

INSTITUTO TECNOLÓGICO SUPERIOR DEL SUR DE GUANAJUATO



ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE

Opción 2: Titulación Integral – **Tesis profesional**

Elaborada por:

Pablo Cruz Chavez

Que presenta para obtener el título de:

INGENIERO EN SISTEMAS AUTOMOTRICES

Asesor:

MTW. Juan Oswaldo López Fuentes

ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE

Elaborada por:

Pablo Cruz Chavez

Aprobado por.

MTW. Juan Oswaldo López Fuentes
Docente de la carrera de Ingeniería en sistemas automotrices
Asesor de Tesis profesional

Revisado por.

Ing. Rogelio Orozco Martínez
Docente de la carrera de Ingeniería en sistemas automotrices
Revisor de Tesis profesional

Revisado por.

MTW. David Morales Orozco
Docente de la carrera de Ingeniería en sistemas automotrices
Revisor de Tesis profesional



LIBERACIÓN DE PROYECTO PARA LA TITULACIÓN INTEGRAL

Uriangato, Gto., 18/agosto//2025

Asunto: Liberación de proyecto para la titulación integral

M.C. José Gabriel Aguilera González
Director Académico
ITSUR
PRESENTE

Por este medio informo que ha sido liberado el siguiente proyecto para la titulación integral:

Nombre de estudiante y/o egresado(a): Pablo Cruz Chávez	
Carrera: Ingeniería en Sistemas Automotrices	Núm. de control: T20120124
Nombre del proyecto: Arduino-Ride prototipo robot humanoide	
Producto: Tesis Profesional	

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestras y nuestros egresados.

ATENTAMENTE

Ing. J. Trinidad Tapia Cruz
Subdirector Académico
ITSUR



INST. TECNOLÓGICO SUPERIOR
DEL SUR DE GUANAJUATO
DEPARTAMENTO ACADÉMICO

La comisión revisora ha tenido a bien aprobar la reproducción de este trabajo.

M.T.W. Juan Oswaldo López Fuentes Asesor (a)	Ing. Rogelio Orozco Martínez Revisor(a)* 1	M.T.W. David Morales Orozco Revisor(a)* 2

Agradecimientos:

Agradezco al Instituto Tecnológico Superior del Sur de Guanajuato por el conocimiento recibido durante el transcurso de los semestres y por permitirme realizar mi residencia profesional en el plantel, también agradezco a los docentes de la carrera de ingeniería de sistemas automotrices por la impartición de las asignaturas para mi aprendizaje. Agradezco a mi padre, madre y a mi hermana por el apoyo brindado y su apoyo incondicional motivándome durante el transcurso de mi carrera profesional.

Dedicatoria:

Dedico este trabajo a mi familia por su apoyo incondicional durante el transcurso del mismo, también a mis docentes que me han impartido el conocimiento y aprendizaje, esto para mi es un gran logro ya que no desistí del objetivo a pesar de los obstáculos que se presentaron durante el transcurso.

Tabla de contenido

Capítulo 1	11
Introducción.	11
Capítulo 2	14
Marco teórico (Antecedentes).	14
Capítulo 3	23
Planteamiento del problema	23
3.1. Identificación.	23
3.2. Justificación.	23
3.3. Alcance.	23
Capítulo 4	25
Objetivos	25
4.1. Objetivos generales.	25
4.2. Objetivos específicos.	25
Capítulo 5	27
Metodología	27
Capítulo 6	42
Resultados	42
Capítulo 7	79
Análisis de Resultados.....	79
Capítulo 8	83
Conclusiones y trabajo a futuro.....	83
Referencias bibliográficas	85
Anexos	92

Índice de figuras

Fig. 1. Hardware para el Robot Humanoide.....	35
Fig. 2. Componentes de alimentación y sensores del Robot Humanoide.	37
Fig. 3. Etapas del proyecto ARduinoRide – Prototipo Robot Humanoide.	38
Fig. 4. Impresora Kingroon K3ps utilizada para imprimir los modelos 3D del prototipo Robot Humanoide.	46
Fig. 5. Componentes electrónicos utilizados en el proyecto ArduinoRide prototipo Robot Humanoide.	47
Fig. 6. Componentes de alimentación del prototipo del Robot Humanoide.	48
Fig. 7. Sensores implementados en el Robot Humanoide.	49
Fig. 8. Interfaz del Software Blender.....	50
Fig. 9. Modelos realizados en Blender durante la capacitación.	50
Fig. 10. Aplicaciones de realidad aumentada (RA) realizadas en Unity durante la capacitación.	52
Fig. 11. Programación en bloques de las aplicaciones de realidad aumentada....	53
Fig. 12. Modelos 3D de la parte superior del torso del Robot Humanoide.	55
Fig. 13. Modelos 3D de la parte inferior del torso del Robot Humanoide.	56
Fig. 14. Modelos 3D de la mano izquierda del Robot Humanoide.	57
Fig. 15. Modelos 3D de la pierna izquierda del Robot Humanoide.	58
Fig. 16. Modelos 3D de la pierna izquierda del robot Humanoide continuación....	58
Fig. 17. Modelos 3D de la cabeza del prototipo Robot Humanoide.	59
Fig. 18. Modelos 3D para los componentes electrónicos del Robot Humanoide. .	60
Fig. 19. Modelos 3D de los componentes electrónicos, de alimentación y sensores del prototipo.....	61
Fig. 20. Piezas del Robot humanoide en 3D.....	62
Fig. 21. Modelo 3D en el software Blender del Robot Humanoide.....	62
Fig. 22. Prototipos de aplicaciones de realidad aumentada.....	63
Fig. 23. Menú inicial de la aplicación del manual de armado del robot humanoide.	64

Fig. 24. Escena del paso 6 de la aplicación de realidad aumentada del robot humanoide.....	65
Fig. 25. Escenas de los botones 2 y 3 del menú inicial de la aplicación de realidad aumentada.	65
Fig. 26. Escenas del 4 botón del menú inicial de la aplicación.	66
Fig. 27. Código de la escena del prototipo de mover cubos.	67
Fig. 28. Código de la escena del prototipo de marcadores.	68
Fig. 29. Código de la escena del menú inicial.....	68
Fig. 30. Programación de los pasos del armado del robot humanoide, paso 6.....	70
Fig. 31. Programación de la escena de los menús de selección de pasos, Pasos Uno.....	70
Fig. 32. Programación de la escena del marcador del robot humanoide.	71
Fig. 33. Código de la programación de las escenas.	72
Fig. 34. Modelos 3D configurados para impresión 3D en UltiMarket Cura.	73
Fig. 35. Modelos 3D impresos en PLA en la impresora Kingroon K3ps.....	73
Fig. 36. Armado del robot humanoide.....	75
Fig. 37. Código en Arduino IDE para el movimiento de las manos del robot humanoide.....	75
Fig. 38. Aplicación de control remoto para el robot humanoide.	76
Fig. 39. Escenas de la aplicación de realidad aumentada.	77
Fig. 40. Prototipo y aplicación.....	78
Fig. 41. Posiciones de los servomotores.	98

Índice de tablas

TABLA I Características del Software Blender	29
TABLA II Características del Software Unity	29
TABLA III Características del Software Microsoft Visual Studio	30
TABLA IV Características del Software Arduino IDE.....	30
TABLA V Características del Software UltiMaker Cura	31
TABLA VI Características de la plataforma MIT App Inventor.....	31

ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE.

Resumen:

Durante la realización de este proyecto fue desarrollada una aplicación de realidad aumentada con la implementación de los modelos 3D del prototipo de robot humanoide desarrollado en Blender, se implementaron procesos para la realización y funcionamiento de la aplicación de realidad aumentada. Se plantearon distintos objetivos para el proyecto, el desarrollo de la aplicación, diseño de los modelos 3D, diseño de la interfaz de la aplicación, entre otros que nos llevó a la terminación del proyecto.

Se realizó el diseño y creación de modelos 3D, así como la asignación de color y textura a los modelos del robot humanoide, la aplicación de realidad aumentada fue diseñada y creada la interfaz en Unity, la programación de cada una de las funciones de la aplicación se realizó en Visual Studio a través del lenguaje de programación C#, el prototipo del robot humanoide se imprimió en PLA, se armó y se programó en Arduino IDE para que simulara movimientos del ser humano y se realizaron pruebas a todos los componentes realizados.

Con este proyecto se pudo realizar la aplicación de realidad aumentada que tendrá un impacto en el entorno educativo, principalmente en la Ingeniería de Sistemas Automotrices, el proyecto de residencia proporciona conocimientos clave en la realización de aplicaciones, en el uso de herramientas para el modelado 3D y herramientas para la programación en C#, fortaleciendo habilidades.

Palabras claves: Realidad aumentada, modelado 3D, diseño, programación, robot humanoide.

Abstract:

During this project, an augmented reality application was developed with the implementation of 3D models of the humanoid robot prototype developed in Blender. Processes were implemented for the creation and operation of the augmented reality

application. Various objectives were set for the project: application development, 3D model design, application interface design, and other tasks that led to the completion of the project.

The 3D models were designed and created, as well as color and texture assignments for the humanoid robot models. The augmented reality application was designed and its interface created in Unity. Each of the application's functions was programmed in Visual Studio using the C# programming language. The humanoid robot prototype was printed in PLA, assembled, and programmed in the Arduino IDE to simulate human movements. All components were tested.

This project enabled the development of an augmented reality application that will have an impact on the educational environment, primarily in Automotive Systems Engineering. The residency project provided key knowledge in the development of applications, in the use of 3D modeling tools, and in C# programming tools, strengthening skills.

Keywords: Augmented reality, 3D modeling, design, programming, humanoid robot.

Capítulo 1

Introducción.

En este informe de residencias profesional se desarrolló el proyecto enfocándolo a la integración de los modelos 3D en una aplicación de realidad aumentada para el prototipo del robot humanoide, en el que se emplearon software como Unity para la creación de la aplicación de realidad aumentada y Blender para creación de los modelos 3D del prototipo.

Las razones de la realización de este proyecto surgen de la necesidad de una herramienta de realidad aumentada interactiva para la materia de sistemas embebidos de la carrera de Ingeniería de Sistemas Automotrices del Tecnológico Superior del Sur de Guanajuato, con el fin de preparar y mejorar la enseñanza de los estudiantes y realizar la asignatura más interactiva y dinámica.

Se definió un objetivo general y cierta cantidad de objetivos específicos para la realización de este proyecto tales como el desarrollo de la aplicación de realidad aumentada, el diseño y desarrollo de modelos 3D para el prototipo del robot humanoide, programación de la aplicación y la realización de pruebas, con una metodología realizada en distintas etapas para la realización del proyecto de una manera sistemática y fácil de interpretar, para una buena eficacia de cada etapa del proyecto.

Este trabajo de la realización del proyecto de residencias tiene contribución a la mejora de las técnicas de aprendizaje de la asignatura, proporcionando herramientas para interactuar de manera digital en la enseñanza de la misma. Los resultados obtenidos tienen un impacto en distintos campos de aprendizaje y conocimiento para los estudiantes ya que nos proporcionan herramientas que interactúan de manera digital y con el entorno que nos rodea.

En el Capítulo 1 que es el actual capítulo es la introducción en el cual se introduce al espectador sobre este documento del proyecto ARduinoRide – Prototipo Robot Humanoide en el cual se da una breve explicación al lector sobre cada una parte del contenido de cada capítulo de este informe de prácticas profesionales, así como una breve explicación de las razones por la que se realizó este proyecto y también se habla de los objetivos que se plantearon para realizar este proyecto.

En el Capítulo 2 se trata sobre el marco teórico y antecedentes en el cual se encuentran antecedentes de otros proyectos similares que se realizaron en el pasado, así como términos sobre la realidad aumentada, que es cada software y su tipo de herramienta, así como otros conceptos muy relevantes para el entendimiento de lo que consta este proyecto.

El Capítulo 3 es el planteamiento del problema en el cual se explica la identificación, justificación y el alcance de este proyecto ARduinoRide – Prototipo Robot Humanoide en el cual se da a conocer las razones y motivos por el cual fue necesario la realización de este proyecto, así como para que será implementado el producto de los resultados del mismo.

El Capítulo 4 es la declaración del objetivo general y de los objetivos específicos los cuales al cumplirse todos se logra alcanzar el objetivo general el cual es el desarrollo de la aplicación de realidad aumentada del prototipo del robot humanoide.

El Capítulo 5 es la metodología del proyecto en el cual se describe y explica donde se realizó el proyecto, así como los materiales y herramientas que podrían ser utilizadas en la realización del proyecto, así como las etapas que se definieron para la realización del mismo.

En el capítulo 6 se trata de los resultados obtenidos en cada una de las etapas que se realizaron durante el proyecto para lograr los objetivos planteados para alcanzar los productos finales deseados del proyecto, como la aplicación de realidad aumentada y el prototipo del robot humanoide.

El Capítulo 7 es el análisis de los resultados en el cual se comparan los resultados con los antecedentes y la descripción de como se alcanzaron todos los objetivos que fueron definidos.

El Capítulo 8 se dan las conclusiones a las que se llegó con la realización de este proyecto, así como las propuestas para trabajo a futuro que se pueden implementar en este proyecto para continuarlo con el y agregar mejoras al producto final.

Sobre todo, esto trata este proyecto lo cual es una breve explicación de todo lo que va encontrar en cada capítulo de este documento del proyecto ARduinoRide – Prototipo Robot Humanoide.

Capítulo 2

Marco teórico (Antecedentes).

REALIDAD AUMENTADA

[1] menciona que:

La realidad aumentada (AR) o Augmented Reality, (AR) designa a una representación de la realidad, visualizada a través de un dispositivo tecnológico, con información digital añadida por éste. Se combinan así elementos físicos tangibles con elementos virtuales, creándose así una realidad aumentada en tiempo real. Complementa la percepción del mundo real con capas de información digital, que se superponen a la realidad, a la percepción del mundo físico.

En [1], la realidad aumentada necesita para poder generar imágenes u objetos marcadores también llamados activadores o disipadores, estos son imágenes que contienen un símbolo o código y esto hace que el dispositivo pueda proyectar las imágenes u objetos en el espacio que detecta la cámara del dispositivo. Los marcadores se clasifican en niveles por su complejidad, nivel 0: hiperenlaces código QR, nivel 1: Marcadores imágenes blanco y negro, nivel 2: no hay marcadores y nivel 3: visión aumentada.

Antecedentes

Realidad Aumentada en la Educación

[2] menciona que:

El proyecto Magic Book del grupo activo HIT de Nueva Zelanda. El alumno lee un libro real a través de un visualizador de mano y ve sobre las páginas reales contenidos virtuales. De esta manera cuando el alumno ve una escena de Realidad Aumentada que le gusta puede introducirse dentro de la escena y experimentarla en un entorno virtual inmersivo.

[2] menciona que:

Instituciones del prestigio como Massachusetts Institute of Technology (MIT) y Harvard están desarrollando en sus programas y grupos de Educación aplicaciones de Realidad

Aumentada en formato de juegos; estos juegos buscan involucrar a los estudiantes de educación secundaria en situaciones que combinan experiencias del mundo real con información adicional que se les presenta en sus dispositivos móviles. juego de exterior Environmental Detectives que utiliza PDA con sistema GPS y el juego de interior Mystery @ The Museum que se basa en la tecnología Wi-Fi. También han desarrollado juegos para enseñar materias de matemáticas y ciencias, y todos ellos están orientados a trabajar de forma colaborativa entre los estudiantes.

En [2], En Europa se han desarrollado distintos proyectos en los cuales se desarrollan y diseñan aplicaciones de realidad aumentada para la educación, entre estos proyectos están CONNECT, CREATE Y ARiSE, estas son herramientas de presentaciones 3D y con interacción que facilitan el aprendizaje de materias de todas las ciencias ya que los alumnos pueden interactuar con los objetos virtuales en su entorno real y logran un aprendizaje experimentado.

Grupo Multimedia EHU

[2] menciona que:

El Grupo Multimedia EHU trabaja en el área de educación superior online desde hace diez años, centrando sus esfuerzos en la creación de video/multimedia como un elemento integral del proceso de enseñanza, en particular en el área de Ingeniería. El grupo ha incluido en su actividad investigadora el diseño y el desarrollo en el entorno de educación online de contenidos multimedia utilizando la Realidad Aumentada. La plataforma elegida para desarrollar los contenidos multimedia basados en Realidad Aumentada ha sido AMIRE. La Realidad Aumentada establecerá un puente entre los conceptos teóricos y la realización física de los experimentos con los dispositivos reales.

Aplicaciones Actuales

En [3], las aplicaciones que tiene la realidad aumentada son como las gafas de Microsoft Hololens que genera la combinación de lo físico y lo virtual, también esta los Google Glass. Otros ejemplos son como los medios que usa Ikea que tiene un catálogo de muebles que te permite ubicar los muebles en tu casa y ver cómo puedes acomodar los muebles en tu casa a través de la realidad aumentada, otra

empresa que usa la realidad aumentada es Volkswagen con Augmented Reality Apps que sirve para comparar el uso de energía de sus automóviles.

Educación

[4] menciona que:

El ámbito educativo ha adoptado la realidad aumentada de manera permanente entre sus recursos tecnológicos. Son numerosas las aplicaciones educativas que tienen como base la realidad aumentada y que son utilizadas tanto en las aulas como fuera de ellas. Se utilizan en proyectos de clase. Aprendizajes experimentales prácticamente todas las disciplinas tienen una parte experimental que pueden realizarse con realidad aumentada facilitando en gran medida el aprendizaje y el desarrollo de destrezas transversales. En medicina, donde el uso de las Google glass de forma experimental hace un par de años fue muy mediático, en arquitectura e ingenierías, la posibilidad de realizar y ver modelos en 3D de diferentes edificios y construcciones es muy útil en el aprendizaje del alumno.

Robot Humanoide

En [5], Es un robot el cual se construye y da forma para ser como un cuerpo humano, se diseña para poder tener distintas funcionales, como de entretenimiento, para fines experimentales de investigación para estudiar su funcionamiento imitando el comportamiento humano, para interactuar con herramientas, en la salud y en la educación. Se componen principalmente de un torso, dos brazos, dos piezas y una cabeza, hay distintos tipos de formas de robots humanoides como que son solo la mitad superior del robot o los androides que son contruidos estéticamente como un humano, también son utilizados para la atención del cliente como en recepción y en líneas de producción de automóviles.

Herramientas de Software

Según [6], Blender es un software CAD de modelaje 2D y 3D que surgió como un proyecto comunitario, el cual fue creado por Ton Roosendaal, que es un neerlandés, que es director de arte y desarrollador de software, el 2 de enero de 1994 fue cuando se crearon los primeros archivos de Blender y al principio Blender era una aplicación

interna de NeoGeo y por lo que el 2 de enero de 1994 se considera el aniversario de Blender. La primera versión de Blender que fue lanzada como un software independiente fue en enero de 1995, Blender fue lanzado un 13 de octubre de 2002 como un software de código abierto, mejor conocido como Licencia Pública General GNU, actualmente Blender es controlado por la Fundación Blender que es financiado casi totalmente por donaciones, se decidió usar este software CAD para la realización de los modelos porque es de libre licencia, así que no tendremos que gastar recursos económicos para la realización de las piezas y es un entorno que se puede entender fácilmente por el usuario.

Según [7], Unity es un motor de desarrollo, de juegos y aplicaciones, ya que es un software de programación, ya que con esto se puede diseñar, crear y realizar el funcionamiento del entorno de una aplicación o videojuego. Sus principales funciones o formas en las que se emplea son en motor gráfico para renderizar gráficos 2D y 3D, programación, animación y el motor físico que simule ambientes de la realidad, este software nos permite crear aplicaciones para distintos sistemas operativos, como Android, IOS, Windows, MacOS, Linux y también nos permite crear aplicaciones para dispositivos de realidad virtual (RV), se decidió utilizar este motor de videojuegos para crear la aplicación de realidad aumentada porque esta tiene una compatibilidad con distintos sistemas operativos, ya que tiene licencia libre y de paga, por lo que se usara la licencia libre para no tener gastos de recursos económicos para realizar la aplicación.

Según [8], Visual Studio es un software de programación IDE, creativa que se usa para crear, editar, depurar y compilar el código hasta llegar a crear alguna aplicación. Este software puede crear aplicaciones para distintos sistemas operativos de dispositivos, ya que Visual Studio es de un código de fuente ligero, este software se puede ejecutar en Windows, Linux y macOS, en este software IDE utiliza distintos lenguajes de programación, los cuales son Java, Python, Go, .NET, C++ y C#, este programa fue lanzado el 1 de mayo de 1997, se decidió utilizar esta herramienta porque viene integrada como herramientas para programación

de Unity y también porque se puede programar en C#, ya que este lenguaje es fácil de programar.

Según [9], Arduino es una plataforma de software IDE la cual es de código abierto, también la parte de hardware se llama Arduino, el lenguaje de programación Arduino está basada en Wiring y el Arduino IDE que está basado en Processing, fue creado en el Instituto de Diseño de Interacción de Ivrea de Italia en el año de 2005, para ser usado como herramienta para crear prototipos para estudiantes. Este software puede ser utilizado en distintos sistemas operativos los cuales son macOS, Windows y Linux. Se decidió utilizar esta plataforma de programación por su entorno claro y sencillo, también por ser de código abierto y esto hace que no se tengan gastos económicos para realizar la programación del Robot Humanoide en la placa ESP32.

En [10], UltiMarker Cura es un software que su función es la impresión 3D, su licencia es gratuita y su manera de uso es fácil de implementar, es recomendado por millones de usuarios gracias a su confiabilidad a la hora de su implementación, tiene un gran ajuste de los modelos 3D con mas de 400 configuraciones para obtener buenos resultados de impresión y de cortes en los modelos, se puede usar este software en distintos sistemas operativos los cuales son Windows, macOS y Linux, permite integrar complementos CAD de SolidWorks, Autodesk Inventor, Siemens NX, entre otros más, admite archivos compatibles como STL, OBJ, X3D, 3MF, BMP, GIF, JPG y PNG. Su utilización es para preparar el modelo 3D para su impresión con las configuraciones deseadas, se configura la velocidad y la calidad para imprimir, su utilización de este software para el proyecto es para la impresión de los modelos 3D estructurales y estéticos del robot, los cuales se configuran los parámetros de impresión para mandarlos a imprimir en la impresora 3D, ya que su licencia es gratuita y es fácil de usar esta plataforma.

En [11], MIT App Inventor es una plataforma de programación visual e intuitivo con el que permite crear aplicaciones funcionales para distintos dispositivos, como teléfonos Android, iPhones y tabletas con sistemas operativos Android y iOS. Se

puede crear una aplicación hasta en menos de 30 minutos, su programación se basa en bloques y hace que se facilite la creación de aplicaciones complejas y que tengan un alto impacto en un periodo de tiempo mucho más cortos. Esta plataforma tiene más de 6 millones de usuarios que están registrados y más de un millón de visitantes mensuales de 195 países que llegan a crear casi 30 millones de aplicaciones. Se usará esta plataforma en la creación de la aplicación para el control remoto del robot para nuestro proyecto, ya que es una plataforma gratuita, es fácil de implementar y su único requisito es tener conexión a una red de internet para poder usarla.

En [12], Unity Learn es una plataforma desarrollada por Unity para el aprendizaje y mejorar las habilidades en la creación de aplicaciones, videojuegos y aplicaciones de realidad virtual, ya que tiene tutoriales, cursos y certificaciones para personas principiantes o desarrolladores expertos, como tutoriales y proyectos gratuitos para aprender desde cero, certificaciones oficiales para habilidades y recursos para educadores para la enseñanza de Unity en escuelas. Se decidió utilizar esta plataforma para el proyecto, ya que es gratuita y nos proporciona cursos para adquirir el aprendizaje que se debe de tener para la realización del proyecto.

Software IDE:

[13] menciona que:

Un IDE o entorno de desarrollo integrado es una aplicación de tipo software que combina en un solo lugar todas las herramientas necesarias para realizar un proyecto de desarrollo de software. Ofrece una interfaz que permite escribir código, organizar grupos de texto y automatizar las tareas redundantes de programación. Poseen un editor, un compilador, un des bloqueador, y las funcionalidades de compleción de código o de gestión de código genérico. Los más avanzados ofrecen también funcionalidades de Data Visualization, de Tracing o de referencias cruzadas.

Software CAD:

[14] menciona que:

El software de diseño asistido por ordenador, mayormente conocido por las siglas CAD que provienen del inglés Computer-Aided Design, es un software para crear y editar

modelos bidimensionales y tridimensionales de objetos físicos. Aunque también puede encontrarse como CADD, Computer-Aided Design & Drafting. Esta tecnología permite la agilización del trabajo, automatizando los procesos manuales del proceso de diseño de producto, reduciendo errores, ganando velocidad y aumentando la calidad. Con esto las empresas consiguen así una mayor eficacia y productividad, puesto que permite visualizar de manera previa el producto final y además jugar de una forma interactiva con diferentes diseños, sin necesidad de crear un número elevado de prototipos. Existen 2 tipos de software CAD, el 2D y el 3D. El primero de ellos, CAD 2D trabaja con dibujos técnicos bidimensionales simples que suelen ser la base para otros proyectos mayores, en cambio el 3D permite crear dibujos tridimensionales con mayor precisión y detalle y que además muestran el espacio de trabajo y la profundidad, por lo que ofrece una visión más real de los objetos.

Motor de desarrollo de juegos y aplicaciones:

En [15], Un motor de desarrollo de juegos y aplicaciones o en inglés game engine, son software creado para optimizar el desarrollo de aplicaciones y juegos, estos llevan un bucle de juego, un motor de renderizado 2D o 3D, un motor de renderizado debe de administrar el almacenamiento y memoria para los modelos 3D e imágenes que se mostraran en la pantalla, también los software game engine tienen bucle de juego integrado, motor de audio, de animación, herramientas de realidad virtual, distintos componentes visuales e inteligencia artificial (IA), los motores de desarrollo más conocidos y usados son Unity, Unreal Engine, Amazon Lumberyard, GameMaker, entre otros más.

Impresión 3D:

[16] menciona que:

La impresión 3D es el proceso de creación de objetos mediante el depósito de capas de material unas sobre otras. La impresión 3D se denomina fabricación aditiva (AM) en lugar de los métodos sustractivos tradicionales, como el fresado CNC, cuando se utiliza para la producción industrial. Esta tecnología existe desde hace unas cuatro décadas, inventada a principios de los años 80. Aunque la impresión 3D empezó siendo una técnica lenta y costosa, los amplios avances tecnológicos han hecho que las tecnologías AM actuales sean más asequibles y rápidas que nunca.

Manufactura adictiva:

En [17], La Manufactura adictiva (MA) es una rama de los procesos de fabricación de más reciente uso industrial y es una herramienta de prototipado rápido, en la actualidad usa distintos materiales como polímeros, termoplásticos y algunos cerámicos y metales, actualmente la manufactura adictiva se clasifica como parte de los procesos de manufactura digital con el auge de la revolución industrial 4.0 y es clave de la misma. La ASTM (American Society for Testing and Materials) clasifica a la manufactura avanzada en 7 familias, las cuales son Fotopolimerización, Fusión por cama de polvos, inyección de aglutinante, Inyección de material, Laminación, Extrusión de material, Deposición directa de energía y también está el híbrido.

PLA:

En [18], El PLA es un termoplástico conocido como ácido láctico o poliláctico, su nombre químico es ácido 2-hidroxipropanoico, su materia prima del PLA es el maíz, la yuca o la caña de azúcar ya que son materias primas ricas en almidón, sus orígenes se dan alrededor de 1780 ya que lo descubrió el químico Scheelle, después el científico Wallace Carothers creó la sustancia con un peso molecular muy bajo, y en los años 90 se implementó en el filamento para la impresión 3D, se usa en la impresión 3D para hacer figuras decorativas, maquetas y prototipos, su rango de temperatura de impresión es entre 190-220 °C, tiene una resistencia mecánica baja, ya que es frágil y a la vez duro, tiene un olor más agradable y no es tóxico. Existen distintos tipos de PLA, los cuales son PLA 850, PLA 870, PLA LW que son de baja densidad, su fórmula química es $C_3H_4O_2$.

Lenguaje C#:

En [19], C# es un lenguaje de programación y su pronunciación correcta es C Sharp, este es un lenguaje de código abierto, también es multiplataforma, fue desarrollado por Microsoft tomando las mejores partes de los lenguajes C y C++, este lenguaje se orientó a objetos para la plataforma NET y después se le agregó la creación de código, sus características principales son su sencillez de uso, su modernidad, la seguridad para evitar errores, extensibilidad, su compatibilidad de sintaxis con el

lenguaje C, C++, Java, entre otros más y su eficiencia al usar, sus plataformas de desarrollo para el lenguaje de programación son como el Visual Studio Code que es gratuito y es multiplataforma ya que se puede usar en Windows, MacOS y Linux, también esta Visual Studio que su licencia de pago, la plataforma recomendada para utilizar el C# es Microsoft Visual Studio, C# es de los 5 lenguajes más usados para la creación de aplicaciones.

Microcontrolador ESP32:

En [20], Es un microcontrolador de desarrollo creado por la empresa Espressif Systems, tiene una gran potencia y es versátil, incluye conectividad WiFi y Bluetooth, es ideal para una amplia gama de proyectos como para Internet de las cosas (IoT). Sus ventajas son que consume poca energía, su conectividad ya que facilita la conexión con otros dispositivos y a internet, también tiene distintos canales de comunicación por lo que lo hacen muy práctico. Para su programación se puede realizar en distintos softwares como Arduino IDE, ESP-IDF, LuaNode, MicroPython, entre otros más y se puede programar en distintos lenguajes de programación como C, C++, Python y JavaScript.

Internet de las cosas:

[21] menciona que:

El Internet de las Cosas (IoT) es la red de objetos conectados a Internet: dispositivos físicos, vehículos, electrodomésticos y otros artículos con componentes electrónicos y software que permiten a estos objetos recopilar e intercambiar datos. IoT está conectando a personas y empresas de todo el mundo al permitirles interactuar con objetos cotidianos de una forma más inteligente. En 1982, una máquina de Coca-Cola modificada en la Universidad Carnegie Mellon se convirtió en el primer aparato conectado a Internet. A éste le siguió una serie de dispositivos con conexión a Internet que continúan hoy en día, incluidos frigoríficos conectados a Internet y dispositivos informáticos portátiles como las Google Glass.

Capítulo 3

Planteamiento del problema

3.1. Identificación.

Esta aplicación surge de la necesidad de tener una herramienta de realidad aumentada interactiva para la carrera de ingeniería en sistemas automotrices del tecnológico superior del sur de Guanajuato, esta aplicación será de mucha ayuda al realizarse para la asignatura de sistemas embebidos, ya que se tiene el problema de que falta la herramienta para poder enseñar y preparar mejor a los estudiantes en los sistemas embebidos para hacer la asignatura más interactiva para poder realizar y replicar sistemas embebidos durante el curso de la asignatura, con esto dar mejores herramientas para los estudiantes para la industria 4.0 en su formación.

3.2. Justificación.

El proyecto de la aplicación tendrá un impacto de manera muy directa en la carrera Ingeniería en Sistemas Automotrices del Instituto Tecnológico Superior del Sur de Guanajuato, tanto en semestres inferiores como en semestres superiores, ya que se podrá diferir la complejidad de los elementos desarrollados facilitando y con esto dar mejores herramientas para la utilización de algunos elementos de la industria 4.0, como es impresión 3D, manejo de internet de las cosas y la realidad aumentada, ayudando a los estudiantes a fomentar sus habilidades y conocimientos en sistemas embebidos que son importantes en su formación y utilidad en la industria actual.

3.3. Alcance.

El alcance del proyecto de la aplicación de realidad aumentada del Robot Humanoide es para ser usada solo en dispositivos con sistema operativo Android ya que no será posible usarlo en otros sistemas operativos como IOS ya que no se tendrá el soporte necesario para otros sistemas operativos al estar realizando el proyecto de la aplicación de realidad aumentada, además de momento solo tendrá

alcance para estudiantes de la carrera de Ingeniería en Sistemas Automotrices del Tecnológico Nacional de México campus Sur de Guanajuato (ITSUR), ya que se está desarrollando en la división de esta carrera. La conclusión de este proyecto se dará con la terminación de la aplicación terminada con todos los modelos necesarios del Robot Humanoide, los cuales son las partes que nos ayudara a ser los soportes de los servomotores, para que el robot tenga movimiento, también los modelos que nos servirán de enlaces para los servomotores y así poder hacer que el robot tenga movimiento en los pies y manos, los modelos de la parte del cuerpo donde van unos servomotores y todos los circuitos necesarios para el funcionamiento del robot, para la enseñanza de la materia de sistemas embebidos o de alguna otra materia que tenga relación con este tema.

Capítulo 4

Objetivos

4.1. Objetivos generales.

Desarrollar una aplicación de realidad aumentada en la plataforma Unity para el sistema ARduinoRide del prototipo Robot Humanoide, para impactar positivamente en la formación de estudiantes, al facilitar elementos complejos e impulsar sus habilidades tecnológicas en la impresión 3D, la realidad aumentada y la industria 4.0 en la Ingeniería de Sistemas Automotrices.

4.2. Objetivos específicos

- Fortalecer y ampliar los conocimientos sobre el modelaje en 3D mediante la utilización de Blender para comprender los aspectos técnicos y artísticos para la creación de los modelos.
- Realizar una capacitación de manera integral en la plataforma Unity Learn, para poder adquirir los conocimientos prácticos y teóricos necesarios para la creación de la aplicación de realidad aumentada.
- Diseñar y elaborar los modelos 3D con el software CAD Blender destinados para el prototipo del Robot Humanoide para que sea funcional y preciso el diseño final.
- Diseñar y desarrollar la interfaz gráfica de la aplicación de realidad aumentada en el motor de videojuegos Unity, para garantizar una experiencia intuitiva y funcional.
- Realizar la programación en la plataforma IDE Visual Studio en C# para la interfaz gráfica desarrollada en Unity para habilitar las funcionalidades de los botones de cambio de modelos 3D, movimiento de los modelos 3D y los cambios de escena.
- Desarrollar la programación en la plataforma IDE Arduino para la interacción humano / máquina (IHM) del prototipo Robot Humanoide, para garantizar una fluida y eficiente comunicación del Robot Humanoide.

- Ejecutar pruebas integrales a la aplicación de realidad aumentada en conjunto del prototipo Robot Humanoide para evaluar el desempeño de esta, para garantizar su estabilidad y funcionalidad.
- Documentar de manera sistemática el proceso del desarrollo del Robot Humanoide y de la aplicación de realidad aumentada.
- Concluir el proyecto de la realidad aumentada con la entrega de la aplicación con la versión optimizada y funcional para sistemas embebidos, cumpliendo con los requisitos pedidos.

Capítulo 5

Metodología

Lugar donde se realizó el trabajo

El proyecto del Robot Humanoide se llevó a cabo en la sala de investigación y posgrado del Instituto Tecnológico Superior del Sur de Guanajuato, que se encuentra ubicado en Uriangato, Guanajuato, ya que en este espacio se encuentra el equipo de cómputo y conexión a Internet que fue proporcionado para realizar el proyecto, la sala está dentro de la comunidad educativa ya que esto facilita la retroalimentación del profesor, esta sala fue seleccionada ya que cuenta con lo necesario para la realización del proyecto y por su cercanía a otros laboratorios que fueron requeridos como el laboratorio de sistemas automotrices para la impresión de las piezas del Robot.

Materiales

1. Desarmador de puntas:

Como herramientas se utilizaron un desarmador multiusos de puntas intercambiables, es de plástico el mango y las puntas son de acero, su función fue para adaptarse a los distintos tipos de tornillos para facilitar el ensamble y ajuste del Robot Humanoide, por lo que esta herramienta fue muy fundamental para hacer el armado de las piezas y la sujeción de los servomotores.

2. Pinzas:

Las pinzas que se utilizaron son de uso general ya que con estas se puede cortar, sujetar y manipular elementos, son de acero con sus mangos recubiertos de plástico tipo goma antideslizante, estas se utilizaron para cortar cables, pelar los cables y realizar algunas conexiones de cables, además también para sujetar los servomotores durante el armado, por lo que son muy esenciales para el ensamblaje y funcionamiento del Robot.

Material

1. PLA:

Acido Poli láctico es el nombre del PLA, el cual fue el filamento para la impresión 3D de las piezas del Robot Humanoide, se usó un rollo de 1 kg, que es un polímero que se usa para la manufactura adictiva y su temperatura de impresión ronda los 220 °C, esto hace que sea fácil de implementar y tiene un buen terminado estético, tiene una buena resistencia y versatilidad para la creación de prototipos que no serán sometidos a esfuerzos mecánicos o térmicos, ya sea a utilizado con anterioridad este material.

2. Cautín y soldadura:

Se uso un cautín ya que es una herramienta de punta metálica para soldar, alcanza una temperatura entre 300 °C y 450 °C, se implementó para soldar conexiones entre los componentes eléctricos y cables, para que tuviera un buen contacto todo el circuito del Robot, la soldadura que se utilizó es una aleación de estaño y plomo, se utilizó resina para mejorar la conexión.

3. Cables:

Los cables son conductores eléctricos que están aislados con plástico, se utilizaron de un calibre delgado ya que cumple con las características de los componentes para la transmisión de señales y de energía, también es más fácil de manipular.

4. Cables Dupont:

Estos son cables flexibles que tienen terminales macho-macho, hembra-hembra o combinados macho-hembra, estos son muy utilizados en prototipos, ya que facilitan las conexiones y son muy rápidas las conexiones con la placa de desarrollo que se implementara.

5. Cinta de aislar:

La cinta se usó como dice su nombre para aislar las conexiones entre cables o componentes electrónicos, ya que con esto se reduce el riesgo de cortos

circuitos en el Robot, para garantizar la seguridad eléctrica y de quien lo manipule.

Software

1. Blender

TABLA I
Características del Software Blender

Características	Descripción
Nombre del Software	Blender
Versión	4.3.2
Licencia	Código Abierto, General Public License (GPL)
Plataforma compatible	Windows, macOS, Linux
Requerimientos	Sistema operativo Windows 10 o Windows 11, CPU de 64 bits y 8 Cores, RAM de 8GB y recomendado 32 GB de RAM, tarjeta gráfica con OpenGL de 8 GB VRAM recomendada y 1 GB de espacio disponible de disco duro.
Funcionalidad	Utilizado para la creación de modelos 3D, para la aplicación de las texturas a los modelos 3D del Robot Humanoide.

En [22], [Tabla I], se muestran las características del software Blender utilizado para el modelaje de las piezas en 3D.

2. Unity:

TABLA II
Características del Software Unity

Características	Descripción
Nombre del Software	Unity
Versión	6000.0.34f1
Licencia	Unity Personal (Gratuita) y Unity Pro o Plus (de pago)
Plataforma compatible	Windows, macOS y Linux
Requerimientos	Sistema operativo Windows 8, 10 y 11, CPU de 64 bits, RAM de 8 GB, tarjeta gráfica compatible con DirectX 11, Vulkan
Funcionalidad	Utilizado para programar, diseñar y probar la aplicación de realidad aumentada (AR) mediante AR Foundation

En [23], [Tabla II], se muestran las características del software Unity utilizado para la creación de la aplicación de realidad aumentada (RA).

3. Microsoft Visual Studio

TABLA III
Características del Software Microsoft Visual Studio

Características	Descripción
Nombre del Software	Microsoft Visual Studio
Versión	Visual Studio 2022
Licencia	Community Edition (gratuita), Professional y Enterprise (de pago)
Plataforma compatible	Windows, macOS
Requerimientos	Sistema operativo Windows 10 y 11, CPU de 64 bits, RAM 8 GB mínimo y 16 GB de RAM recomendada, tarjeta gráfica que admita una resolución mínima de pantalla de WXGA (1366 x 768) y 20 Gb de espacio en el disco duro.
Funcionalidad	Utilizado para la programación en C# de las escenas de la aplicación de realidad aumentada en el software Unity

En [24], [Tabla III], se muestran las características para el funcionamiento del software Microsoft Visual Studio para la programación de la aplicación de realidad aumentada.

4. Arduino IDE

TABLA IV
Características del Software Arduino IDE

Características	Descripción
Nombre del Software	Arduino IDE
Versión	2.3.1 Versión estable
Licencia	Código abierto y gratuita
Plataforma compatible	Windows, macOS, Linux
Requerimientos	Sistema operativo Windows 10 y 11, CPU de 64 bits, RAM superior de 4 GB y espacio en el disco duro superior a 1 GB
Funcionalidad	Utilizado para la programación, compilación del código para el controlador para el funcionamiento del Robot Humanoide

En [25], [Tabla IV], se muestran las especificaciones del Software Arduino IDE para la programación del Robot para su funcionamiento.

5. UltiMaker Cura

TABLA V
Características del Software UltiMaker Cura

Características	Descripción
Nombre del Software	UltiMaker Cura
Versión	5.3.0
Licencia	Código abierto y gratuita
Plataforma compatible	Windows, Linux y macOS
Requerimientos	Sistema operativo Windows 10 y 11, CPU de 64 bits, RAM de 8 GB, tarjeta gráfica compatible con OpenGL 4.1 y espacio de 2 GB en el disco duro
Funcionalidad	Utilizado para la configuración de los modelos 3D para imprimir en la impresora de PLA o ABS para el Robot

En [26], [Tabla V], se muestran las características del software para la configuración de los modelos para la impresión.

6. MIT App Inventor

TABLA VI
Características de la plataforma MIT App Inventor

Características	Descripción
Nombre del Software	MIT App Inventor
Versión	Ultima versión disponible en línea
Licencia	Código abierto y gratuita
Plataforma compatible	Windows, macOS, Linux, pero se basa en el navegador con los cuales es compatible Mozilla Firefox 3.6 o superior, Apple Safari 5.0 o superior, Google Chrome 4.0 o superior
Requerimientos	Conexión a internet y navegador actualizado
Funcionalidad	Utilizado para crear la aplicación mediante programación de bloque para el control del Robot

En [27], [Tabla VI], se muestran las especificaciones de la plataforma MIT App Inventor para la creación de la aplicación para controlar el funcionamiento del Robot.

Hardware

1. Computadora:

La computadora que se requirió para la realización de este proyecto debe de cumplir con distintos requerimientos para el funcionamiento adecuado de los distintos software que se utilizaron durante el desarrollo del proyecto, preferentemente que la computadora fuera de escritorio para un mejor funcionamiento, debía de tener un CPU de 64 bits y que el procesador fuera un Intel Core i5, un AMD RYZEN 5 o superior, memoria RAM de al menos 16 GB, una tarjeta gráfica compatible con Open GL y DirectX 11, una VRAM de 8 GB, un disco duro de estado sólido (SSD) de al menos 256 GB y el sistema operativo Windows 11, esto con el fin de evitar inconvenientes a la hora de la realización del proyecto, de que se cerraran los programas o no se pudieran realizar algunas acciones en los software que se implementaron durante la realización del proyecto del Robot Humanoide.

2. Impresora Kingroon kp3s:

La impresora que se necesitó para la impresión de los modelos en 3D en material PLA para el Robot Humanoide, la cual fue la impresora de la marca Kingroon, modelo kp3s, sus características son: En [28], su espacio de impresión es de 180 * 180 * 180 mm, es un espacio pequeño pero suficiente, tiene un controlador TMC225 que es el encargado de la impresión, precisión y estabilidad, tiene una superficie de impresión magnética extraíble, que hace que las impresiones se adhieran bien y se puedan retirar fácilmente, su pantalla táctil es de 2.8 in que es desde donde se maneja la impresora para la configuración para realizar las impresiones de los modelos 3D.

Componentes electrónicos

3. ESP32:

En [29], [30], es un microcontrolador con diseño robusto que puede funcionar con distintas temperaturas las cuales oscilan entre -40 °C y 125 °C, tiene un consumo de energía bajo ya que está diseñado para dispositivos móviles,

dispositivos electrónicos y aplicaciones de Internet de las cosas (IoT), tiene un chip híbrido Wi-Fi y Bluetooth que hace que pueda funcionar como un sistema independiente completo, con esto puede interactuar con otros sistemas para agregar funcionalidad Wi-Fi y Bluetooth a través de distintas interfaces. Sus características que tiene son una alimentación de 5 Vcc, tiene 32 pines de GPIO, una memoria flash de 32M bit, su Bluetooth es 4.2 y su Wi-Fi 802.11b/g/n, tiene interfaz UART, SPI, SDIO, I2C, PWM, 12S, IR, ADC, DAC, tiene una frecuencia de 2412 – 2484 MHz y sus dimensiones son de 25.6 * 50 mm, por lo que esta placa fue la encargada de todo el funcionamiento del prototipo del Robot.

4. PCA9685:

En [31], es un controlador de 16 canales que fue creado para controlar leds, pero también permite controlar servomotores, genera una señal PWM con reloj interno y de 12 bits, se comunica mediante la interfaz I2C y solo requiere 2 pines para controlar todas las señales PWM las cuales son SDA y SCL. Sus características son que tiene una alimentación de 3-5 V, tiene 16 canales de comunicación para controlar hasta 16 servomotores, tiene una frecuencia de salida de 40 – 1600 Hz, salidas configurables push-pull o drenador abierto, rápida respuesta para la habilitación y des habilitación de las salidas y tiene un diseño que permite tener varias tarjetas en un mismo bus que se puede obtener hasta un máximo de 992 salidas PWM para controlar esa cantidad de servomotores.

5. Convertidor de voltaje 300W 20A:

En [32], en un regulador de voltaje Step-down, que tiene una entrada mínima de voltaje de 6V y un voltaje máximo de entrada de 40V, un voltaje mínimo de salida de 1.2V y una salida máxima de voltaje de 36V, su corriente máxima de salida es de 20A, a la temperatura que funciona es desde -40 °C a 85 °C, con una eficiencia de conversión de hasta el 97%, su onda de salida es menor a 50mV, tiene una protección contra cortocircuitos, el regulador tiene 2 disipadores de calor para mejorar la disipación del calor, se

recomienda cableado de 30A de valor del cable y sus medidas son 60 * 53 * 27 mm, se utilizó para controlar la energía que consuman los servomotores del Robot Humanoide para evitar que se llegue a quemar el controlador PCA9685.

6. Servomotores MG995:

En [33], Es un servomotor con un alto torque de hasta 11 kg-cm, sus especificaciones son que tiene un voltaje de operación de 4.8V a 6.6V, su velocidad de operación es de 0.20 s/60° (4.8V), 0.16 s/60° (6.6V), su torque es de 9.4 kgf*cm (4.8V), 11 kgf*cm (6.6V), tiene un ángulo de rotación de 180°, sus dimensiones son 40.7 * 19.7 * 42.9 mm, el largo del cable es de 32 cm y tiene un peso de 55 g, se utilizaran para que el Robot se desplace y realice las funciones de movimiento, es importante el peso de los motores porque puede hacer que el Robot tuviera buena estabilidad o no la tuviera y así mismo el consumo de energía para mantenerlo de pie.

7. Controlador de voltaje LM2596 DC-DC HW-411:

En [34], El Step Down LM2596 es un dispositivo que puede regular o disminuir el voltaje de entrada de un circuito que tiene una fuente de alimentación de mayor voltaje, tiene un trimpot multivuelta (potenciómetro) para ajustar el voltaje el trimpot permite dar 25 vueltas para ajustar, esto hace que sea más fácil configurar la salida del módulo en el parámetro necesario o en el que se desee. Sus características son que es un regulador de voltaje Step Down, tiene 4 pines los cuales son IN-, IN+, OUT- y OUT+, tiene unas dimensiones de 43 * 21 * 14 mm, su voltaje de entrada es de 4 a 35 V y su voltaje de salida es de 1.23 a 30 V, su corriente máxima es de 3A, pero lo recomendado es de 2.5A, la potencia de salida es de 25 W, tiene una eficiencia del 92%, una frecuencia de trabajo de 150KHz, tiene protección de corto circuito de hasta 5A, su temperatura de funcionamiento va desde - 45 °C a +85 °C. Se utilizo en el proyecto como su nombre lo dice para regular el voltaje que alimentara a la placa del microcontrolador ESP32 para evitar

que se fuera a quemar o producir algún corto circuito al momento del funcionamiento del Robot Humanoide y pueda funcionar correctamente.

Los componentes de hardware que se implementaran para el armado y funcionamiento del Robot Humanoide son los elementos que podemos ver en la Fig. 1, los cuales son la impresora Kingroon Kp3s para la impresión 3D de los modelos, el microcontrolador ESP32 para controlar el funcionamiento del Robot, el controlador PCA9685 para las señales de pulso (PWM) de los servomotores, el controlador de voltaje 300W 20^a para controlar el voltaje de salida, los servomotores MG995 para el funcionamiento del Robot y el controlador de voltaje LM2596 DC-DC HW-411 para la alimentación del microcontrolador ESP32.

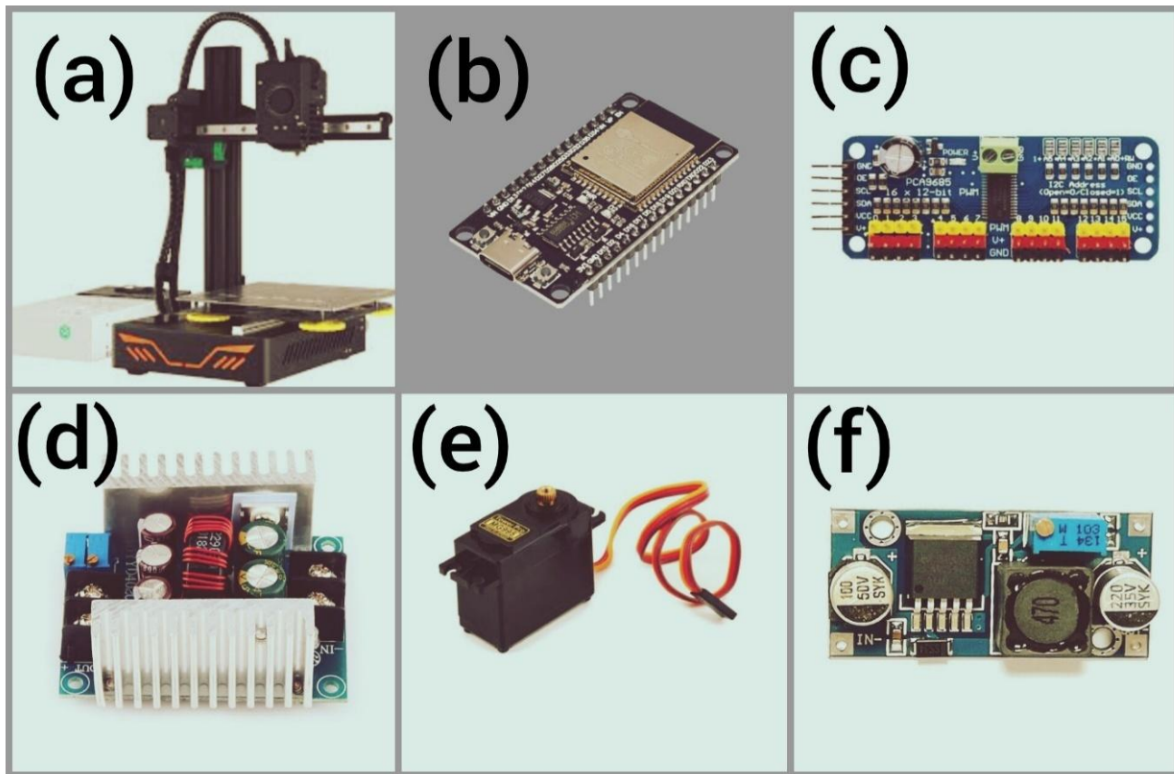


Fig. 1. Hardware para el Robot Humanoide. (a) Impresora Kingroon kp3s, (b) Microcontrolador ESP32, (c) Controlador PCA9685, (d) Controlador de voltaje 300W 20A, (e) Servomotor MG995, (f) Controlador de voltaje LM2596 DC-DC HW-411 [28], [30], [31], [32], [33], [34].

Componentes de alimentación

8. Batería 18650:

En [35], [36], Las baterías 18650 son una batería cilíndrica, el 18 significa que tiene un diámetro de 18 mm, el 65 nos dice que tiene 65 mm de altura y

el 0 significa que es cilíndrica, se pueden encontrar en capacidades desde 2000 mAh a 3500 mAh, son más grandes que las pilas AA y pueden contener hasta cuatro veces la potencia que una pila AA. Son ampliamente utilizadas ya que almacenan una gran cantidad de energía con una larga vida útil que se recargan muchas veces, son usadas en muchos dispositivos, como linternas, bocinas, juguetes, bancos de energía, herramientas eléctricas, dispositivos médicos y más dispositivos. Sus principales características son un voltaje de 3.7 Vcc, unas dimensiones de 65 mm de alto y 18 mm de diámetro, una capacidad de almacenamiento de 2200 mAh, hechas de encapsulado metálico y su elemento activo de iones de litio. Se decidió utilizar este tipo de batería en el proyecto por su voltaje y capacidad para alimentar a los circuitos y por ser recargables.

9. Interruptor:

En [37], el interruptor switch balancín con piloto on/off es un interruptor para controlar equipos eléctricos de una manera segura. Sus características son que soporta un voltaje de 127 V, una corriente de hasta 16 A una frecuencia de 60 HZ, su temperatura de uso ronda entre 25 °C a 85 °C, tiene 2 polos de alimentación, tiene 2 posiciones el interruptor una de encendido y otra de apagado, está hecho de material de cobre. Se decidió implementar para apagar y encender la alimentación del circuito, ya que es importante tener un interruptor.

Sensores

10. Sensor ultrasónico:

En [38], El sensor ultrasónico de distancia HC-SR04 es el sensor más utilizado en proyectos educativos para realizar distintas mediciones de distancias, se deben de utilizar en lugares que no sean húmedos y superficies planos. Sus características son que tiene una alimentación de 5V, tiene 4 pines los cuales son Vcc, Trigger, Echo y el GND, tiene un rango de medición de 2 cm a 400 cm, su corriente de alimentación es de 15 mA, una señal de disparo de 10 uS, una apertura de pulso de 15° y sus

dimensiones son de 45 * 20 *15 mm. Se decidió implementar este sensor para el Robot para que nos muestre la distancia a la que se encuentre de objetos y así tomar decisiones

11. Sensor de humedad:

En [39], El sensor de humedad y temperatura DHT11, detecta y mide la temperatura hasta los 50 °C y la humedad hasta 90 % RH, es muy utilizado en proyectos para automatización esto a través de microcontroladores o placas de desarrollo. Sus características son su alimentación de 3.3 a 5.5 V, un rango de humedad de 20% a 90% RH, un rango de humedad de 0°C a 50°C, tiene una resolución de humedad de 1% RH, también tiene una resolución de temperatura de 1°C y su tamaño es de 12 * 25 * 6 mm. Se utilizo para el proyecto para medir los parámetros de humedad y temperatura del ambiente que rodea el entorno del Robot Humanoide.

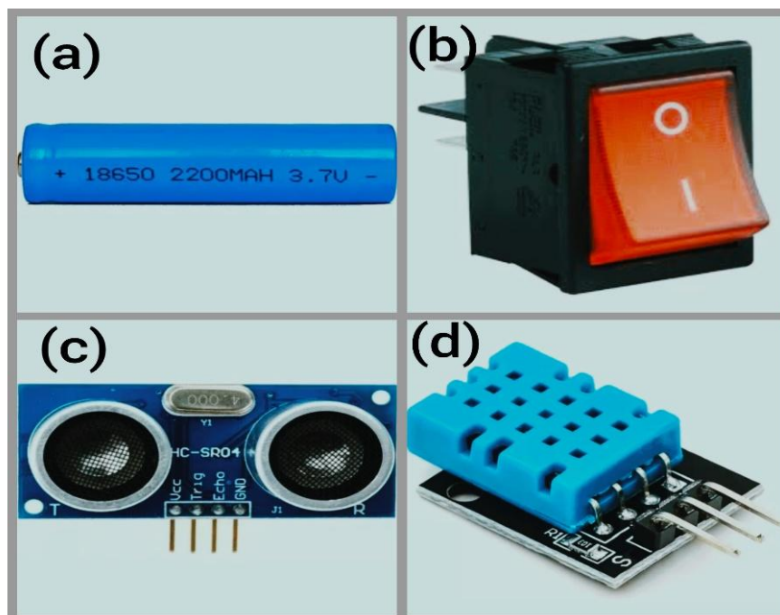


Fig. 2. Componentes de alimentación y sensores del Robot Humanoide. (a) Batería 18650, (b) Interruptor switch balancín con piloto on/off, (c) Sensor ultrasónico HC-SR04 de distancia y (d) Sensor de humedad y temperatura DHT11 [35], [37], [38], [39].

Los componentes de alimentación y sensores que se implementaron en el prototipo Robot Humanoide son los que podemos observar en la Fig. 2, los cuales son la batería 18650 para energizar el circuito, el interruptor switch con piloto on/off para

energizar y des energizar el circuito, el sensor ultrasónico HC-SR04 para medir la distancia y el sensor DHT11 para medir la humedad y la temperatura.

Metodología del proyecto

La metodología del proyecto de ARduinoRide del prototipo Robot Humanoide está dividida en distintas etapas, las cuales unas son más extensas y complicadas que otras, pero todas las etapas son igual de importantes para poder realizar y desarrollar este proyecto, por lo que a continuación se detallara y explicara cada etapa del proyecto ARduinoRide del prototipo Robot Humanoide, como podemos ver en la Fig. 3, las cuales son un total de 7 etapas y son las siguientes:

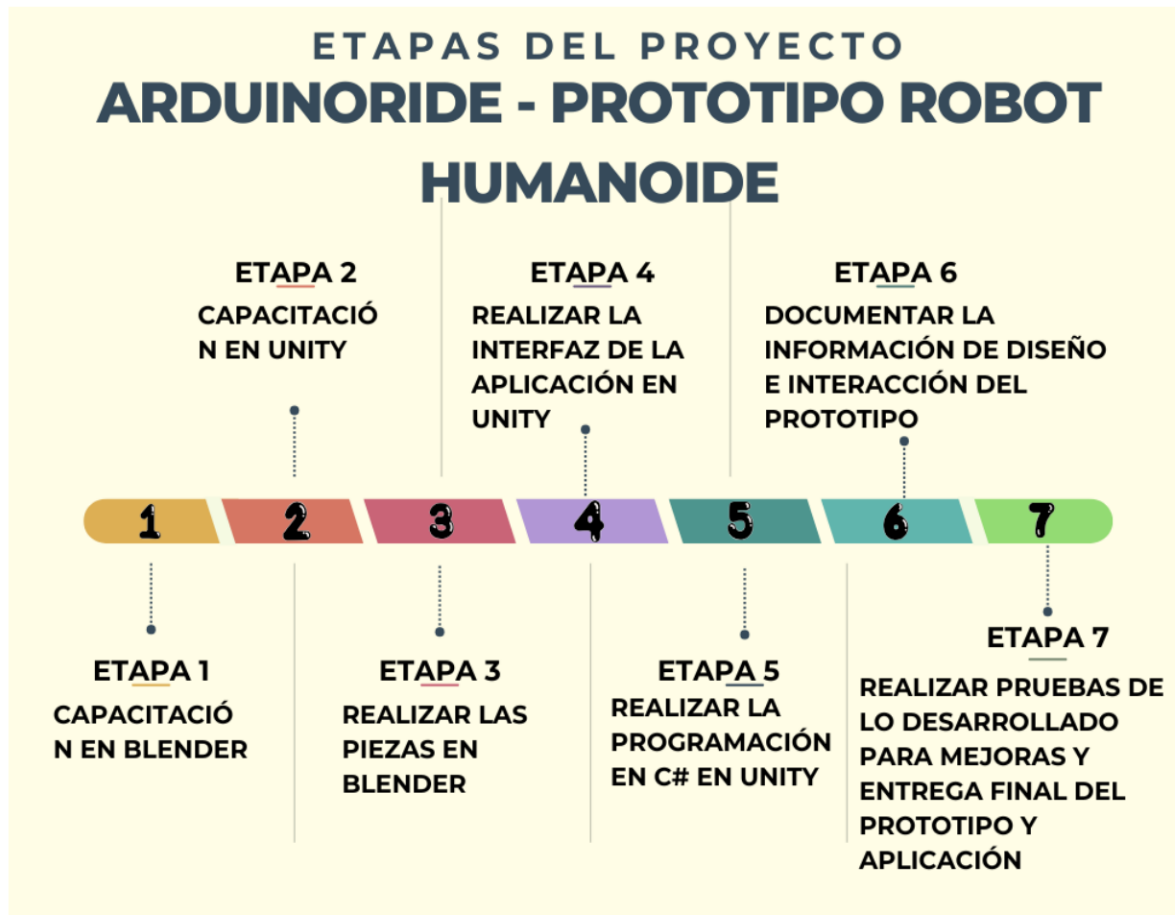


Fig. 3. Etapas del proyecto ARduinoRide – Prototipo Robot Humanoide.

Primera etapa, Capacitación en Blender:

La cual consistió en tomar un tutorial sobre la interfaz de Blender y tutoriales de modelaje 3D de distintos tipos de objetos 3D, ya sean modelos relacionados con el

proyecto o modelos no relacionados al proyecto, estos tutoriales de Blender se tomaron a través de la plataforma de videos YouTube de Google, esto con el fin de poder conocer las distintas herramientas y modeladores que se encuentran en Blender para poder diseñar, desarrollo, dar textura y color a las piezas que utilizamos en el prototipo de Robot Humanoide.

Segunda etapa, Capacitación en Unity:

La realización de esta capacitación se tomó un curso a través de la plataforma de Unity Learn, la cual pertenece a Unity, tuvimos que realizar una cuenta para poder tomar el curso, en la plataforma buscamos un curso de realidad aumentada para dispositivos móviles para esto tuvimos que asignarlo y tuvimos que instalar la aplicación de Unity Hub y la versión necesaria para el curso, esto con el fin de saber utilizar las herramientas de creación de aplicaciones y de la realidad aumentada de Unity para la creación de la aplicación de realidad aumentada que se implementó en el proyecto.

Tercera etapa, Realización de piezas en Blender:

Consistió en diseñar y creación de todas las piezas del Robot Humanoide en el software de diseño 3D Blender, las cuales fueron las piezas que formaron el torso del robot, las piezas que formaron los brazos, las piezas de los pies, las piezas de la cabeza del Robot, los modelos de los elementos electrónicos que se utilizaron en el proyecto y también se diseñaron las piezas en las que van a ir empotrados todos los componentes electrónicos del Robot Humanoide para su adecuado funcionamiento y terminado estético, también se realizaron piezas estéticas para darle un terminado más presentable y no solo funcional al Robot Humanoide, una vez realizados todos los modelos se les dio textura y color para exportarlos en formato fbx para implementarlos en Unity, también se exportaron en formato STL para imprimirlos en la impresora 3D con filamento PLA.

Cuarta etapa, Desarrollo de la interfaz de la aplicación en Unity:

En esta etapa se realizaron y diseñaron las escenas de la aplicación utilizando Unity que fueron lo que el usuario final pueda ver e interactuar en la aplicación, ya que en

estas escenas se agregaron los modelos 3D de distintas partes del Robot Humanoide y también de los componentes electrónicos, mediante botones o marcadores de realidad aumentada se pudo interactuar con ellos, esto se realizó para fines didácticos de la aplicación ya que tiene pasos para el armado del robot humanoide y otras características importantes, también en otra escena pudimos visualizar el modelo 3D del Robot Humanoide a través de marcadores y estas escenas serán de realidad aumentada, que es lo que se estuvo implementando y desarrollado durante el proyecto.

Quinta etapa, Programación en C#:

La programación se llevó a cabo para que en las escenas que se desarrollaron para la interfaz de la aplicación en Unity tenga la funcionalidad de los botones y marcadores de realidad aumentada, y pueda mostrarse en realidad aumentada y se pudo interactuar con ella, ya que sin la programación adecuada resultaría obsoleta la aplicación, por eso esta programación en C# se realizó en el software de Visual Studio que pertenece al desarrollador Microsoft que nos fue una herramienta muy útil para creación de la aplicación en Unity y nos sirvió para programar todas las escenas diseñadas y pudimos implementar la aplicación de realidad aumentada para el prototipo del Robot Humanoide.

Sexta etapa, Documentación de la información del diseño e interacción del prototipo:

En esta se desarrolló la impresión de los modelos del Robot Humanoide creados en Blender en la impresora 3D en material PLA, para posteriormente se comenzó a armar el Robot y se fue documentando como colocar las piezas y los servomotores así como a la posición del ángulo al que deben de ir colocados los servomotores, una vez armado se procedió a realizar la programación del robot utilizando el software Arduino IDE para el microcontrolador ESP32 que se utilizó para ser el cerebro del Robot y el controlador PCA9685 para mandar las señales PWM a los servomotores para su funcionamiento, también se desarrolló una aplicación en la plataforma MIT App Inventor para que funcione de control remoto, para controlar el

Robot, una vez realizada la programación se podrá conectar con la aplicación de realidad aumentada para su control.

Séptima etapa, Realización de pruebas de lo desarrollado para mejoras y entrega final del prototipo y de la aplicación:

En esta etapa se usó tanto la aplicación de realidad aumentada y el Robot Humanoide en distintas situaciones y condiciones con el fin de probar y de que no existieran errores o se llegue a tener bugs en la aplicación, si se llegó a encontrar alguna anomalía se corrigió y se pulió cualquier cosa para tener una aplicación fluida y un adecuado movimiento y control del Robot, una vez que se implementó las mejoras se procedió a entregar la aplicación de realidad aumentada y el Robot Humanoide para su uso didáctico de los alumnos de la carrera de Ingeniería de Sistemas Automotrices del Tecnológico Superior del Sur de Guanajuato.

Las herramientas que se utilizaron para la realización de este proyecto fueron en su mayoría Software de diseño y programación ya que principalmente todo se desarrolló de manera virtual, solo el prototipo del Robot Humanoide fue en físico, y que se imprimió en 3D.

Capítulo 6

Resultados

El proyecto ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE fue desarrollado en la sala de investigación y posgrado como se mencionó en el capítulo 5 Metodología, lo que se realizó en esta sala fue la parte virtual, todo lo que se tenía que realizar con software, para la parte física se tuvo que usar el laboratorio de sistemas automotrices para poder realizar las impresiones 3D de los modelos del Robot Humanoide y posteriormente realizar el armado y programación del mismo.

Los materiales que fueron implementados para la realización del proyecto fueron el desarmador de puntas para el armado del Robot Humanoide y el ajuste de tornillos de cabeza plana para punta de desarmador Phillips y de punta Allen hexagonal. También se implementaron en el proyecto las pinzas de uso general para la realización, se usaron para pelar y cortar los cables necesarios para realizar las conexiones entre los componentes, también se implementaron para detener los servomotores al armado y para soldar los componentes, además se usaron para retirar los soportes a algunas de las piezas impresas, además también fue necesario utilizar un cutter para retirar los soportes a las piezas del Robot.

El material que fue necesario implementar para la realización de este proyecto fue, PLA el cual se usó para las impresiones 3D de los modelos del Robot humanoide, se usó menos de un rollo de 1 kilogramo, ya que se imprimieron algunas piezas las cuales en su mayoría son piezas de tamaño pequeño, la mayoría de las piezas fueron proporcionadas por el asesor externo y unas piezas fueron impresas con PLA otorgado por la coordinación de Sistemas Automotrices. Se uso el cautín, la pasta para soldar y la soldadura de plomo y estaño, para realizar las conexiones se soldaron los componentes electrónicos con los cables correspondientes y necesarios para su adecuado funcionamiento, además también se soldaron cables con cables donde fue requerido ya que se usan para dar alimentación a todos los componentes electrónicos del prototipo.

Se implementó cable calibre número 20 de dos distintos colores para el positivo y negativo, para evitar confusiones y algún inconveniente, esto para realizar las conexiones necesarias de los componentes que se tuvieron que soldar con el cautín para el funcionamiento del Robot. Además, se utilizó cables DuPont de distintos colores para poder identificar fácilmente cada señal que se tuvo que conectar entre componentes electrónicos para la transmisión de las señales, para esto en su mayoría se utilizaron cables hembra-hembra y algunos macho-macho, pero en menor proporción. También se implementó cinta de aislar, para aislar los cables que se soldaron con otros cables con esto para evitar cortos circuitos en el funcionamiento del Robot Humanoide.

Software

El software que se utilizó para el desarrollo del proyecto ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE fue el que se mencionó y se hizo referencia en el Capítulo 5 Metodología, los únicos inconvenientes que se tuvo con el software fue que se tuvo que utilizar otras versiones de las aplicaciones ya que unas fue necesario utilizar una versión en específico para poder realizar el proyecto y en otras ya se tenía instalado el software y fue funcional para lo que se requería en el proyecto de residencias.

- **Blender:** para Blender se utilizó la versión especificada en el Capítulo 5 Metodología, la cual es la versión 4.3.2, con la licencia no se tuvo problemas ya que es gratuita, se utilizó para diseñar y crear las piezas del Robot, realizar en 3D los modelos de los componentes electrónicos que se implementaron en el prototipo, también para darle color y textura a los modelos y exportarlos en formato STL para imprimirlos y en formato fbx para importarlos a Unity, se tuvieron inconvenientes al momento de agregar el color y las texturas ya que el software se cerraba solo a causa de que la computadora no cumplía con todos los requerimientos, por lo que se tuvo que utilizar una laptop para poder realizar esa actividad y se utilizó la misma versión de Blender 4.3.2.

- **Unity:** para Unity se utilizaron dos versiones distintas a la que se mencionó en el Capítulo 5 Metodología por lo que no se usó esa versión, esto se debe a que para realizar la capacitación se tuvo que utilizar una versión anterior la cual es la 2021.3.1f1, ya que no se podría realizar y después se usó otra versión para realizar algunos prototipos de aplicaciones de realidad aumentada la cual es la versión 2022.2.16f1, con la licencia no se tuvo inconvenientes ya que se utilizó la licencia personal que es gratuita, se utilizó para crear, diseñar y desarrollar la aplicación de realidad aumentada para el prototipo del Robot Humanoide, en este caso no se tuvieron inconvenientes ya que todo funcionó correctamente con este software.
- **Microsoft Visual Studio:** para Visual Studio se usó una versión diferente a la propuesta en el Capítulo 5 Metodología, la versión que se implementó fue la siguiente 2019, con la licencia no existió inconveniente ya que también se utilizó la licencia gratuita Community Edition, se usó este software para la programación de C# para las escenas y botones para la interacción de la aplicación de realidad aumentada del prototipo, con este software no hubo inconvenientes a la hora de utilizarlo en el proyecto.
- **Arduino IDE:** en el caso de la aplicación del Arduino IDE se usó otra versión ya que no existía una gran diferencia entre la versión que se utilizó y la propuesta en el Capítulo 5 Metodología, la versión que se utilizó es la 2.3.3, este software se implementó en el proyecto para realizar la programación del Robot Humanoide para que funcionara correctamente y poder interactuar con el mismo y poder cargar ese código en la placa ESP32, no existieron problemas con esta aplicación ya que funcionó correctamente y la licencia del software es gratuita.
- **UltiMaker Cura:** para este software se utilizó la versión que se describió en el Capítulo 5 Metodología, la licencia no hubo problema alguno ya que es gratuita como los softwares anteriores, se utilizó para importar los archivos STL de los modelos 3D del Robot, configurarlos en el espacio de la cama de

impresión, configurar todos los parámetros como el relleno y los soportes, para posteriormente guardar la configuración en un archivo gcode, para mandar a imprimirlos a la impresora 3D, no se tuvo inconvenientes con este software al implementarlo ya que funcionó correctamente al realizar todas las actividades del proyecto que fueron necesarias con esta aplicación.

- **MIT App Inventor:** para el uso de esta plataforma se usó la versión más reciente ya que se usa en línea y siempre se implementa la más reciente, además se utilizó con el navegador de Google Chrome, con la licencia no se tuvo problema ya que es gratuita, solo se necesita realizar una cuenta para poder utilizarla, se usó para crear la aplicación para el control del Robot Humanoide para mandar las instrucciones y controlar el funcionamiento, con esta plataforma no se tuvieron inconvenientes a la hora de utilizarla ya que funciona de una buena manera.

Hardware

- **Computadora:** la computadora que fue proporcionada por la institución para realizar el proyecto ARDUINORIDE – PROTOTIPO ROBOT HUMANOIDE tiene estas características las cuales tiene un procesador AMD PRO A12-9800E R7, 12 COMPUTE CORES 4C +8G, con una velocidad base de reloj de 3.10 GHz, de 4 núcleos de 64 bits, tiene una memoria RAM de 8 GB, un disco duro SSD de 119 GB, no tiene tarjeta gráfica, pero tiene gráficos integrados en el procesador de AMD RADEON R7 Graphics con una VRAM de 1007 MB, con sistema operativo Windows 10, con esta computadora se realizó todo lo que se tenía que realizar en Unity, Microsoft Visual Studio, Arduino IDE, UltiMaker Cura y MIT App Inventor ya que con esta computadora sirvió correctamente para realizar todas las actividades del proyecto, solo se tuvo inconvenientes con el software de Blender ya que la computadora no cumplía con los requisitos mínimos de Blender y solo se pudo modelar las piezas del Robot y los componentes electrónicos, las texturas y colores no se pudieron agregar ya que Blender se cerraba al iniciar

con eso, para solucionar ese problema fue necesario utilizar otra computadora para realizar esas actividades.

La computadora que se utilizó para dar textura y color es una laptop LENOVO ideapad 3 que cuenta con estas características tiene un procesador AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, con velocidad base de reloj de 2.10 GHz, de 4 núcleos de 64 bits, tiene una memoria RAM de 8 GB con velocidad de 2400 MHz, un disco duro SSD de 239 GB, no tiene tarjeta gráfica, pero tiene gráficos integrados en el procesador la cual es la AMD Radeon™ Vega 8 Graphics y una memoria VRAM de 2 GB, con sistema operativo Windows 11 Home con versión 24H2, con la laptop utilizando Blender se agregó las texturas y el color a los modelos y se exportaron a los formatos necesarios y con eso se concluyó las actividades de Blender.



Fig. 4. Impresora Kingroon K3ps utilizada para imprimir los modelos 3D del prototipo Robot Humanoide.

- **Impresora 3D:** la impresora que se utilizó para imprimir las piezas del Robot Humanoide fue la que se mencionó en el Capítulo 5 Metodología la cual es la impresora Kingroon K3ps como podemos ver en la Fig. 4 en la cual se pudieron imprimir todas las piezas del Robot tanto las estructurales y estéticas, para imprimirlas no hubo problemas ya que todas las piezas son de menor tamaño que la cama de impresión ya que sus medidas son de 180

* 180 * 180 mm, tampoco hubo problemas con la impresora a la hora de realizar de imprimir ya que no existieron problemas con el filamento PLA o con la temperatura, ya que se pueden despegarse las piezas y no imprimirse bien.

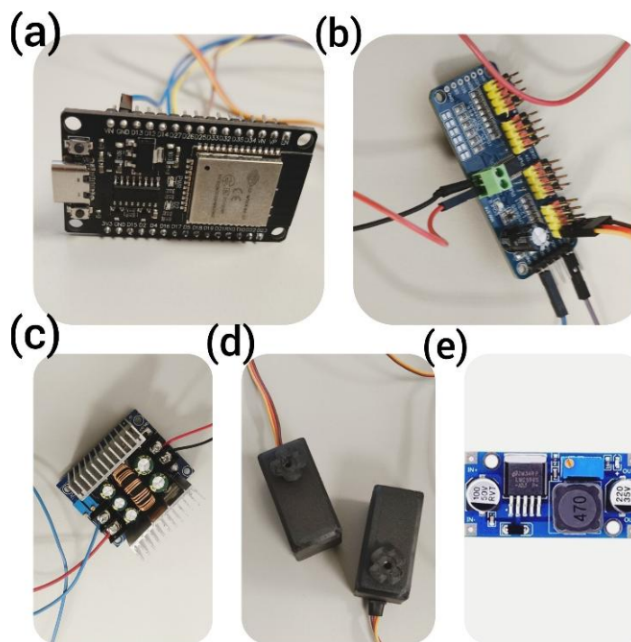


Fig. 5. Componentes electrónicos utilizados en el proyecto ArduinoRide prototipo Robot Humanoide. (a) Microcontrolador ESP32, (b) Controlador PCA9685, (c) Convertidor de voltaje 300W 20A Step-down, (d) Servomotor Mg995, (e) Controlador de voltaje LM2596 DC-DC HW411.

- **Componentes electrónicos:** los componentes que se implementaron para realizar el circuito y configurar el movimiento del Robot Humanoide son los que se mencionaron en el Capítulo 5 Metodología, como podemos observar en la Fig. 5, los cuales son el microcontrolador ESP32 el cual se implementó para ser el cerebro del Robot, se programó para mandar las señales de pulso PWM de los servomotores y controlar el movimiento del mismo, además se configuro los pines de salida para su comunicación con los otros componentes. El controlador PCA9685 se usó para controlar los servomotores a través de las señales PWM que llegan a través de los pines SDA y SCL que se conectaron a la placa ESP32. Convertidor de voltaje 300W 20A Step-down se implementó para regular el voltaje de alimentación para el controlador PCA9685 que alimenta los servomotores, para evitar cortos

circuitos o que se queme el controlador. Los Servomotores MG995 se utilizaron para darle el movimiento a los pies, brazos y torso del Robot Humanoide, para esto fue necesario implementar 18 servomotores para su funcionamiento. Además, se implementó el controlador de voltaje LM2596 DC-DC HW-411 para regular el voltaje de alimentación de la placa ESP32 para evitar cortos circuitos y daños a la placa para así asegurar un buen funcionamiento del prototipo.

- **Componentes de alimentación:** Los componentes que se utilizaron para la alimentación del Robot, los podemos observar en la Fig. 6, y los cuales son 3 baterías 18650 para alimentar todo el circuito y con estas alcanzar el nivel de voltaje necesario. Se utilizó el apagador de switch balancín con piloto on/off para encender y apagar el Robot cuando no se esté usando y así no estar retirando las baterías cuando no este funcionamiento.

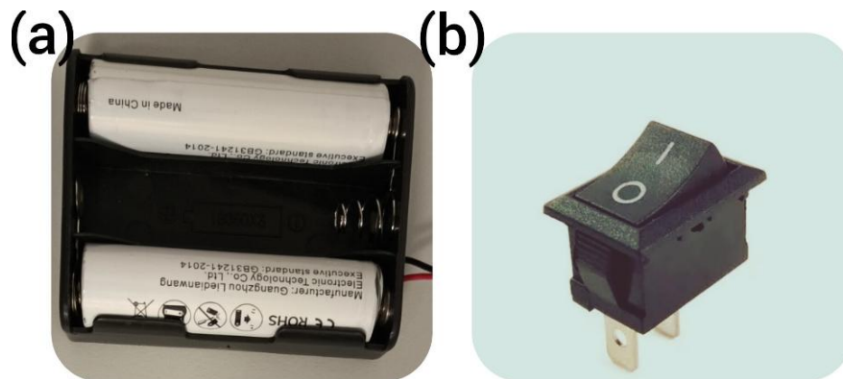


Fig. 6. Componentes de alimentación del prototipo del Robot Humanoide. (a) Baterías 18650, (b) Apagador on/off switch de balancín.

- **Sensores:** Los sensores que se utilizaron para el prototipo del Robot Humanoide son los descritos en el Capítulo 5 Metodología, como podemos ver en la Fig. 7, los cuales son el sensor ultrasónico de distancia HC-SR04 para medir la distancia que exista enfrente del Robot respecto a un objeto y así mismo tomar decisiones que están programadas en la placa ESP32. También se usó el sensor de humedad y temperatura DHT11 para medir la temperatura y humedad del entorno en el que se encuentre el Robot y así poder visualizarse en una interfaz. Estos sensores no son los más adecuados

para un Robot Humanoide, pero son los más fácil de implementar para un prototipo por esa razón son implementados en él.

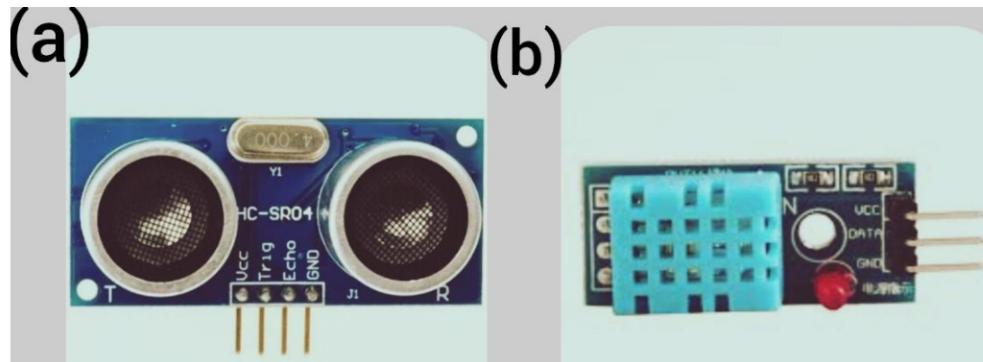


Fig. 7. Sensores implementados en el Robot Humanoide. (a) Sensor ultrasónico de distancia HC-SR04, (b) Sensor de humedad y temperatura DHT11.

Resultados de las etapas de la metodología.

Los resultados a los que se llegó con la realización del proyecto ARDUINIRIDE – PROTOTIPO ROBOT HUMANOIDE son los que se presentan y describen a continuación, los cuales son los productos de las siete etapas descritas de lo que se realizaría, mencionadas y descritas en el Capítulo 5 Metodología.

En la primera etapa del proyecto fue la capacitación en Blender en la cual se adquirieron los conocimientos necesarios en este software para realizar el proyecto, la capacitación se dividió en dos partes, en la primera parte fue conocer la interfaz de Blender, comenzar a realizar modelos 3D desde cero a partir de un cubo, utilizando las diferentes herramientas como extruir, cortar, escalar, insertar caras, agregar biseles al objeto y entre otras, también se aprendió como usar los diferentes modos de edición como el objet mode y el edit mode, para la realización de la capacitación de la primera parte se tomaron tutoriales de modelaje y creación de objetos en Blender a través de la plataforma de videos YouTube de Google como se había mencionado en el Capítulo 5 Metodología. Para la segunda parte de la capacitación fue realizar modelos 3D desde líneas o a partir de una imagen, utilizando modificadores como solidificar, booleano para realizar cortes geométricos, bevel para agregar redondez a todo el modelo 3D para no este pixelada, simetría, entre otros más modificadores, además se aprendió como

agregar el color y las texturas a los objetos 3D, para realizar esta parte de la capacitación en Blender también se tomaron tutoriales en YouTube.

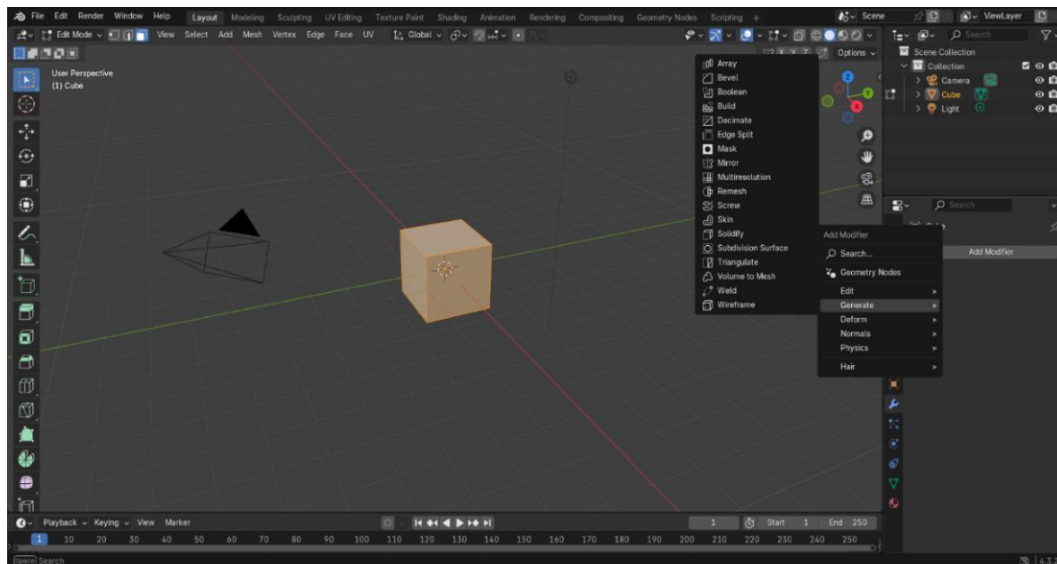


Fig. 8. Interfaz del Software Blender.

En la Fig. 8. Podemos observar la interfaz del software de diseño asistido por computadora Blender en la cual contiene un cubo, una cámara y una lampara que siempre esta predeterminada al crear un nuevo proyecto general, la ventana que está abierta es la de edit mode, en la cual se encuentran en la parte izquierda las herramientas de escala, extruir, cortar, mover, rotar, insertar caras, entre otras, en la parte derecha se muestra la ventana de los modificadores que son muy importantes para el diseño final de los modelos 3D que se realizaron durante este proyecto.

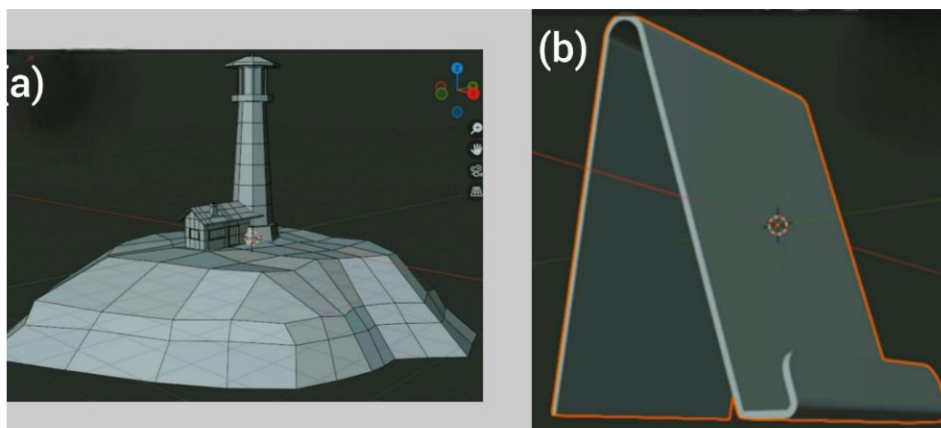


Fig. 9. Modelos realizados en Blender durante la capacitación. (a) Modelo 3D de faro, (b) Modelo 3D de un soporte de celular.

En la Fig. 9. Se observan dos modelos 3D realizados en Blender, los cuales son un faro y un soporte de celular, aunque estos modelos no tengan relación geométrica con el Robot, estos modelos realizados durante la capacitación implementan las herramientas y modificadores, con estos se adquirió el conocimiento necesario para diseñar y realizar los modelos 3D del Robot Humanoide.

En la segunda etapa del proyecto fue la capacitación en Unity la cual consistió en tomar un curso para la creación de aplicaciones de realidad aumentada, el curso se tomó a través de la plataforma Unity Learn que pertenece a Unity, para poder acceder al curso se hizo una cuenta y se pudo tomar el curso ya que la plataforma de Unity Learn es gratuita, el curso consistió en conocer la aplicación de Unity para poder crear un nuevo proyecto y agregar las escenas de realidad aumentada con todo sus componentes necesarios para su adecuado funcionamiento, además de como exportar la APK de la aplicación creada para el sistema operativo Android, también se pueden exportar a otros sistemas operativos pero el proyecto está limitado solo al sistema Android, también se aprendieron más funciones importantes para realizar la aplicación del proyecto como exportar e importar prefabs que son objetos 3D, agregar y navegar entre escenas, escalar, rotar, entre otras funciones más, se realizaron distintas aplicaciones durante el curso como una que se mueve un objeto 3D en la realidad aumentada, una aplicación de filtros de lentes, un juego de realidad aumentada entre otros.

En la Fig. 10. Se observan las distintas aplicaciones que se realizaron durante la capacitación en Unity, las cuales fueron 5 aplicaciones, la primera consta de como visualizar objetos 3D a través de la realidad aumentada mediante marcadores que son imágenes o códigos QR, la segunda aplicación es de como interactuar con los modelos 3D a través de RA utilizándolos en una persona, que son los filtros de lentes que se pueden encontrar en muchas aplicaciones de redes sociales o fotos, en la tercera aplicación es la más simple de todas y la primera que se creó, esta consiste de agregar un modelo 3D y agregarle movimiento automático, para cuando se visualice en RA tenga movimiento, la cuarta aplicación consiste en como visualizar los modelos 3D a través de detención de planos, estos planos se

visualizan una vez que se detecte una superficie plana y puede ser vertical o horizontal según la necesidad de lo que se dese visualizar en realidad aumentada, la quinta y última aplicación es un juego de capturar ovejas mediante un OVNI en RA el cual era un juego de computadora y se tuvo que importar a realidad aumentada, para lograr esto se utilizó la detección de planos para visualizar los modelos 3D del juego, después se realizó la programación de los botones para el funcionamiento de la jugabilidad del juego, esta se realizó en programación de bloques. Con esta capacitación en Unity Learn se adquirió todo el aprendizaje necesario para realizar el proyecto de realidad aumentada del prototipo Robot Humanoide.

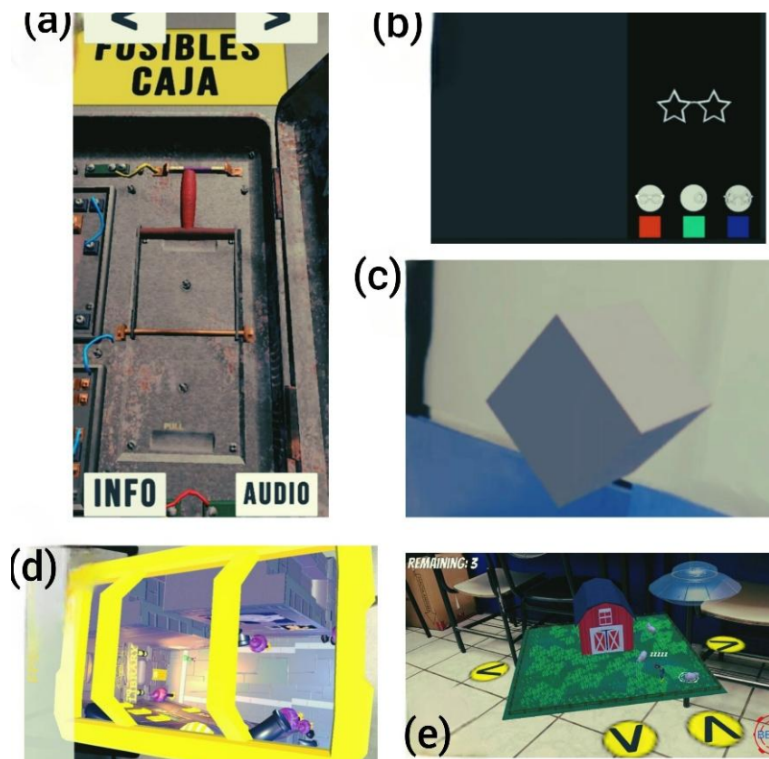


Fig. 10. Aplicaciones de realidad aumentada (RA) realizadas en Unity durante la capacitación. (a) Aplicación RA con marcadores, (b) Aplicación RA de un filtro de lentes, (c) Aplicación RA de un cubo en movimiento, (d) Aplicación RA de un portal a través de detección de planos, (e) Juego RA de atrapar ovejas con un OVNI.

En la Fig. 11. Podemos observar la programación en bloques que se llevó durante la capacitación, esta programación se realiza en Unity y no es necesario otro software para realizar la programación de los botones o movimientos, pero tiene sus limitaciones ya que para poder interactuar con más objetos 3D ya es muy

complicado realizarlo y es más práctico utilizar otro software como Microsoft Visual Studio para programar en el lenguaje C#.

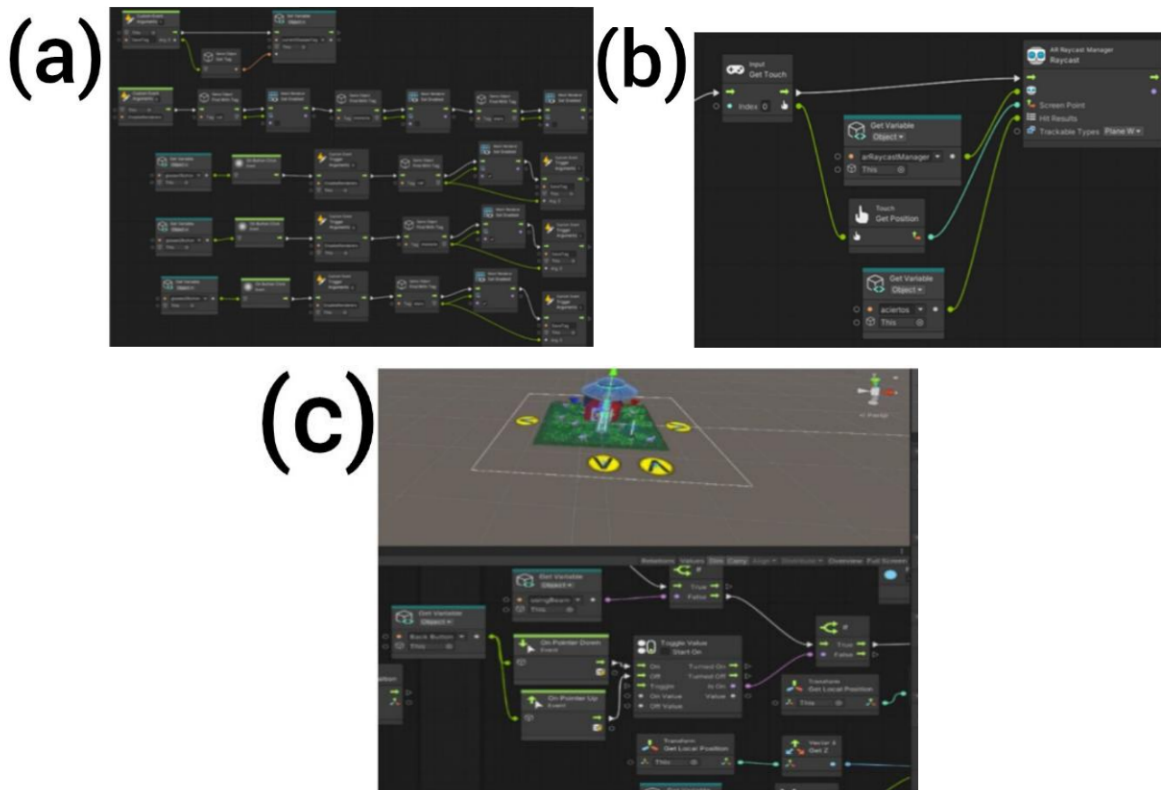


Fig. 11. Programación en bloques de las aplicaciones de realidad aumentada. (a) Programación en bloques de la aplicación de RA de los filtros de lentes, (b) Programación en bloques de la detección de planos, (c) Programación en bloques de los botones del juego de capturar ovejas a través de un OVNI.

En la tercera etapa del proyecto fue el diseño y realización de algunas piezas del robot en el software Blender, la mayoría de piezas del robot fueron proporcionadas por el encargado del proyecto el MTW. David Morales Orozco, las cuales se compraron en el sitio web hashrobotics.gumroad.com [40], por lo que se diseñaron y crearon solo algunas piezas del robot, la parte superior del torso del robot es la parte que nos da las dimensiones del torso superior por lo que fue lo primero que se creó y realice y diseñe la parte trasera del torso superior, esto con el fin de que sea proporcional, luego es la parte inferior del torso, para posteriormente pasar a los modelos 3D de la pierna derecha e izquierda del robot, para después pasar a los modelos de los brazos derecho e izquierdo y por último la cabeza del mismo, que son todos los modelos estructurales del robot, que están

conformados con los objetos 3D donde se colocaron los servomotores para su funcionamiento y las piezas estructurales que permitirán el movimiento del robot humanoide, además por último se crearon modelos 3D de algunos componentes electrónicos que lleva el prototipo para su funcionamiento y también se hicieron otros modelos 3D para contener los componentes electrónicos del robot.

Los modelos 3D que fueron proporcionado y se realizaron se guardaron por la parte a la que pertenecen y se les asigno un nombre en inglés que tiene que ver a la parte a la que pertenece, esto se realizó así para poder identificar las piezas ya que se tuvieron que imprimir en 3D y así evitar confusiones a la hora de imprimir o armar el prototipo, una vez que se realizaron todos los modelos en el software CAD Blender se procedió a darles color y textura a las piezas del robot, para las piezas del torso superior e inferior, las piernas derecha e izquierda, los brazos derecho e izquierdo y la cabeza del robot fue rápido realizar esta actividad ya que solo se implementaron dos colores, amarillo para darle color a las piezas donde van colocados los servomotores y a la parte del frente y trasera del torso superior y además también se utilizó para los ojos del robot, todos los modelos restantes de las partes antes mencionadas se les asigno el color negro, para los modelos que se realizaron para contener algunos componentes electrónicos también solo se les asigno un solo color el cual fue amarillo, posteriormente se procedió a asignarles el color y textura a los modelos 3D de los componentes electrónicos que lleva el robot para su funcionamiento, estos modelos fueron adquiridos del sitio web GRABCAD, que es una página donde puedes descargar modelos 3D de distintos componentes y su licencia es gratuita [41], esta parte si fue más tardada ya que fue más entretenida ya que los componentes se les asigno el color tal como son en realidad en físico, para darles color se tomaron de referencia las imágenes de los componentes que se describieron en el Capítulo 5 Metodología, ya que estos modelos están constituidos por muchos objetos y a cada uno se le asigno el color real por eso fue más tardado, esta actividad de esta etapa del proyecto, una vez que se terminó de dar color y textura todos los modelos 3D se exportaron en formato fbx, ya que son requeridos para importarlos a la aplicación de realidad aumentada que se realizara

en Unity en las dos siguientes etapas del proyecto, también se exportaron todos los modelos 3D a formato STL, con excepción de los modelos de los componentes electrónicos, ya que para poder imprimirlos en 3D se requieren en formato STL.

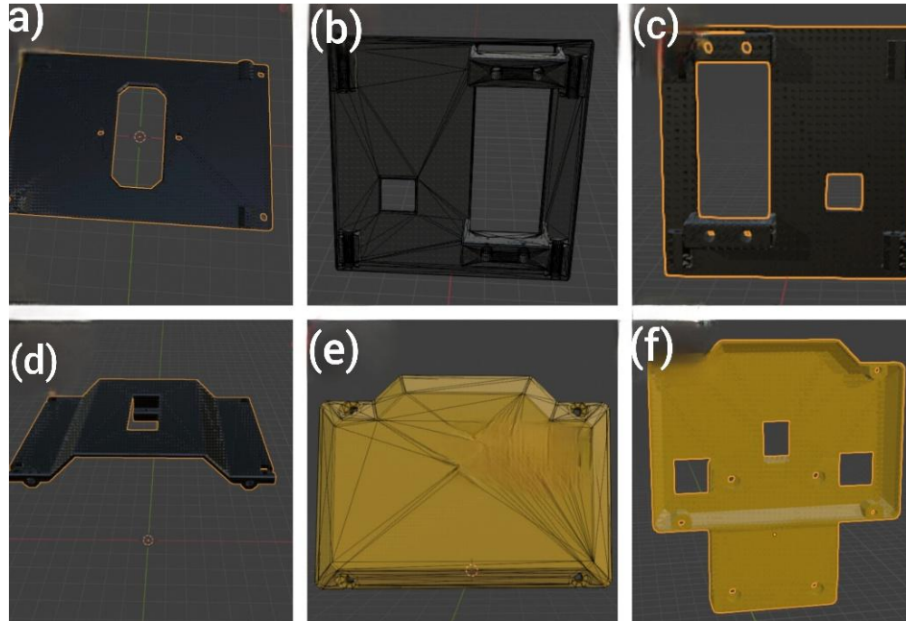


Fig. 12. Modelos 3D de la parte superior del torso del Robot Humanoide. (a) body top 1, (b) body top 2, (c) body top 3, (d) body top 4, (e) body top front, (f) Base convertidor and Body Top Back.

En la Fig. 12. Se pueden observar los modelos 3D de la parte superior del torso que fueron proporcionados y el modelo que se desarrolló el Base convertidor and Body Top Back y se les dio el color y textura a estos modelos, está conformado por 6 piezas las cuales forman como un cubo todo el torso superior, las piezas son el body top 1 que es la base y va unida al torso inferior, el body top 2 y body top 3 son las partes laterales del torso, donde van colocados los servomotores para el movimiento y conexión de los brazos del robot, el body top 4 es la parte superior del robot y va montada la cabeza, el body top front es la parte del frente del torso y la base convertidor and Body Top Back es la tapa trasera del robot y donde va colocado el convertidor de voltaje 300W 20A, en el torso superior van a estar colocados la mayoría de los componentes electrónicos del Robot Humanoide.

En la Fig. 13. Podemos observar los modelos 3D de la parte inferior del torso del robot que fueron proporcionados y se les asignó la textura y el color a estos modelos, los cuales son 4 piezas las cuales forman el torso, las piezas son body

bottom 1 en la cual van colocados dos servomotores para el movimiento y conexión de las piernas, el body bottom 2 es la parte superior del torso inferior y va ensamblada junto con el torso superior para unir las dos partes, el body bottom back es la parte trasera del torso inferior en la cual se tienen dos orificios para el cableado, el body bottom front es la parte de enfrente, en esta parte del torso inferior del robot humanoide también van colocados componentes electrónicos como en la parte superior del torso y cableado de los servomotores y de los componentes.

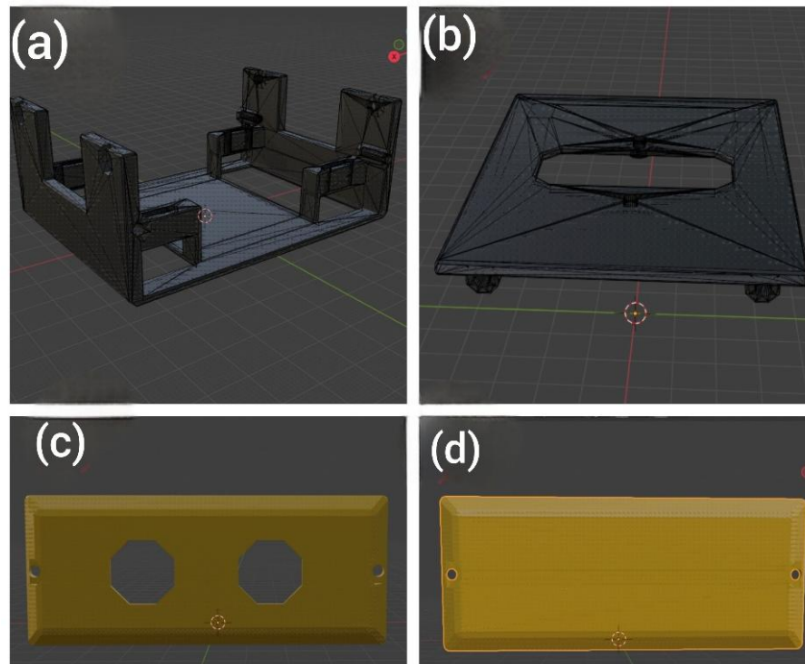


Fig. 13. Modelos 3D de la parte inferior del torso del Robot Humanoide. (a) body bottom 1, (b) body bottom 2, (c) body bottom back, (d) body bottom front.

En la Fig. 14. Se observan los modelos 3D proporcionados de la mano izquierda del robot humanoide la cual está conformada por 8 piezas, las cuales son el left hand 1 1, left hand 1 2 y el left hand 1 3 las cuales conforman la parte superior del hombro, el left hand 2 es la parte superior del brazo y es la pieza en la que van colocados dos servomotores para el movimiento de las manos, las otras piezas que son left hand 3 1, left hand 3 2 y left hand 3 3 forman la parte inferior del brazo y por último el left hand 3 4 es el modelo de la mano y con esto se completa toda la mano izquierda del robot. La mano derecha del robot también está constituida por 8 piezas que son iguales ya que son simétricas a estas y se nombraron como right hand y con una numeración como estos modelos 3D.

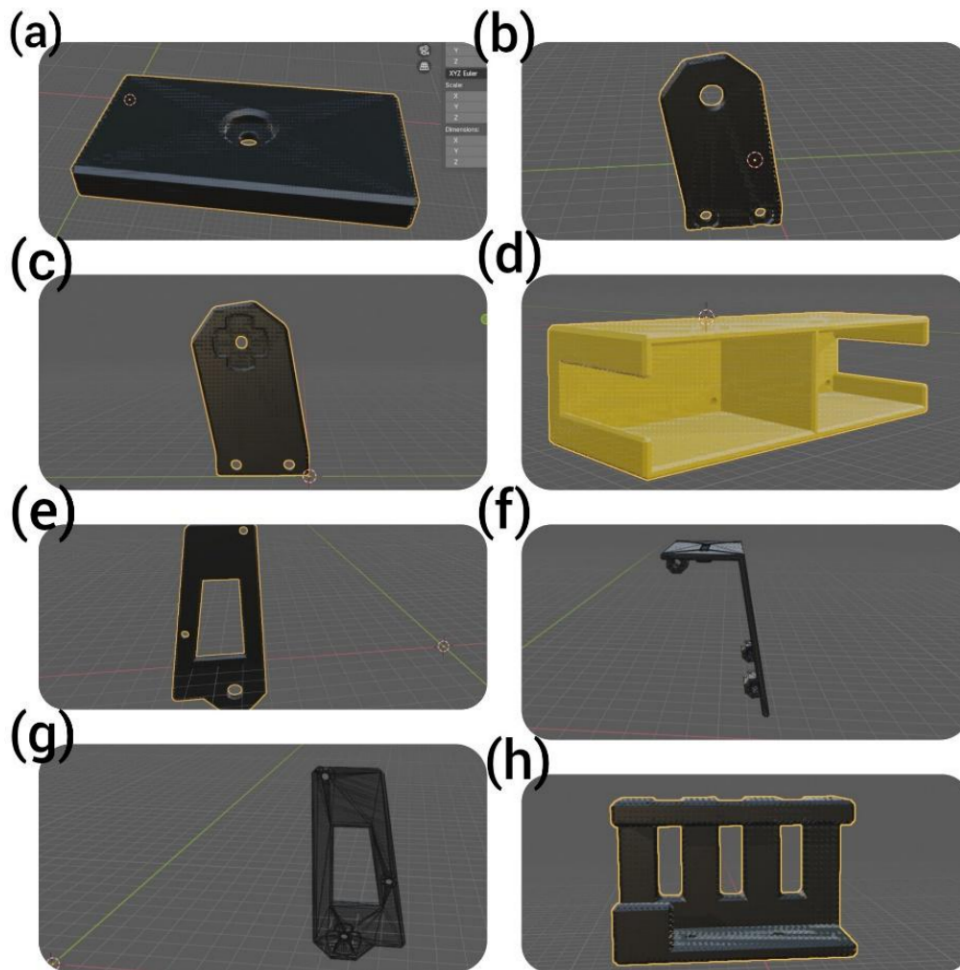


Fig. 14. Modelos 3D de la mano izquierda del Robot Humanoide. (a) left hand 1 1, (b) left hand 1 2, (c) left hand 1 3, (d) left hand 2, (e) left hand 3 1, (f) left hand 3 2, (g) left hand 3 3, (h) left hand 3 4.

La pierna izquierda del robot humanoide está constituida por 14 piezas, las cuales son los modelos 3D que fueron proporcionados y se observan en la Fig. 15. Los cuales son 9 modelos que se muestran en la imagen, los cuales left leg 1 1, left leg 1 2 y left leg 1 3 conforman el pie del Robot, el left leg 2 representa la parte del tobillo de una pierna y en esa pieza se encuentran dos servomotores, el left leg 3 1, left leg 3 2 y left leg 3 3 representa la parte de la pierna donde se encuentra el peroné y la tibia de una pierna humana y el left leg 4 1 con el left 4 2 en conjunto del left leg 4 3 que se muestra en la Fig. 16 es la parte de la rodilla con una parte donde se encuentra el fémur de una pierna humana, además en el left leg 4 1 va colocado un servomotor para el movimiento del robot como si fuera una rodilla de un ser humano.

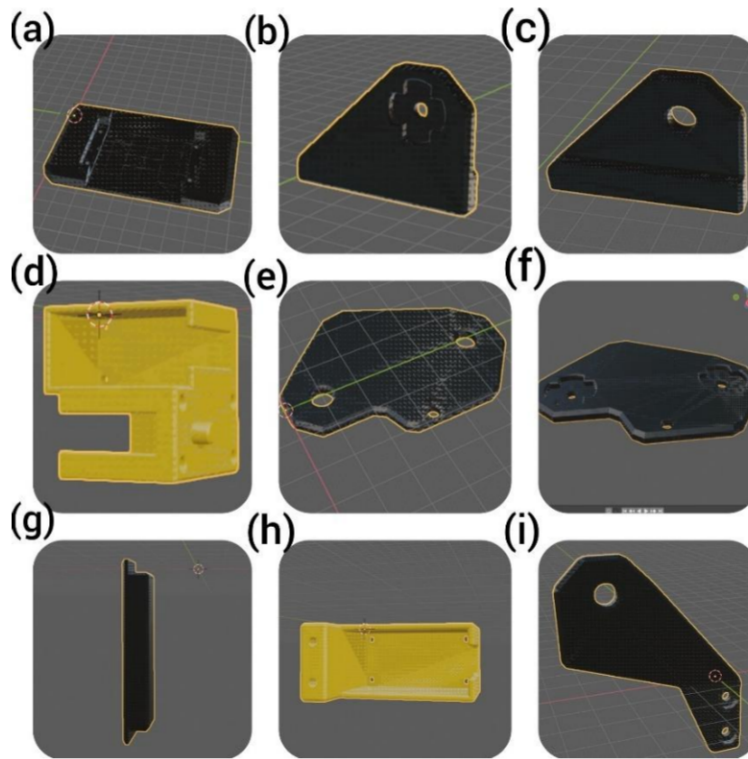


Fig. 15. Modelos 3D de la pierna izquierda del Robot Humanoide. (a) left leg 1 1, (b) left leg 1 2, (c) left leg 1 3, (d) left leg 2, (e) left leg 3 1, (f) left leg 3 2, (g) left leg 3 3, (h) left leg 4 1, (i) left leg 4 2.

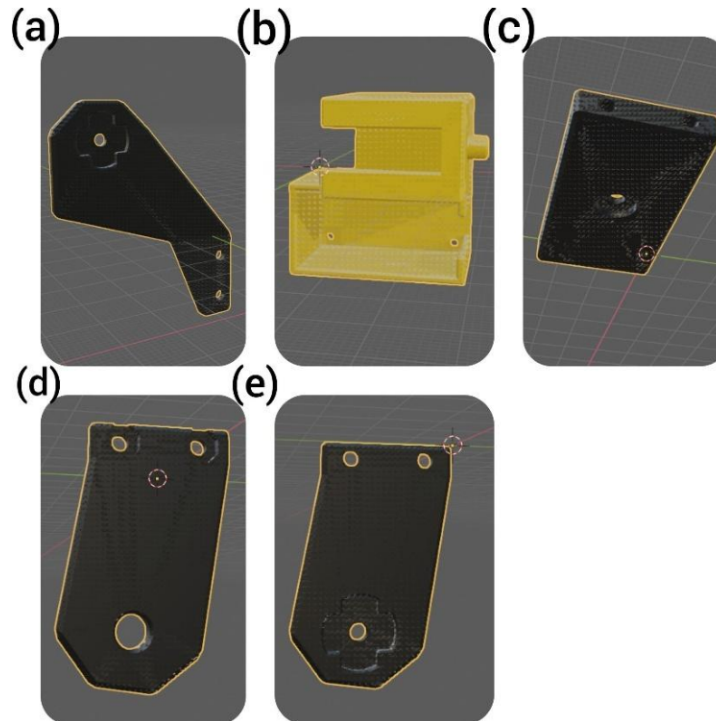


Fig. 16. Modelos 3D de la pierna izquierda del robot Humanoide continuación. (a) left leg 4 3, (b) left leg 5, (c) left leg 6 1, (d) left leg 6 2, (e) left leg 6 3.

En la Fig. 16 podemos observar los últimos 5 modelos 3D de las 14 piezas de la pierna izquierda, la pieza left leg 5 lleva dos servomotores para conectar el movimiento superior de la rodilla con la parte superior de la pierna, por ultimo las piezas left leg 6 1, left leg 6 2 y left leg 6 3 forman la parte superior de la pierna y es la unión para poder unir la pierna con todo el torso del Robot. La pierna derecha del robot también está constituida por 14 piezas que son iguales ya que son simétricas a estas y se nombraron como right leg y con una numeración como estos modelos 3D de esta pierna.

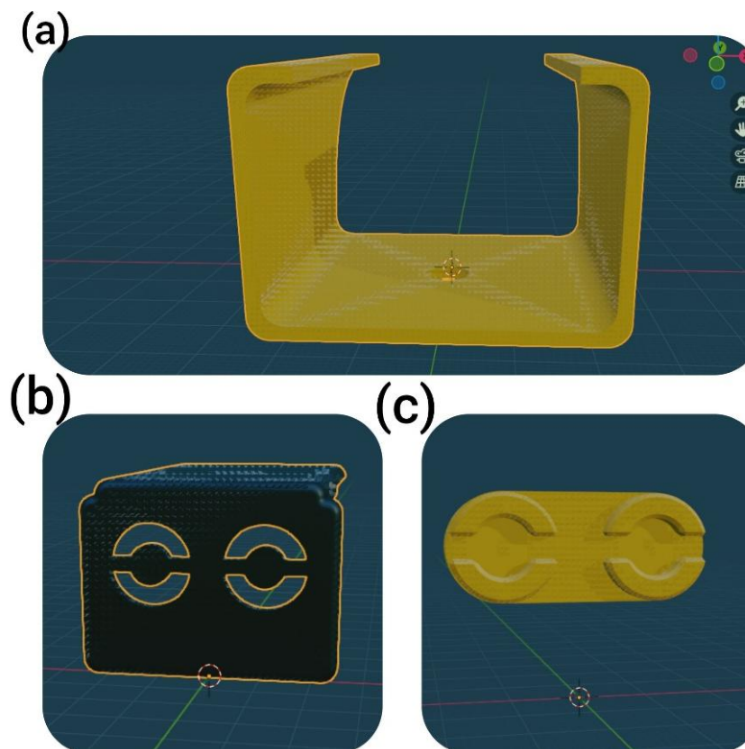


Fig. 17. Modelos 3D de la cabeza del prototipo Robot Humanoide. (a) head 1, (b) head 2, (c) head led.

La cabeza del Robot Humanoide está conformada por 3 piezas las cuales se proporcionaron y podemos observar en la Fig. 17, las cuales son head 1 que es la estructura principal de la cabeza, el head 2 es la parte superior y frontal y por último el head led son los ojos que van ensamblados en el head 2 y todas las piezas conforman la cabeza al robot, la cabeza va unida al robot mediante un servomotor para que tenga movimiento o también puede estar unida al cuerpo sin tener movimiento y es la manera que se va armar el prototipo.

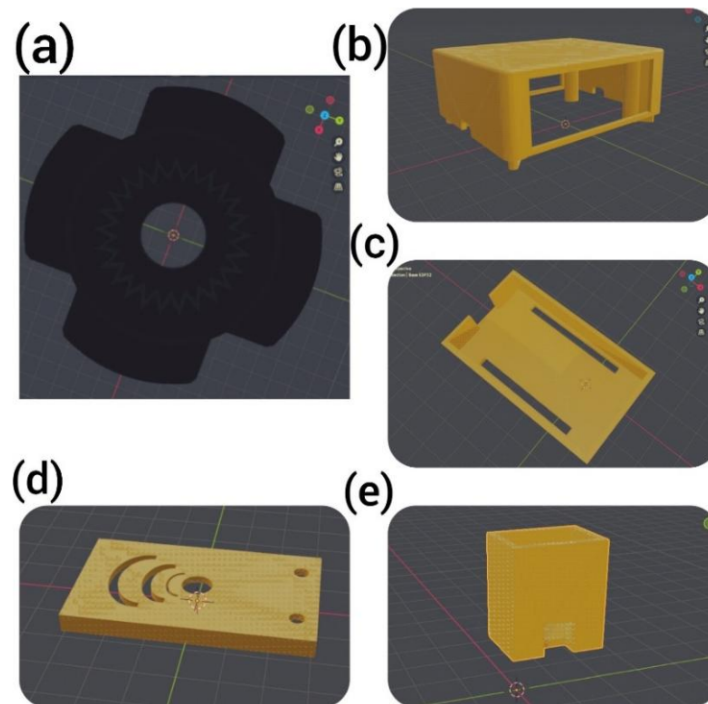


Fig. 18. Modelos 3D para los componentes electrónicos del Robot Humanoide. (a) mg995 horn, (b) Tapa Convertidor 300W 20A, (c) Base ESP32, (d) Tapa ESP32, (e) Apagador tapa.

En la Fig. 18 se muestran parte de los modelos que se realizaron para ser la carcasa de algunos componentes electrónicos o el modelo proporcionado para ser la conexión del movimiento, la pieza mg995 horn tiene una función muy importante para el prototipo ya que es la parte encargada de transmitir el movimiento del servomotor a la pieza que debe mover para que el robot tenga movimiento y desplazamiento, la Tapa Convertidor 300W 20A como su nombre lo indica es para tapar el convertidor de voltaje Step-down que va colocado en la parte trasera del torso superior, la Base ESP32 y la Tapa ESP32 es la carcasa donde va ir colocado el microcontrolador ESP32 que está a la vez ira en la cabeza del robot, por último el Apagador tapa es donde se va a colocar el apagador on/off switch balancín para poder encender y apagar el prototipo, se va a colocar en la parte trasera del torso superior.

Los componentes electrónicos, de alimentación y sensores que son empleados en el prototipo, podemos observar sus modelos 3D en la Fig. 19 a los cuales se les dio las textura y color durante esta etapa del proyecto, a los componentes electrónicos que se les agrego el color fue al microcontrolador ESP32, el controlador PCA9685,

el convertidor de voltaje step-down 300W 20A, al servomotor MG995 y al controlador de voltaje LM2596 DC-DC HW411, también a los componentes de alimentación como la batería 18650 y el apagador on/off switch y además a los sensores HC-SR04 y el sensor DHT11, se les agrego la textura y el color para exportarlos en formato fbx para ser importados al software de Unity para crear la aplicación de realidad aumentada, en especial para ser compartidas para el proyecto [42] ARduinoRide - Prototipos Base.

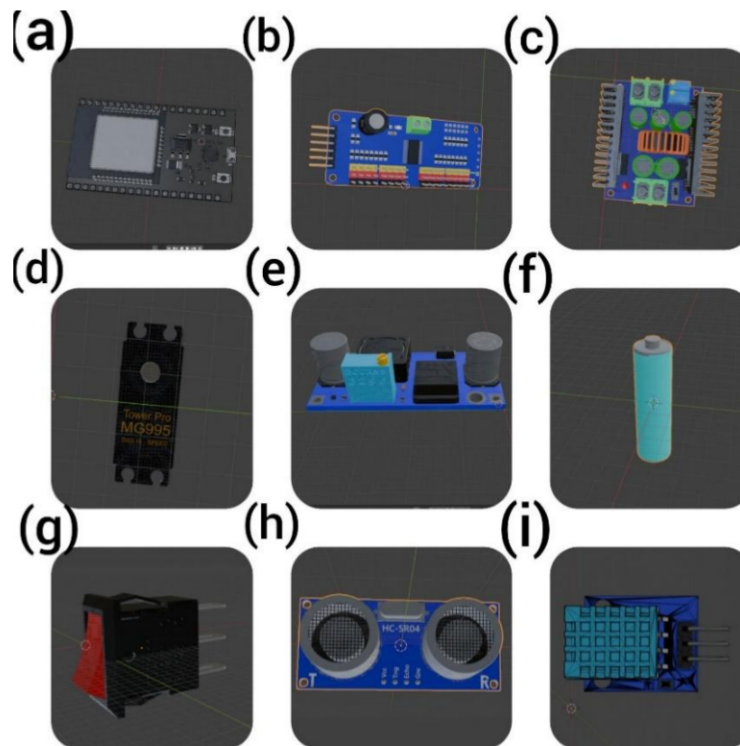


Fig. 19. Modelos 3D de los componentes electrónicos, de alimentación y sensores del prototipo. (a) Microcontrolador ESP32, (b) Controlador PCA9685, (c) Convertidor de voltaje step-down 300W 20A, (d) servomotor MG995, (e) Convertidor de voltaje LM2596 DC-DC HW411, (f) Batería 18650, (g) Apagador on/off switch balancín, (h) Sensor ultrasónico HC-SR04, (i) Sensor de humedad y temperatura DHT11.

Los modelos 3D que conforman todo el prototipo del robot humanoide se clasifican en distintas partes, las cuales son el torso inferior y superior, la cabeza, las manos izquierda y derecha, las piernas izquierda y derecha, como los podemos ver en la Fig. 20, la razón de juntar todos los modelos en el software Blender por sección es para importar cada extremidad del robot en formato fbx para ser implementados en la aplicación de realidad aumentada que se desarrollara en la siguiente etapa del proyecto.

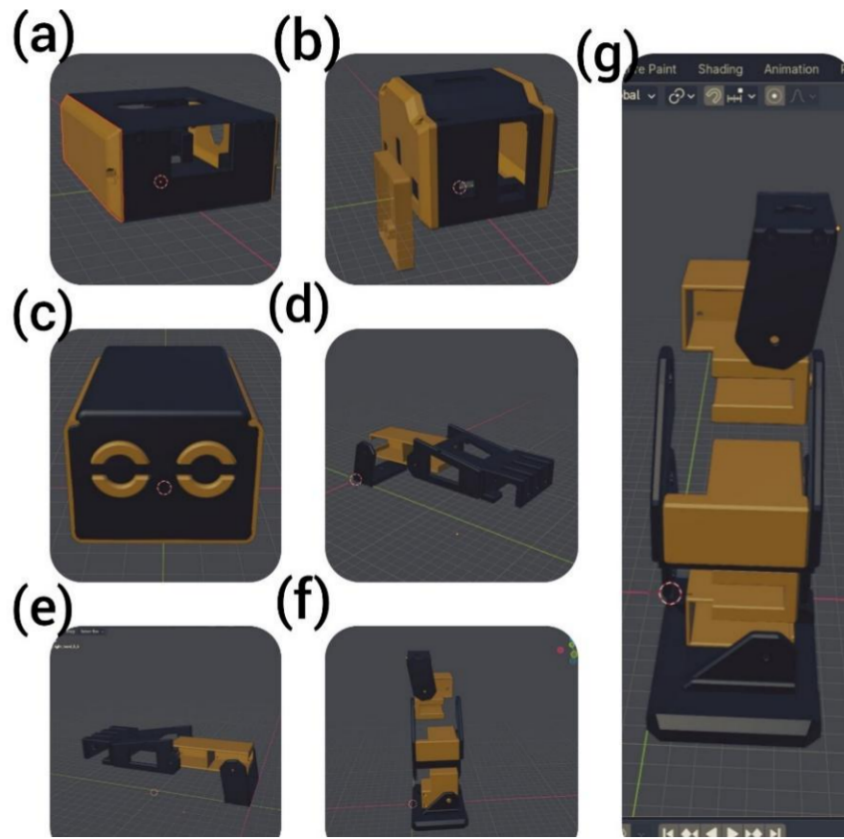


Fig. 20. Piezas del Robot humanoide en 3D. (a) Torso inferior, (b) Torso superior, (c) Cabeza, (d) Mano izquierda, (e) Mano derecha, (f) Pierna izquierda, (g) Pierna derecha.



Fig. 21. Modelo 3D en el software Blender del Robot Humanoide.

En la Fig. 21 podemos observar todos los modelos 3D de los que está conformado el robot humanoide, todas las piezas fueron importadas a un solo archivo de Blender en formato fbx una vez que se les asignó el color y la textura, para ir acomodándolos modelos hasta armar el robot humanoide completo, ya que es requerido para la aplicación de realidad aumentada, una vez que se terminó de armar se exportó en formato fbx.

La cuarta etapa del proyecto fue el diseño y desarrollo de la interfaz de la aplicación de realidad aumentada, lo que se realizó fue crear prototipos de aplicaciones utilizando algunos de los niveles en los que se clasifica la realidad aumentada, como de nivel 1 que es realidad aumentada basada en marcadores y nivel 2 que es realidad aumentada sin marcadores, estos prototipos se realizaron para poder saber cuál era la mejor opción para lo que se deseaba realizar de aplicación para el robot humanoide, como podemos observar en la Fig. 22, donde se muestran 3 prototipos, el prototipo (a) y (b) fue desarrollado con el nivel 2 de realidad aumentada y el (c) fueron desarrollados con el nivel 1, y las versiones de Unity fue 2021.3.1f1 para el prototipo (a) y (c), para el (b) se implementó la versión 2022.2.16f1.

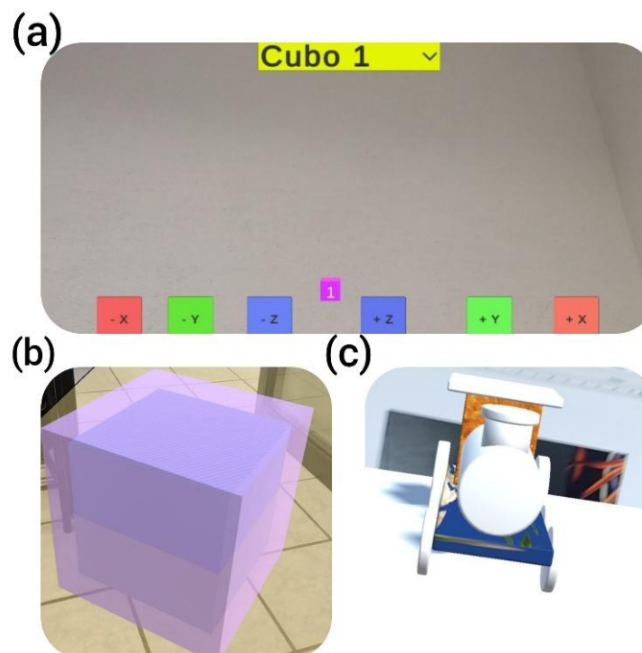


Fig. 22. Prototipos de aplicaciones de realidad aumentada. (a) Aplicación mover cubos, (b) Aplicación Interactuar Cubos, (c) Aplicación marcadores.

Una vez que se realizaron los prototipos se comenzó a diseñar la interfaz de las escenas de la aplicación del manual del armado del robot y a realizar las escenas, para esta aplicación se usó la versión de Unity 2021.3.1f1, primero se creó una escena para ser el menú inicial donde se configuraron 4 botones de opciones y en cada opción nos lleva a otra escena distinta, el primer botón es el del manual de armado donde se podrán visualizar cada uno de los pasos del armado en manera ascendente, el segundo botón nos lleva a dos escenas donde son un menú de cada uno de los pasos, el tercer botón nos llevara a la escena donde se muestra el modelo 3D del robot humanoide en realidad aumentada a través de un marcador y por último el cuarto botón se nombró como información y este nos lleva a una escena que es un menú que contiene otros tres botones y además en la escena principal se agregó el botón para salir de la aplicación, como se puede ver en la Fig. 23.

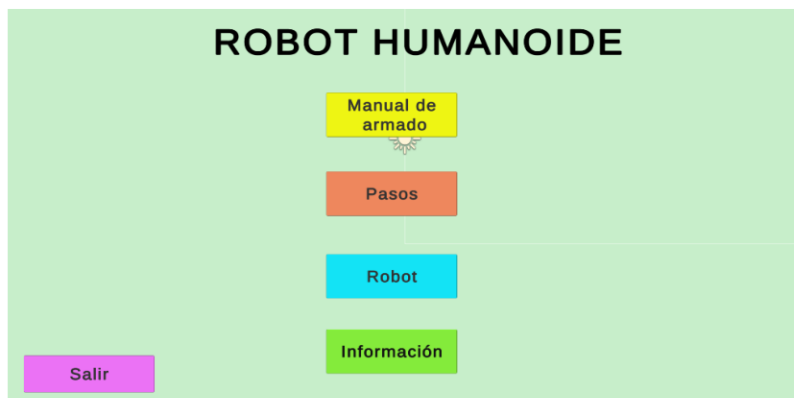


Fig. 23. Menú inicial de la aplicación del manual de armado del robot humanoide.

Se diseñaron y crearon 26 escenas para los pasos del manual del armado del robot humanoide ya que se decidió realizar el armado en 26 pasos, todas las escenas son muy parecidas entre sí, el diseño de estas escenas se agregaron seis botones para el movimiento del modelo 3D del paso correspondiente en X, Y y Z positivo y en X, Y y Z negativo, también se agregaron dos botones para rotar el modelo 3D en Y en el sentido horario y antihorario, también se agregó un botón para regresar al menú inicial y se agregaron dos botones uno para ir al paso anterior y otro al paso siguiente, además otro botón para poder ir al menú de los pasos, se agregó un título para identificar el paso y se agregó las instrucciones de los pasos en un cuadro de

texto y por último el modelo 3D de cada paso del armado del robot humanoide y todas estas escenas son de realidad aumentada, como se puede ver en la Fig. 24.

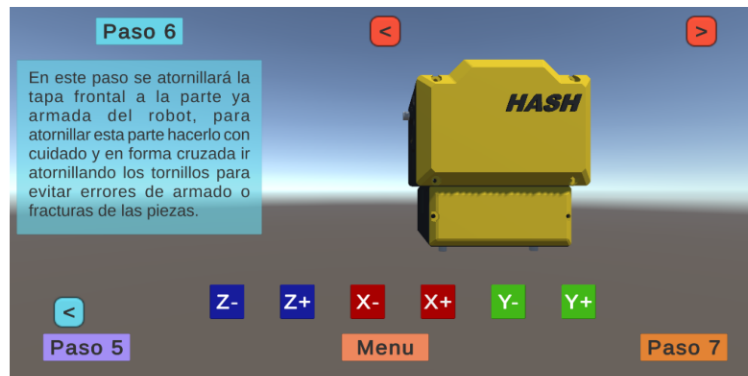


Fig. 24. Escena del paso 6 de la aplicación de realidad aumentada del robot humanoide.

En las dos escenas del menú de los pasos del armado del robot se agregaron un botón por cada escena de los pasos de armado, estos botones nos llevaran a su escena correspondiente del paso asignado y así poder interactuar con el paso que desee. En la escena del tercer botón del robot se mostrará el robot humanoide a través de marcadores, el marcador será colocado en el prototipo del robot humanoide y una vez que lo detecte la cámara del celular lo podremos visualizar, como lo podemos observar en la Fig. 25.

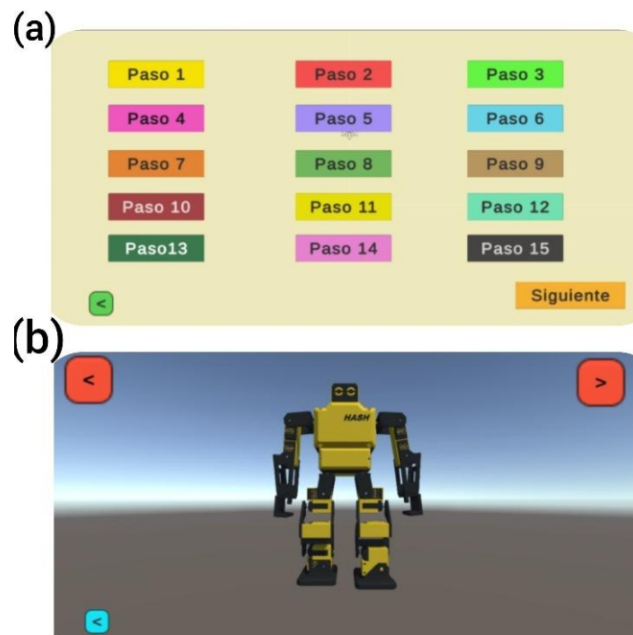


Fig. 25. Escenas de los botones 2 y 3 del menú inicial de la aplicación de realidad aumentada. (a) Escena del menú de pasos, (b) Escena de la realidad aumentada del robot humanoide a través de marcadores.

En la escena del menú de información los tres botones tendrán distintas funciones, el primero nos llevará a una escena donde se mostrará una imagen que contiene los ángulos a los que deben de ir cada uno de los servomotores, el segundo botón tiene un código en Arduino para servir de apoyo para acomodar los ángulos de los servomotores y el ultimo botón son los créditos de la aplicación de realidad aumentada del manual de armado del robot humanoide como se ve en la Fig. 26 y con esto terminamos esta etapa.

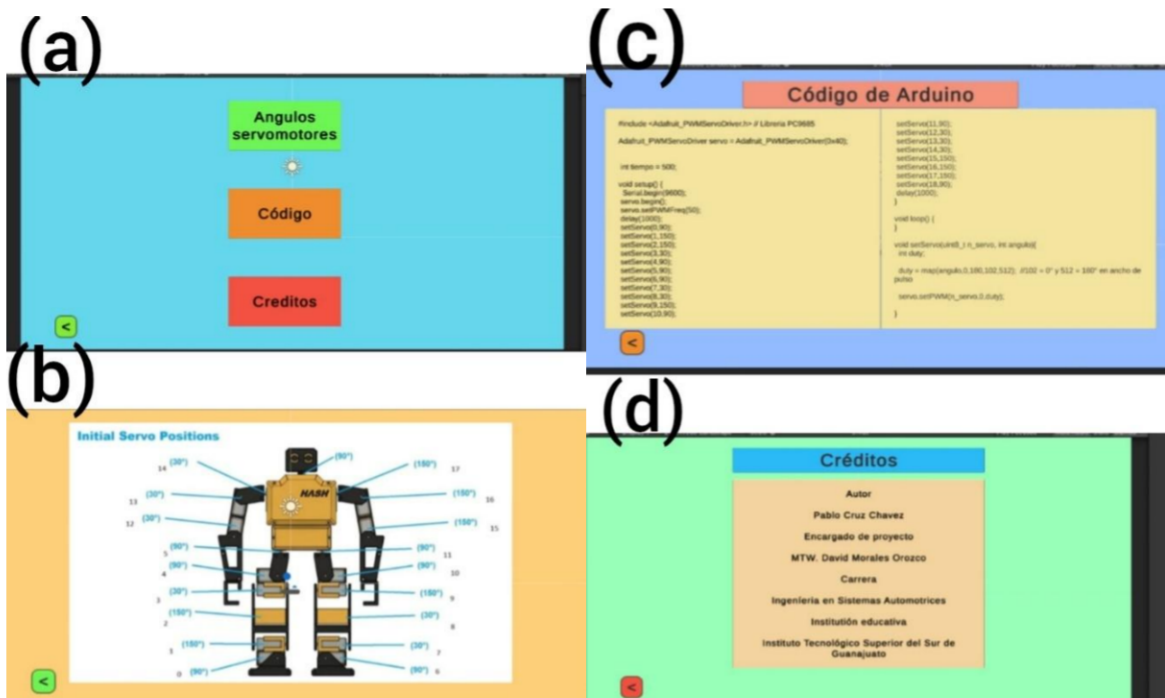


Fig. 26. Escenas del 4 botón del menú inicial de la aplicación. (a) Escena del botón de información, (b) Escena del botón ángulos servomotores, (c) Escena del botón código, (d) Escena del botón créditos.

La quinta etapa del proyecto consiste en la programación de las escenas en C#, en esta etapa se llevó a cabo la programación de las escenas de las aplicaciones de los prototipos como de la aplicación final, esto para agregar la funcionalidad de todos los botones que se diseñaron y se desarrollaron en la etapa anterior del proyecto, la programación se realizó en la versión 2019 de Microsoft Visual Studio, la programación de las aplicaciones de los prototipos no fue muy complicado realizar ya que cada prototipo solo tiene una escena, en la Fig. 27 podemos observar parte de los códigos que se utilizaron para la programación del

prototipo de mover los cubos, en (a) el código es para el botón de activar el cubo 1 o cubo 2, en (b) es el código del movimiento de los cubos, primero se agregan las librerías, luego se declararon todas las variables de los botones, de los cubos y del código anterior, luego un ciclo if para activar el cubo 1 o cubo 2 y se fue haciendo cada función de movimiento, hasta que se terminó de hacer cada una de las funciones de movimiento.

(a)

```
1 using UnityEngine;
2 using UnityEngine.UI;
3 using TMPro;
4
5 public class DropdownItemSelector : MonoBehaviour
6 {
7     public TMP_Dropdown Cubos;
8     public GameObject cube1;
9     public GameObject cube2;
10
11     void Start()
12     {
13         Cubos.onValueChanged.AddListener(delegate { DropdownValueChanged(Cubos); });
14     }
15
16     void DropdownValueChanged(TMP_Dropdown change)
17     {
18         if (change.value == 0)
19         {
20             cube1.SetActive(true);
21             cube2.SetActive(false);
22         }
23         else if (change.value == 1)
24         {
25             cube1.SetActive(false);
26             cube2.SetActive(true);
27         }
28     }
29 }
30
31 }
```

(b)

```
1 using UnityEngine;
2 using UnityEngine.UI;
3 using TMPro;
4
5 public class Movercubos : MonoBehaviour
6 {
7     public TMP_Dropdown Cubos;
8     public GameObject cube1;
9     public GameObject cube2;
10
11     public Button Xplus;
12     public Button Xminus;
13     public Button Yplus;
14     public Button Yminus;
15     public Button Zplus;
16     public Button Zminus;
17
18     private GameObject activeCube;
19     private float movedis = 0.1f;
20
21     void Start()
22     {
23         Cubos.onValueChanged.AddListener(delegate { DropdownValueChanged(Cubos); });
24
25         Yplus.onClick.AddListener(MoveUp);
26         Yminus.onClick.AddListener(MoveDown);
27         Xminus.onClick.AddListener(MoveLeft);
28         Xplus.onClick.AddListener(MoveRight);
29         Zplus.onClick.AddListener(MoveForward);
30         Zminus.onClick.AddListener(MoveBackward);
31
32         DropdownValueChanged(Cubos); // Ensure the correct cube is active at start
33     }
34 }
```

Fig. 27. Código de la escena del prototipo de mover cubos. (a) Código para el botón de selección de uno de los dos cubos, (b) Código del movimiento donde se agregan las librerías y declaración de variables para el movimiento de los cubos.

En el prototipo 2 de la interacción de cubos no se realizó con código si no con componentes de una librería que se instaló y esa fue la razón de usar otra versión, en el prototipo 3 de marcadores se utilizó el siguiente código que se muestra una parte de el en la Fig. 28 en el cual se declararon las librerías y variables para poder realizar las funciones para que reconozca las imágenes para ser utilizados como marcadores que fue utilizada para la activación de los modelos 3D que son implementados en este prototipo, para después agregar las funciones para que se visualicen los modelos 3D de este prototipo de aplicación que nos servirá para .

```
1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using UnityEngine.XR.ARFoundation;
6 using UnityEngine.XR.ARSubsystems;
7
8 public class MultiTargetsManager : MonoBehaviour
9 {
10     [SerializeField] private ARTrackedImageManager aRTrackedImageManager;
11     [SerializeField] private GameObject[] aRModelsToPlace;
12
13     private Dictionary<string, GameObject> aRModels = new Dictionary<string, GameObject>();
14     private Dictionary<string, bool> modelState = new Dictionary<string, bool>();
15     // Start is called before the first frame update
16     void Start()
17     {
18         foreach (var aRModel in aRModelsToPlace)
19         {
20             GameObject newARModel = Instantiate(aRModel, Vector3.zero, Quaternion.identity);
21             newARModel.name = aRModel.name;
22             aRModels.Add(newARModel.name, newARModel);
23             newARModel.SetActive(false);
24             modelState.Add(newARModel.name, false);
25         }
26     }
27 }
```

Fig. 28. Código de la escena del prototipo de marcadores.

Una vez que se realizó el diseño y desarrollo de las escenas en Unity se procedió a realizar la programación de todas las escenas por la escena que se comenzó a realizar la programación fue la del menú inicial, en la cual se declararon las librerías necesarias para el funcionamiento, para luego declarar las variables de los 4 botones y del botón de salir, para proceder a realizar la función de cada botón para ir a la escena deseada y crear la función para salir de la aplicación del manual de armados del robot humanoide, en la Fig. 29 podemos observar parte del código de la escena del menú inicial de la aplicación de realidad aumentada.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 using UnityEngine.UI;
6
7 // Script de Unity (1 referencia de recurso) | 0 referencias
8 public class MenuInicial : MonoBehaviour
9 {
10     public Button Instrucciones;
11     public Button Pasos;
12     public Button RobotMarket;
13     public Button Informacion;
14     public Button Salir;
15
16     // Mensaje de Unity | 0 referencias
17     void Start()
18     {
19         // Asignar funciones a los botones
20         Instrucciones.onClick.AddListener(Paso1);
21         Pasos.onClick.AddListener(PasosUno);
22         RobotMarket.onClick.AddListener(RobotMarkets);
23         Informacion.onClick.AddListener(informacion);
24         Salir.onClick.AddListener(ExitApplication);
25     }
26 }
```

Fig. 29. Código de la escena del menú inicial.

Luego se continuo con la programación de cada una de las escenas de los pasos del manual de armado, para esta parte de programación fue bastante entretenido y un poco complejo, ya que se tuvieron que realizar un archivo de código para cada escena de cada paso y son un total de 26 pasos así que se realizaron 26 scripts, los códigos de cada escena son muy parecidos entre si ya que en todos se declararon las librerías para posteriormente declarar la velocidad de movimiento la cual es un valor flotante que multiplica la velocidad por default, el valor fue de 0.1, también se declaró la velocidad de rotación que también es un valor flotante el cual es 25, para esta velocidad funciona igual que la anterior, posteriormente se declararon cada uno de los botones, primero los seis botones de movimiento para luego los dos botones de rotación, para después declarar el botón de regresar que nos lleva al menú inicial y también se declaró el botón de menú que nos lleva al menú de pasos, todas las declaraciones anteriores se encuentran en todas las escenas, además también se declaró el botón de ir al paso siguiente este botón se declaró en todas las escenas menos en la última, así como también se declaró el botón de paso anterior que solo no está en el primer paso, además se debe de declarar el game object para que pueda funcionar correctamente, este cambia en cada escena, luego se declara cada función de cada uno de los botones, para luego crear cada ciclo void de cada función de movimiento, rotación y en el ciclo de los botones de regresar, paso siguiente, paso anterior y menú de pasos se agrega en el ciclo void el nombre de la escena a la que nos debe de llevar cada botón, en la Fig. 30 podemos observar parte del código de la escena del paso 6.

Posteriormente se continuo con la programación de los menús de los botones de los pasos, se realizó lo mismo que la programación de las escenas anteriores declarar las librerías y variables de los botones que nos llevan a los pasos, de regresar al menú inicial y un botón de ir al siguiente o regresar al anterior página del menú de pasos, luego se declara cada función de cada uno de los botones, para luego crear cada ciclo void de cada función de los botones para que nos lleve a cada escena, se debe de agregar en el ciclo void el nombre de cada escena a la

que nos debe llevar el botón, en la Fig. 31 se puede ver el código del menú de Pasos Uno.

(a)

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using TMPro;
6  using UnityEngine.SceneManagement;
7
8  @ Script de Unity (1 referencia de recurso) | 0 referencias
9  public class Paso6 : MonoBehaviour
10 {
11     public float moveSpeed = 0.1f;
12     public float rotationSpeed = 25f;
13
14     // Botones declarados
15     public Button xmas;
16     public Button xmenos;
17     public Button ymas;
18     public Button ymenos;
19     public Button zmas;
20     public Button zmenos;
21     public Button rotarmas;
22     public Button rotarmenos;
23     public Button regresar;
24     public Button siguiente;
25     public Button anterior;
26     public Button pasos;
27     public GameObject Paso6s;

```

(b)

```

28 void Start()
29 {
30     // Asignamos funciones a los botones
31     xmas.onClick.AddListener(MoveXmas);
32     xmenos.onClick.AddListener(MoveXmenos);
33     ymas.onClick.AddListener(MoveYmas);
34     ymenos.onClick.AddListener(MoveYmenos);
35     zmas.onClick.AddListener(MoveZmas);
36     zmenos.onClick.AddListener(MoveZmenos);
37     rotarmas.onClick.AddListener(RotateXmas);
38     rotarmenos.onClick.AddListener(RotateXmenos);
39     regresar.onClick.AddListener(Regresa);
40     siguiente.onClick.AddListener(Next);
41     anterior.onClick.AddListener(Back);
42     pasos.onClick.AddListener(Submenu);
43 }
44
45 1 referencia
46 void MoveXmas()
47 {
48     Paso6s.transform.Translate(Vector3.left * moveSpeed);
49 }
50
51 1 referencia
52 void MoveXmenos()
53 {
54     Paso6s.transform.Translate(Vector3.right * moveSpeed);
55 }
56
57 1 referencia
58 void MoveYmas()
59 {
60     Paso6s.transform.Translate(Vector3.up * moveSpeed);
61 }
62

```

Fig. 30. Programación de los pasos del armado del robot humanoide, paso 6. (a) Declaración de librerías y variables, (b) Declaración de las funciones de los botones y ciclos void de las funciones.

(a)

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  using UnityEngine.UI;
6
7  @ Script de Unity (1 referencia de recurso) | 0 referencias
8  public class PasosUno : MonoBehaviour
9  {
10     public Button paso1;
11     public Button paso2;
12     public Button paso3;
13     public Button paso4;
14     public Button paso5;
15     public Button paso6;
16     public Button paso7;
17     public Button paso8;
18     public Button paso9;
19     public Button paso10;
20     public Button paso11;
21     public Button paso12;
22     public Button paso13;
23     public Button paso14;
24     public Button paso15;
25     public Button regresar;
26     public Button siguiente;

```

(b)

```

27 void Start()
28 {
29     // Asignar funciones a los botones
30     paso1.onClick.AddListener(Paso1);
31     paso2.onClick.AddListener(Paso2);
32     paso3.onClick.AddListener(Paso3);
33     paso4.onClick.AddListener(Paso4);
34     paso5.onClick.AddListener(Paso5);
35     paso6.onClick.AddListener(Paso6);
36     paso7.onClick.AddListener(Paso7);
37     paso8.onClick.AddListener(Paso8);
38     paso9.onClick.AddListener(Paso9);
39     paso10.onClick.AddListener(Paso10);
40     paso11.onClick.AddListener(Paso11);
41     paso12.onClick.AddListener(Paso12);
42     paso13.onClick.AddListener(Paso13);
43     paso14.onClick.AddListener(Paso14);
44     paso15.onClick.AddListener(Paso15);
45     regresar.onClick.AddListener(Regresa);
46     siguiente.onClick.AddListener(Next);
47 }
48
49 1 referencia
50 public void Paso1()
51 {
52     SceneManager.LoadScene("Paso1"); // Reem
53 }
54
55 1 referencia
56 public void Paso2()
57 {
58     SceneManager.LoadScene("Paso2");
59 }
60
61 1 referencia
62 public void Paso3()
63 {
64     SceneManager.LoadScene("Paso3");
65 }
66

```

Fig. 31. Programación de la escena de los menús de selección de pasos, Pasos Uno. (a) Declaración de librerías y variables, (b) Declaración de las funciones de los botones y ciclos void de las funciones.

Luego se procedió a programar la escena donde se visualizará el robot humanoide a través de un marcador, para lograr esto se agregaron dos componentes uno para agregar la imagen que nos sirve de marcador y el otro componente para que cuando se active la cámara del dispositivo y se visualice la imagen que se imprimió y se colocó en el robot se active el modelo 3D del mismo y podemos interactuar con él, para eso se realizó la programación en C# la cual se puede observar en la Fig. 32, primero se declararon las librerías, la velocidad de rotación y los tres botones que son rotar a la derecha, a la izquierda y de regresar al menú inicial, también se declaró el game object para el modelo, luego se declaró la función de cada botón para posteriormente realizar el ciclo void de cada función, una vez que se terminó de realizar la programación se puede interactuar con el modelo 3D en la cual lo podemos rotar para observarlo desde distintas posiciones.

(a)

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using TMPro;
6 using UnityEngine.SceneManagement;
7
8 @ Script de Unity (1 referencia de recurso) | 0 referencias
9
10 public class RobotMarket : MonoBehaviour
11 {
12     public float rotationSpeed = 25f;
13
14     public Button rotarmas;
15     public Button rotarmenos;
16     public Button regresar;
17     public GameObject Paso26a;
18
19 @ Mensaje de Unity | 0 referencias
20
21 void Start()
22 {
23     rotarmas.onClick.AddListener(RotateXmas);
24     rotarmenos.onClick.AddListener(RotateXmenos);
25     regresar.onClick.AddListener(Regresa);
26 }
27 }
```

(b)

```
23 void RotateXmas()
24 {
25     Paso26a.transform.Rotate(Vector3.down * rotationSpeed);
26 }
27
28 void RotateXmenos()
29 {
30     Paso26a.transform.Rotate(Vector3.up * rotationSpeed);
31 }
32
33 public void Regresa()
34 {
35     SceneManager.LoadScene("MenuInicial");
36 }
37
38 }
```

Fig. 32. Programación de la escena del marcador del robot humanoide. (a) Declaración de librerías, variables y funciones de los botones, (b) Ciclos void de las funciones.

Al final se programaron la escena de información la cual tiene 3 botones, para esta se declararon las librerías, variables, funciones de los botones y se crearon los ciclos de cada función de los botones, luego se programaron las 3 escenas de los botones de la escena de información las cuales son ángulos servomotores, código de los servomotores y créditos de la aplicación para esto fue muy similar la programación ya que solo se programó un solo botón para cada escena el cual fue para regresar a la escena anterior de información, en la Fig. 33 podemos observar parte del código de las 4 escenas programadas.

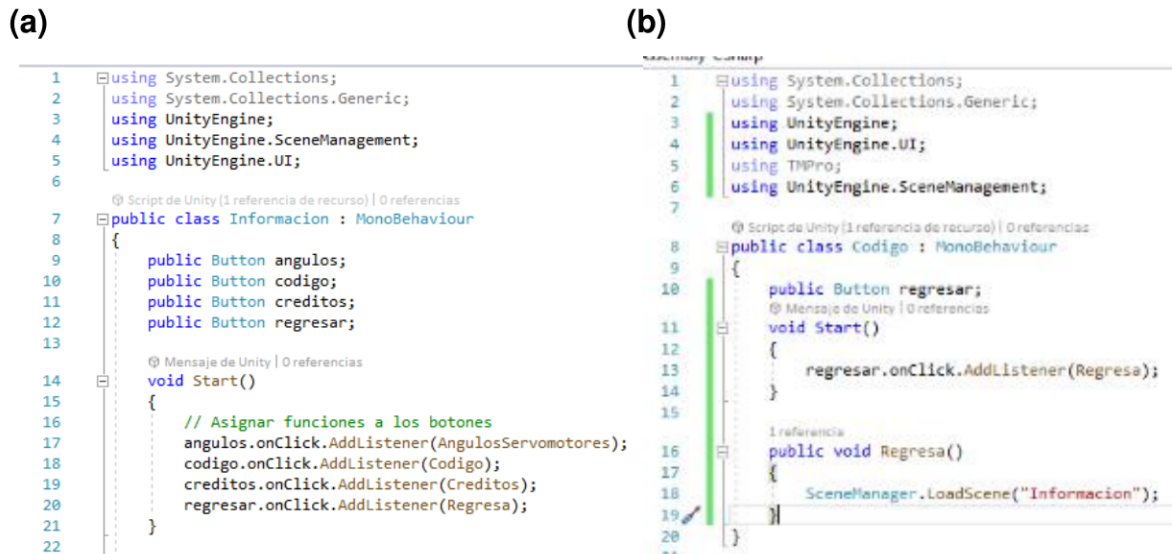


Fig. 33. Código de la programación de las escenas. (a) Código de la escena de información, (b) Código de la escena de Código de Arduino para el armado del robot humanoide.

La sexta etapa del proyecto es la documentación de la información del diseño e interacción del prototipo, lo que se realizó en la etapa primero fue la impresión de los modelos que se realizaron para el robot humanoide, para una vez tener todas las piezas del robot se comenzó a realizar el armado del robot humanoide y también se realizó un manual para el armado del robot, este manual fue implementado en la aplicación de realidad aumentada, también se realizó programación en Arduino para colocar los servomotores en los ángulos necesarios y para el funcionamiento del robot y se desarrolló la aplicación para el control del robot en la plataforma MIT App Inventor.

Para poder mandar a imprimir a la impresora 3D, primero se tuvieron que configurar los modelos 3D en el software UltiMarket Cura se configuro la impresora para luego acomodar en la cama de la impresora los modelos, primero se configuro la base del convertidor en un archivo y en otro la tapa del convertidor, la base y tapa del ESP32, además también se configuro la caja del apagador en otro archivo, se seleccionaron los parámetros deseados como el soporte, tipo de soporte, cantidad de relleno y grosor de impresión, una vez realizado eso se realizó la transformación de los archivos al formato que necesita la impresora, al igual que se genera una simulación, los archivos se guardaron en formato .gcode, en la Fig. 34 podemos

observar la simulación de impresión de los cuatro primeros modelos configurados para imprimirlos.

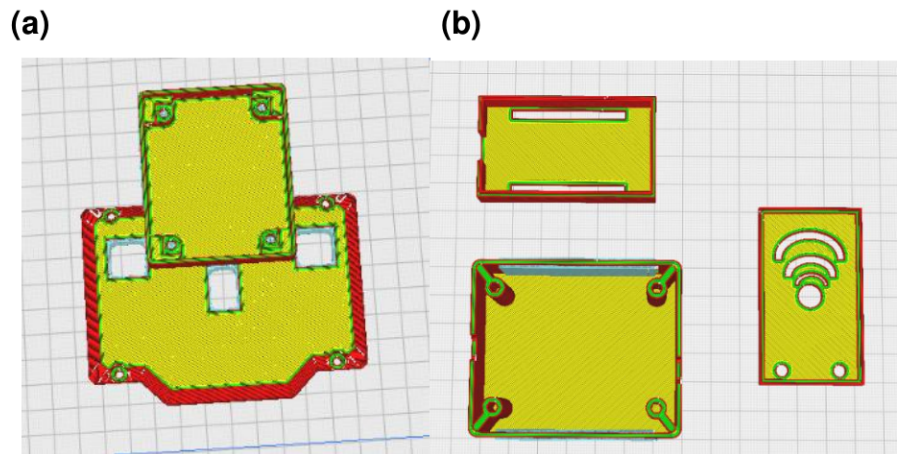


Fig. 34. Modelos 3D configurados para impresión 3D en UltiMarket Cura. (a) Base convertidor, (b) Tapa convertidor, base y tapa ESP32.

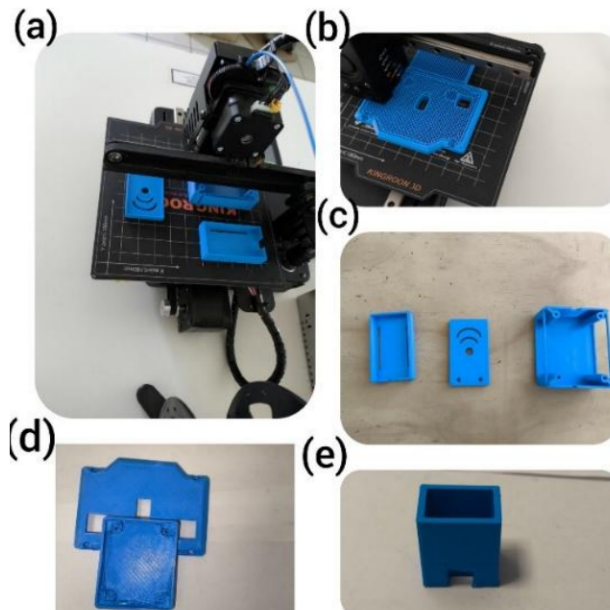


Fig. 35. Modelos 3D impresos en PLA en la impresora Kingroon K3ps. (a) Imprimiendo la tapa del convertidor, base y tapa del ESP32, (b) Imprimiendo la base del convertidor, (c) Modelos impresos de la tapa del convertidor, tapa y base del ESP32, (d) Modelo impreso de la base del convertidor, (e) Modelo impreso del contenedor del apagador del robot.

Posteriormente se mandó a imprimir cada archivo a la impresora 3D Kingroon K3ps que pertenece a la coordinación de sistemas automotrices, los primeros dos archivos que son los que se muestran en la Fig. 34 tardaron en imprimirse un total de cuatro horas y treinta minutos aproximadamente y la otra pieza tardo como

cuarenta y cinco minutos, una vez que se realizaron las impresiones se procedió a limpiar las piezas ya que algunas piezas tenían material de soporte, en la Fig. 35 podemos observar la impresora 3D Kingroon K3ps durante la impresión de alguno de los modelos y los modelos una vez terminados de imprimirlos.

Una vez que se terminó de imprimir todas las piezas se comenzó a armar el robot humanoide, para esto se tuvo que realizar un código en Arduino IDE para configurar cada uno de los servomotores al ángulo que deben de ser colocados para poder tener un buen funcionamiento, este código puede ser encontrado en la sección de anexos por si gusta consultarlo, anexo 2, al igual que una imagen con el número y ángulo que se asignó a cada servomotor es la Fig. 41, anexo 3, por lo primero que se empezó a armar fue el torso inferior del robot en el cual se agregaron dos servomotores que nos servirán para conectar las piernas y darles movimiento, los servomotores y las piezas todas fueron atornilladas con cuidado y con los tornillos adecuados, posteriormente se procedió a armar el torso superior se juntó al torso inferior atornillando la parte superior del torso inferior con la base del torso superior, luego se atornillaron las demás piezas junto con dos servomotores que serán la conexión a las manos del robot, con la excepción de la parte trasera, terminando de armar el torso se comenzó a armar la mano derecha se fue armando de la parte del hombro hacia la parte de la palma de la mano, se agregaron dos servomotores uno en la parte del hombro y otro en la parte del codo, luego se procedió a armar la mano izquierda la cual se armó de la misma manera que la mano derecha, una vez armadas las dos manos se atornillaron al torso del robot, después se armó la pierna derecha la cual se comenzó a armar de la parte inferior hasta llegar a la parte superior, en la pierna se ensamblaron un total de cinco servomotores los cuales se colocaron al ángulo necesario, para posteriormente armar la pierna izquierda de la misma manera que la pierna derecha solo que cambiando el ángulo de los servomotores, luego que se armaron las piernas se colocaron al torso del robot, para posteriormente armar la cabeza del robot humanoide y con esto tenerlo armado casi en su totalidad como se muestra en la Fig. 36, además también se realizó el manual para armarlo el cual se encuentre en la sección de anexos, anexo 1.

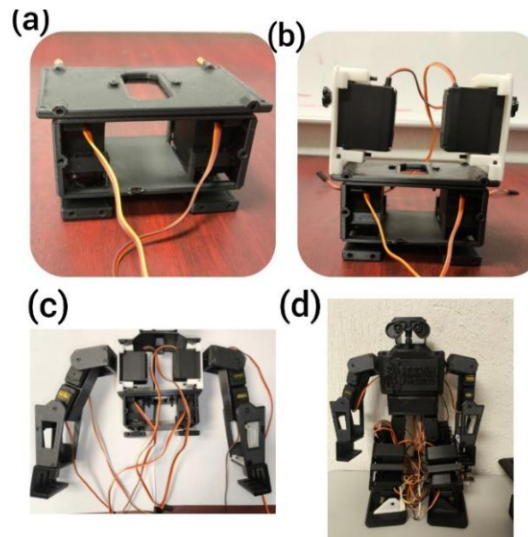


Fig. 36. Armado del robot humanoide. (a) Torso inferior, (b) Torso superior, (c) Manos del robot, (d) Piernas y cabeza del robot.

Posteriormente se realizó un código en Arduino IDE para que el robot humanoide tenga movimiento del cual lo podemos observar una parte en la Fig. 37 este código hace que los servomotores de las manos tenga movimiento, cada servomotor se desplaza cinco grados cada que se mueve, así que cada servomotor se mueve por separado, se realizó con un ciclo for la programación del movimiento de los servomotores para los cuales se realizaron dos ciclos for para cada servomotor, uno para que se desplace hasta determinado ángulo y otro para que regrese a la posición inicial.

```
(a)
1  #include <Adafruit_PWMServoDriver.h> // Libreria PC9685
2
3  Adafruit_PWMServoDriver servo = Adafruit_PWMServoDriver(0x40);
4
5
6  int tiempo = 500;
7
8  void setup() {
9    Serial.begin(9600);
10   servo.begin();
11   servo.setPWMFreq(50);
12   delay(1000);
13   setServo(0,30);
14   setServo(1,30);
15   setServo(2,150);
16   setServo(3,150);
17   setServo(4,30);
18   setServo(5,150);
19   delay(1000);
20 }
21

(b)
75
76 for(int i = 60; i>=30; i = i-5){
77   setServo(4,i);
78   delay(tiempo);
79 }
80
81 for(int i = 120; i<=150; i = i+5){
82   setServo(5,i);
83   delay(tiempo);
84 }
85
86 void setServo(uint8_t n_servo, int angulo){
87   int duty;
88
89   duty = map(angulo,0,180,102,512); //102 =
90
91   servo.setPWM(n_servo,0,duty);
92
93
94 }
```

Fig. 37. Código en Arduino IDE para el movimiento de las manos del robot humanoide. (a) Declaración de la librería y variables, (b) Funciones finales del ciclo void Loop y función del ciclo void setServo.

También se realizó el diseño de la interfaz y programación de la aplicación en MIT App Inventor, que nos sirve de control remoto para el funcionamiento del robot humanoide, para así poder ejecutar ciertas partes del código cuando se mande una instrucción para esto se diseñó la interfaz en la cual se agregó un botón de lista con el cual podremos conectar la aplicación a la placa ESP32, tres botones uno para activar la función de caminar, otro para activar la función de parar y en el último para activar la función de posición inicial del robot, para la programación de los botones esta se realizó en bloques en MIT App Inventor en la cual se agregaron las funciones de cada botón, el botón de caminar mandara un 1, el botón parar mandara un 0 y el botón posición inicial un -1, en la Fig. 38 podemos observar el diseño de la aplicación y parte del código.

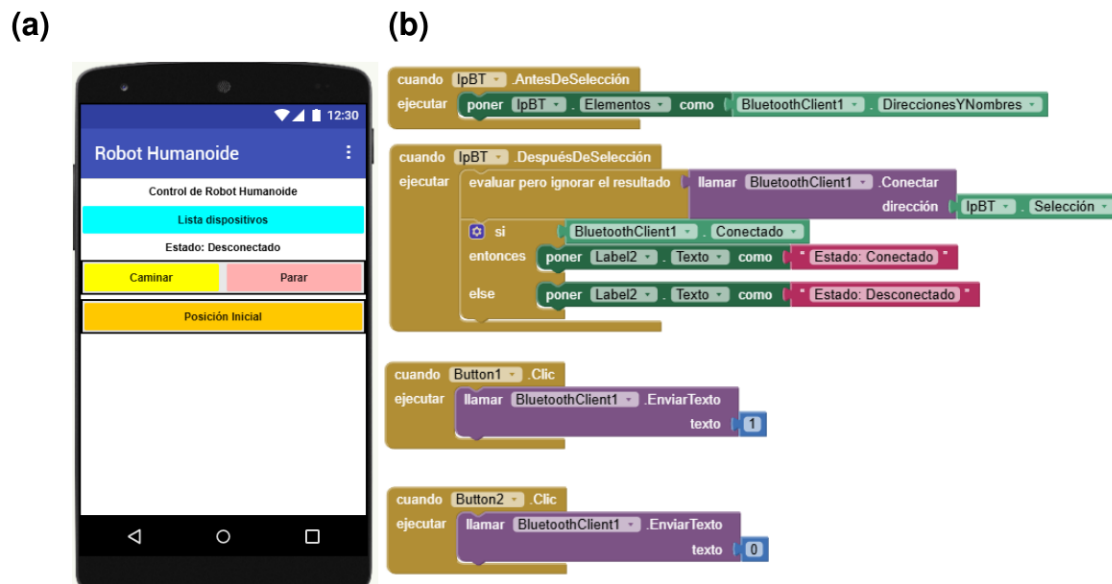


Fig. 38. Aplicación de control remoto para el robot humanoide. (a) Interfaz de la aplicación, (b) Programación en bloques de la aplicación.

La séptima etapa del proyecto es la realización de las pruebas de lo desarrollado para mejoras y la entrega final del prototipo y de la aplicación, en lo que consistió esta etapa fue probar la aplicación de realidad aumentada y probar el funcionamiento del robot humanoide en los distintos modos, al igual también probar la aplicación de control remoto para el robot humanoide, para verificar que no existen errores o bugs en la aplicación de realidad aumentada y la del control

remoto, así como errores en el robot, para proceder a corregirlos y poder entregar ser entregadas las aplicaciones y el robot humanoide.

Lo primero que se realizó fue hacer pruebas a la aplicación de realidad aumentada que consistió en verificar que los botones funcionaran correctamente que nos lleven a la escena para el que fue diseñado al igual que los botones de regresar y el de salir funcionen bien, además que en las escenas de realidad aumentada funcione correctamente la cámara, los modelos 3D se visualicen correctamente y que los botones de desplazamiento y de rotación lo hagan correctamente en la dirección establecida, también se verificó que la escena del marcador de la escena del robot humanoide se visualice correctamente el modelo 3D, en esta parte si se tuvieron problemas porque el robot salía en distintas posiciones al detectar el marcador, en la Fig. 39 se pueden observar unas escenas de la aplicación de realidad aumentada del manual de armado del prototipo del robot humanoide, además para saber la compatibilidad de la aplicación con el dispositivo celular consultarlo en ARCore <https://developers.google.com/ar/devices?hl=es-419> [43].

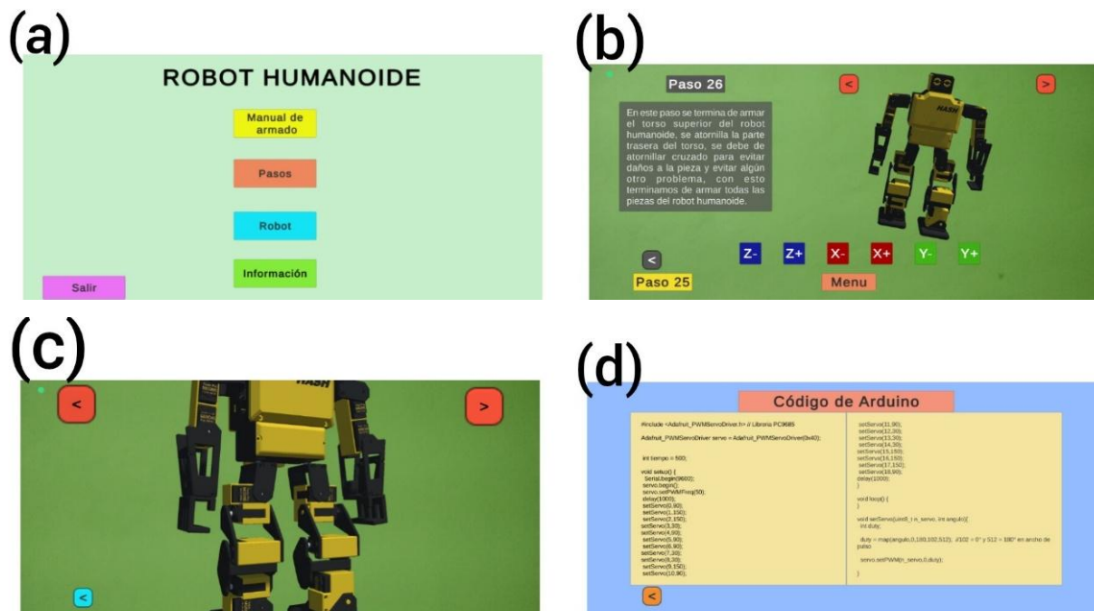


Fig. 39. Escenas de la aplicación de realidad aumentada. (a) Escena del menú inicial, (b) Escena del paso 26, (c) escena del marcador del robot humanoide, (d) Escena del código de Arduino para los servomotores.

Posteriormente se realizaron pruebas al prototipo del robot humanoide que funcione correctamente el código para observar el movimiento del robot, que se detenga y

mandarlo a la posición inicial y verificar que no existan errores a la hora de ejecutar el código, también se verificó que funcionara correctamente la aplicación del control remoto para el control del robot, se verificó que los botones mandaran correctamente las instrucciones al robot, una vez que se terminó de realizar la pruebas no se encontraron errores, en la Fig. 40 podemos observar al robot humanoide y la aplicación del control remoto, una vez que se corrigió todo se procedió a entregar la aplicación de realidad aumentada, la aplicación del control remoto para el robot y el prototipo del robot humanoide.

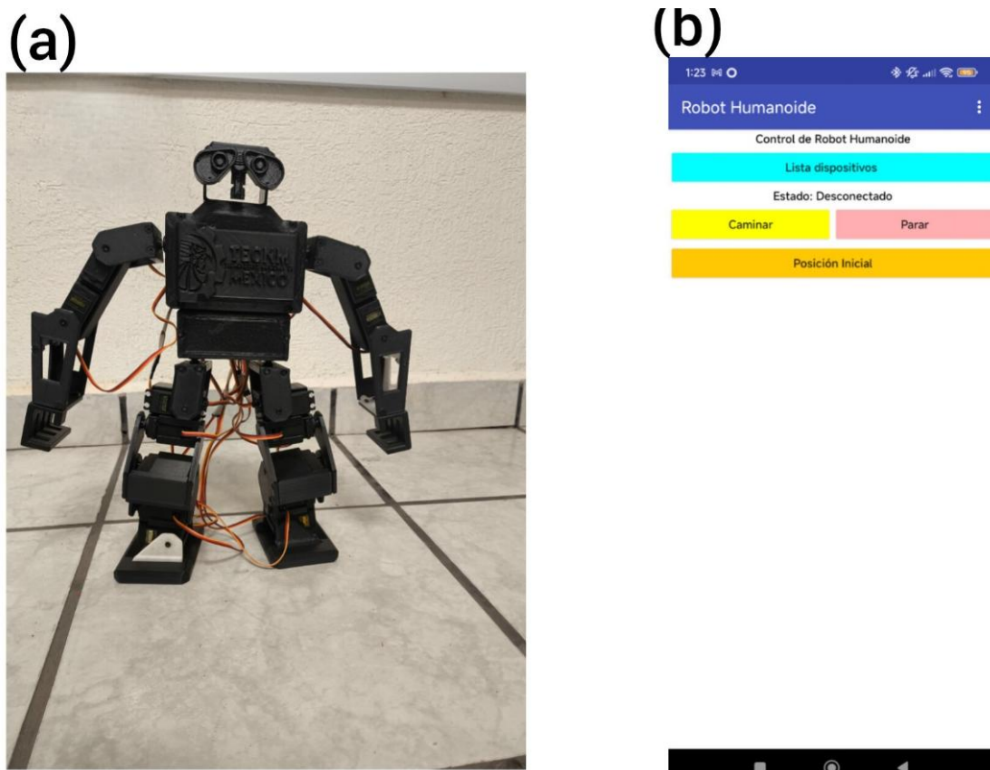


Fig. 40. Prototipo y aplicación. (a) Prototipo del robot humanoide, (b) Aplicación del control remoto del robot humanoide.

Capítulo 7

Análisis de Resultados

Con los resultados obtenidos y descriptos en el Capítulo 6 Resultados, se presenta el análisis de una manera detallada de los aspectos clave de la realización de este proyecto ARduinoRide – prototipo robot humanoide, primero se comparará los resultados obtenidos con los antecedentes para poder saber la similitud del proyecto con los resultados de otras investigaciones y con los objetivos que se plantearon para este proyecto.

Los resultados obtenidos son similares a los antecedentes con respecto a la función de la aplicación de realidad aumentada, ya que en los antecedes son aplicaciones que son empleadas en el ámbito educativo al igual que la que se realizó, la aplicación tendrá implementación en este ámbito en específico para solo una materia, la cual es sistemas embebidos para la carrera de Ingeniería en sistemas automotrices en la cual se tendrá las herramientas para armar el prototipo del robot humanoide, las aplicaciones de los antecedentes algunas son en específico para solo una materia y otras tienen la funcionalidad para la enseñanza de distintas materias para el aprendizaje.

Los objetivos que se describieron en el Capítulo 4 todos fueron alcanzados con los resultados, el primer objetivo fue fortalecer y ampliar los conocimientos sobre el modelaje 3D mediante la utilización de Blender, se alcanzó este objetivo a través de la etapa uno de los resultados en la cual se tuvo la capacitación en Blender en la cual se conoció la interfaz de este software además también se realizaron distintos modelos 3D para comprender los aspectos técnicos y artísticos para la creación de los futuros modelos que se crearon.

El segundo objetivo fue realizar la capacitación de manera integral a través de la plataforma de Unity Lear, este objetivo se cumplió a través de la etapa dos de los resultados en la cual se tomó un curso para la creación de aplicaciones de realidad

aumentada, se crearon distintos tipos de prototipos de realidad aumentada utilizando algunos de los niveles de realidad aumentada como el nivel de marcadores y el nivel de realidad aumentada sin marcadores y con esto se adquirieron los conocimientos prácticos y teóricos necesarios para crear la aplicación de realidad aumentada.

El tercer objetivo fue el diseño y elaboración de los modelos 3D del robot humanoide utilizando el software CAD Blender, este objetivo fue alcanzado a través de la etapa tres en la cual se realizó el diseño y creación de algunos modelos 3D del robot humanoide para que sea funcional y preciso el diseño final, así como darles la textura y color a todos los modelos del robot y también la importación de los modelos en formato STL y fbx.

El cuarto objetivo fue diseñar y desarrollar la interfaz gráfica de la aplicación de realidad aumentada en el motor de videojuegos Unity, este objetivo se cumplió a través de la etapa cuatro de los resultados en la cual se realizaron todas las escenas de la aplicación del manual del robot humanoide en realidad aumentada, en la cual se agregó diseño, botones y la interfaz para la funcionalidad de la aplicación, en las escenas de menú inicial, menú de pasos, menú de información y de las escenas de los ángulos de los servomotores, del código de los servomotores y los créditos de la aplicación, también se agregó la funcionalidad de las cámaras de realidad aumentada a las escenas de los pasos así como los modelos 3D y también a la escena del marcador del robot.

El quinto objetivo fue la realización de la programación en la plataforma IDE Visual Studio en C# para la interfaz gráfica desarrollada en Unity para habilitar las funcionalidades, este objetivo se alcanzó a través de la etapa cinco de los resultados en la cual se realizó la programación de cada escena de la aplicación para esto se realizó un script por separado para cada escena, además el código fue similar, primero fue la declaración de librerías, nombre del script, declaración de variables, declaración de las funciones de los botones y por ultimo las funciones de cada ciclo

void para el funcionamiento de los botones de cambio de escena, de movimiento y rotación de los modelos 3D.

El sexto objetivo fue el desarrollo de la programación en la plataforma IDE Arduino para la interacción humano / maquina (IHM) del prototipo Robot Humanoide, este objetivo se cumplió a través de la etapa seis de los resultados en la cual se armó el robot humanoide, se realizó la programación para el movimiento del robot humanoide en Arduino y se desarrolló la aplicación para el control remoto del robot humanoide, para una fluida y eficiente comunicación con el robot.

El séptimo objetivo fue el ejecutar las pruebas integrales a la aplicación de realidad aumentada en conjunto del prototipo del robot humanoide para evaluar el desempeño de esta, este objetivo se alcanzó a través de la etapa siete en la cual se realizaron las pruebas a la aplicación de realidad aumentada del manual de armado del robot humanoide, al prototipo del robot humanoide y a la aplicación del control remoto, los errores que surgieron fueron corregidos y una vez esto las aplicaciones tiene estabilidad y una buena funcionalidad.

El octavo objetivo es la documentación de manera sistemática del proceso de desarrollo del robot humanoide y de la aplicación de realidad aumentada, este objetivo se alcanzó con la etapa seis de los resultados en la cual se realizó la documentación del manual del armado del robot humanoide para ser implementado en la aplicación de realidad aumentada. El noveno objetivo fue la conclusión del proyecto de realidad aumentada con la entrega de la aplicación de realidad aumentada con la versión optimizada y funcional para sistemas embebidos, este objetivo se logró con la etapa siete de los resultados en la cual se entregaron la aplicación de realidad aumentada y del control remoto y la entrega del prototipo del robot humanoide.

El objetivo general es el desarrollo de una aplicación de realidad aumentada en la plataforma Unity para el sistema de ARduinoRide del prototipo Robot Humanoide, para el impacto positivo en la formación de estudiantes, al facilitar elementos

complejos e impulsar sus habilidades tecnológicas en la impresión 3D, la realidad aumentada y la industria 4.0 en la ingeniería de Sistemas Automotrices, se cumplió el objetivo general a través del alcance de todos los objetivos específicos, con esto en conjunto se logró desarrollar la aplicación de realidad aumentada del manual del robot humanoide en la cual se emplearon los modelos 3D del robot humanoide desarrollados en el software Blender en el cual se les dio el color y textura a las piezas utilizadas, se programó la aplicación de realidad aumentada para ser utilizadas todas sus funciones, para después ser probada la aplicación y ser entrega para sistemas embebidos.

Capítulo 8

Conclusiones y trabajo a futuro

Durante el desarrollo del proyecto de residencia profesional ARduinoRide – Prototipo Robot Humanoide, se logró cumplir con todos los objetivos que se plantearon al inicio del proyecto, se consolidaron conocimientos en la realización y programación de aplicaciones y modelo 3D a través del software Unity, Visual Studio y Blender, además se adquirieron habilidades en el manejo de herramientas digitales.

El impacto que se tuvo con este proyecto fue la creación de la aplicación de realidad aumentada del manual de armado del robot humanoide para la materia de sistemas embebidos de la carrera de Ingeniería en Sistemas Automotrices para la mejora del aprendizaje del alumnado y de la impartición de la materia, en conclusión, ha sido una experiencia muy gratificante ya que se realizaron todo lo planeado al principio del proyecto y se adquirieron nuevas competencias que servirán para próximos desafíos y metas en el ámbito laboral en el futuro.

Los conocimientos adquiridos durante el transcurso de la carrera de Ingeniería en Sistemas Automotrices que fueron implementados para la realización de este proyecto fue programación, dibujo asistido por computadora y circuitos eléctricos, todos estos conocimientos fueron adquiridos en materias como programación básica, programación aplicada, sistema embebidos, fundamentos de dibujo, análisis de circuitos eléctricos, electrónica analógica y entre otras materias, todos estos conocimientos fueron muy indispensables para realizarlo y para la capacitación de las herramientas usadas en el proyecto.

El trabajo a futuro que se puede continuar realizando con este proyecto es mejorar la programación en Arduino para el movimiento del robot humanoide hacerla mucho más fluida y los movimientos hacerlos lo más parecidos a un ser humano, además se le pueden agregar otro tipo de movimiento como agacharse o bailar, además a parte de los sensores que se integraron como el sensor ultrasónico HC-SR04 y el

sensor de humedad y temperatura DHT11 se pueden agregar sensores más afines para un robot humanoide para un mejor funcionamiento del mismo.

También para trabajo a futuro de este proyecto se encuentra la aplicación de realidad aumentada del manual de armado del robot humanoide, lo que se puede agregar es nuevas escenas de como conectar los futuros sensores, así como mejorar o cambiar la forma en que se muestran los modelos 3D en las escenas de los pasos de armado del robot, además se pueden agregar escenas en la aplicación en la cual nos muestre los datos recopilados de los sensores.

En la aplicación de control remoto del robot humanoide se puede mejorar la interfaz de cómo se muestran los botones y así como agregar nuevos botones para las funciones que pueden ser agregadas al prototipo del robot humanoide. Estas son algunas cosas que pueden trabajarse a futuro en el proyecto ARduinoRide – Prototipo Robot Humanoide.

Referencias bibliográficas

- [1] J. L. Ordóñez, «Realidad Virtual y Realidad Aumentada,» *Revista Digit. de ACTA*, vol. 1, nº 1, pp. 1- 21, 2020.
- [2] M. O. K. E. C. R. y. J. C. O. X. Basogain, «“Realidad Aumentada en la Educación: una tecnología emergente”,» Bilbao, 2007.
- [3] C. R. Bello, «“La realidad aumentada: lo que debemos conocer”,» *TIA*, vol. 5, nº 2, pp. 257-261, 2017.
- [4] B. SEVILLA, «“REALIDAD AUMENTADA en Educación”,» *Gab. Tele-Educacion Del Vicerrectorado Serv. Tecnol. Univ. Politec. Madr.*, vol. 1, nº 1, pp. 1-39, 2017.
- [5] «“Robots humanoides ¿Cuál es su función?”,» Fundación Aquae , [En línea]. Available: <https://www.fundacionaquae.org/creando-robots-humanoides/> . [Último acceso: 2025 abril 2025].
- [6] «“About — blender.org”,» blender.org. , 01 marzo 2025. [En línea]. Available: <https://www.blender.org/about/>. [Último acceso: 21 marzo 2025].
- [7] «“Qué es Unity y para qué sirve”,» MasterD: Oposiciones, Cursos y FP., 2025. [En línea]. Available: <https://www.masterd.es/blog/que-es-unity-3d-tutorial>. [Último acceso: 22 marzo 2025].
- [8] «“Visual Studio: IDE y Editor de código para desarrolladores de software y Teams”,» Visual Studio., [En línea]. Available: <https://visualstudio.microsoft.com/es/#vs-section>. [Último acceso: 27 marzo 2025].

- [9] «“What is Arduino?”», Arduino - Home., [En línea]. Available: https://www.arduino.cc/en/Guide/Introduction/?_gl=1*jt3buo*_up*MQ..*_ga*MjAxMDY5MDk5Mi4xNzQzMTEwNzM5*_ga_NEXN8H46L5*MTc0MzExMDczOC4xLjAuMTc0MzExMDczOC4wLjAuMjA. [Último acceso: 28 marzo 2025].
- [10] «“UltiMaker Cura”», UltiMaker, [En línea]. Available: <https://ultimaker.com/es/software/ultimaker-cura/>. [Último acceso: 13 abril 2025].
- [11] «“About Us”», MIT App Inventor, [En línea]. Available: <https://appinventor.mit.edu/about-us>. [Último acceso: 13 abril 2025].
- [12] «“Learn”», Unity, [En línea]. Available: <https://unity.com/es/learn>. [Último acceso: 13 abril 2025].
- [13] «“IDE : ¿Qué es un Entorno de Desarrollo Integrado?”», Formación en ciencia de datos | DataScientest.com., [En línea]. Available: <https://datascientest.com/es/ide-que-es#:~:text=Un%20IDE%20o%20entorno%20de,las%20tareas%20>. [Último acceso: 28 marzo 2025].
- [14] «“Potencia tu Diseño con Creo 11 y la Definición Basada en Modelos (MBD)”», 9altitudes | From digitalization to digital transformation., [En línea]. Available: <https://9altitudes.es/aprende-y-conecta/articulos/que-es->. [Último acceso: 30 marzo 2025].
- [15] «“¿Qué es un motor de juegos? | Blog CC”», Universidad Europea Creative Campus, [En línea]. Available: <https://creativecampus.universidadeuropea.com/blog/motor-juegos/>. [Último acceso: 13 abril 2025].

- [16] «“Impresión 3D”,» Dassault Systèmes., [En línea]. Available: <https://www.3ds.com/es/make/guide/process/3d-printing>. [Último acceso: 31 marzo 2025].
- [17] «“¿Qué es la Manufactura Aditiva? - ▷ CIDESI”,» ▷ CIDESI - Generando valor a través del conocimiento, [En línea]. Available: <https://cidesi.com/que-es-la-manufactura-aditiva/>. [Último acceso: 14 abril 2025].
- [18] «“Guía de Uso: Filamento PLA para impresora 3D”,» Impresoras3d.com, [En línea]. Available: <https://www.impresoras3d.com/filamento-pla-consejos-caracteristicas-y-mucho-mas/>. [Último acceso: 14 abril 2025].
- [19] «“¿Qué es C# en programación y para que sirve? - Besoftware”,» Besoftware, [En línea]. Available: <https://bsw.es/que-es-c/> . [Último acceso: 14 abril 2025].
- [20] «“ESP32 Microcontrolador Wifi y Bluetooth para proyectos IOT”,» ElectroHobby, [En línea]. Available: <https://www.electrohobby.es/138-esp32?srsId=AfmBOorRdpmEVRooUvO1e8bn4H6R01nAXc-qefN-Zoedqi3T9MUf6-V3>. [Último acceso: 14 abril 2025].
- [21] V. Yadav., «“Qué es el Internet de las cosas: Definición y ejemplos de uso”,» GlobalSign by GMO , [En línea]. Available: <https://www.globalsign.com/es/blog/que-es-el-internet-de-las-cosas>. [Último acceso: 16 abril 2025].
- [22] «“Requirements — blender.org”,» blender.org., [En línea]. Available: <https://www.blender.org/download/requirements>. [Último acceso: 04 Abril 2025].
- [23] «“Inicie sus proyectos creativos y descargue el Unity Hub | Unity”,» Unity, [En línea]. Available: <https://unity.com/es/download>. [Último acceso: 05 abril 2025].

- [24] «“Requisitos del sistema de VisualÂ StudioÂ 2022”,» Microsoft Learn: Build skills that open doors in your career., [En línea]. Available: <https://learn.microsoft.com/esmx/visualstudio/releases/2022/system-requirements>. [Último acceso: 05 abril 2025].
- [25] «“Software”,» Arduino - Home., [En línea]. Available: <https://www.arduino.cc/en/software>. [Último acceso: 06 abril 2025].
- [26] «“Support Community”,» Support Community, [En línea]. Available: https://support.ultimaker.com/s/article/1667410778209?_gl=1*17lo3vj*_ga*MTExNTkxMzI5LjE3NDM5MTk0NTk.*_ga_CJM2DTBWYF*MTc0MzIxOTQ1OS4xLjAuMTc0MzIxOTQ1OS42MC. [Último acceso: 06 abril 2025].
- [27] «“System Requirements”,» MIT App Inventor, [En línea]. Available: <https://appinventor.mit.edu/explore/content/system-requirements> . [Último acceso: 06 abril 2025].
- [28] «“KP3S 3.0”,» 深圳市启庞科技有限公司, [En línea]. Available: <https://es.kingroon.com/products/3d-printer-kp3s>. [Último acceso: 06 abril 2025].
- [29] «“ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems”,» Wireless SoCs, Software, Cloud and IoT Solutions | Espressif Systems, [En línea]. Available: <https://www.espressif.com/en/products/socs/esp32>. [Último acceso: 06 abril 2025].
- [30] «“Placa de programación tipo ESP32 Bluetooth + Wi-Fi Ster”,» Electrónica Steren México, [En línea]. Available: [20] “Placa de programación tipo ESP32 Bluetooth + Wi-Fi Ster”. Electrónica Steren México. Accedido el 6 de abril de 2025. [En línea]. Disponible: <https://www.steren.com.mx/placa-de->

programacion-tipo-esp32.html?srsId=AfmBOopjWb8va3894F-n13MXsrMs2nD_WR6i. [Último acceso: 06 abril 2025].

- [31] «“PCA9685 Módulo Controlador de Servos 16 canales – Novatronic”,» Novatronic – Impulsamos tus ideas, innovamos el futuro, [En línea]. Available: <https://novatronicec.com/index.php/product/pca9685-driver-para-servos-16-c>. [Último acceso: 06 abril 2025].
- [32] «“Regulador Voltaje Salida 6v-40v 20a 300w Convertidor Buck - \$ 215”,» Tienda de electrónica - Beeggle Technology., [En línea]. Available: <https://www.beeggleshop.com/MLM-785310861-regulador-voltaje-salida-6v-40v-20a-300>. [Último acceso: 07 abril 2025].
- [33] «“Servomotor Tower Pro MG995 - SANDOROBOTICS”,» SANDOROBOTICS, [En línea]. Available: <https://sandorobotics.com.mx/producto/mg995/>. [Último acceso: 07 abril 2025].
- [34] «“Convertidor De Voltaje Dc A Dc Lm2596 Dc-dc Hw-411 Arduino - Robot Electronica”,» Robot Electronica, [En línea]. Available: <https://robot.com.ve/product/convertidor-de-voltaje-dc-a-dc-lm2596-dc-dc-hw-411-arduino/?srs>. [Último acceso: 16 abril 2025].
- [35] «“Una guía completa de la batería recargable 18650”,» CM BATTERIES, [En línea]. Available: <https://www.cmbatteries.com/es/una-guía-completa-para-la-batería-recargable-18650/>. [Último acceso: 19 abril 2025].
- [36] «“Batería recargable Li-ion 2200 mAh tipo 18650”,» Electrónica Steren México, [En línea]. Available: https://www.steren.com.mx/bateria-recargable-li-ion-2200-mah-tipo-18650.html?srsId=AfmBOop0i_XbU-syDe9mFY1PtjpGjHEw. [Último acceso: 19 abril 2025].

- [37] «“Interruptor switch balancín con piloto on/off 10 Piezas”,» Mitzu, [En línea]. Available: https://www.mitzu.com/interruptor-switch-balancin-con-piloto-on-off-10-piezas.html?srsId=AfmBOoorwmjnwrsRcxmfuZTdHgwk_v57lhTE. [Último acceso: 19 abril 2025].
- [38] «“Sensor de Distancia Ultrasónico HC-SR04”,» 330ohms, [En línea]. Available: <https://www.330ohms.com/products/sensor-ultrasonico-hc-sr04?srsId=AfmBOop3xvN9M8viBwjPX-geRCgmR7DCG78ZOmZb6qLzB7yDavPNL4Mx>. [Último acceso: 19 abril 2025].
- [39] «“Sensor de temperatura y humedad (DHT11) Steren Tienda e”,» Electrónica Steren México, [En línea]. Available: [39] “Sensor de temperatura y humedad (DHT11) Steren Tienda e”. Electrón<https://www.steren.com.mx/sensor-de-temperatura-y-humedad.html?srsId=AfmBOoqmPSZQljVeteF6f0DAnly8NwkDx12S>. [Último acceso: 19 abril 2025].
- [40] «Gumroad», [En línea]. Available: <https://hashrobotics.gumroad.com/l/hashv3humanoid>. [Último acceso: 01 Agosto 2025].
- [41] «“GrabCAD Making Additive Manufacturing at Scale Possible”,» GrabCAD Making Additive Manufacturing at Scale Possible, [En línea]. Available: <https://grabcad.com/home>. [Último acceso: 02 mayo 2025].
- [42] A. O. G. Espitia, «ARduinoRide - Prototipos Base», Uriangato, Guanajuato, 2025.
- [43] «Dispositivos compatibles con ARCore | Google for Developers», Google for Developers, [En línea]. Available:

<https://developers.google.com/ar/devices?hl=es-419>. [Último acceso: 18 mayo 2025].

Anexos

Anexo 1

Manual de armado del prototipo del robot humanoide.

Paso 1:

Para este paso se comienza configurando el ángulo de los dos servomotores en el código de Arduino, este código lo puede visualizar en la sección de Información, el ángulo al que van los servomotores es a 90° y son el número 5 y 11 del total de motores, estos motores van atornillados a la base del torso inferior.

Paso 2:

Se termina de armar el torso inferior atornillando la tapa superior a la base, para después atornillar la parte frontal y posteriormente la parte trasera del torso inferior, tenga cuidado de utilizar los tornillos adecuados y no apretar mucho, dejar de apretar cuando entre el tornillo o se sienta una mayor resistencia.

Paso 3:

Se debe de atornillar la base del torso superior encima de la tapa del torso inferior, esto se debe de realizar con cuidado y colocarlo bien ya que se colocarán los demás componentes del torso superior.

Paso 4:

Se atornillarán a la parte ya armada del robot las paredes laterales del torso superior para posteriormente colocar y atornillar los dos servomotores que conectaron los brazos del robot, el servomotor derecho va a un ángulo de 30° y el servomotor izquierdo a un ángulo de 150°.

Paso 5:

En este paso se atornillará la tapa superior a la parte ya armada del robot, para atornillar esta parte hacerlo con cuidado y en forma cruzada ir atornillando los tornillos para evitar errores de armado o fracturas de las piezas.

Paso 6:

En este paso se atornillará la tapa frontal a la parte ya armada del robot, para atornillar esta parte hacerlo con cuidado y en forma cruzada ir atornillando los tornillos para evitar errores de armado o fracturas de las piezas.

Paso 7:

En este paso se comenzará a armar la mano derecha del robot humanoide, se atornillarán las 3 piezas superiores de la mano derecha que una vez armada toda la mano se unirá al torso superior, recuerda atornillar cuidadosamente.

Paso 8:

En este paso se debe de agregar el servomotor a un ángulo de 30° grados en el componente que simula la parte del codo al hombro y atornillarlo a la parte que se armó en el paso anterior.

Paso 9:

En este paso se debe ensamblar el otro servomotor de la mano derecha a un ángulo de 30° grados, se coloca en el otro extremo de la parte que simula del codo al hombro y al servomotor se le agrega el componente que sirve de conexión del movimiento a las piezas.

Paso 10:

En este paso se arma la parte inferior de la mano las cuales son cuatro piezas, van atornilladas así que atornillar con cuidado, una vez armada la parte inferior de la mano procede a armar esa parte con la parte del paso anterior, hazlo con cuidado para no dañar las piezas.

Paso 11:

En este paso es armar la mano izquierda del robot, para esto repite los pasos del 7 al 10 siguiendo las mismas instrucciones, solo el primer servomotor debe de ir a 150° grados y el segundo servomotor a 150° grados.

Paso 12:

En este paso se ensambla la mano derecha al torso del robot, para esto el servomotor se coloca a un ángulo de 30° grados y se coloca la pieza que sirve de conector de movimiento a las piezas y posteriormente se debe de atornillar cuidadosamente.

Paso 13:

En este paso se ensambla la mano izquierda al robot, el servomotor se coloca a un ángulo de 150° y también se coloca la pieza conectora de movimiento que lo transmite a las piezas y se debe de atornillar con cuidado.

Paso 14:

Este paso se comenzará armar la pierna derecha del robot, se comenzará de la parte inferior, se unirán estas tres piezas, las cuales deben de llevar los tornillos adecuados y atornillar cuidadosamente para no dañar las piezas.

Paso 15:

En este paso se van a colocar dos servomotores en la pieza superior al pie armado en el paso anterior, el primer servomotor va ir a un ángulo de 90° y va ir atornillado al pie, el segundo servomotor va ir colocado a un ángulo de 150° grados, recuerda hacerlo con cuidado.

Paso 16:

En este paso se arman las tres piezas inferiores a la rodilla, se atornillarán entre sí para juntar las piezas debes hacerlo con cuidado, una vez armado se atornilla al segundo servomotor del paso anterior que va a un ángulo de 150° grados.

Paso 17:

En este paso se va a colocar la pieza que simula la rodilla de la pierna y en esta pieza va colocado un servomotor a un ángulo de 150° grados, posteriormente se agregó la pieza conectora al servomotor para luego atornillar a la parte ensamblada anteriormente.

Paso 18:

En este paso se atornillan dos piezas que van en la parte superior de la rodilla, recuerda estas piezas las debes de atornillar con cuidado porque podría hacer que el tornillo no apriete bien o romper alguna pieza.

Paso 19:

En este paso debes de ensamblar dos servomotores en la siguiente pieza de la pierna el servomotor inferior va a un ángulo de 30° grados y el otro a un ángulo de 90° grados, luego debes de colocar la pieza conectora a cada servomotor y atornillar la parte inferior de la pieza a toda la pierna ya armada.

Paso 20:

En este paso debes de agregar las últimas tres piezas de la pierna derecha, estas piezas son las superiores de la pierna derecha, las debes de atornillar con cuidado para evitar errores a la hora del armado y con esto terminamos de armar la pierna derecha.

Paso 21:

En este paso es armar la pierna izquierda del robot, para esto repite los pasos del 14 al 20 siguiendo las mismas instrucciones, solo los servomotores van a distintos ángulos de la parte inferior a superior, el 1 va a 90°, el 2 a 30°, el 3 a 30°, el 4 a 150° y el 5 a 90° grados.

Paso 22:

En este paso se ensambla la pierna derecha al torso del robot, para esto el servomotor se coloca a un ángulo de 90° grados y se coloca la pieza que sirve de conector de movimiento a las piezas y posteriormente se debe de atornillar cuidadosamente.

Paso 23:

En este paso se ensambla la pierna izquierda de robot, el servomotor se coloca a un ángulo de 90° y también se coloca la pieza conectora de movimiento que lo transmite a las piezas y se debe de atornillar con cuidado.

Paso 24:

En este paso se tiene que armar la cabeza, está conformada por 3 piezas, primero se ensamblan los ojos a la parte frontal de la cabeza, para luego ensamblarla la otra pieza, para así terminar de armar la cabeza del robot.

Paso 25:

En este paso se ensambla la cabeza del robot al torso superior, puede ser ensamblada con pines o también puede ser ensamblada colocando un servomotor, este debe de tener un ángulo de 90° grados, recuerda hacerlo con cuidado.

Paso 26:

En este paso se termina de armar el torso superior del robot humanoide, se atornilla la parte trasera del torso, se debe de atornillar cruzado para evitar daños a la pieza y evitar algún otro problema, con esto terminamos de armar todas las piezas del robot humanoide.

Anexo 2

Código para la colocación de los servomotores en su ángulo.

```
#include <Adafruit_PWMServoDriver.h> // Libreria PC9685

Adafruit_PWMServoDriver servo = Adafruit_PWMServoDriver(0x40);

int tiempo = 500;

void setup() {

  Serial.begin(9600);

  servo.begin();

  servo.setPWMPFreq(50);

  delay(1000);

  setServo(0,90);
```

```
setServo(1,150);  
  
setServo(2,150);  
  
setServo(3,30);  
  
setServo(4,90);  
  
setServo(5,90);  
  
setServo(6,90);  
  
setServo(7,30);  
  
setServo(8,30);  
  
setServo(9,150);  
  
setServo(10,90);  
  
setServo(11,90);  
  
setServo(12,30);  
  
setServo(13,30);  
  
setServo(14,30);  
  
setServo(15,150);  
  
setServo(16,150);  
  
setServo(17,150);  
  
setServo(18,90);  
  
delay(1000);  
  
}  
  
void loop() {
```

```
}
```

```
void setServo(uint8_t n_servo, int angulo){  
  
    int duty;  
  
    duty = map(angulo,0,180,102,512); //102 = 0° y 512 = 180° en ancho de pulso  
  
    servo.setPWM(n_servo,0,duty);  
  
}
```

Anexo 3

Imagen de la posición del ángulo de los servomotores.

Initial Servo Positions

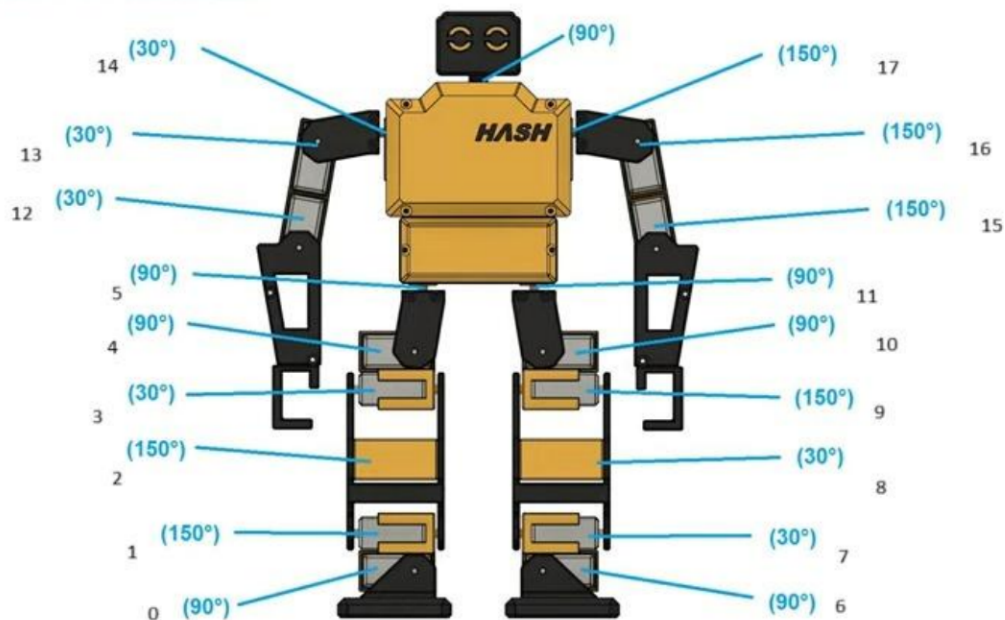


Fig. 41. Posiciones de los servomotores