55	02:11	02:02	01:56	01:58	02:00
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 3	5	Tiempo de ejecución	03:30	02:45	03:02	02:43	02:55	02:59
		Si cumple		X		X	X	3
		No cumple	X		X			2
Niño 4	4	Tiempo de ejecución	02:25	02:17	02:21	02:14	02:05	02:16
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	02:12	02:19	02:17	02:14	02:07	02:14
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de ejecución	02:21	02:34	02:31	02:22	02:27	02:27
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 7	4	Tiempo de ejecución	01:54	01:57	02:07	02:05	02:04	02:01
		Si cumple	X	X	X	X	X	5

		No cumple						0
Niño 8	5	Tiempo de ejecución	02:07	02:18	02:23	02:31	02:26	02:21
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	02:35	02:29	02:34	02:41	02:38	02:35
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	02:47	02:41	02:52	02:55	02:43	02:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 11	5	Tiempo de ejecución	02:26	02:39	02:21	02:32	02:37	02:31
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 12	5	Tiempo de ejecución	02:19	02:25	02:31	02:29	02:21	02:25
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 13	5	Tiempo de ejecución	02:01	02:09	02:19	02:11	02:07	02:09
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	02:29	02:31	02:39	02:43	02:34	02:35
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 15	5	Tiempo de ejecución	02:47	02:41	02:43	02:43	02:45	02:44
		Si cumple	X	X	X	X	X	5
		No cumple						0

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.1.4. Ejercicio 4

Para el cuarto ejercicio, según el marco teórico se estableció este ejercicio como parte de la activación de los reflejos del infante acompañado a la coordinación de sus extremidades inferiores al momento de atrapar el balón plástico de 10 pulgadas, los resultados de esta actividad se reflejan en las siguientes tablas.

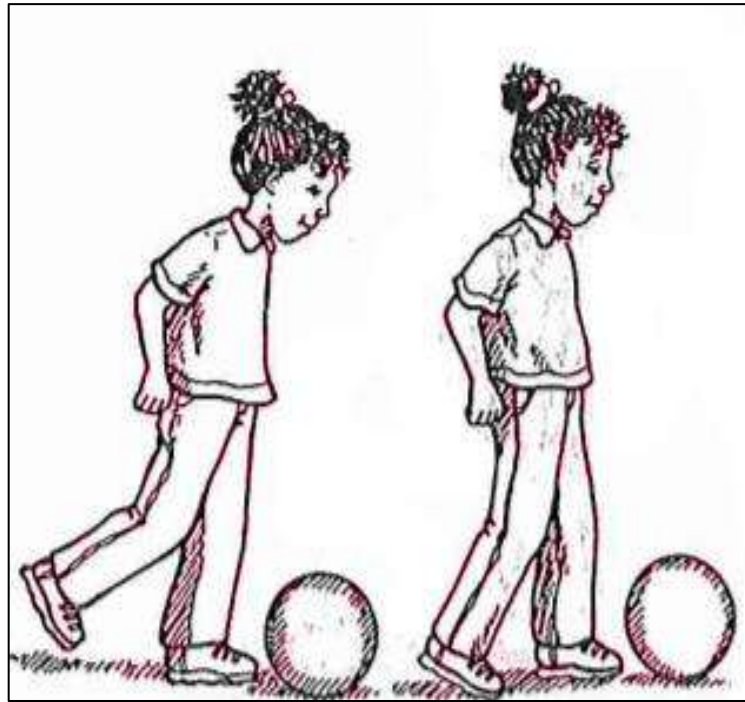


Ilustración 4-5: Ejercicio 4, kick, etapa evaluación

Fuente: (PINTEREST, 2020)

Tabla 4-5: Evaluación ejercicio 4 de los 15 niños

	Edad	Detalle	Repeticion 1	Repeticion 2	Repeticion 3	Repeticion 4	Repeticion 5	Promedio
Niño 1	4	Tiempo de ejecución	03:32	03:25	03:44	03:38	03:41	03:36
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 2	5	Tiempo de ejecución	03:33	03:40	03:39	03:23	03:52	03:37
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 3	5	Tiempo de ejecución	03:43	03:41	03:21	03:30	03:51	03:37
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 4	4	Tiempo de ejecución	03:42	03:41	03:45	03:54	03:57	03:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	03:36	03:39	03:42	03:45	03:52	03:43
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de ejecución	03:45	03:54	03:47	03:42	03:51	03:48
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 7	4	Tiempo de ejecución	03:25	03:29	03:36	03:25	03:31	03:29
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 8	5	Tiempo de ejecución	03:42	03:45	03:42	03:44	03:41	03:43

		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	03:58	03:48	03:46	03:56	03:54	03:52
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	03:42	03:37	03:39	03:46	03:41	03:41
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 11	5	Tiempo de ejecución	03:41	03:49	03:46	03:45	03:48	03:46
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 12	5	Tiempo de ejecución	03:58	03:54	03:51	03:54	03:56	03:55
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 13	5	Tiempo de ejecución	03:48	03:42	03:46	03:58	03:59	03:51
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	03:48	03:46	03:51	03:53	03:52	03:50
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 15	5	Tiempo de ejecución	03:42	03:38	03:35	03:32	03:29	03:35
		Si cumple	X	X	X	X	X	5
		No cumple						0

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.1.5. Ejercicio 5

Para el ejercicio 5 se establece la actividad Overhand throw o lanzamiento de la pelota con una mano como se ilustra en la ilustración 5-5, se estimó un tiempo promedio de 4 minutos para realizar las 5 repeticiones, los resultados de esta actividad se expresan en la siguiente tabla 5-5.

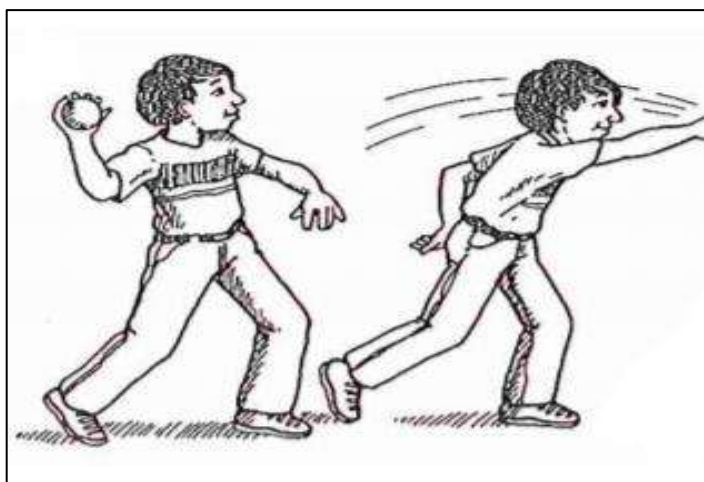


Ilustración 5-5: Ejercicio 5, Overhand Throw, etapa evaluación

Fuente: (PINTEREST, 2020)

Tabla 5-5: Evaluación ejercicio 5 de los 15 niños

	Edad	Detalle	Repetición 1	Repetición 2	Repetición 3	Repetición 4	Repetición 5	Promedio
Niño 1	4	Tiempo de ejecución	03:18	03:35	03:31	03:48	03:54	03:37
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 2	5	Tiempo de ejecución	03:11	02:58	03:09	03:43	02:53	03:11
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 3	5	Tiempo de ejecución	03:10	02:54	02:33	02:32	02:47	02:47
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 4	4	Tiempo de ejecución	03:14	03:21	03:19	03:09	03:12	03:15
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	03:23	03:22	03:29	03:31	03:27	03:26
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de ejecución	03:31	03:40	03:42	03:37	03:32	03:36
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 7	4	Tiempo de ejecución	03:16	03:11	03:07	03:18	03:17	03:14
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 8	5	Tiempo de ejecución	03:27	03:31	03:42	03:36	03:32	03:34
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	03:18	03:24	03:29	03:18	03:31	03:24
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	03:26	03:23	03:15	03:12	03:17	03:19
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 11	5	Tiempo de ejecución	03:40	03:39	03:46	03:43	03:48	03:43
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 12	5	Tiempo de ejecución	03:36	03:32	03:35	03:31	03:37	03:34
		Si cumple	X	X	X	X	X	5
		No cumple						0

Niño 13	5	Tiempo de ejecución	03:48	03:42	03:46	03:58	03:59	03:51
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	03:25	03:26	03:29	03:32	03:21	03:27
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 15	5	Tiempo de ejecución	03:42	03:37	03:35	03:21	03:29	03:33
		Si cumple	X	X	X	X	X	5
		No cumple						0

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.1.6. Ejercicio 6

Para el ejercicio 6 se el salto largo horizontal ilustrado en la ilustración 6-5, se estimó un tiempo de 5 minutos para realizar las 5 repeticiones, los resultados de esta actividad se reflejan en la siguiente tabla.

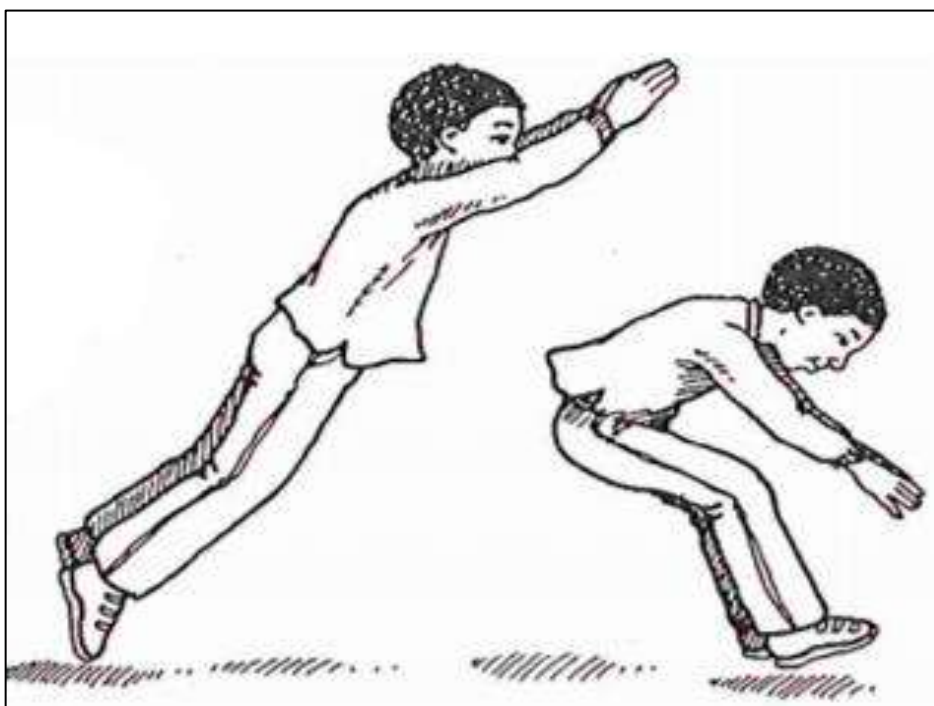


Ilustración 6-5: Ejercicio 6, Saltos largos horizontales, etapa evaluación

Fuente: (JUEGOSINFANTILES, 2020)

Tabla 6-5: Evaluación ejercicio 6 de los 15 niños

	Edad	Detalle	Repetición 1	Repetición 2	Repetición 3	Repetición 4	Repetición 5	Promedio
Niño 1	4	Tiempo de ejecución	05:33	05:11	04:43	04:49	04:52	05:02
		Si cumple			X	X	X	3
		No cumple	X	X				2
Niño 2	5	Tiempo de ejecución	04:43	04:35	04:22	03:43	03:54	04:15

		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 3	5	Tiempo de ejecución	04:22	04:43	04:20	04:12	03:54	04:18
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 4	4	Tiempo de ejecución	03:55	03:59	04:13	04:15	04:21	04:09
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 5	4	Tiempo de ejecución	04:18	04:24	03:50	03:52	03:49	04:03
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 6	4	Tiempo de ejecución	04:31	04:29	04:14	04:26	04:17	04:23
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 7	4	Tiempo de ejecución	04:12	04:15	04:19	04:22	04:24	04:18
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 8	5	Tiempo de ejecución	04:25	04:27	04:23	04:31	04:22	04:26
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 9	5	Tiempo de ejecución	04:02	04:11	03:52	03:49	03:54	03:58
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 10	5	Tiempo de ejecución	03:48	03:53	03:51	03:37	04:01	03:50
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 11	5	Tiempo de ejecución	03:32	03:37	03:42	03:43	03:41	03:39
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 12	5	Tiempo de ejecución	03:52	03:52	03:46	03:32	03:49	03:46
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 13	5	Tiempo de ejecución	03:29	03:32	03:36	03:48	03:45	03:38
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 14	5	Tiempo de ejecución	04:29	04:33	04:35	04:21	04:17	04:27
		Si cumple	X	X	X	X	X	5
		No cumple						0
Niño 15	5	Tiempo de ejecución	04:27	04:29	04:27	04:21	04:32	04:27
		Si cumple	X	X	X	X	X	5
		No cumple						0

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.2. Resumen de la evaluación

Una vez culminada la etapa de la recolección de los datos en cada una de las pruebas se analiza que el cumplimiento de los objetivos establecidos se ha cumplido con éxito total, se refleja en la

tabla 7-5. Se visualiza que los tiempos promedios para cada uno de los ejercicios mantienen tiempos similares para niños de edades de 4 y 5 años. Mientras la tabla 8-5 muestra el número de veces que un niño cumple o no el ejercicio planteado con la utilización del prototipo.

Tabla 7-5: Tiempos de ejecución con el prototipo ejecutándose

		Niño 1	Niño 2	Niño 3	Niño 4	Niño 5	Niño 6	Niño 7	Niño 8	Niño 9	Niño 10	Niño 11	Niño 12	Niño 13	Niño 14	Niño 15
Ejercicio 1	Repetición 1	03:10	02:46	02:57	02:48	02:59	02:50	03:12	02:56	02:39	02:48	03:06	03:02	02:31	02:43	02:58
	Repetición 2	02:40	03:02	02:23	02:37	02:51	03:07	03:03	02:43	02:35	02:52	02:54	03:01	02:27	02:49	02:41
	Repetición 3	02:39	03:12	02:24	02:41	02:39	02:57	02:52	02:39	02:49	02:58	02:47	02:47	02:47	02:42	02:45
	Repetición 4	02:43	02:55	02:31	02:51	02:45	02:55	02:57	02:52	02:45	02:50	03:10	02:49	02:35	02:48	03:12
	Repetición 5	02:41	02:43	02:18	02:49	02:40	03:02	03:01	02:47	02:29	02:42	02:45	02:42	02:25	02:40	02:56
Ejercicio 2	Repetición 1	03:20	02:51	02:43	02:59	02:42	03:12	03:01	02:41	02:41	02:56	02:57	02:31	02:58	02:45	02:47
	Repetición 2	03:10	02:55	02:20	02:51	02:50	03:02	03:07	02:43	02:32	02:59	02:48	02:28	02:46	02:50	02:56
	Repetición 3	02:58	02:45	02:50	02:57	02:56	03:05	02:58	02:51	02:46	03:06	02:47	02:37	02:41	02:57	02:40
	Repetición 4	02:55	02:47	03:03	02:51	02:49	02:54	02:53	02:46	02:51	03:01	03:01	02:40	02:52	03:12	02:51
	Repetición 5	02:56	02:42	02:43	02:45	02:42	02:58	02:59	02:53	02:38	02:45	02:43	02:41	02:50	03:05	02:45
Ejercicio 3	Repetición 1	02:04	01:55	03:30	02:25	02:12	02:21	01:54	02:07	02:35	02:47	02:26	02:19	02:01	02:29	02:47
	Repetición 2	02:12	02:15	02:45	02:17	02:19	02:34	01:57	02:18	02:29	02:41	02:39	02:25	02:09	02:31	02:41
	Repetición 3	02:09	02:02	03:02	02:21	02:17	02:31	02:07	02:23	02:34	02:52	02:21	02:31	02:19	02:39	02:43
	Repetición 4	02:02	01:56	02:43	02:14	02:14	02:22	02:05	02:31	02:41	02:55	02:32	02:29	02:11	02:43	02:43
	Repetición 5	02:11	01:58	02:55	02:05	02:07	02:27	02:04	02:26	02:38	02:43	02:37	02:21	02:07	02:34	02:45
Ejercicio 4	Repetición 1	03:32	03:33	03:43	03:42	03:36	03:45	03:25	03:42	03:58	03:42	03:41	03:51	03:48	03:48	03:42
	Repetición 2	03:25	03:40	03:41	03:41	03:39	03:54	03:29	03:45	03:48	03:37	03:41	03:54	03:42	03:46	03:38
	Repetición 3	03:44	03:39	03:21	03:45	03:42	03:47	03:36	03:42	03:46	03:39	03:46	03:51	03:46	03:51	03:35
	Repetición 4	03:38	03:32	03:30	03:54	03:45	03:42	03:25	03:44	03:56	03:46	03:41	03:54	03:58	03:53	03:32
	Repetición 5	03:41	03:52	03:51	03:57	03:55	03:51	03:33	03:41	03:54	03:41	03:41	03:56	03:59	03:52	03:29
Ejercicio 5	Repetición 1	03:18	03:11	03:10	03:14	03:23	03:31	03:16	03:27	03:18	03:26	03:40	03:33	03:48	03:25	03:42
	Repetición 2	03:35	02:58	02:54	03:21	03:29	03:40	03:11	03:33	03:29	03:29	03:39	03:33	03:42	03:26	03:37
	Repetición 3	03:31	03:09	02:33	03:19	03:29	03:42	03:07	03:40	03:29	03:15	03:46	03:33	03:46	03:29	03:35
	Repetición 4	03:48	03:43	02:32	03:09	03:31	03:33	03:18	03:33	03:18	03:12	03:43	03:33	03:51	03:33	03:32
	Repetición 5	03:54	02:53	02:47	03:12	03:27	03:32	03:17	03:33	03:33	03:17	03:48	03:33	03:59	03:21	03:32
Ejercicio 6	Repetición 1	05:33	04:43	04:22	03:55	04:18	04:31	04:12	04:25	04:02	03:48	03:33	03:52	03:29	04:29	04:27
	Repetición 2	05:11	04:35	04:43	03:59	04:24	04:29	04:15	04:27	04:11	03:53	03:37	03:52	03:32	04:33	04:29

	Repetición 3	04:43	04:22	04:20	04:13	03:50	04:14	04:19	04:23	03:52	03:51	03:42	03:46	03:36	04:35	04:27
	Repetición 4	04:49	03:43	04:12	04:15	03:52	04:26	04:22	04:31	03:49	03:37	03:43	03:32	03:48	04:21	04:21
	Repetición 5	04:52	03:54	03:54	04:21	03:49	04:17	04:24	04:22	03:54	04:01	03:41	03:49	03:45	04:17	04:32

Realizado por: Fajardo C., Yanchapanta W., 2022.

Tabla 8-5: Número de veces que se cumple o no cada ejercicio, con el prototipo en ejecución

		Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6	Promedio	Porcentaje
Si cumple	Niño 1	4	3	5	5	5	3	4,17	69%
	Niño 2	2	5	5	5	5	5	4,50	75%
	Niño 3	5	4	3	5	5	5	4,50	75%
	Niño 4	5	5	5	5	5	5	5,00	83%
	Niño 5	5	5	5	5	5	5	5,00	83%
	Niño 6	3	2	5	5	5	5	4,17	69%
	Niño 7	2	3	5	5	5	5	4,17	69%
	Niño 8	5	5	5	5	5	5	5,00	83%
	Niño 9	5	5	5	5	5	5	5,00	83%
	Niño 10	5	3	5	5	5	5	4,67	78%
	Niño 11	3	4	5	5	5	5	4,50	75%
	Niño 12	3	5	5	5	5	5	4,67	78%
	Niño 13	5	5	5	5	5	5	5,00	83%
	Niño 14	5	3	5	5	5	5	4,67	78%
	Niño 15	4	5	5	5	5	5	4,83	81%
No cumple	Niño 1	1	2	0	0	0	2	0,83	14%
	Niño 2	3	0	0	0	0	0	0,50	8%
	Niño 3	0	1	2	0	0	0	0,50	8%
	Niño 4	0	0	0	0	0	0	0,00	0%
	Niño 5	0	0	0	0	0	0	0,00	0%
	Niño 6	2	3	0	0	0	0	0,83	14%
	Niño 7	3	2	0	0	0	0	0,83	14%
	Niño 8	0	0	0	0	0	0	0,00	0%
	Niño 9	0	0	0	0	0	0	0,00	0%
	Niño 10	0	2	0	0	0	0	0,33	6%
	Niño 11	2	1	0	0	0	0	0,50	8%
	Niño 12	2	0	0	0	0	0	0,33	6%
	Niño 13	0	0	0	0	0	0	0,00	0%
	Niño 14	0	2	0	0	0	0	0,33	6%
	Niño 15	1	0	0	0	0	0	0,17	3%

Realizado por: Fajardo C., Yanchapanta W., 2022.

5.3. Ejecución de ejercicios sin el prototipo

Al ser este un prototipo nuevo y al no contar con un dispositivo similar en el mercado con el cual comparar las muestras obtenidas con el prototipo en ejecución, se procede a realizar los mismos ejercicios, con los mismos quince niños participantes, pero en este caso sin la guía de la

pantalla la misma que indica la correcta postura y la verificación automática del cumplimiento del ejercicio o no.

También se precisa de un cronometraje externo medido por nosotros para comparar tiempos de ejecución de cada uno de los ejercicios. A continuación, la tabla 9-5, recoge un resumen de los datos obtenido de cada una de las pruebas.

Tabla 9-5: Resumen de ejecución de ejercicios sin el prototipo

		Niño 1	Niño 2	Niño 3	Niño 4	Niño 5	Niño 6	Niño 7	Niño 8	Niño 9	Niño 10	Niño 11	Niño 12	Niño 13	Niño 14	Niño 15
Ejercicio 1	Repetición 1	03:34	03:16	03:04	03:00	02:50	03:12	03:11	02:47	02:56	03:08	03:27	03:22	03:32	03:32	03:15
	Repetición 2	03:47	03:44	03:16	03:10	02:57	03:19	03:07	02:51	02:59	03:02	03:32	03:25	03:31	03:35	03:11
	Repetición 3	03:27	03:31	02:51	03:21	02:50	03:06	03:00	02:59	03:04	02:54	03:36	03:26	03:24	03:41	02:52
	Repetición 4	03:15	03:18	02:55	03:16	03:01	03:07	03:11	03:01	03:07	02:50	03:31	03:24	03:20	03:40	03:27
	Repetición 5	03:55	03:21	03:11	03:12	03:15	03:14	03:01	03:06	03:13	02:56	03:29	03:20	03:32	03:42	03:18
Ejercicio 2	Repetición 1	04:32	03:45	03:24	03:47	03:56	03:31	04:15	03:05	03:21	04:12	03:21	05:02	03:27	03:18	04:12
	Repetición 2	04:51	03:21	03:05	03:32	04:12	03:22	04:18	03:50	03:16	04:22	03:32	04:46	03:36	03:25	03:46
	Repetición 3	03:49	03:37	03:17	03:46	03:50	03:27	04:19	03:37	03:35	04:36	03:42	04:39	03:39	03:27	04:01
	Repetición 4	04:43	03:19	03:13	03:48	03:46	03:12	04:22	03:28	03:40	04:15	04:05	04:47	03:21	03:14	03:52
	Repetición 5	04:11	03:28	03:11	03:41	03:49	03:16	04:01	03:26	03:36	04:04	04:25	04:30	03:32	03:29	03:44
Ejercicio 3	Repetición 1	03:34	02:51	03:55	03:17	03:55	04:21	03:26	03:04	04:15	03:17	03:21	03:42	03:26	03:21	03:12
	Repetición 2	03:47	03:05	03:55	03:34	04:25	04:36	03:47	03:46	04:23	03:36	03:16	03:32	03:14	03:35	04:26
	Repetición 3	03:27	03:17	04:03	03:56	04:27	04:19	03:42	03:50	04:17	03:26	03:26	03:56	03:36	03:44	04:47
	Repetición 4	03:15	02:56	03:49	03:48	03:56	04:26	03:36	04:15	04:27	03:14	03:26	03:32	03:36	03:32	03:58
	Repetición 5	03:55	02:54	03:53	03:42	03:40	04:31	03:37	04:14	04:14	03:15	03:15	03:41	03:32	03:57	03:42
Ejercicio 4	Repetición 1	04:17	03:51	04:12	04:15	04:44	04:33	04:21	04:50	03:36	03:25	04:17	04:01	04:14	03:44	04:29
	Repetición 2	04:36	04:16	04:04	04:20	04:35	04:02	04:03	04:35	03:43	03:47	04:02	03:43	04:19	03:52	04:39
	Repetición 3	03:44	03:57	03:51	04:23	04:33	04:36	04:48	03:20	03:42	04:32	03:56	04:10	04:00	04:00	04:52
	Repetición 4	03:38	04:11	03:44	04:16	04:41	03:47	04:55	04:42	03:39	03:53	04:22	04:12	04:12	03:50	04:47
	Repetición 5	03:41	04:06	04:03	04:12	04:26	03:33	04:50	04:47	03:42	03:38	04:13	03:29	04:15	04:14	05:02
Ejercicio 5	Repetición 1	04:28	03:51	03:31	03:40	03:49	04:15	04:00	03:30	04:30	03:56	04:43	03:55	03:37	04:31	03:28
	Repetición 2	04:51	03:52	03:12	04:05	04:01	04:00	04:21	03:38	04:14	03:50	04:49	03:46	04:21	04:00	03:29
	Repetición 3	03:56	04:03	03:06	04:16	03:58	04:02	04:32	03:50	04:24	03:40	04:46	03:52	04:23	04:45	03:46
	Repetición 4	03:46	04:07	03:16	04:19	04:19	04:16	04:22	04:13	04:18	03:39	04:41	03:21	04:21	04:00	04:12
	Repetición 5	04:37	03:53	03:02	04:20	03:40	04:10	04:26	04:22	04:31	03:38	04:33	04:15	04:33	04:12	04:37

Ejercicio 6	Repetición 1	05:44	04:31	04:51	04:30	04:59	04:27	05:02	05:50	04:40	04:28	04:36	04:44	05:15	05:34	05:35
	Repetición 2	05:37	04:41	04:48	04:36	05:17	04:32	05:17	05:30	04:36	04:23	04:22	04:48	05:14	05:22	05:25
	Repetición 3	05:11	04:36	04:50	04:40	05:14	04:16	05:21	05:26	04:44	04:25	04:39	04:47	05:03	05:15	05:11
	Repetición 4	04:40	04:11	04:41	04:46	05:02	05:02	04:40	05:29	04:58	04:29	04:47	04:37	05:26	05:04	05:09
	Repetición 5	05:45	03:56	03:54	04:50	05:04	05:32	04:48	04:50	05:01	04:40	04:42	04:28	05:14	05:00	04:52

Realizado por: Fajardo C., Yanchapanta W., 2022.

En el caso del cumplimiento o no de cada uno de los ejercicios propuestos sin la utilización del prototipo, tenemos la tabla 10-5, la misma que resume los casos de éxitos y de fracasos de cada uno de los participantes.

Tabla 10-5: Número de veces que se cumple un ejercicio sin la utilización del prototipo

		Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6	Promedio	Porcentaje
Si cumple	Niño 1	0	0	0	3	2	1	1,00	17%
	Niño 2	0	0	3	2	3	5	2,17	36%
	Niño 3	2	0	0	2	5	5	2,33	39%
	Niño 4	0	0	0	0	1	5	1,00	17%
	Niño 5	3	0	0	0	3	1	1,17	19%
	Niño 6	0	0	0	0	0	3	0,50	8%
	Niño 7	0	0	0	0	0	2	0,33	6%
	Niño 8	3	0	0	0	3	1	1,17	19%
	Niño 9	2	0	0	0	0	4	1,00	17%
	Niño 10	3	0	0	0	5	5	2,17	36%
	Niño 11	0	0	0	0	0	5	0,83	14%
	Niño 12	0	0	0	0	4	5	1,50	25%
	Niño 13	0	0	0	0	1	0	0,17	3%
	Niño 14	0	0	0	0	0	0	0,00	0%
	Niño 15	1	0	0	0	3	1	0,83	14%
No cumple	Niño 1	5	5	5	2	3	4	4,00	67%
	Niño 2	5	5	2	3	2	0	2,83	47%
	Niño 3	3	5	5	3	0	0	2,67	44%
	Niño 4	5	5	5	5	4	0	4,00	67%
	Niño 5	2	5	5	5	2	4	3,83	64%
	Niño 6	5	5	5	5	5	2	4,50	75%
	Niño 7	5	5	5	5	5	3	4,67	78%
	Niño 8	2	5	5	5	2	4	3,83	64%
	Niño 9	3	5	5	5	5	1	4,00	67%
	Niño 10	2	5	5	5	0	0	2,83	47%
	Niño 11	5	5	5	5	5	0	4,17	69%
	Niño 12	5	5	5	5	1	0	3,50	58%
	Niño 13	5	5	5	5	4	5	4,83	81%
	Niño 14	5	5	5	5	5	5	5,00	83%
	Niño 15	4	5	5	5	2	4	4,17	69%

Realizado por: Fajardo C., Yanchapanta W., 2022.

CAPÍTULO VI

6. VALIDACIÓN DEL PROTOTIPO

En este capítulo se detalla la validación del prototipo con sus pruebas respectivas.

6.1. Evaluación estadística

Dado que las muestras de tiempos obtenidos en las tablas 7-5 y 9-5, son relacionadas entre sí, se optó por usar la prueba T de Student para la validación del dispositivo. Además, se usó la prueba de Shapiro-Wilk para verificar si los datos son normales, siendo el resultado positivo.

Para la obtención de las muestras se obtuvieron los promedios de las repeticiones de cada niño por ejercicio, consiguiendo 15 muestras en total por cada ejercicio como se observa en las tablas 1-6 y 2-6, con las cuales se procedió a realizar la prueba T de Student, comparando los datos obtenidos con el prototipo en ejecución y sin el prototipo. En la tabla 3-6 se muestra una tabla resumen con los valores obtenidos de la prueba T de Student, aplicada a cada ejercicio. Se puede notar que el valor de t para una cola es menor que el nivel de significancia ($\alpha=0.05$). La comparación es estadísticamente significativa por tal razón se concluye que el dispositivo muestra una mejora en los tiempos de ejecución de los ejercicios realizados por los niños entre 4 y 5 años.

Tabla 1-6: Promedios de los ejercicios con el prototipo

	Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6
Niño 1	167	184	128	216	217	302
Niño 2	176	168	120	217	191	255
Niño 3	151	164	179	217	167	258
Niño 4	165	173	136	228	195	249
Niño 5	167	168	134	223	206	243
Niño 6	178	182	147	228	216	263
Niño 7	181	180	121	209	194	258
Niño 8	168	167	141	223	214	266
Niño 9	157	162	155	232	204	238
Niño 10	170	177	168	221	199	230
Niño 11	176	171	151	226	223	219
Niño 12	173	155	145	235	214	226
Niño 13	152	169	129	231	231	218
Niño 14	163	178	155	230	207	267
Niño 15	175	168	164	215	213	267

Realizado por: Fajardo C., Yanchapanta W., 2022.

Tabla 2-6: Promedios de los ejercicios sin el prototipo

	Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6
Niño 1	231	265	216	239	260	323
Niño 2	206	210	181	244	237	263
Niño 3	183	194	234	239	193	277
Niño 4	192	223	219	257	248	280
Niño 5	180	235	245	276	237	307
Niño 6	192	202	267	238	249	286
Niño 7	186	255	218	273	260	302

Niño 8	177	209	229	283	235	325
Niño 9	184	210	259	216	264	288
Niño 10	178	258	201	221	223	269
Niño 11	211	229	201	258	283	277
Niño 12	203	285	221	231	230	281
Niño 13	208	211	208	254	255	314
Niño 14	218	203	218	236	258	315
Niño 15	193	235	241	286	234	314

Realizado por: Fajardo C., Yanchapanta W., 2022.

Tabla 3-6: Resumen de valores de la Prueba T de Student

	Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6
Media	0,00194429	0,001979321	0,001677469	0,00258534	0,002384568	0,002900309
Varianza	1,12575E-08	8,48036E-09	4,01492E-08	7,1847E-09	3,22596E-08	6,69225E-08
P(T<=t) una cola	1,90618E-05	7,56137E-07	1,95858E-09	0,000652471	1,86103E-07	9,96276E-07

Realizado por: Fajardo C., Yanchapanta W., 2022.

CAPÍTULO VII

7. EVALUACIÓN ECONÓMICA

En este capítulo se detalla los costos presentados en la elaboración del prototipo.

7.1. Costos

Se realiza un análisis general de los costos efectuados en el diseño y construcción de un dispositivo electrónico de estimulación temprana motriz gruesa para niños de 4 a 5 años empleando técnicas de visión artificial.

7.1.1. Costos directos

Son los que se relacionan directamente para la realización o fabricación del modelo prototipo.

7.1.1.1. Costos implementación del prototipo

Tabla 1-7: Costos electrónicos

Cantidad	Elemento	V. unitario (USD)	V. Total (USD)
1	Computador	800	800
1	RASPBERRY PI 4 MODEL B 8gb RAM	179	179
1	MicroSD 32gb 10 class	29	29
1	Web cam	24	24
1	Monitor 19"	80	80
1	Mouse + teclado	30	30
TOTAL			1142

Realizado por: Fajardo C., Yanchapanta W., 2022.

7.1.1.2. Costos implementación estructural

Tabla 2-7: Costos estructurales

Cantidad	Elemento	V. unitario (USD)	V. Total (USD)
1	Case acrílico	30	30
TOTAL			30

Realizado por: Fajardo C., Yanchapanta W., 2022.

7.1.1.3. Costos directos totales

Tabla 3-7: Costos directos totales

Descripción	V. Total
Costos electrónicos	1112
Costos mecánicos	30
Costo directo total	1172

Realizado por: Fajardo C., Yanchapanta W., 2022.

7.1.2. Costos indirectos

Tabla 4-7: Costos indirectos

N°	Descripción	V. Unitario (USD)	V. Total (USD)
1	Imprevistos	50	50
Total			50

Realizado por: Fajardo C., Yanchapanta W., 2022.

7.1.3. Costo total

Tabla 5-7: Costos totales

N°	Descripción	V. Unitario (USD)	V. Total (USD)
1	Costos directos totales	1133,00	1172
2	Costos indirectos totales	200,00	50
Costo Total			1222

Realizado por: Fajardo C., Yanchapanta W., 2022.

7.1.4. Comparativa costos

Comparando los costos de desarrollar nuestro dispositivo con diferentes dispositivos y robots ya presentes en el mercado, podemos establecer una referencia entre todos ellos. Cabe destacar que el dispositivo que en su aplicación es más parecido a nuestro dispositivo es el juguete inteligente Pushaq puesto que está basado en un controlador Raspberry Pi, sensores ultrasónicos y una cámara para capturar imagen. Además, interactúa cognitivamente con el niño a través de cuentos, juegos y reconocimiento facial (SALINAS SALINAS & CHAMORRO CARRERA, 2017).

El resto de los dispositivos son robots que se encuentran en el mercado y la mayoría de ellos se enfocan en desarrollar capacidades cognitivas de los niños a través de la programación de estos, pero no tienen una verificación para saber si los niños ejecutan las actividades sean estas físicas o no. A continuación, la tabla 6-7, y el gráfico 1-7, muestran la variación de costos de los diferentes dispositivos.

Tabla 6-7: Costos de dispositivos relacionados inteligentes que utilizan visión artificial

Dispositivo	Nuestro prototipo	Juguete Pushaq	Robot Teebot	Robot Dash&Dot	Robot Zowi	Robot Bee-Bot	Moway Scratch
Costo	\$ 1222	\$ 1500	\$ 500	\$3800	\$ 110	\$ 116	\$ 239

Realizado por: Fajardo C., Yanchapanta W., 2022.

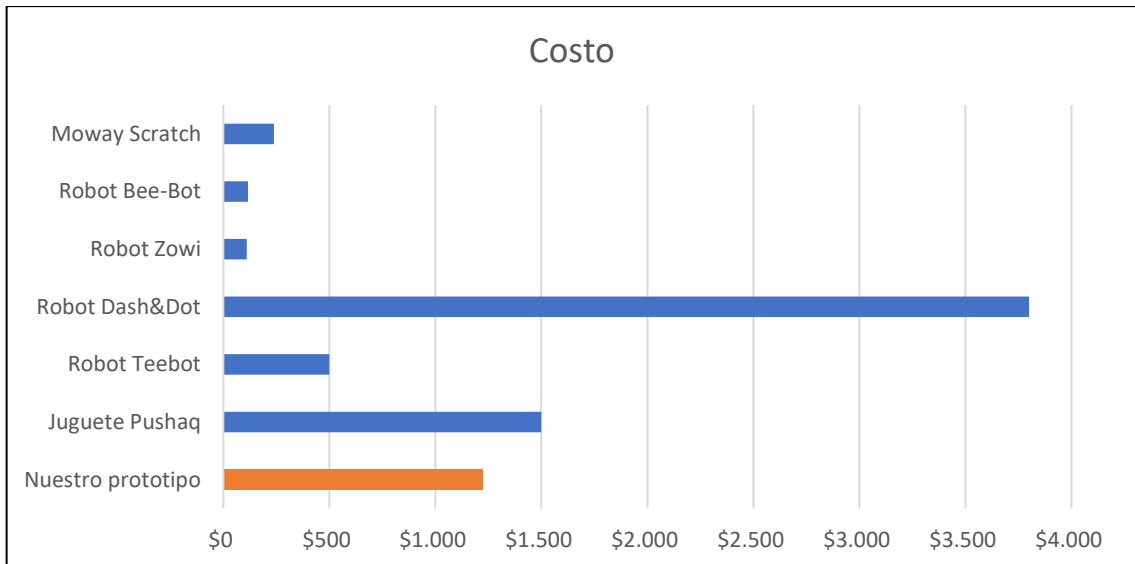


Gráfico 1-7: Comparación prototipo vs diferentes dispositivos en el mercado

Realizado por: Fajardo C., Yanchapanta W., 2022.

Aunque nuestro prototipo es de coste elevado, posee la ventaja de verificar que el niño realice todas las actividades impuestas por el mismo. Además de que estimula la motricidad gruesa de los niños participantes de las pruebas de funcionamiento.

CONCLUSIONES

- Según el análisis realizado en el estado del arte acerca a los diferentes métodos de medición objetiva de un ejercicio para mejorar la motricidad gruesa en un infante entre 4 años y 5 años se determina la metodología de medición de los ángulos de desplazamiento que tiene cada articulación, y la única manera de hacerlo en tiempo real es utilizando técnicas de visión artificial.
- Se establecieron los requisitos de diseño del software para su posterior prueba en una tarjeta embebida, de esta manera se evaluó el rendimiento que tiene nuestra solución a esta problemática, y la mejor opción de tarjetas para el propósito de visión artificial es el Raspberry pi en nuestro caso la versión 4 con 8 GB de Ram, de esta manera se pueden procesar 30 fotogramas por segundo sin perder la eficiencia de la lectura sensorial de la cámara.
- La metodología aplicada en este trabajo de titulación encaminó a que el prototipo cumpla con los parámetros establecidos que contribuyeron de manera óptima con los requerimientos del proyecto.
- El algoritmo empleado en este trabajo de titulación se basa en la codificación con Python utilizando frameworks y librerías como, OpenCV, Keras, Numpy, Mediapipe, el interface del panel de selección está estructurado con Tkinter,
- En las pruebas realizadas del sistema con 15 infantes, entre 4 y 5 años, el prototipo fue capaz de reconocer las articulaciones principales del cuerpo a una distancia promedio de 241cm, entre los 6 ejercicios establecidos cada infante pasó con satisfacción cada actividad en un tiempo promedio de 3min 40 segundos, cumpliendo con una fluidez de 4 a 10 fotogramas por segundo, la retroalimentación del infante con esta solución fue exitosa y de cierta manera flexible para despertar el interés de fortalecer el movimiento correcto de las articulaciones y despertar el interese de ejercitarse jugando.

RECOMENDACIONES

- Para una mejor respuesta en cuanto a la escalabilidad del proyecto en agregar más ejercicios a la rutina, es necesario contar con un procesamiento diferente, es decir, un levantamiento sensorial en el plano espacial, ya que el primer inconveniente que se evidenció con esta propuesta fue el planteamiento de ejercicios que ejecuten las actividades sin que las extremidades se sobrepongan, ya que la captura de las imágenes se las está haciendo de una manera plana en 2D.
- Se recomienda utilizar una pared con fondo de un color uniforme en una tonalidad mate, donde no se refleje la luz artificial o natural, con el objetivo de mejorar las respuestas ante el reconocimiento y no generar falsos cuadros que el algoritmo puede asociar con puntos de interés sin serlo.
- Es necesario respetar la distancia de 2.5m para enfocar al infante en la mitad de la ventana del programa, de esta manera se toma un encuadre de un primer plano sin perder puntos relevantes como los pies o manos.
- Para evitar un bajo rendimiento o latencia en la toma de los fotogramas en actividades prolongadas es necesario diseñar un sistema de disipación de calor para la tarjeta Raspberry pi, de esta manera aseguramos un trabajo efectivo y continuo sin que el procesador sufra un estrés.

BIBLIOGRAFÍA

ADAMS, B.J., et al, Comparing video games and laparoscopic simulators in the development of laparoscopic skills in surgical residents. *Journal of Surgical Education*, vol. 69, no. 6, pp. 714-717. ISSN 1878-7452. DOI 10.1016/j.jsurg.2012.06.006.

AHMET, C. *Artificial Intelligence: How Advance Machine Learning Will Shape The Future Of Our World*. S.l.: Shockwave Publishing via PublishDrive.

AKIN, S., 2019. Fine Motor Skills, Writing Skills and Physical Education Based Assistive Intervention Program in Children at Grade 1. *Asian Journal of Education and Training*, vol. 5, no. 4, pp. 518-525. ISSN 2519-5387.

ALOIMONOS, Y. Purposive, qualitative, active vision. *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 1-2. ISSN 1049-9660. DOI 10.1016/1049-9660(92)90080-M.

ANTOLÍN, M., et al. *La estimulación temprana y el desarrollo infantil*. Buenos Aires: Cadiex.

AYALA, S. La estimulación Temprana: Ventajas, desventajas, e información relevante | Asociación Mexicana de Psicoterapia y Educación. *Asociación Mexicana de Psicoterapia y Educación* [en línea]. [Consulta: 26 febrero 2021]. Disponible en: <https://www.psycoedu.org/la-estimulacion-temprana-ventajas-desventajas-e-informacion-relevante/>.

BARRENO, Z. & MACIAS, J. Estimulación temprana para potenciar la inteligencia psicomotriz: importancia y relación. *Revista Ciencia UNEMI*, vol. 8, no. 15, pp. 9.

BART, O., HAJAMI, D. & BAR-HAIM, Y. Predicting school adjustment from motor abilities in kindergarten. *Infant and Child Development*, vol. 16, no. 6, pp. 597-615. ISSN 1522-7219. DOI <https://doi.org/10.1002/icd.514>.

BISHOP, C. *Pattern Recognition and Machine Learning* [en línea]. Nueva York: Springer-Verlag. [Consulta: 15 mayo 2020]. Ciencias de la información y estadística. ISBN 978-0-387-31073-2. Disponible en: <https://www.springer.com/gp/book/9780387310732>.

BRAGANÇA MORAES, A. Computer-based application to assess gross motor skills using Microsoft Kinect sensor [en línea]. Thesis. S.l.: Laurentian University of Sudbury. [Consulta: 2 marzo 2021]. Disponible en: <https://zone.biblio.laurentian.ca/jspui/handle/10219/3225>.

BROWNLEE, J. *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. S.l.: Machine Learning Mastery.

BURNHAM, L., et al. BTEC Level 2 Technical Diploma Children's Play, Learning and Development Early Years Assistant Learner Handbook with ActiveBook. 1st edition. S.l.: Pearson Education. ISBN 978-1-292-19708-1.

BURTON, A.W. y MILLER, D.E. *Movement Skill Assessment*. S.l.: Human Kinetics. ISBN 978-0-87322-975-3.

CAMPBELL, F.A., et al. Adult outcomes as a function of an early childhood educational program: an Abecedarian Project follow-up. *Developmental Psychology*, vol. 48, no. 4, pp. 1033-1043. ISSN 1939-0599. DOI 10.1037/a0026644.

CENTER ON THE DEVELOPING CHILD. InBrief: The Science of Early Childhood Development. *Center on the Developing Child at Harvard University* [en línea]. [Consulta: 26 febrero 2021]. Disponible en: <https://developingchild.harvard.edu/resources/inbrief-science-of-eed/>.

CHOI, J.Y., et al. A comparative study of preprocessing mismatch effects in color image based face recognition. *Pattern Recognition*, vol. 44, no. 2, pp. 412-430. ISSN 0031-3203. DOI 10.1016/j.patcog.2010.08.020.

COBOS, F.S. Proyectos de productos editoriales multimedia. ARGN0110. S.l.: IC Editorial. ISBN 978-84-17224-53-0.

DARLING-CHURCHILL, K.E. y LIPPMAN, L. Early childhood social and emotional development: Advancing the field of measurement. *Journal of Applied Developmental Psychology*, vol. 45, pp. 1-7. ISSN 0193-3973. DOI 10.1016/j.appdev.2016.02.002.

DEHGHAN, L., et al. The Relationship Between Fine Motor Skills and Social Development and Maturation. *Iranian Rehabilitation Journal*, vol. 15, no. 4, pp. 407-414. DOI 10.29252/nrip.irj.15.4.407.

DEMAAGD, K., et al. Practical Computer Vision with SimpleCV: *The Simple Way to Make Technology See*. S.l.: O'Reilly Media, Inc. ISBN 978-1-4493-4383-5.

DZITAC, I., et al. Fuzzy Logic Is Not Fuzzy: World-renowned Computer Scientist Lotfi A. Zadeh. *International Journal of Computers Communications & Control*, vol. 12, no. 6, pp. 748. ISSN 1841-9836, 1841-9836. DOI 10.15837/ijccc.2017.6.3111.

ELGENDY, M. *Deep Learning for Vision Systems*. Edición: 1. S.l.: Manning Publications. ISBN 978-1-61729-619-2.

ELLIOTT, A. Early Childhood Education: Pathways to quality and equity for all children. *Australian Education Review* [en línea], Disponible en: <https://research.acer.edu.au/aer/4>.

GARCÍA, U. Introducción a las Redes Neuronales Pt. I. *Medium* [en línea]. [Consulta: 2 marzo 2021]. Disponible en: <https://medium.com/futurelabmx/introducci%C3%B3n-a-las-redes-neuronales-pt-i-a73f87933f8e>.

GIALAMAS, A., et al. Quality of childcare influences children's attentiveness and emotional regulation at school entry. *The Journal of Pediatrics*, vol. 165, no. 4, pp. 813- 819.e3. ISSN 1097-6833. DOI 10.1016/j.jpeds.2014.06.011.

GILLARD. NCC Spiritual and Moral Development (1993). [en línea]. [Consulta: 25 enero 2021]. Disponible en: <http://www.educationengland.org.uk/documents/ncc1993/smdev.html>.

GÓMEZ, G. Estimulación temprana en el desarrollo infantil. [en línea]. Quetzaltenango: Universidad Rafael Landívar. [Consulta: 26 enero 2020]. Disponible en: <http://biblio3.url.edu.gt/Tesario/2014/05/84/Gomez-Georgina.pdf>.

GONZALEZ, R. & WOODS, R. *Digital Image Processing*. 3. Upper Saddle River, N.J: Pearson. ISBN 978-0-13-168728-8.

GUPTA, P. & KULKARNI, N. An Introduction of Soft Computing Approach over Hard Computing. *International Journal of Latest Trends in Engineering and Technology*, vol. 3, no. 1, pp. 5.

HEATH, S. *Embedded Systems Design*. S.l.: Elsevier. ISBN 978-0-08-047756-5.

IJJINA, E.P. & CHALAVADI, K.M., 2016. Human action recognition using genetic algorithms and convolutional neural networks. *Pattern Recognition*, vol. 59, pp. 199-212. ISSN 0031-3203. DOI 10.1016/j.patcog.2016.01.012.

INDRAYANI, N. Language Development at Early Childhood. *International Conference on Education (IECO) Proceeding*. University of Muhammadiyah Jember: s.n., pp. 279-289. ISBN 978-602-698-821-8.

KWON, S.; & O'NEILL, M. Socioeconomic and Familial Factors Associated with Gross Motor Skills among US Children Aged 3-5 Years: The 2012 NHANES National Youth Fitness Survey. *International Journal of Environmental Research and Public Health*, vol. 17, no. 12. ISSN 1660-4601. DOI 10.3390/ijerph17124491.

LEFA, B. The Piaget theory of cognitive development: an educational implications. *Educational Psychology*, vol. 1, pp. 9.

LEVINE, M.D., et al. *Developmental variation and learning disorders*. Cambridge, MA, US: Educators Publishing Service. Developmental variation and learning disorders. ISBN 978-0-8388-1992-0.

LUCICH, M. Building Baby's Intelligence: Why Infant Stimulation Is So Important. *California Childcare Health Program* [en línea]. [Consulta: 15 octubre 2020]. Disponible en: <https://cchp.ucsf.edu/content/building-babys-intelligence-why-infant-stimulation-so-important>.

LUDEÑA, E.; & LISSETH, D. La estimulación temprana y su incidencia en el aprendizaje de los niños y niñas de primer año de educación básica de la escuela "san tarsicio" del cantón macará, periodo lectivo 2012-2013. En: Accepted: 2013-05-20T20:16:18Z [en línea], [Consulta: 1 marzo 2021]. Disponible en: <https://dspace.unl.edu.ec/handle/123456789/241>.

LUGARESI, C., et al. MediaPipe: A Framework for Perceiving and Processing Reality. , pp. 4.

MALINA, R.M. Motor Development during Infancy and Early Childhood: Overview and Suggested Directions for Research. *International Journal of Sport and Health Science*, vol. 2, pp. 50-66. DOI 10.5432/ijshs.2.50.

MARR, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* [en línea]. S.l.: The MIT Press. [Consulta: 5 febrero 2020]. ISBN 978-0-262-28961-0. Disponible en: <https://mitpress.universitypressscholarship.com/view/10.7551/mitpress/9780262514620.001.0001/upso-9780262514620>.

MCCALL, J. Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, vol. 184, no. 1, pp. 205-222. ISSN 0377-0427. DOI 10.1016/j.cam.2004.07.034.

MULLEN, R.J., et al. Ant algorithms for image feature extraction. *Sistemas expertos con aplicaciones: una revista internacional de*, vol. 40, no. 11, pp. 4315-4332. ISSN 0957-4174. DOI 10.1016/j.eswa.2013.01.020.

MURALIDHARAN, R. Early childhood stimulation. *The Indian Journal of Pediatrics*, vol. 59, no. 6, pp. 669-674. ISSN 0019-5456, 0973-7693. DOI 10.1007/BF02859396.

NANTHAGOPAL, A.P.; & SUKANESH, R. Wavelet statistical texture features-based segmentation and classification of brain computed tomography images. *IET Image Processing*, vol. 7, no. 1, pp. 25-32. ISSN 1751-9667. DOI 10.1049/iet-ipr.2012.0073.

NELSON, C.A.; & BOSQUET, M. Neurobiology of fetal and infant development: Implications for infant mental health. *Handbook of infant mental health, 2nd ed.* New York, NY, US: The Guilford Press, pp. 37-59. ISBN 978-1-57230-515-1.

NEWELL, A., et al. Stacked Hourglass Networks for Human Pose Estimation. En: B. LEIBE, J. MATAS, N. SEBE y M. WELLING (eds.), *Computer Vision – ECCV 2016* [en línea]. Cham: Springer International Publishing, Lecture Notes in Computer Science, pp. 483-499. [Consulta:

2153-DBRA-UPT-2022

21 abril 2021]. ISBN 978-3-319-46483-1. Disponible en: http://link.springer.com/10.1007/978-3-319-46484-8_29.

OPENCV-PYTHON. Wrapper package for OpenCV python bindings. [en línea], 2021. C++, Python. S.l.: s.n. [Consulta: 10 mayo 2021]. Disponible en: <https://github.com/skvark/opencv-python>.

PALMER, J. Social and Emotional Development in Early Learning Settings. , pp. 9.

PÉREZ, D.S.; & BROMBERG, F. Segmentación de imágenes en viñedos para la medición autónoma de variables vitícolas. *XVIII Congreso Argentino de Ciencias de la Computación* [en línea]. S.l.: s.n., [Consulta: 12 mayo 2020]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/23598>.

PI, R. Teach, Learn, and Make with Raspberry Pi. *Raspberry Pi* [en línea]. [Consulta: 5 mayo 2021]. Disponible en: <https://www.raspberrypi.org/>.

PIEK, J.P., et al. The role of early fine and gross motor development on later motor and cognitive ability. *Human Movement Science*, vol. 27, no. 5, pp. 668-681. ISSN 0167-9457. DOI 10.1016/j.humov.2007.11.002.

POSE. Mediapipe [en línea], 2021. [Consulta: 21 abril 2021]. Disponible en: <https://google.github.io/mediapipe/solutions/pose.html>.

PRINCE, S.J.D. Computer Vision: Models, Learning, and Inference. S.l.: Cambridge University Press. ISBN 978-1-107-01179-3.

RECALDE, D. La estimulación temprana y su influencia en el desarrollo psicomotor de los niños de 2 a 3 años del centro de estimulación prenatal y temprana baby place de Guayaquil en el periodo lectivo 2017 - 2018 [en línea]. Ecuador: Universidad Laica Vicente Rocafuerte de Guayaquil. [Consulta: 26 febrero 2021]. Disponible en: <http://repositorio.ulvr.edu.ec/handle/44000/2070>.

RICHLAND, L., et al. Cognitive Development. . S.l.: s.n., pp. 143-146. ISBN 978-1-4833-4627-4.

RICHTER, D. Flash Memories [en línea]. Dordrecht: Springer Netherlands. [Consulta: 28 abril 2021]. Springer Series in Advanced Microelectronics. ISBN 978-94-007-6081-3. Disponible en: <http://link.springer.com/10.1007/978-94-007-6082-0>.

ROBERTS, L.G. Machine perception of three-dimensional solids [en línea]. Thesis. S.l.: Massachusetts Institute of Technology. [Consulta: 5 febrero 2020]. Disponible en: <https://dspace.mit.edu/handle/1721.1/11589>.

SEEFELDT, V. Developmental motor patterns: Implications for elementary school physical education. , vol. 36, no. 6, pp. 314-323.

SHIJIE, J., et al. Research on data augmentation for image classification based on convolution neural networks. *2017 Chinese Automation Congress (CAC)* [en línea]. Jinan: IEEE, pp. 4165-4170. [Consulta: 14 abril 2021]. ISBN 978-1-5386-3524-7. DOI 10.1109/CAC.2017.8243510. Disponible en: <http://ieeexplore.ieee.org/document/8243510/>.

SHONKOFF, J.P. Investment in Early Childhood Development Lays the Foundation for a Prosperous and Sustainable Society. , pp. 4.

SNYDER, D. Early Childhood Development— Basic Concepts. [en línea]. Baltimore: Catholic Relief Services. [Consulta: 25 noviembre 2020]. Disponible en: <https://www.crs.org/sites/default/files/crs-files/module-two-resource-guide.pdf>.

STRICKLAND, J. *Data Analytics Using Open-Source Tools*. S.l.: Lulu.com. ISBN 978-1-365-21384-7.

SUGGATE, S., et al. Children's fine motor skills in kindergarten predict reading in grade 1. *Early Childhood Research Quarterly*, vol. 47, pp. 248-258. ISSN 1873-7706(Electronic),0885-2006(Print). DOI 10.1016/j.ecresq.2018.12.015.

TENSORFLOW/TFJS-MODELS. GitHub [en línea], 2019. [Consulta: 21 abril 2021]. Disponible en: <https://github.com/tensorflow/tfjs-models>.

TOMPSON, J., et al. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. , pp. 9.

TOSHEV, A.; & SZEGEDY, C. DeepPose: Human Pose Estimation via Deep Neural Networks., pp. 8.

ULRICH, D. *Test of gross motor development-2*. S.l.: s.n.

VERGARA, O., et al. Automatic Product Quality Inspection Using Computer Vision Systems. En: J.L. GARCÍA, A.A. MALDONADO y G. CORTES (eds.), *Lean Manufacturing in the Developing World: Methodology, Case Studies and Trends from Latin America*. Cham: Springer International Publishing, pp. 135-156. ISBN 978-3-319-04951-9.

VYGOTSKY, L.S. *Mind in Society: The Development of Higher Psychological Processes*. S.l.: Harvard University Press. ISBN 978-0-674-07668-6.

WALCZAK, S.; & CERPA, N. Artificial Neural Networks. En: R.A. MEYERS (ed.), *Encyclopedia of Physical Science and Technology (Third Edition)* [en línea]. New York:

Academic Press, pp. 631-645. [Consulta: 8 febrero 2021]. ISBN 978-0-12-227410-7. Disponible en: <https://www.sciencedirect.com/science/article/pii/B0122274105008371>.

WANDELL, B.A., et al. Common principles of image acquisition systems and biological vision. *Proceedings of the IEEE*, vol. 90, no. 1, pp. 5-17. ISSN 1558-2256. DOI 10.1109/5.982401.

WEEDIA BRAND SERVICE. Inteligencia artificial débil vs fuerte: ¿hasta dónde llega la tecnología? (infografía). *Ecosistema Huawei* [en línea]. [Consulta: 18 mayo 2020]. Disponible en: <https://ecosistemahuawei.xataka.com/inteligencia-artificial-debil-vs-fuerte-donde-llega-otra-infografia/>.

NUMPY. NumPy v1.20 Manual. [en línea], 2021. [Consulta: 10 mayo 2021]. Disponible en: <https://numpy.org/doc/stable/user/whatisnumpy.html>.

XIAO, B.; & WANG, G. Generic radial orthogonal moment invariants for invariant image recognition. *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1002-1008. ISSN 1047-3203. DOI 10.1016/j.jvcir.2013.06.017.

ZADEH, L. Fuzzy logic a personal perspective. *Fuzzy sets and Systems*, vol. 281, pp. 4-20. DOI <https://doi.org/10.1016/j.fss.2015.05.009>.

ZHANG, K. Spirituality and early childhood special education: exploring a ‘forgotten’ dimension. *International Journal of Children’s Spirituality*, vol. 17, no. 1, pp. 39-49. ISSN 1364-436X. DOI 10.1080/1364436X.2012.660475.

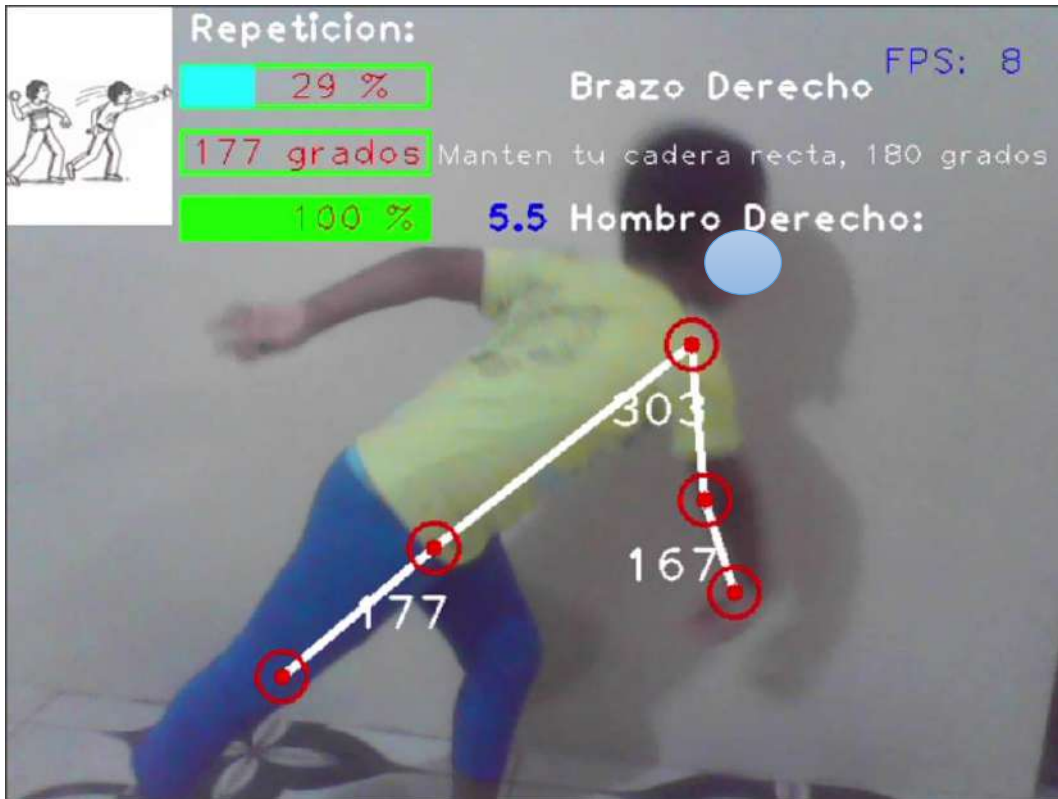


ANEXOS

ANEXO A: DISPOSITIVO RASPBERRY CON LOS PERIFÉRICOS CONECTADOS Y EJECUTÁNDOSE



ANEXO B: PRUEBAS EJECUTADAS POR NIÑO DE 5 AÑOS



ANEXO C: PRUEBAS EJECUTADAS POR NIÑA DE 5 AÑOS

Repetición: 100 % 9.5 Cadera Derecha FPS: 8
100 % 12.5 Cadera Izquierda
92 % 0.5 Derecho:
Lo Lograste
171
215
189

Repetición: 100 % 4.0 Brazo Derecho FPS: 8
179 grados Manten tu codo correcta, 180 grados
154
179

**ANEXO D: PRUEBAS REALIZADAS A NIÑOS EN UNIDAD EDUCATIVA
QUISAPINCHA**



ANEXO E: PANTALLA INICIAL DE LA INTERFACE DEL PROGRAMA DE ESTIMULACIÓN MOTRIZ

DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO ELECTRÓNICO DE ESTIMULACIÓN TEMPRANA MOTRIZ GRUESA PARA NIÑOS DE 4 A 5 AÑOS EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL



INDICACIONES:

- 1.- **Esoger el ejercicio correspondiente en cada imagen y el ejercicio objetivo.**
- 2.- **Esperar 15 segundos hasta que se inicie el programa**
- 3.- **Ubicarse frente a la cámara e iniciar el ejercicio**



0. nose	17. left_pinky
1. left_eye_inner	18. right_pinky
2. left_eye	19. left_index
3. left_eye_outer	20. right_index
4. right_eye_inner	21. left_thumb
5. right_eye	22. right_thumb
6. right_eye_outer	23. left_hip
7. left_ear	24. right_hip
8. right_ear	25. left_knee
9. mouth_left	26. right_knee
10. mouth_right	27. left_ankle
11. left_shoulder	28. right_ankle
12. right_shoulder	29. left_heel
13. left_elbow	30. right_heel
14. right_elbow	31. left_foot_index
15. left_wrist	32. right_foot_index
16. right_wrist	

Primer Ejercicio



Segundo Ejercicio



Tercer Ejercicio



Cuarto Ejercicio



Quinto Ejercicio



Sexto Ejercicio



Realizado por: Cristian Fajardo Wilmer Yanchapanta

ANEXO F: OFICIO UNIDAD EDUCATIVA QUISAPINCHA



ESPOCH

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

Ambato, 17 de mayo de 2022

MSc.

Ernesto López

RECTOR (E) DE LA UNIDAD EDUCATIVA QUISAPINCHA

Presente. -

De mi consideración:

Reciba un cordial y atento saludo deseándole el mejor de los éxitos en las funciones que tan acertadamente dirige.

Por medio del presente me dirijo hacia usted para que me permita realizar pruebas en niños de 4 a 5 años correspondiente a inicial 2 con el fin de validar el dispositivo de la tesis llamada **“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO ELECTRÓNICO DE ESTIMULACIÓN TEMPRANA MOTRIZ GRUESA PARA NIÑOS DE 4 A 5 AÑOS EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL”**.

Seguro de contar con su valiosa aceptación, anticipo mis agradecimientos.

Atentamente,

Wilmer Javier Yanchapanta Tamaquiza

Estudiante de Ingeniería en Electrónica y Automatización

CI: 1804874822

Cel: 0983819556



Dirección: Panamericana Sur km 1 1/2,

www.espoch.edu.ec

Teléfono: 593 (03) 2 998200

Código Postal: EC060155

Escaneado con CamScanner

2153-DBRA-UPT-2022

ANEXO G: PROGRAMACIÓN EN PHYTON

```
import cv2
import numpy as np
import time
import configuracionPose as pm

cap = cv2.VideoCapture(0) # Encender video desde webcam
#cap = cv2.VideoCapture("BD_videos/prueba3.mp4") # Encender video desde Carpeta precargado

detector = pm.poseDetector()
count=0
count2=0
dir=0
dir2=0

pTime = 0

while True:
    success, img= cap.read() # Activar para video
    img = detector.findPose(img) # Activar para vid
    #img = cv2.imread("BD_videos/configuracion2.jpg") # Activar para imagen
    #img = cv2.resize(img, (600,400)) # Activar para video, Configurar Tamano
    img = detector.findPose(img, False) # Si se desea eliminar otras zonas solo agregar (img,
    False)
    lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos
    print(lmList)
    if len(lmList) !=0:

        # _____ Escoger los 3 puntos de cada articulacion -----
        #Brazo Derecho
        angle = detector.findAngle(img, 12, 14, 16)
        #Pierza Izquierda
        angle2 = detector.findAngle(img, 24, 26, 28)

        #Acciones con los angulos para las barras porcentuales
        porcentaje = np.interp(angle, (140, 72), (0, 100))
        barra = np.interp(angle, (72, 140), (105, 255))
        porcentaje2 = np.interp(angle2, (200, 280), (0, 100))
        barra2 = np.interp(angle2, (200, 280), (105, 255))
        #print(angle, porcentaje)

        #Numero de repeticiones
        #Articulacion 1 Repeticion
        color = (255, 255, 0)
        if porcentaje == 100:
            color = (0, 255, 0)
            if dir == 0:
                count += 0.5
                dir = 1
            if porcentaje == 0:
                color = (0, 255, 0)
            if dir==1:
                count += 0.5
                dir = 0

        #Articulacion 2 Repeticion
        if porcentaje2 == 100:
            if dir2 == 0:
                count2 += 0.5
```

```

dir2 = 1
if porcentaje2 == 0:
    if dir2==1:
        count2 += 0.5
        dir2 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (260, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)
# Barra de progreso Dibujar
#Articulacion 1
cv2.putText(img, "Arti 1: ", (20, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255), 2)
cv2.rectangle(img, (105,60), (255, 35), (0, 255, 0), 2)
cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)
#Articulacion 2
cv2.putText(img, "Arti 2: ", (20, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255), 2)
cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
cv2.rectangle(img, (105, 100), (int(barra2), 75), (255, 255, 0), cv2.FILLED)
cv2.putText(img, f'{int(porcentaje2)} %', (170, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Premios de metas
if count >= 10 :
    cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)

if count2 >= 10 :
    cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)

#Reiniciar conteo
if lmList[13][2] < lmList[11][2] !=0:
    count = 0
    count2 = 0

#Acciones

# Imprimir FPS
cTime = time.time()
fps = 1 / (cTime - pTime)
pTime = cTime

# Informacion en Ventana, FPS
cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)

cv2.imshow("Ejercicio 1 Correr", img)
cv2.waitKey(30)

```

```

import cv2
import mediapipe as mp
import time
import math

class poseDetector():

    def __init__(self, mode=False, upBody=False, smooth=True,
                 detectionCon=0.5, trackCon=0.5):

        self.mode = mode
        self.upBody = upBody
        self.smooth = smooth
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpDraw = mp.solutions.drawing_utils
        self.mpPose = mp.solutions.pose
        self.pose = self.mpPose.Pose(self.mode, self.upBody, self.smooth,
                                     self.detectionCon, self.trackCon)

    def findPose(self, img, draw=False):          #Detecta todo el cuerpo
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.pose.process(imgRGB)
        if self.results.pose_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, self.results.pose_landmarks,
                self.mpPose.POSE_CONNECTIONS,
                self.mpDraw.DrawingSpec (color=(245,117,66), thickness=2,
                circle_radius=4),
                self.mpDraw.DrawingSpec (color=(245,66,230), thickness=2, circle_radius=2)
                )

        return img

    def findPosition(self, img, draw=True):          #Dibuja en los puntos relevantes
        self.lmList = []
        if self.results.pose_landmarks:
            for id, lm in enumerate(self.results.pose_landmarks.landmark):
                h, w, c = img.shape
                #print(id, lm)                                #Imprime, id del punto relevante y su posicion
                cx, cy = int(lm.x * w), int(lm.y * h)          #Comprime en un arreglo, ID mas el punto en
                X Y
                self.lmList.append([id, cx, cy])
                if draw:
                    cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)

        return self.lmList

    def findAngle(self, img, p1, p2, p3, draw=True):

        # Obtener los puntos relevantes 3 para poder identificarlos
        x1, y1 = self.lmList[p1][1:]
        x2, y2 = self.lmList[p2][1:]
        x3, y3 = self.lmList[p3][1:]

        # Calcula el angulo entre los puntos escogidos
        angle = math.degrees(math.atan2(y3 - y2, x3 - x2) -
                               math.atan2(y1 - y2, x1 - x2))

```

```

if angle < 0 : angle += 360

# print(angulo)                                #imprime los angulos

# Draw
if draw:
    cv2.line(img, (x1, y1), (x2, y2), (255, 255, 255), 3)
    cv2.line(img, (x3, y3), (x2, y2), (255, 255, 255), 3)
    cv2.circle(img, (x1, y1), 5, (0, 0, 255), cv2.FILLED)
    cv2.circle(img, (x1, y1), 15, (0, 0, 255), 2)
    cv2.circle(img, (x2, y2), 5, (0, 0, 255), cv2.FILLED)
    cv2.circle(img, (x2, y2), 15, (0, 0, 255), 2)
    cv2.circle(img, (x3, y3), 5, (0, 0, 255), cv2.FILLED)
    cv2.circle(img, (x3, y3), 15, (0, 0, 255), 2)
    cv2.putText(img, str(int(angle)), (x2 - 50, y2 + 50),
                cv2.FONT_HERSHEY_PLAIN, 2, (255, 255, 255), 2)
return angle

def main():
    cap = cv2.VideoCapture(0)
    pTime = 0
    detector = poseDetector()
    while True:
        success, img = cap.read()
        img = detector.findPose(img)
        #lmList = detector.findPosition(img, draw=True)
        #if len(lmList) != 0:
            #print(lmList[14])
            #cv2.circle(img, (lmList[14][1], lmList[14][2]), 15, (245,117,66), cv2.FILLED)

        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

        cv2.putText(img, str(int(fps)), (70, 50), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 0), 3)

        cv2.imshow("Pruebas", img)
        cv2.waitKey(1)

if __name__ == "__main__":
    main()
#!/usr/bin/python
import sys
import os
from tkinter import *
from PIL import Image
from PIL import ImageTk

#crear las imagenes
imagenAnchuraMaxima=150
imagenAltruaMaxima=150
#Abrir imagen
img=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\portada.png')
img2=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\indicaciones.png')
img3=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\pose.png')
imgje1=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\0.png')
imgje2=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\11.png')

```



```

imgeje3=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\22.png')
imgeje4=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\4.png')
imgeje5=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\5.png')
imgeje6=Image.open(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\6.png')
#modificar tamaño para ajustar a la ventana
img.thumbnail((imagenAnchuraMaxima,imagenAltruaMaxima), Image.ANTIALIAS)
img2.thumbnail((330,90), Image.ANTIALIAS)
img3.thumbnail((300,160), Image.ANTIALIAS)
imgeje1.thumbnail((100,132), Image.ANTIALIAS)
imgeje2.thumbnail((100,132), Image.ANTIALIAS)
imgeje3.thumbnail((100,132), Image.ANTIALIAS)
imgeje4.thumbnail((100,132), Image.ANTIALIAS)
imgeje5.thumbnail((100,132), Image.ANTIALIAS)
imgeje6.thumbnail((100,132), Image.ANTIALIAS)

#crear la ventana
ventana=Tk()
ventana.title("Tesis Estimulacion Motriz")
ventana.geometry("800x700")
ventana.configure(bg='white')
ventana.iconbitmap(r'C:\Users\sianw\Desktop\ProyectoPajiYT\img\Espoch.ico')

#Titulos Principales
TituloPrincipal01=Label(ventana, text="DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO
ELECTRÓNICO DE ", font='Helvetica 10 bold', bg='white').pack()
TituloPrincipal=Label(ventana, text="ESTIMULACIÓN TEMPRANA MOTRIZ GRUESA PARA
NIÑOS DE 4 A 5 AÑOS", font='Helvetica 10 bold', bg='white').pack()
TituloPrincipal1=Label(ventana, text="EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL",
font='Helvetica 10 bold', bg='white').pack()

#Realizado por
TituloPrincipal2=Label(ventana, text="Realizado por: ")
TituloPrincipal2.place(x=10, y=665, width=120, height=40)
TituloPrincipal2.configure(bg='white')
TituloPrincipal3=Label(ventana, text="Cristian Fajardo")
TituloPrincipal3.place(x=110, y=665, width=120, height=40)
TituloPrincipal3.configure(bg='white')
TituloPrincipal4=Label(ventana, text="Wilmer Yanchapanta")
TituloPrincipal4.place(x=225, y=665, width=120, height=40)
TituloPrincipal4.configure(bg='white')

#Convertir la imagen en un objeto
tkimage= ImageTk.PhotoImage(img)
tkimage2= ImageTk.PhotoImage(img2)
tkimage3= ImageTk.PhotoImage(img3)
tkimage33= ImageTk.PhotoImage(imgeje1)
tkimage4= ImageTk.PhotoImage(imgeje2)
tkimage5= ImageTk.PhotoImage(imgeje3)
tkimage6= ImageTk.PhotoImage(imgeje4)
tkimage7= ImageTk.PhotoImage(imgeje5)
tkimage8= ImageTk.PhotoImage(imgeje6)
#montar imagen en un label
label=Label(ventana, image=tkimage, width=imagenAnchuraMaxima, height=imagenAltruaMaxima)
label.configure(bg='white')
label.place(x=20, y=100)
labelimg2=Label(ventana, image=tkimage2, width=330, height=90)
labelimg2.configure(bg='white')
labelimg2.place(x=180, y=120)
labelimg3=Label(ventana, image=tkimage3, width=300, height=160)

```

```

labelimg3.configure(bg='white')
labelimg3.place(x=500, y=85)
#labes para las imagenes de los ejercicios
#ejercicio1
labelimgejer1=Label(ventana, image=tkimage33, width=100, height=130)
labelimgejer1.configure(bg='white')
labelimgejer1.place(x=100, y=330)
#ejercicio2
labelimgejer2=Label(ventana, image=tkimage4, width=100, height=130)
labelimgejer2.configure(bg='white')
labelimgejer2.place(x=350, y=330)
#ejercicio3
labelimgejer3=Label(ventana, image=tkimage5, width=100, height=130)
labelimgejer3.configure(bg='white')
labelimgejer3.place(x=627, y=330)
#ejercicio4
labelimgejer4=Label(ventana, image=tkimage6, width=100, height=130)
labelimgejer4.configure(bg='white')
labelimgejer4.place(x=100, y=530)
#ejercicio5
labelimgejer5=Label(ventana, image=tkimage7, width=100, height=130)
labelimgejer5.configure(bg='white')
labelimgejer5.place(x=350, y=530)
#ejercicio6
labelimgejer6=Label(ventana, image=tkimage8, width=100, height=130)
labelimgejer6.configure(bg='white')
labelimgejer6.place(x=627, y=530)

```

```

def Primer_Ejercicio():
    os.system('python correr_Eje1.py')

```

```

def Segundo_Ejercicio():
    os.system('python Pelota_Eje2.py')

```

```

def Tercer_Ejercicio():
    os.system('python fortnite_Eje3.py')

```

```

def Cuarto_Ejercicio():
    os.system('python fortnite_Eje3.py')

```

```

def Quinto_Ejercicio():
    os.system('python fortnite_Eje3.py')

```

```

def Sexto_Ejercicio():
    os.system('python fortnite_Eje3.py')

```

```

B = Button(ventana, text="Primer Ejercicio", command=Primer_Ejercicio)
B.place(x=80, y=280, width=120, height=40)

```

```

C = Button(ventana, text="Segundo Ejercicio", command=Segundo_Ejercicio)
C.place(x=340, y=280, width=120, height=40)

```

```

D = Button(ventana, text="Tercer Ejercicio", command=Tercer_Ejercicio)
D.place(x=620, y=280, width=120, height=40)

```

```

E = Button(ventana, text="Cuarto Ejercicio", command=Primer_Ejercicio)
E.place(x=80, y=480, width=120, height=40)

```

```
F = Button(ventana, text="Quinto Ejercicio", command=Primer_Ejercicio)
F.place(x=340, y=480, width=120, height=40)
```

```
G = Button(ventana, text="Sexto Ejercicio", command=Primer_Ejercicio)
G.place(x=620, y=480, width=120, height=40)
```

```
ventana.mainloop()
```

```
# This is a sample Python script.
```

```
# Press Mayús+F10 to execute it or replace it with your code.
```

```
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
```

```
def print_hi(name):
```

```
    # Use a breakpoint in the code line below to debug your script.
```

```
    print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
```

```
# Press the green button in the gutter to run the script.
```

```
if __name__ == '__main__':
```

```
    print_hi('PyCharm')
```

```
# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

```
import os
```

```
import cv2
```

```
import numpy as np
```

```
import time
```

```
import configuracionPose as pm
```

```
from playsound import playsound
```

```
# Encender video desde webcam
```

```
cap = cv2.VideoCapture(0)
```

```
# detector de pose
```

```
detector = pm.poseDetector()
```

```
#variables contador de repeticiones
```

```
count=0
```

```
count2=0
```

```
dir=0
```

```
dir2=0
```

```
#variable para contador fps
```

```
pTime = 0
```

```
#overlay de imagen
```

```
folderPath = "img"
```

```
myList = os.listdir(folderPath)
```

```
print(myList)
```

```
overlayList = []
```

```
for imPath in myList:
```

```
    image = cv2.imread(f'{folderPath}/{imPath}')
```

```
    overlayList.append(image)
```

```
print(len(overlayList))
```

```
while True:
```

```
    success, img= cap.read()
```

```
    # Activar para video
```

```

img = detector.findPose(img) # Activar para vid
 # Activar para imagen
 # Activar para video, Configurar Tamano
img = detector.findPose(img, False) # Si se desea eliminar otras zonas solo agregar (img,
False)
lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos

print(lmList)
if len(lmList) !=0:

    # _____ Escoger los 3 puntos de cada articulacion -----
    #Pierna Derecho
    angle = detector.findAngle(img, 12, 24, 26)
    #Pie Derecho
    angle2 = detector.findAngle(img, 24, 26, 28)

    # Acciones con los angulos para las barras porcentuales
    porcentaje = np.interp(angle, (97, 140), (0, 100))
    barra = np.interp(angle, (97, 140), (105, 255))
    porcentaje2 = np.interp(angle2, (190, 284), (0, 100))
    barra2 = np.interp(angle2, (190, 284), (105, 255))

    #print(angle, porcentaje)

#Numero de repeticiones
#Articulacion 1 Repeticion
color = (255, 255, 0)
if porcentaje == 100:
    color = (0, 255, 0)
    if dir == 0:
        count += 0.5
        dir = 1
    if porcentaje == 0:
        color = (0, 255, 0)
        if dir==1:
            count += 0.5
            dir = 0

# Articulacion 2 Repeticion
color2 = (255, 255, 0)
if porcentaje2 == 100:
    color2 = (0, 255, 0)
    if dir2 == 0:
        count2 += 0.5
        dir2 = 1
    if porcentaje2 == 0:
        color2 = (0, 255, 0)
        if dir2 == 1:
            count2 += 0.5
            dir2 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)
# Barra de progreso Dibujar

```

```

    #Articulacion 1
    cv2.putText(img, "Muslo Derecho", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
    cv2.rectangle(img, (105, 60), (255, 35), (0, 255, 0), 2)
    cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
    cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
    cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

    # Articulacion 2
    cv2.putText(img, "Rodilla Derecha", (340, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
    cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
    cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)
    cv2.putText(img, f'{int(porcentaje2)} %', (170, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
    cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

    # Premios de metas
    if count >= 10 and count2 >= 10:
        cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)
        playsound('mario-bros vida.mp3')

    #Reiniciar conteo
    if lmList[13][2] < lmList[11][2] !=0:
        count = 0
        count2 = 0

    #Acciones

    # Imprimir FPS
    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime

    # Informacion en Ventana, FPS
    cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
    cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

    # Imprimir imagen

    img[0:132, 0:100] = overlayList[0]

    #ABRE LA VENTANA PRINCIPAL
    cv2.imshow("Ejercicio 1 Correr en el propio sitio", img)

    cv2.waitKey(30)
import os
import cv2
import numpy as np
import time
import configuracionPose as pm
from playsound import playsound

    # Encender video desde webcam

```

```

cap = cv2.VideoCapture(0)
# detector de pose
detector = pm.poseDetector()
#variables contador de repeticiones
count=0
count2=0
dir=0
dir2=0
#variable para contador fps
pTime = 0

#overlay de imagen
folderPath = "img"
myList = os.listdir(folderPath)
print(myList)
overlayList = [3]
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)
print(len(overlayList))

while True:
    success, img= cap.read()                # Activar para video
    img = detector.findPose(img)            # Activar para vid
    #img = cv2.imread("BD_videos/configuracion2.jpg") # Activar para imagen
    #img = cv2.resize(img, (600,400))        # Activar para video, Configurar Tamano
    img = detector.findPose(img, False)    # Si se desea eliminar otras zonas solo agregar (img,
False)
    lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos

    print(lmList)
    if len(lmList) !=0:

        # _____ Escoger los 3 puntos de cada articulacion -----
        #Brazo Derecho
        angle = detector.findAngle(img, 12, 14, 16)
        #Hombro, cadera rodilla Parte derecha del cuerpo
        angle2 = detector.findAngle(img, 12, 24, 26)

        #Acciones con los angulos para las barras porcentuales
        porcentaje = np.interp(angle, (97, 140), (0, 100))
        barra = np.interp(angle, (97, 140), (105, 255))
        porcentaje2 = np.interp(angle2, (190, 284), (0, 100))
        barra2 = np.interp(angle2, (190, 284), (105, 255))
        #print(angle, porcentaje)

    #Numero de repeticiones
    #Articulacion 1 Repeticion
    color = (255, 255, 0)
    if porcentaje == 100:
        color = (0, 255, 0)
        if dir == 0:
            count += 0.5
            dir = 1
    if porcentaje == 0:
        color = (0, 255, 0)
        if dir==1:
            count += 0.5

```

```

dir = 0

# Articulacion 2 Repeticion
color2 = (255, 255, 0)
if porcentaje2 == 100:
    color2 = (0, 255, 0)
    if dir2 == 0:
        count2 += 0.5
        dir2 = 1
    if porcentaje2 == 0:
        color2 = (0, 255, 0)
        if dir2 == 1:
            count2 += 0.5
            dir2 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)
# Barra de progreso Dibujar

#Articulacion 1
cv2.putText(img, "Brazo Derecho", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 60), (255, 35), (0, 255, 0), 2)
cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Articulacion 2
cv2.putText(img, "Manten tu cadera recta, 180 grados", (260, 95),
cv2.FONT_HERSHEY_PLAIN, 1.2, (255, 255, 255), 1)
cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)
cv2.putText(img, f'{int(angle2)} grados', (110, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
#cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Premios de metas
if count >= 10:
    cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)
    playsound('mario-bros vida.mp3')

# Penalización por no tener la cadera recta
if angle2>=0 and angle2<=170:
    count = 0
    cv2.putText(img, "Manten tu cadera recta", (100, 200), cv2.FONT_HERSHEY_PLAIN, 1.5,
(255, 255, 255), 2)
    playsound('sonidoincorrecto.mp3')

#Reiniciar conteo
if lmList[13][2] < lmList[11][2] !=0:
    count = 0
    count2 = 0

#Acciones

```

```

    # Imprimir FPS
    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime

    # Informacion en Ventana, FPS
    cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
    cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

    # Imprimir imagen

    img[0:132, 0:100] = overlayList[3]

    #ABRE LA VENTANA PRINCIPAL
    cv2.imshow("Ejercicio 2 Pelota", img)

cv2.waitKey(30)

import os
import cv2
import numpy as np
import time
import configuracionPose as pm
from playsound import playsound

# Encender video desde webcam
cap = cv2.VideoCapture(0)
# detector de pose
detector = pm.poseDetector()
#variables contador de repeticiones
count=0
count2=0
count3=0
dir=0
dir2=0
dir3=0
#variable para contador fps
pTime = 0

#overlay de imagen
folderPath = "img"
myList = os.listdir(folderPath)
print(myList)
overlayList = [4]
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)
print(len(overlayList))

while True:
    success, img= cap.read()
    img = detector.findPose(img)
    #img = cv2.imread("BD_videos/configuracion2.jpg")
    #img = cv2.resize(img, (600,400))
    img = detector.findPose(img, False)

```



```

lmList = detector.findPosition(img, False)           # Si se desea eliminar los dibujos en los puntos
print(lmList)
if len(lmList) !=0:

    # _____ Escoger los 3 puntos de cada articulacion -----
    #Cadera Derecha
    angle = detector.findAngle(img, 12, 24, 26)

    #Cadera Izquierda
    angle2 = detector.findAngle(img, 11, 23, 25)

    #Brazo Derecho
    angle3 = detector.findAngle(img, 12, 14, 16)

    #Acciones con los angulos
    porcentaje = np.interp(angle, (182, 195), (0, 100))
    barra = np.interp(angle, (182, 195), (105, 255))
    porcentaje2 = np.interp(angle2, (165, 173), (0, 100))
    barra2 = np.interp(angle2, (165, 173), (105, 255))
    porcentaje3 = np.interp(angle3, (72, 180), (0, 100))
    barra3 = np.interp(angle3, (72, 180), (105, 255))

    #print(angle, porcentaje)

#Numero de repeticiones
#Articulacion 1 Repeticion
color = (255, 255, 0)
if porcentaje == 100:
    color = (0, 255, 0)
    if dir == 0:
        count += 0.5
        dir = 1
    if porcentaje == 0:
        color = (0, 255, 0)
        if dir==1:
            count += 0.5
            dir = 0

#Articulacion 2 Repeticion
color2 = (255, 255, 0)
if porcentaje2 == 100:
    color2 = (0, 255, 0)
    if dir2 == 0:
        count2 += 0.5
        dir2 = 1
    if porcentaje2 == 0:
        color2 = (0, 255, 0)
        if dir2==1:
            count2 += 0.5
            dir2 = 0

# Articulacion 3 Repeticion
color3 = (255, 255, 0)
if porcentaje3 == 100:
    color3 = (0, 255, 0)
    if dir3 == 0:
        count3 += 0.5
        dir3 = 1
    if porcentaje3 == 0:
        color3 = (0, 255, 0)

```

```

    if dir3 == 1:
        count3 += 0.5
        dir3 = 0

    #print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

    # Informacion en Ventana, Numero de repeticiones
    cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)

    # Barra de progreso Dibujar
    #Articulacion 1
    cv2.putText(img, "Cadera Derecha ", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
    cv2.rectangle(img, (105,60), (255, 35), (0, 255, 0), 2)
    cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
    cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
    cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)
    #Articulacion 2
    cv2.putText(img, "Cadera Izquierda: ", (340, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
    cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
    cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)
    cv2.putText(img, f'{int(porcentaje2)} %', (170, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
    cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)
    # Articulacion 3
    cv2.putText(img, "Brazo Derecho: ", (340, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
    cv2.rectangle(img, (105, 140), (255, 115), (0, 255, 0), 2)
    cv2.rectangle(img, (105, 140), (int(barra3), 115), color3, cv2.FILLED)
    cv2.putText(img, f'{int(porcentaje3)} %', (170, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
    cv2.putText(img, f'{count3}', (290, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

    # Premios de metas
    if count >= 10 :
        cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)
        playsound('mario-bros vida.mp3')

    if count2 >= 10 :
        cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)

    #Reiniciar conteo
    if lmList[13][2] < lmList[11][2] !=0:
        count = 0
        count2 = 0
        count3 = 0

    #Acciones

    # Imprimir FPS
    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime

```

```

# Informacion en Ventana, FPS
cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

# Imprimir imagen
img[0:132, 0:100] = overlayList[4]

# ABRE LA VENTANA PRINCIPAL
cv2.imshow("Ejercicio 3 Fortnite", img)

cv2.waitKey(30)

import os
import cv2
import numpy as np
import time
import configuracionPose as pm
from playsound import playsound

# Encender video desde webcam
cap = cv2.VideoCapture(0)
# detector de pose
detector = pm.poseDetector()
#variables contador de repeticiones
count=0
count2=0
dir=0
dir2=0
#variable para contador fps
pTime = 0

#overlay de imagen
folderPath = "img"
myList = os.listdir(folderPath)
print(myList)
overlayList = [5]
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)
print(len(overlayList))

while True:
    success, img= cap.read() # Activar para video
    img = detector.findPose(img) # Activar para vid
    #img = cv2.imread("BD_videos/configuracion2.jpg") # Activar para imagen
    #img = cv2.resize(img, (600,400)) # Activar para video, Configurar Tamano
    img = detector.findPose(img, False) # Si se desea eliminar otras zonas solo agregar (img,
False)
    lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos

    print(lmList)
    if len(lmList) !=0:

        # _____ Escoger los 3 puntos de cada articulacion -----
        #Pierna Derecho
        angle = detector.findAngle(img, 12, 24, 26)

```

```

#Pie Derecho
angle2 = detector.findAngle(img, 24, 26, 28)

# Acciones con los angulos para las barras porcentuales
porcentaje = np.interp(angle, (173, 188), (0, 100))
barra = np.interp(angle, (173, 188), (105, 255))
porcentaje2 = np.interp(angle2, (198, 260), (0, 100))
barra2 = np.interp(angle2, (198, 260), (105, 255))

#print(angle, porcentaje)

#Numero de repeticiones
#Articulacion 1 Repeticion
color = (255, 255, 0)
if porcentaje == 100:
    color = (0, 255, 0)
    if dir == 0:
        count += 0.5
        dir = 1
    if porcentaje == 0:
        color = (0, 255, 0)
        if dir==1:
            count += 0.5
            dir = 0

# Articulacion 2 Repeticion
color2 = (255, 255, 0)
if porcentaje2 == 100:
    color2 = (0, 255, 0)
    if dir2 == 0:
        count2 += 0.5
        dir2 = 1
    if porcentaje2 == 0:
        color2 = (0, 255, 0)
        if dir2 == 1:
            count2 += 0.5
            dir2 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)
# Barra de progreso Dibujar

#Articulacion 1
cv2.putText(img, "Muslo Derecho", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 60), (255, 35), (0, 255, 0), 2)
cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Articulacion 2
cv2.putText(img, "Rodilla Derecha", (340, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)

```

```

cv2.putText(img, f'{int(porcentaje)} %', (170, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Premios de metas
if count >= 10 and count2 >= 10:
    cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)
    playsound('mario-bros vida.mp3')

#Reiniciar conteo
if lmList[13][2] < lmList[11][2] !=0:
    count = 0
    count2 = 0

#Acciones

# Imprimir FPS
cTime = time.time()
fps = 1 / (cTime - pTime)
pTime = cTime

# Informacion en Ventana, FPS
cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

# Imprimir imagen

img[0:132, 0:100] = overlayList[5]

#ABRE LA VENTANA PRINCIPAL
cv2.imshow("Ejercicio 4 Patear un Balon", img)

cv2.waitKey(30)

import os
import cv2
import numpy as np
import time
import configuracionPose as pm
from playsound import playsound

# Encender video desde webcam
cap = cv2.VideoCapture(0)
# detector de pose
detector = pm.poseDetector()
#variables contador de repeticiones
count=0
count2=0
count3=0
dir=0
dir2=0
dir3=0
#variable para contador fps
pTime = 0

```

```

#overlay de imagen
folderPath = "img"
myList = os.listdir(folderPath)
print(myList)
overlayList = [6]
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)
print(len(overlayList))

while True:
    success, img= cap.read()                # Activar para video
    img = detector.findPose(img)            # Activar para vid
    #img = cv2.imread("BD_videos/configuracion2.jpg") # Activar para imagen
    #img = cv2.resize(img, (600,400))        # Activar para video, Configurar Tamano
    img = detector.findPose(img, False)     # Si se desea eliminar otras zonas solo agregar (img,
False)
    lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos

    print(lmList)
    if len(lmList) !=0:

        # _____ Escoger los 3 puntos de cada articulacion -----
        #Brazo Derecho
        angle = detector.findAngle(img, 12, 14, 16)
        #Hombro, cadera rodilla Parte derecha del cuerpo
        angle2 = detector.findAngle(img, 12, 24, 26)
        # Brazo Derecho
        angle3 = detector.findAngle(img, 24, 12, 14)

        #Acciones con los angulos para las barras porcentuales
        porcentaje = np.interp(angle, (149, 210), (0, 100))
        barra = np.interp(angle, (149, 210), (105, 255))
        porcentaje2 = np.interp(angle2, (190, 284), (0, 100))
        barra2 = np.interp(angle2, (190, 284), (105, 255))
        porcentaje3 = np.interp(angle3, (72, 180), (0, 100))
        barra3 = np.interp(angle3, (72, 180), (105, 255))
        #print(angle, porcentaje)

        #Numero de repeticiones
        #Articulacion 1 Repeticion
        color = (255, 255, 0)
        if porcentaje == 100:
            color = (0, 255, 0)
            if dir == 0:
                count += 0.5
                dir = 1
            if porcentaje == 0:
                color = (0, 255, 0)
                if dir==1:
                    count += 0.5
                    dir = 0

        # Articulacion 2 Repeticion
        color2 = (255, 255, 0)
        if porcentaje2 == 100:
            color2 = (0, 255, 0)

```

```

    if dir2 == 0:
        count2 += 0.5
        dir2 = 1
if porcentaje2 == 0:
    color2 = (0, 255, 0)
    if dir2 == 1:
        count2 += 0.5
        dir2 = 0

# Articulacion 3 Repeticion
color3 = (255, 255, 0)
if porcentaje3 == 100:
    color3 = (0, 255, 0)
    if dir3 == 0:
        count3 += 0.5
        dir3 = 1
if porcentaje3 == 0:
    color3 = (0, 255, 0)
    if dir3 == 1:
        count3 += 0.5
        dir3 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)

# Barra de progreso Dibujar

#Articulacion 1
cv2.putText(img, "Brazo Derecho", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 60), (255, 35), (0, 255, 0), 2)
cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje)} %', (170,55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
#cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Articulacion 2
cv2.putText(img, "Manten tu cadera recta, 180 grados", (260, 95),
cv2.FONT_HERSHEY_PLAIN, 1.2, (255, 255, 255), 1)
cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)
cv2.putText(img, f'{int(angle2)} grados', (110, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
#cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Articulacion 3
cv2.putText(img, "Hombro Derecho: ", (340, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (255,
255, 255), 2)
cv2.rectangle(img, (105, 140), (255, 115), (0, 255, 0), 2)
cv2.rectangle(img, (105, 140), (int(barra3), 115), color3, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje3)} %', (170, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
cv2.putText(img, f'{count3}', (290, 135), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Premios de metas
if count3 >= 10:
    cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)

```

```

    playsound('mario-bros vida.mp3')

# Penalización por no tener la cadera recta
    if angle2>=0 and angle2<=170:
        count = 0
        cv2.putText(img, "Manten tu cadera recta", (100, 200), cv2.FONT_HERSHEY_PLAIN, 1.5,
(255, 255, 255), 2)
        playsound('sonidoincorrecto.mp3')

#Reiniciar conteo
    if lmList[13][2] < lmList[11][2] !=0:
        count = 0
        count2 = 0
        count3 = 0

#Acciones

    # Imprimir FPS
    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime

    # Informacion en Ventana, FPS
    cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
    cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

    # Imprimir imagen

    img[0:132, 0:100] = overlayList[6]

    #ABRE LA VENTANA PRINCIPAL
    cv2.imshow("Ejercicio 5 Lanzamiento de Pelota con la mano derecha", img)

    cv2.waitKey(30)
import os
import cv2
import numpy as np
import time
import configuracionPose as pm
from playsound import playsound

# Encender video desde webcam
cap = cv2.VideoCapture(0)
# detector de pose
detector = pm.poseDetector()
#variables contador de repeticiones
count=0
count2=0
dir=0
dir2=0
#variable para contador fps
pTime = 0

#overlay de imagen
folderPath = "img"
myList = os.listdir(folderPath)
print(myList)

```



```

overlayList = [7]
for imPath in myList:
    image = cv2.imread(f'{folderPath}/{imPath}')
    overlayList.append(image)
print(len(overlayList))

while True:
    success, img= cap.read()                # Activar para video
    img = detector.findPose(img)            # Activar para vid
    #img = cv2.imread("BD_videos/configuracion2.jpg") # Activar para imagen
    #img = cv2.resize(img, (600,400)) # Activar para video, Configurar Tamano
    img = detector.findPose(img, False)    # Si se desea eliminar otras zonas solo agregar (img,
False)
    lmList = detector.findPosition(img, False) # Si se desea eliminar los dibujos en los puntos

    print(lmList)
    if len(lmList) != 0:

        # _____ Escoger los 3 puntos de cada articulacion -----
        #Cadera derecha
        angle = detector.findAngle(img, 12, 24, 26)
        #Brazo Derecho
        angle2 = detector.findAngle(img, 12, 14, 22)

        # Acciones con los angulos para las barras porcentuales
        porcentaje = np.interp(angle, (75, 170), (0, 100))
        barra = np.interp(angle, (75, 170), (105, 255))
        porcentaje2 = np.interp(angle2, (190, 284), (0, 100))
        barra2 = np.interp(angle2, (190, 284), (105, 255))

        #print(angle, porcentaje)

#Numero de repeticiones
        #Articulacion 1 Repeticion
        color = (255, 255, 0)
        if porcentaje == 100:
            color = (0, 255, 0)
            if dir == 0:
                count += 0.5
                dir = 1
            if porcentaje == 0:
                color = (0, 255, 0)
                if dir==1:
                    count += 0.5
                    dir = 0

        # Articulacion 2 Repeticion
        color2 = (255, 255, 0)
        if porcentaje2 == 100:
            color2 = (0, 255, 0)
            if dir2 == 0:
                count2 += 0.5
                dir2 = 1
            if porcentaje2 == 0:
                color2 = (0, 255, 0)
                if dir2 == 1:
                    count2 += 0.5

```

```

dir2 = 0

#print(int(angle), int(porcentaje), count, int(angle2), int(porcentaje))

# Informacion en Ventana, Numero de repeticiones
cv2.putText(img, "Repeticion: ", (110, 20), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255),
2)
# Barra de progreso Dibujar

#Articulacion 1
cv2.putText(img, "Cadera Derecha", (340, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 60), (255, 35), (0, 255, 0), 2)
cv2.rectangle(img, (105, 60), (int(barra), 35), color, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje)} %', (170, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 255))
cv2.putText(img, f'{count}', (290, 55), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Articulacion 2
cv2.putText(img, "Brazo Derecho", (340, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255,
255), 2)
cv2.rectangle(img, (105, 100), (255, 75), (0, 255, 0), 2)
cv2.rectangle(img, (105, 100), (int(barra2), 75), color2, cv2.FILLED)
cv2.putText(img, f'{int(porcentaje2)} %', (170, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0,
255))
cv2.putText(img, f'{count2}', (290, 95), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 2)

# Premios de metas
if count >= 10 and count2 >= 10:
cv2.putText(img, "Lo Lograste", (100, 200), cv2.FONT_HERSHEY_PLAIN, 5, (255, 255, 255),
5)
    playsound('mario-bros vida.mp3')

#Reiniciar conteo
if lmList[13][2] < lmList[11][2] !=0:
    count = 0
    count2 = 0

#Acciones

# Imprimir FPS
cTime = time.time()
fps = 1 / (cTime - pTime)
pTime = cTime

# Informacion en Ventana, FPS
cv2.putText(img, "FPS:", (530, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)
cv2.putText(img, str(int(fps)), (600, 40), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 0, 0), 1)

# Imprimir imagen

img[0:132, 0:100] = overlayList[7]

#ABRE LA VENTANA PRINCIPAL
cv2.imshow("Ejercicio 1 Correr en el propio sitio", img)

cv2.waitKey(30)

```



**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO**

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE**



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 22/11/2022

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: CRISTHIAN GEOVANNY FAJARDO ORTIZ WILMER JAVIER YANCHAPANTA TAMAQUIZA
INFORMACIÓN INSTITUCIONAL
Facultad: INFORMÁTICA Y ELECTRÓNICA
Carrera: ELECTRÓNICA Y AUTOMATIZACIÓN
Título a optar: INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN
f. Analista de Biblioteca responsable: <p style="text-align: right;">Ing. Fernanda Arévalo M.</p>

