ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

"DESARROLLO DE UN SIMULADOR MEDIANTE REALIDAD VIRTUAL APLICADO A VEHÍCULOS LIGEROS MANUALES PARA ENTRETENIMIENTO."

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR:

MICHAEL STEVEN MORALES ROSARIO

Riobamba – Ecuador

2024

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

"DESARROLLO DE UN SIMULADOR MEDIANTE REALIDAD VIRTUAL APLICADO A VEHICULOS LIGEROS MANUALES PARA ENTRETENIMIENTO."

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR: MICHAEL STEVEN MORALES ROSARIO

DIRECTOR: ING. RAMIRO FERNANDO ISA JARA. PhD

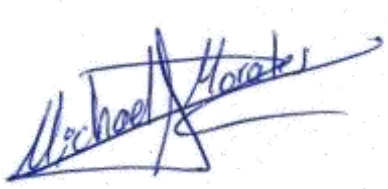
Riobamba – Ecuador

2024

Yo, Michael Steven Morales Rosario, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 03 de abril de 2024



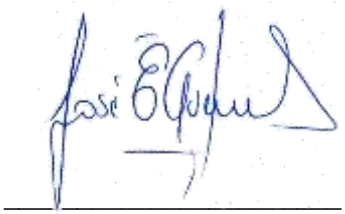
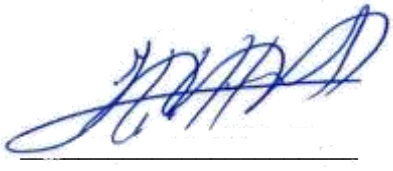
Michael Steven Morales Rosario
2100463310

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, **DESARROLLO DE UN SIMULADOR MEDIANTE REALIDAD VIRTUAL APLICADO A VEHICULOS LIGEROS MANUALES PARA ENTRETENIMIENTO**, realizado por el señor: Michael Steven Morales Rosario, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	**FECHA**
Ing. José Enrique Guerra Salazar **PRESIDENTE DEL TRIBUNAL**		2024-04-03
Ing. Ramiro Fernando Isa Jara. PhD **DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2024-04-03
Ing. Jorge Luis Hernández Ambato. PhD **ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2024-04-03

DEDICATORIA

Este trabajo lo dedico principalmente a mi madre, quien con su paciencia, dedicación, compromiso y amor ha sido pilar fundamental para poder culminar esta meta, agradezco inmensamente todo su apoyo, cada una de sus palabras y oraciones que me han brindado la fortaleza para el día de hoy ser ingeniero.

A mi padre, quien, con su apoyo y sus consejos, me alentó a forjarme metas y a ser una mejor persona cada día.

A mis hermanos Andrés y Heidy, por estar siempre al pendiente, creer en mí, ser mi soporte, brindarme su ayuda cuando lo he necesitado, y por alentarme para lograr este sueño que hoy es una realidad.

Michael

AGRADECIMIENTO

Agradezco primeramente a Dios, por la vida, la salud y la bendición de contar con una familia que me apoya.

A Luisa, por ser un gran apoyo incondicional, por su compañía en los buenos y malos momentos del caminar diario, por sus palabras de ánimos que me dieron fuerzas para no desistir.

Doy gracias a la Escuela Superior Politécnica de Chimborazo por abrirme sus puertas, a los maestros que he tenido el honor de conocer y supieron guiarme y formarme como profesional, especialmente a mi tutor del trabajo de titulación, Ing. Ramiro Fernando Isa Jara, a quien considero y estimo por su valiosa guía y asesoramiento durante la realización de este trabajo.

Michael

ÍNDICE DE CONTENIDO

CAPÍTULO III

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE ANEXOS

RESUMEN

En la actualidad, la realidad virtual tiene diversas aplicaciones innovadoras. En el campo de la ingeniería con el uso de sistemas embebidos ha dado lugar a avances significativos en la simulación. El presente trabajo se enfoca en el desarrollo de un sistema integrado de realidad virtual aplicado al entretenimiento mediante el uso de vehículos ligeros para ciclismo indoor. La metodología se basa en cinco etapas: 1) la etapa de operación del usuario que permite interactuar de forma directa con el sistema; 2) la etapa de adquisición de datos obtenidos mediante sensores para procesarlos mediante la tarjeta de desarrollo; 3) la etapa de simulación y comunicación para la comunicación entre software y Hardware, generando variaciones en el entorno físico y virtual; 4) la etapa de control del motor para generar la resistencia al movimiento de acuerdo a los cambios en el terreno en el entorno virtual. Finalmente, la etapa de registro de variables que consiste en la visualización de las variables velocidad, pendiente y tiempo recopiladas para del análisis del rendimiento. La prueba en los tiempos de comunicación bróker MQTT determinó el servidor más estable, sin perturbación en su desviación estándar. De acuerdo con los resultados, se ha verificado que no existe una latencia considerable en los tiempos de conexión entre ESP32 y Unity. Por medio de la prueba de correlación de Pearson con 0.999988 se otorgó el respaldo al sensado de velocidad. En base a las valoraciones, se ha establecido que el sistema tanto en su software, hardware y calidad de entretenimiento tiene una puntuación positiva en la escala de Likert. De lo analizado se concluye que el simulador de realidad virtual brindó una experiencia inmersiva a las personas, cumpliendo con los objetivos planteados. Esto abre nuevas posibilidades para abordar los desafíos de las tecnologías VR.

Palabras clave: <SIMULADOR DE REALIDAD VIRTUAL>, <COMUNICACIÓN>, <VELOCIDAD>, <FRICCIÓN>, <BASE DE DATOS>.

0357-DBRA-UPT-2024

SUMMARY

Currently, virtual reality has various innovative applications. In the field of engineering, the use of embedded systems has led to significant advances in simulation. The present work focuses on developing an integrated virtual reality system applied to entertainment through lightweight vehicles for indoor cycling. The methodology is based on five stages: 1) the operating stage allows direct interaction with the system; 2) The data acquisition stage involves gathering data obtained through sensors to process them using the development board; 3) The simulation and communication stage facilitates communication between software and hardware, generating variations in the physical and virtual environment; 4) The motor control stage is responsible for generating resistance to movement according to changes in the terrain in the virtual environment. ¿Finally, the variable logging stage involves visualizing the collected variables such as speed, incline, and time for performance analysis. The test on MQTT broker communication times determined the most stable server, with no disturbance in its standard deviation. According to the results, it has been verified that there is no considerable latency in the connection times between ESP32 and Unity. Through the Pearson correlation test with a value of 0.999988, support was given to the speed sensing. Based on the evaluations, it has been established that the system has received a positive score on the Likert scale, both in its software, hardware, and entertainment quality. The analysis concluded that the virtual reality simulator provided an immersive experience to individuals, meeting the set objectives and opening up new possibilities for addressing the challenges of VR technologies.

Keywords: <VIRTUAL REALITY SIMULATOR>, <COMMUNICATION>, <SPEED>, <FRICTION>, <DATABASE>.

0357-DBRA-UPT-2024

Lenin Iván Lara Olivo

0602546103

INTRODUCCIÓN

La realidad virtual permite crear experiencias inmersivas, lo cual, ofrece a las personas la posibilidad de experimentar sensaciones cercanas a las reales dentro de entornos virtuales cuidadosamente diseñados. Estos entornos se construyen seleccionando elementos específicos que aseguran una experiencia atractiva y divertida para el usuario. Sin embargo, en la actualidad, el uso de la realidad virtual se encuentra notablemente subutilizado, limitándose principalmente a la esfera del entretenimiento a través de videojuegos. No obstante, una búsqueda más detallada de las capacidades de los dispositivos de realidad virtual revela un panorama mucho más amplio de aplicaciones dentro de la industria, la medicina, la formación profesional en distintas áreas, entre otras.

La incorporación de dos elementos clave como son la integración del entorno real y el entorno virtual durante las simulaciones, ha permitido generar respuestas significativamente favorables por parte de los usuarios. Esta efectividad se hace evidente en el desarrollo de simuladores de realidad virtual para vehículos ligeros manuales, donde se facilita un intercambio dinámico de información entre la realidad virtual y la física. En particular, un simulador implementado mediante un sistema de fricción mecánica permite a los usuarios experimentar la sensación de un paseo en bicicleta. Mediante el uso de la realidad virtual se puede recrear la experiencia de ciclismo al aire libre bajo una variedad de condiciones ambientales y tipos de terreno, todo ello sin salir de un espacio interior.

Este trabajo de integración curricular se enfoca en un análisis exhaustivo de los materiales y componentes necesarios para la construcción de dicho simulador. Para ello se han realizado múltiples pruebas con la finalidad de validar la funcionalidad del hardware y el software, así como la calidad de la experiencia del usuario, asegurando que todos los aspectos cumplan con los estándares y requisitos previamente establecidos. A través de este estudio, se busca no solo demostrar la viabilidad técnica del proyecto, sino también destacar el potencial de la realidad virtual como herramienta para expandir las fronteras de nuestra interacción con el mundo digital, abriendo nuevas vías para la educación, el entretenimiento y la simulación profesional.

CAPÍTULO I

1. PLANTEAMIENTO DEL PROBLEMA

En el presente capítulo se aborda los antecedentes, el planteamiento del problema, la justificación teórica y aplicativa, los objetivos, tanto general como específicos.

1.1. Antecedentes

(Ijsselsteijn et al. 2004), en su estudio revelan que proporcionar un entorno en el que los usuarios se sientan presentes tiene un impacto positivo en su diversión y en consecuencia en su motivación. Específicamente, en un entorno donde la sensación de presencia era más intensa, los participantes reportaron mayor interés, disfrute, percepción de competencia y control, mientras que (Mestre, Dagonneau y Mercier 2011), indica que los resultados de su estudio evidenciaron un efecto significativo de la intensidad del ejercicio en el rendimiento, medido a través de diversas variables, siendo estas medidas sensibles al perfil de pendiente del terreno simulado, utilizando una fuerza de resistencia variable aplicada a la rueda trasera de la bicicleta para simular la pendiente de la carretera en el entorno virtual.

Para (Ortega, Pedrosa de Lima y Ortega 2017), la unión entre la realidad virtual y los sensores permite una fusión cautivadora entre el mundo real y el mundo virtual. Esta fusión nos brinda la oportunidad de experimentar lo real mientras nos sumergimos en la concreción de la imaginación humana, sin perder la conexión con la realidad.

Según (Velasteguí López 2019), indica que la realidad virtual data desde la década de los ochenta, la cual se encuentra en diversos procesos de actualización enfocados ampliamente al mundo del entretenimiento. (Chessa et al. 2019), manifiesta que, a partir de su establecimiento en 2010, Oculus VR se ha destacado como una empresa de vanguardia en la creación de tecnología de realidad virtual. Específicamente, su revolucionario lanzamiento del Oculus Rift en 2012 ha dejado una huella significativa en la historia de la realidad virtual contemporánea, impulsando el progreso y la creciente aceptación de esta tecnología en los últimos años.

Salas (Salas Noain y Delgado del Castillo 2023), describe que la falta de motivación y la ausencia de resultados a corto plazo son dos de las principales barreras que enfrentan las personas para mantener una rutina de ejercicios regular. Sin embargo, los exergames (videojuegos que combinan el entretenimiento digital con la actividad física) de realidad virtual han demostrado tener un impacto positivo en la motivación de las personas, al permitirles ejercitarse de manera entretenida y sostenida a lo largo del tiempo. La realidad virtual ha demostrado su eficacia de manera notable en el desarrollo de simuladores utilizados para el entrenamiento en diversas áreas como la medicina, rehabilitación, biotecnología e incluso la industria militar, entre otros campos,

reflejan la utilidad y el impacto de esta tecnología en una amplia gama de aplicaciones especializadas (Véliz Vega, Correa Madrigal y Kugurakova 2021).

1.2. Formulación del problema

¿Se puede desarrollar un simulador mediante realidad virtual aplicado a vehículos ligeros manuales para entretenimiento?

1.3. Sistematización del problema

- ¿Cómo se puede aplicar el uso de tecnologías envolventes para la comunicación entre el operador y el entorno virtual 3D?
- ¿Se puede integrar ambientes virtuales hacia el simulador mediante lenguajes de programación?
- ¿Cómo diseñar el sistema electrónico para la interacción del usuario y el entorno virtual generado por el simulador?
- ¿Por qué construir un sistema mecánico para la regulación de velocidad mediante un microcontrolador para generar una experiencia segura?
- ¿Cómo implementar un sistema para registro de información sobre el rendimiento basado en tiempo y distancia para un histórico de entrenamiento?
- ¿Se puede realizar pruebas de validación sobre la funcionalidad y el grado de inmersión del simulador mediante información de los usuarios?

1.4. Justificación

1.4.1. Justificación teórica

(Herrera 2004), argumenta que los exergames, al involucrar movimiento físico, se han convertido en una herramienta efectiva para fomentar un estilo de vida activo, aumentar los niveles de actividad física y, en consecuencia, tener un impacto positivo en la salud y condición física del usuario. Además, se destaca que los exergames implican actividades cognitivamente desafiantes, lo que los convierte en posibles instrumentos para mejorar el rendimiento perceptual.
(Lanningham-Foster et al. 2006), en un estudio llevado a cabo en niños de entre 8 y 12 años, demostraron que el grupo que jugó al videojuego “Dance Dance Revolution” consumió el doble de energía en comparación con el grupo que jugó un videojuego sedentario.

(Staiano y Calvert 2011), evidenciaron que los adolescentes que jugaron la modalidad de tenis en la videoconsola Nintendo Wii gastaron más energía que aquellos que jugaron un juego sedentario. Además, estos mismos jóvenes disfrutaron mucho más de la experiencia y jugaron durante un tiempo prolongado, lo que resultó en un mayor gasto calórico en comparación con los que jugaron de manera individual.

Es importante recalcar la necesidad de mejorar los métodos de entrenamiento físico utilizando tecnologías innovadoras que permitan simular situaciones y escenarios específicos del mundo real en un entorno controlado. En este sentido, el uso de tecnologías de realidad virtual en la práctica deportiva con vehículos ligeros manuales podría proporcionar una mayor experiencia para el usuario, permitiendo mejorar su eficacia y reducir los riesgos de lesiones.

La realidad virtual es una herramienta beneficiosa que nos permite mejorar no solo nuestra percepción cognitiva visual, sino también la capacidad de crear y recrear planos de forma visual, lo que facilita la enseñanza, el recuerdo y el intercambio de experiencias relacionadas con nuestros objetivos específicos (Ortega, Pedrosa de Lima y Ortega 2017).

(Velasteguí López 2019), señala que la creación de entornos virtuales permite trasladar a las personas a un ambiente virtual que simula condiciones reales que enfrentarán en sus aplicaciones diarias. Esto simplifica el proceso de aprendizaje, mejora las habilidades de la persona y evita posibles daños en equipos e instalaciones.

En las últimas décadas, se ha observado una evolución hacia un futuro más sostenible en las ciudades, en la cual se ha prestado una mayor atención al uso de la bicicleta como medio de transporte urbano (Pérez Pujol 2022).

La mejora de la experiencia de ejercicio puede lograrse mediante la incorporación de un videojuego que simule un recorrido en un entorno ficticio o real mientras el usuario genera movimientos para posteriormente ser captados, con el objetivo de ofrecer una experiencia virtual lo más realista posible del ciclismo convencional. En este sentido, se proporciona comparaciones entre diferentes dispositivos para el ensamblaje del hardware, así como comparaciones de los softwares de motores de juegos y modelado gráfico más utilizados en el mercado, lo cual puede ser de gran utilidad para tomar decisiones basadas en las características, ventajas y desventajas de cada uno (Pérez Pujol 2022).

1.4.2. Justificación aplicativa

El proyecto estará enfocado en ofrecer al usuario una forma de entretenimiento al momento de interactuar con un entorno virtual seguro. El sistema contará con un motor gráfico para poder procesar el entorno, al momento que el usuario comience a ciclar se generará movimiento en el mundo virtual el cual tendrá zonas específicas programadas para poder generar una variación en la velocidad de manera controlada para evitar excesos y posibles peligros.

Se podrá adaptar cualquier bicicleta al dispositivo de regulación, sin necesidad de realizar modificaciones físicas en la misma, por lo cual, se estima un valor inferior al mercado al no estar ligados a usar nuevos modelos de bicicletas lanzados al mercado los cuales cuentan con un valor mucho más elevado.

Se tendrá presente un sensor para la adquisición y envío de datos a la tarjeta controladora, misma que se encargará de la conversión de esta información para obtener las diversas variables generadas por el dispositivo y el usuario.

Toda esta información se podrá adquirir a través de una plataforma virtual, para la lectura de las variables, de esta forma el usuario logrará llevar un seguimiento de su progreso en todo instante.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un simulador mediante realidad virtual aplicado a vehículos ligeros manuales para el entretenimiento.

1.5.2. Objetivos Específicos

- Aplicar el uso de tecnologías envolventes para la comunicación entre el operador y el entorno virtual 3D.
- Integrar ambientes virtuales hacia el simulador usando lenguajes de programación.
- Diseñar el sistema electrónico para la interacción del usuario y el entorno virtual generado por el simulador.
- Construir un sistema mecánico para la regulación de velocidad mediante un microcontrolador para generar una experiencia segura.
- Implementar un sistema para registro de información sobre el rendimiento del usuario para un histórico de entrenamiento.
- Realizar pruebas de validación sobre la funcionalidad y el grado de inmersión del simulador mediante información de los usuarios.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se analiza la realidad virtual como una plataforma inmersiva que estimula los sentidos humanos. También se exploran los componentes tecnológicos, como los motores gráficos y las bases de datos, además del uso de IDE de programación. Este enfoque ofrece una base para comprender las tecnologías que respaldan la realidad virtual y su aplicación en diversas áreas.

2.1. Realidad virtual

Se define como una plataforma interactiva que utiliza una base de datos para crear una simulación que involucre uno o varios de los sentidos humanos. Esta simulación es generada por un ordenador y puede ser explorada, visualizada y manipulada en tiempo real mediante la representación de imágenes y sonidos digitales, lo que proporciona una sensación de presencia en el entorno virtual (Gutiérrez 2020).

La finalidad principal de una interfaz de realidad virtual consiste en lograr una inmersión total de los canales sensomotores del ser humano en una vivencia creada a través de la informática. Para ser considerado como realidad virtual, un sistema debe contar con la capacidad de generar de forma digital un entorno tridimensional que permita al usuario interactuar de manera intuitiva y en tiempo real con los objetos que se encuentran dentro de él (García 2000).

Mediante un entorno de realidad virtual el usuario deja de ver el lugar donde se encuentra para "teletransportarse" a otra realidad y poder interactuar con los objetos virtuales que la componen, logrando una experiencia sensorial completa. Es fundamental que haya una interacción entre el usuario y el mundo virtual en el que se está desenvolviendo para que se considere una experiencia de realidad virtual (Valarezo-Guzmán et al. 2023).

La relevancia del diseño de la interfaz se debe a que constituye el esquema principal de navegación dentro del entorno virtual, determinando la forma en que se presenta y se interactúa con él (Herrera del Gener 2022).

En la **Ilustración 2-1** se observa un entorno precargado usando el dispositivo de realidad virtual Oculus Quest 2.

Ilustración 2-1: Entorno precargado de realidad virtual.

Realizado por: Morales, 2023.

2.2. Dispositivos para la realidad virtual

Son elementos tecnológicos diseñados para sumergir al usuario en experiencias inmersivas. A lo largo de los años, diversas empresas han avanzado en sus tecnologías, ofreciendo herramientas como mandos hápticos para involucrar al usuario en entornos virtuales tridimensionales. Estos dispositivos incluyen gafas o cascos de realidad virtual, controladores de movimiento y sensores, creando una experiencia sensorial que simula la realidad y permite una interacción activa con el entorno virtual (Gutiérrez 2020).

2.2.1. Meta - Oculus

Es una empresa líder en tecnología de realidad virtual, reconocida por sus dispositivos como el Oculus Rift, Oculus Quest y Oculus Go. Estos dispositivos ofrecen experiencias inmersivas de alta calidad para juegos, entretenimiento y aplicaciones educativas, transformando la forma en que se interactúa con la tecnología y el mundo digital. Meta también ha desarrollado herramientas para crear contenido en realidad virtual, fomentando el crecimiento y la innovación en este campo tecnológico.(Salas Noain y Delgado del Castillo 2023).

2.2.2. HTC Vive

Es una marca de realidad virtual desarrollada por HTC Corporation en colaboración con la empresa de videojuegos Valve. Los dispositivos VIVE, tales como VIVE Pro y VIVE Cosmos, ofrecen experiencias de realidad virtual de alta gama, con seguimiento preciso de los movimientos

y controladores ergonómicos. Con un enfoque en la calidad visual y la inmersión, HTC VIVE es popular entre los entusiastas de la realidad virtual y los profesionales en aplicaciones empresariales y de diseño (Yaguana 2021).

2.2.3. Sony - PlayStation VR

Los visores de realidad virtual desarrollados por Sony están diseñados específicamente para su uso con las consolas PlayStation 4 y PlayStation 5. Estos visores brindan una experiencia inmersiva que transporta a los usuarios a mundos virtuales, permitiéndoles disfrutar de juegos, videos y experiencias interactivas en 360 grados. Se destacan por su facilidad de uso al integrarse directamente con las consolas PlayStation, lo que facilita su instalación y ofrece una experiencia de usuario fluida (PlayStation 2020).

2.3. Comparación entre dispositivos de realidad virtual

En la **Tabla 2-1** se puede apreciar las características de los dispositivos de realidad virtual, las cuales son más favorables para los dispositivos de Meta, esto debido a que no es dependiente de una plataforma externa, su alta cantidad de librerías hacen que el desarrollo de aplicaciones sea mucho más factible que en los otros dispositivos.

Tabla 2-1: Características de los dispositivos más comunes de realidad virtual.

Característica/ Dispositivo	Meta - Oculus	HTC Vive	Sony – PlayStation Vr
Plataforma	Autónomo, no requiere PC ni consola	Requiere conexión a Pc	Requiere conexión a PlayStation 4 o PlayStation 5
Resolución de pantalla	1440 x 1600 - 1832 x 1920 por ojo	1080 x 1200 - 1832 x 1920 por ojo	960 x 1080 por ojo
Controladores	Oculus Touch	HTC Vive Controllers	PlayStation Move Controllers
Conexión	Inalámbrica o por cable (link a PC)	Por cable (HDMI y USB)	Por cable (conexión a PS4/PS5)
Cantidad de librerías	Alta	Media	Media/Baja
Compatibilidad	Windows, Android	Windows, macOS	PlayStation 4, PlayStation 5
Sensores	Acelerómetro, giroscopio, magnetómetro	Acelerómetro, giroscopio, lighthouse	Acelerómetro, giroscopio
Precio	$299 - $1299	$399 - $1399	$199 - $399

Fuente: (Oculus 2020; PlayStation 2020; Yaguana 2021)

Realizado por: Morales, 2023.

2.4. Motores gráficos

Las funcionalidades de este tipo de motores abarcan la capacidad de renderizar gráficos en 2D y 3D, así como funcionar como motores de sonido y motores de animación. Estos motores permiten infundir vida a su contenido, posibilitando la creación de animaciones tan impresionantes que, como se ha mencionado previamente, se emplean en la industria cinematográfica, llegando en algunos casos a confundir la distinción entre lo real y lo generado digitalmente (Martínez 2021).

2.4.1. Motores gráficos más comunes

2.4.1.1. Unity

Es un motor de desarrollo de videojuegos ampliamente popular y versátil, esencial en la creación de experiencias interactivas en entornos 2D y 3D. Destaca por su accesibilidad y facilidad de uso, siendo preferido tanto por principiantes como por desarrolladores experimentados. Su capacidad multiplataforma simplifica el proceso de desarrollo y distribución, lo que resulta en un ahorro significativo de tiempo y recursos. Además, ofrece una comunidad activa de desarrolladores que comparten conocimientos y recursos, promoviendo la colaboración y facilitando la resolución de problemas (Lidon 2019).

2.4.1.2. GODOT

Es un motor de videojuegos de código abierto y gratuito, adecuado tanto para juegos en 2D como en 3D. Es compatible con sistemas operativos como Windows, Linux, macOS, BSD y Android. Además, permite exportar proyectos a estos sistemas y a diversos dispositivos, incluyendo teléfonos móviles y el formato web HTML5. Entre sus características notables, se destacan su motor gráfico basado en OpenGL ES 2.0 y 3.0 (Galarza 2023), para crear objetos de colisión tanto en 2D como en 3D. Godot también cuenta con su propio motor de física y objetos relacionados, lo que lo hace una elección sólida. Uno de sus puntos fuertes es su motor 2D, que no requiere simulación en un espacio 3D, lo que lo convierte en una opción preferida para desarrollar juegos en 2D (Culiáñez 2023).

2.4.1.3. GameMaker

Es una opción de desarrollo gráfico que puede ser más desafiante, ya que requiere el uso de su propio lenguaje de programación llamado GML (Game Maker Language). Sin embargo, GML

guarda similitudes con lenguajes de programación comunes en el desarrollo de videojuegos, como C++ o C#. Aunque este motor es ampliamente utilizado por desarrolladores que se enfocan en juegos 2D, en realidad no está limitado, ya que su editor de activos y mapas permite la creación de juegos en 3D. Algunos juegos conocidos creados con este motor incluyen Undertale, Spelunky y Hotline Miami, entre otros (Morales 2023).

2.4.1.4. Unreal Engine

Es uno de los motores gráficos más antiguos, habiendo lanzado su primera versión en 1998, y sigue siendo uno de los más populares en la industria. Ha sido utilizado para crear juegos importantes como Gears of War y Mass Effect. En 2021, se lanzó su última versión, Unreal Engine 5, que causó un gran revuelo debido a sus mejoras en gráficos e iluminación, incluyendo el uso de RayTracing. Esta última versión continúa siendo gratuita, pero es importante tener en cuenta los términos y condiciones establecidos por Epic Games para el desarrollo en este motor gráfico (Morales 2023).

2.5. Comparación de distintos motores gráficos

Con el objetivo de obtener una visión más detallada de cada uno de los motores gráficos previamente mencionados, es importante conocer sus características principales, las cuales se muestran en la **Tabla 2-2**. A través de estas características se puede concluir como Unity es el motor con mayores ventajas, esto debido a su lenguaje de programación de baja complejidad, pero muy versátil, su amplia comunidad activa ayuda a los desarrolladores a resolver problemas.

Tabla 2-2: Características de los diferentes motores gráficos más comunes.

Característica/Motor gráfico	Unity	GODOT	GameMaker	Unreal Engine
Lenguaje de Programación	C#	GDScript	GML (GameMaker Language)	C++ y Bluesprints
Comunidad y Soporte	Amplia comunidad y soporte activo	Comunidad creciente y soporte	Comunidad activa y soporte	Amplia comunidad y soporte activo
Plataformas Compatibles	Múltiples, incluyen Windows, Mac, Linux, iOS, Android, consolas, web, AR/VR	Múltiples, incluyen Windows, Mac, Linux, iOS, Android, web, consolas	Principalmente Windows, pero con exportación a otras plataformas	Múltiples, incluyendo Windows, Mac, Linux, iOS, Android, consolas, AR/VR
2D y 3D	Soporta tanto 2D como 3D	Soporta tanto 2D como 3D	Principalmente 2D, pero admite 3D	Soporta tanto 2D como 3D

Facilidad de Uso	Ampliamente considerado fácil de aprender	Fácil de aprender y usar	Relativamente fácil para 2D	Más curva de aprendizaje, pero poderoso
Precio	Gratuito para proyectos pequeños, tarifas variables para ingresos más altos	Código fuente abierto, gratuito	Versión gratuita y licencias pagas para exportación	Gratuito para proyectos pequeños, tarifas variables para ingresos más altos

Fuente: (Lidon 2019; Culiáñez 2023; Morales 2023)

Realizado por: Morales, 2023.

2.6. Tarjetas de desarrollo

Se trata de un chip integrado que puede programarse y está conformado principalmente por tres componentes esenciales: una unidad central de procesamiento (CPU), memorias de solo lectura (ROM) y memorias de acceso aleatorio (RAM), además de dispositivos periféricos de entrada y salida. Estos dispositivos tienen la capacidad de ejecutar las funciones almacenadas en su memoria. Los microcontroladores son componentes electrónicos integrados que combinan en un solo chip las funcionalidades necesarias para llevar a cabo tareas específicas de control y operación. A diferencia de sistemas más complejos, están diseñados para ejecutar un conjunto limitado de instrucciones de manera eficiente, lo que los hace ideales para aplicaciones que requieren automatización o control directo sobre hardware, como en sistemas embebidos o dispositivos electrónicos dedicados (Martinez 2021).

2.6.1. Espressif (ESP)

Son dispositivos que integran en un solo circuito integrado múltiples funciones, lo que reduce tanto el tamaño como el costo total del sistema. Por lo general, incluyen un microcontrolador, memoria flash, módulos de entrada/salida (GPIO), convertidores analógicos a digital (A/D) y circuitos específicos para conectividad WiFi. Además, vienen con el software o firmware necesario para gestionar la mayoría de las comunicaciones por Internet (Espressif Systems 2020).

2.6.2. Arduino

Se trata de una plataforma de prototipado electrónico basada en hardware y software de código abierto, conocida por su flexibilidad y facilidad de uso. Su función principal es interactuar con el entorno mediante la recepción de entradas de una variedad de sensores y la capacidad de controlar motores, luces y otros dispositivos de salida (Arduino 2023).

2.6.3. Raspberry PI

Es una línea de minicomputadoras ampliamente conocida y popular debido a su versatilidad en diversas aplicaciones. Se destacan por su capacidad de conexión a Internet y por su uso en la conexión y supervisión de dispositivos y sistemas a través de sensores. Una característica distintiva es su programación interna basada en software libre, lo que brinda una mayor flexibilidad y posibilidad de personalización (Raspberry pi Foundation 2018).

2.7. Comparación entre las distintas tarjetas de desarrollo

En la **Tabla 2-3** se presentan las características de las distintas tarjetas de desarrollo, en donde se puede destacar a la tarjeta de desarrollo perteneciente a la familia Espressif (ESP), debido a su comunicación inalámbrica tanto de WiFi como de Bluetooth sin necesidad de incorporar módulos externos para realizar esta conexión, su alta compatibilidad con softwares de realidad virtual y su rápida frecuencia de operación lo hacen capaz de ejecutar instrucciones y procesar datos de manera eficiente y rápida.

Tabla 2-3: Características de distintas tarjetas de desarrollo.

Característica /Tarjeta	Espressif (ESP)	Arduino	Raspberry Pi
Microcontrolador	Tensilica	AVR ATmega	ARM Cortex
Pines digitales	17-36	14-54	40
Pines analógicos	1-18	6-12	No posee
Voltaje de alimentación	3.3 V – 5 V	5 V – 12V	5 V
WiFi	Si	Incluyendo un módulo externo	Si
Bluetooth	Si	Incluyendo un módulo externo	Si
Frecuencia de operación	80 MHz – 160 MHz	16 MHz – 84 MHz	700 MHz – 1.5 GHz
Consumo de energía	Variable (bajo)	Bajo	Mayor
Compatibilidad con softwares VR	Alta	Media alta	Media/ Mayor complejidad
RAM	120 KB – 520 KB	2 KB – 96 KB	512 MB – 4GB
Precio	$10 - $70	$9 - $60	$25 - $300

Fuente: (Espressif Systems 2020; Arduino 2023; Raspberry pi Foundation 2018)

Realizado por: Morales, 2023.

2.8. Base de datos

Son sistemas de almacenamiento y gestión de información que permiten organizar datos de manera estructurada para su posterior uso. Es posible almacenar y acceder a información relevante, configuraciones de usuario, estadísticas de rendimiento, registros de actividad y más. Las bases de datos se pueden clasificar en dos categorías principales según la estructura y el modo de almacenamiento de los datos: bases de datos relacionales, también conocidas como SQL (Structured Query Language o Lenguaje de Consulta Estructurada), y bases de datos no relacionales o NoSQL (Not only SQL) (Rangel 2022).

En la **tabla 2-4** se presenta una comparativa que destaca las diferencias clave entre las bases de datos relacionales y no relacionales, donde la base de datos relacional (SQL) destaca debido a su gestión de esquemas, normalización y eficiencia en transacciones. Aunque la base de datos NoSQL ofrece flexibilidad en el esquema y crecimiento de datos, la relacional sigue siendo preferible por su integridad y eficiencia.

Tabla 2-4: Características de las bases de datos relacionales y no relacionales.

Característica/Base de datos	Relacional (SQL)	No relacional (NoSQL)
Gestión de Esquemas	Implica la necesidad de definir y crear esquemas antes de almacenar datos.	No es requerido establecer un esquema previo al almacenamiento de datos.
Normalización	En diversos casos, se encuentra la necesidad de normalizar la base de datos para prevenir redundancias y mantener la integridad de los datos.	La normalización no es obligatoria y, de hecho, puede resultar impráctica para ciertos tipos de datos no estructurados.
Estructura de Datos	Hace uso de tablas para almacenar datos, en las cuales cada fila representa un registro y cada columna, un atributo.	Utiliza colecciones para almacenar documentos que pueden contener objetos y arreglos.
Clave Primaria	Siempre se requiere definir una clave primaria para identificar de manera única cada registro en una tabla.	No es esencial definir explícitamente una clave primaria, ya que el sistema puede generar identificadores automáticamente.
Transacciones	Es muy eficiente en el manejo de transacciones, asegurando la consistencia en la base de datos.	No gestiona transacciones de manera tan rigurosa, dando prioridad a la escalabilidad y disponibilidad.
Crecimiento de Datos	El crecimiento se produce de manera vertical, agregando más columnas a las tablas.	El crecimiento se da de forma horizontal, añadiendo más documentos o nodos.
Espacio en Memoria	Existe la posibilidad de desperdicio de espacio en memoria al tener campos vacíos o nulos.	Hay una situación de desperdicio de espacio en memoria al repetir información en documentos.

Fuente: (Valverde, Portalanza y Mora 2019)

Realizado por: Morales, 2023.

2.9. Rodillo ciclosimulador

Son dispositivos diseñados para proporcionar una experiencia de ciclismo en interiores, permitiendo a los usuarios utilizar sus propias bicicletas en un entorno controlado. La característica distintiva de los rodillos ciclosimuladores es su capacidad para ajustar la resistencia, ofreciendo a los ciclistas la posibilidad de simular diferentes condiciones de terreno y niveles de dificultad. Este ajuste de resistencia contribuye a una experiencia de entrenamiento más desafiante y realista. Estos dispositivos son apreciados por ciclistas tanto aficionados como profesionales, ya que brindan la flexibilidad de entrenar en casa, independientemente de las condiciones climáticas o la disponibilidad de terreno exterior (Briseño 2023).

2.9.1. Tipos de rodillos ciclosimuladores

A continuación, se presentan los tipos de rodillos ciclo simuladores más comunes, junto con una tabla comparativa de sus características.

2.9.1.1. Rodillos de equilibrio

Cuentan con tres cilindros alineados y montados sobre una estructura resistente tal como se muestra en la **Ilustración 2-2**. Al colocar la rueda trasera de la bicicleta sobre estos cilindros y comenzar a pedalear, el ciclista debe mantener el equilibrio para evitar caer. Este tipo de entrenamiento contribuye significativamente a perfeccionar la técnica de pedaleo, la estabilidad y la habilidad para mantener el control sobre la bicicleta, aspectos esenciales para ciclistas que buscan mejorar su destreza en situaciones desafiantes, como giros cerrados o terrenos irregulares (Sánchez 2020).

Ilustración 2-2: Ejemplo del rodillo de equilibrio.

Fuente: (Sánchez 2020)

2.9.1.2. Rodillos magnéticos

Han ganado popularidad entre los aficionados al ciclismo por su accesibilidad y versatilidad. Su estructura se presenta en la **Ilustración 2-3**, la cual funciona mediante imanes ajustables desde el manillar, permitiendo cambiar sus niveles de resistencia. Aunque suelen ser más asequibles, proporcionan una sensación de pedaleo menos realista, lo que podría limitar su uso en entrenamientos más intensos. Son ideales para sesiones ocasionales en casa o como calentamiento, pero para un entrenamiento más exigente, es recomendable explorar otras opciones, ya que pueden desgastarse con facilidad y generar algo de ruido durante su uso (Rutasenbici 2023).

Ilustración 2-3: Ejemplo del rodillo magnético.

Fuente: (Rutasenbici 2023)

2.9.1.3. Rodillos de fluidos

Son elementos de entrenamiento que utilizan fluidos, como el aceite, para simular la resistencia al pedalear. Se componen de un tambor de acero conectado al eje trasero de la bicicleta sumergido en un fluido viscoso tal como se muestra en la **Ilustración 2-4**.

Al pedalear, este fluido crea una resistencia progresiva y natural, aumentando la intensidad conforme se incrementa la velocidad. Esta estructura proporciona una sensación de pedaleo más suave y realista en comparación con otros rodillos. Además ofrece un funcionamiento más silencioso (Sánchez 2020).

Ilustración 2-4: Ejemplo del rodillo de fluidos.

Fuente: (Briseño 2023)

2.9.1.4. Rodillos de transmisión directa

Estos dispositivos se conectan directamente al cuadro trasero de la bicicleta. Estos rodillos sustituyen la rueda trasera por un sistema de transmisión que se acopla directamente al cuadro de la bicicleta como se presenta en la **Ilustración 2-5**, ofreciendo una experiencia de pedaleo más realista al simular una conexión directa con la cadena y los engranajes. Al no depender de la rueda trasera, evitan el desgaste de los neumáticos y proporcionan una sensación de pedaleo precisa y consistente. Suelen ser más estables y ofrecen resistencia ajustable para adaptarse a distintos niveles de entrenamiento (Bicio 2021).

Ilustración 2-5: Ejemplo del rodillo de transmisión directa

Fuente: (Bicio 2021)

2.9.2. Comparación entre los tipos de rodillos ciclo simuladores

En la **Tabla 2-5** se presentan las características de los distintos tipos de rodillos ciclo simuladores. Donde el rodillo magnético se destaca como la mejor opción para la mayoría de los ciclistas debido a su equilibrio entre precio, rendimiento y facilidad de uso.

Tabla 2-5: Características de los rodillos ciclo simuladores

Característica/Rodillo	De equilibrio	Magnético	De fluidos	De transmisión directa
Precio	Bajo	Bajo-Medio	Medio-Alto	Alto
Ruido	Alto	Bajo	Bajo	Bajo
Sensación de pedaleo	Poco realista	Realista	Realista	Muy realista
Compatibilidad con aplicaciones	No	Sí	Sí	Sí
Simulación de pendientes	No	Limitada	Sí	Sí
Mantenimiento	Bajo	Bajo	Bajo	Alto
Facilidad de uso	Alta	Alta	Alta	Media
Portabilidad	Alta	Alta	Media	Baja

Fuente: (Sánchez 2020; Rutasenbici 2023; Briseño 2023; Bicio 2021)

Realizado por: Morales, 2024

2.10. Motores de corriente continua

Son dispositivos electromecánicos que convierten la energía eléctrica en movimiento rotativo, esto mediante la interacción de un campo magnético generado por imanes permanentes o electroimanes y una corriente eléctrica que circula por bobinas ubicadas en el rotor o en el estator del motor (Briseño 2023).

2.10.1. Motor paso a paso

Son de los motores más usados en diversos trabajos de electrónica y demás aplicaciones, como impresoras 3D, máquinas de grabado láser y robots. Su distintivo radica en su tamaño compacto y su habilidad para efectuar movimientos precisos y en pasos pequeños estas características lo vuelven idóneo para controlar movimientos detallados en una variedad de proyectos creativos y de automatización (Quintana 2023).

2.10.2. Servomotor

Es un dispositivo destinado a gestionar la posición de un eje en un instante específico, permite encendidos y apagados suaves, además de un posicionamiento altamente preciso. Estos motores tienen la capacidad de producir una cantidad significativa de potencia y son conocidos por su eficiencia energética notable. Su estructura incluye un motor de corriente continua (DC), un conjunto de engranajes y un circuito electrónico de control que monitorea y dirige sus operaciones (Quintana 2023).

2.11. Comparación entre los tipos de motores

La **Tabla 2-6** presenta la comparativa entre las características del motor paso a paso y el servomotor, donde se selecciona el tipo de motor paso a paso debido a su mayor torque a velocidades bajas, lo cual será necesario debido a que la cantidad de revoluciones estimadas por un ciclista es baja, por lo tanto, la velocidad de accionamiento de este no se verá influenciado en su torque.

Tabla 2-6: Características del motor paso a paso y servomotor.

Característica /Tipo de motor	Paso a paso	Servomotor
Retroalimentación	Sin retroalimentación	Con enconder o sensor de posición
Codificador	No requiere	Requiere
Precisión	Menor precisión	Mayor precisión en posición y velocidad
Velocidad	Velocidad fija	Velocidad variable según la carga
Torque	Mayor torque a baja velocidad	Menor torque a velocidades bajas, constante a altas velocidades
Precio	$5 - $200	$3 - $1000

Fuente: (Quintana 2023)

Realizado por: Morales, 2024.

2.12. Sensor para adquisición de datos

Es un dispositivo que detecta y convierte magnitudes físicas en señales eléctricas que pueden ser procesadas por un sistema electrónico permitiendo monitorear y controlar variables del entorno para tomar (Guzman y Sanchez 2020).

2.12.1. Sensor magnético

Comúnmente usado para identificar campos magnéticos, ya sea mediante la atracción o repulsión generada por imanes o energía eléctrica. Su función principal consiste en proporcionar mediciones precisas, exactas y sensibles de cambios en el entorno magnético. En la **Ilustración 2-6** se evidencia el sensor de efecto Hall de dos hilos junto con el imán para la generación de interrupciones. En la industria, este dispositivo desempeña diversas funciones, destacándose por su capacidad para detectar niveles de precisión, su aplicación en sistemas de alarmas, la determinación precisa de posiciones, el control de velocidad y la automatización de puertas y

ventanas. Entre las ventajas clave de los sensores magnéticos se encuentran su instalación rápida y sencilla, la seguridad en las conexiones, la capacidad para detectar objetos sin contacto directo, y su amplia disponibilidad en el mercado, lo que contribuye a su asequibilidad. Estas características convierten a los sensores magnéticos en herramientas versátiles y accesibles para una variedad de aplicaciones industriales y comerciales (Industrias GSL 2021).

Ilustración 2-6: Ejemplo de sensor magnético

Realizado por: Morales, 2024.

2.12.2. Sensor óptico

Es también conocido como fotocompuerta, combina un LED emisor infrarrojo con un detector infrarrojo para registrar la interrupción de la ruta de luz tal como se evidencia en la **Ilustración 2-7**. Este dispositivo se basa en la capacidad de percibir cambios en la intensidad de la luz infrarroja cuando la ruta de luz se ve bloqueada. La función principal de un sensor óptico reside en su habilidad para medir la posición y velocidad cuando se vincula con temporizadores y objetivos específicos que alternan entre opacidad y transparencia. Su aplicación abarca diversos campos y destaca en situaciones donde la detección precisa de obstáculos o cambios en la posición es fundamental. Este tipo de sensor resulta particularmente valioso en entornos industriales y en sistemas automatizados donde la exactitud y la confiabilidad en la detección son críticas (Velásquez et al. 2021).

Ilustración 2-7: Ejemplo de sensor óptico

Realizado por: Morales, 2024.

2.12.3. Codificador rotatorio

Utiliza principios de detección fotoeléctrica, electromagnética capacitiva o inductiva para identificar la posición y los cambios en un objeto. Convierte esta información en señales eléctricas para su salida, que se utilizan como retroalimentación en el control de movimiento. Su diseño se observa en la **Ilustración 2-8**. Es capaz de transformar el desplazamiento y la velocidad angular en impulsos eléctricos para su salida digital, lo que lo hace útil para medir la velocidad y controlarla con precisión. Este tipo de sensor es ampliamente utilizado en la industria para el posicionamiento preciso de motores (Qiaoqi Wang 2018).

Ilustración 2-8: Ejemplo de codificador rotatorio

Fuente: (Qiaoqi Wang 2018)

2.13. Comparación entre los sensores de adquisición de datos

En la **Tabla 2-7** se presentan las principales características de algunos sensores de adquisición de datos, destacando el sensor magnético por varias razones. Ofreciendo una alta precisión, su sensibilidad ajustable de 1 mT a 100 mT permite una detección precisa de campos magnéticos. También es altamente durable y fácil de instalar. Además, de tener una excelente relación entre calidad y precio.

Tabla 2-7: Características de los tipos de sensores para adquisición de datos

Característica/ Tipo	Sensor magnético	Sensor óptico	Codificador rotatorio
Tipo de detección	Campo magnético	Luz infrarroja	Posición mecánica
Precisión	±0.5% a ±2%	±0.1% a ±1%	±0.01% a ±0.05%
Rango de medición	0.5 mm a 10 m	1 mm a 100 m	0° a 360°
Sensibilidad	Ajustable de 1 mT a 100 mT	Ajustable de 1 lux a 1000 lux	Ajustable de 1 pulso por revolución (PPR) a 10000 PPR
Costo	$0.50 a $70	$1 a $100	$5 a $500

Durabilidad	Alta	Media	Alta
Facilidad de instalación	Alta	Media	Baja

Fuente: (Guzman y Sanchez 2020; Qiaoqi Wang 2018; Velásquez et al. 2021; Industrias GSL 2021)
Realizado por: Morales, 2024.

2.14. Sensores para el cálculo de sentido de giro

Son dispositivos utilizados para medir la orientación angular de un objeto en relación con un sistema de referencia. Estos sensores pueden proporcionar información sobre la inclinación, la orientación o el movimiento angular de un objeto en una o varias dimensiones (Galarza 2023).

2.14.1. Sensor de campo electromagnético

Está diseñado para detectar y medir la intensidad y dirección de los campos magnéticos en su entorno. Funciona mediante principios físicos y electromagnéticos para convertir la información del campo magnético en señales eléctricas que pueden ser procesadas y utilizadas para diversos fines. Estos sensores son esenciales en aplicaciones como la navegación, donde se emplean para determinar la orientación y posición relativa respecto a campos magnéticos terrestres o artificiales. También se utilizan en la detección de metales, el control de motores y la medición de corriente eléctrica, entre otras aplicaciones. Los sensores de campo electromagnético pueden variar en tamaño, principio de funcionamiento y precisión, y la elección del sensor adecuado depende de la aplicación específica y los requisitos de medición (Arias 2020).

2.14.2. Sensor acelerómetro

Es un dispositivo que mide la aceleración experimentada por un objeto en movimiento. Puede detectar cambios en la velocidad lineal en uno, dos o tres ejes, dependiendo de su diseño. Los acelerómetros se utilizan en una variedad de aplicaciones, como en dispositivos electrónicos para detectar la orientación de la pantalla, en sistemas de navegación para determinar la posición y en equipos médicos para monitorear la actividad física (Palacios, Jiménez y Pérez 2023).

2.15. Comparación entre los sensores para la adquisición de ángulos

En la **Tabla 2-8** se muestran las diferentes características de el sensor de campo electromagnético y el sensor acelerómetro. Definiendo al sensor de campo electromagnético como una mejor opción debido a su capacidad para medir una amplia gama de campos magnéticos, su precisión confiable y su sensibilidad ajustable para detectar cambios sutiles. Además, tiene un consumo de

energía similar al del acelerómetro y un precio asequible, lo que lo hace ideal para diversas aplicaciones.

Tabla 2-8: Comparación entre los sensores para cálculo de sentido de giro

Característica /Sensor	Campo electromagnético	Acelerómetro
Rango de medición	0.1 mT - 10 T	±2 g - ±16 g
Precisión	±0.5% - ±2%	±0.01 g - ±0.1 g
Sensibilidad	0.1 mV/mT - 10 mV/mT	0.01 g/LSB - 0.1 g/LSB
Voltaje de alimentación	3.3 V – 5 V	3.3 V – 5 V
Consumo de energía	1 mA – 10 mA	1 mA – 10 mA
Interfaz	Analógica, digital (I2C)	Digital (I2C)
Precio	$0.50 - $70.00	$3.00 - $520.00

Fuente: (Arias 2020; Palacios, Jiménez y Pérez 2023)

Realizado por: Morales, 2024.

2.16. Protocolos de comunicación

Son conjuntos de reglas y convenciones que permiten que dos o más dispositivos se comuniquen entre sí. Estas reglas definen cómo se deben enviar, recibir y procesar los datos para garantizar una comunicación efectiva y confiable. Los protocolos especifican aspectos como el formato de los mensajes, la secuencia de intercambio de mensajes, la detección y corrección de errores, y la gestión de la conexión (Chambers 2021).

2.16.1. MQTT

Es una forma de comunicación asincrónica utilizada en entornos de comunicación entre máquinas (M2M), especialmente en el contexto del Internet de las cosas. Opera a través de un modelo de publicación y suscripción para intercambiar mensajes. Se destaca por ser ligero, lo que lo hace ideal para su implementación en redes con ancho de banda limitado y alta latencia. Su flexibilidad permite su aplicación en una variedad de dispositivos y servicios dentro del ecosistema del IoT. Además, se compone de un Broker MQTT el cual opera como intermediario para la comunicación entre dispositivos y aplicaciones, siendo el encargado de recibir, filtrar y distribuir mensajes, permitiendo que los dispositivos suscritos a ciertos tópicos envíen y reciban información de manera eficiente y organizada. Esto permite que el broker MQTT asegure una entrega confiable de mensajes en entornos con múltiples dispositivos conectados (Biot 2020).

2.16.2. HTTP

Este protocolo define la forma en que los mensajes son formulados y transmitidos, y cómo los servidores y los navegadores web responden a diversas solicitudes y comandos. Opera en un modelo de cliente-servidor, donde un cliente, generalmente un navegador web, realiza una solicitud a un servidor web que hospeda un recurso, y el servidor responde con los datos solicitados. HTTP es un protocolo sin estado, lo que significa que cada solicitud se procesa de forma independiente, sin que el servidor mantenga información sobre las solicitudes anteriores del mismo cliente (Bracho y Guañuna 2023).

2.17. Comparación entre los protocolos de comunicación

En la **Tabla 2-9** se presenta una comparativa entre el protocolo de comunicación MQTT y HTTP. Donde el protocolo MQTT se destaca como uno de los mejores protocolos de comunicación debido a su eficiencia en el uso de ancho de banda y energía, su capacidad para manejar grandes cantidades de dispositivos de forma escalable, su baja latencia que lo hace ideal para aplicaciones en tiempo real, y sus opciones avanzadas de seguridad para garantizar la protección de la comunicación.

Tabla 2-9: Comparación entre los protocolos de comunicación

Característica /Protocolo	MQTT	HTTP
Modelo de comunicación	Publicación/Suscripción	Solicitud/Respuesta
Uso principal	Comunicación IoT	Servidores web
Eficiencia	Eficiente en ancho de banda y energía.	Menos eficiente, requiere conexiones nuevas.
Seguridad	Admite seguridad a través de TLS/SSL para cifrado de extremo a extremo	Requiere medidas adicionales como HTTPS para asegurar la comunicación
Escalabilidad	Altamente escalable para muchos dispositivos.	Menos escalable, requiere conexiones separadas.

Fuente: (Chambers 2021; Bracho y Guañuna 2023; Biot 2020)

Realizado por: Morales, 2024.

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se detallan los requerimientos que se deben cumplir dentro del proyecto, además se establece la concepción de la arquitectura, las etapas que lo conforman, y adicionalmente se describen los algoritmos y las conexiones que se requieren para su implementación.

3.1. Requerimientos

A continuación, se presentan los requerimientos necesarios para el desarrollo de un simulador usando realidad virtual aplicado a vehículos ligeros manuales para entretenimiento:

3.1.1. Requerimientos de hardware

- La tarjeta de desarrollo de la familia Espressif contará con una conexión inalámbrica WiFi que permitirá enlazarse con un servidor MQTT.
- Los sensores de tipo magnético y de campo electromagnético realizarán la adquisición de información de variables físicas para cálculos de velocidad y sentido del giro.
- Se utilizará un rodillo ciclo simulador de tipo magnético para generar el efecto de resistencia al movimiento de la rueda trasera de la bicicleta.
- El motor paso a paso, y su controlador permitirán gestionar el rodillo en función de las órdenes de la tarjeta de desarrollo.

3.1.2. Requerimientos de software

- La interfaz de realidad virtual se desarrollará en Unity y permitirá mediante el dispositivo Oculus experimentar ambientes tales como rectas, curvas de $(0°, 45°]$ geométricos y pendientes de [-45°, 45°] geométricos en un solo escenario.
- El sistema podrá ejecutarse tanto en sistemas operativo Windows como en Android.
- Proporcionar una experiencia inmersiva mediante una relación entre el hardware y el software de simulación en los diferentes ambientes definidos.
- Implementar una base de datos que permita el registro de distintos usuarios.
- Durante cada simulación se deberá almacenar y visualizar la velocidad, pendiente y tiempo obtenidas por usuario.

3.2. Concepción de la arquitectura general del proyecto

3.2.1. Etapas

En la **Ilustración 3-1** se puede visualizar las cinco etapas esenciales para llevar a cabo el proyecto las cuales son: operación del usuario, adquisición de datos, simulación y comunicación, control de motor, registro de variables.

- **Etapa de operación del usuario. –** Dentro de esta etapa el usuario interactúa de forma directa con el sistema establecido en base al vehículo ligero. Incluye los preparativos físicos requeridos para el correcto funcionamiento de la simulación.
- **Etapa de adquisición de datos. -** Está compuesta por un sensor para la detección de campos magnéticos mediante la creación de pulsos, los cuales son necesarios para ser procesados como interrupciones a través de la tarjeta de desarrollo.
- **Etapa de simulación y comunicación. -** Se encarga de la simulación del terreno virtual y la comunicación entre los diversos componentes tanto del software como del Hardware, procesando los datos adquiridos para la generación de cambios en el entorno físico y virtual.
- **Etapa de control de motor. –** Permite crear la experiencia de resistencia al movimiento al usuario, mediante los cambios en el terreno virtual durante la sesión de entretenimiento. Mediante un controlador se puede obtener un correcto funcionamiento del motor.
- **Etapa de registro de variables. –** Por medio de esta etapa se logra la visualización de las variables velocidad, pendiente y tiempo, recopiladas durante el periodo de simulación, y el operador puede hacer uso de este registro para el análisis de su rendimiento.

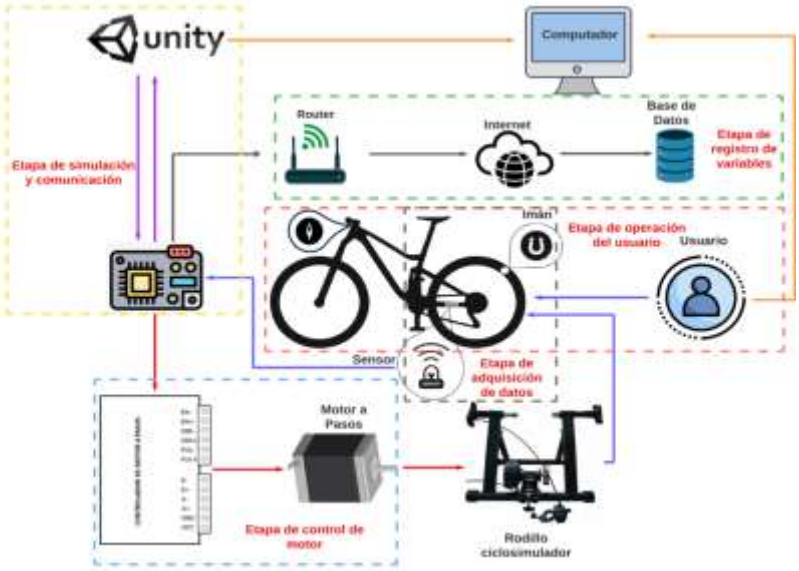


Ilustración 3-1: Concepción general del proyecto.

Realizado por: Morales, 2023.

3.3. Selección del Hardware para el simulador de realidad virtual

A continuación, se presentan las características más relevantes de los componentes hardware empleados en el simulador de realidad virtual.

3.3.1. Oculus Quest 2

El dispositivo de realidad virtual autónomo de Meta mostrado en la **Ilustración 3-2**, ofrece seguimiento de 6 grados de libertad para total movilidad en entornos virtuales. Es independiente de ordenadores o consolas, permitiendo su uso en cualquier lugar. Con una resolución de 1832 x 1920 píxeles por ojo y una tasa de refresco de 90Hz, proporciona una experiencia inmersiva y fluida. Tiene 64 GB de almacenamiento y controladores ergonómicos con retroalimentación háptica y respuesta táctil, ofreciendo precisión y comodidad para usuarios diestros y zurdos. Los mandos se conectan de forma inalámbrica, brindando libertad de movimiento sin cables (Salas Noain y Delgado del Castillo 2023).

Ilustración 3-2: Dispositivo Oculus Quest 2

Realizado por: Morales, 2023.

En la **Tabla 3-1** se muestran las principales características con las que cuenta el dispositivo de realidad virtual seleccionado.

Tabla 3-1: Características de Oculus Quest 2

Característica/Dispositivo	Oculus Quest 2
Plataforma	Autónomo, no requiere PC ni consola
Resolución de pantalla	1832 x 1920 por ojo
Tasa de refresco	72 Hz (compatible con 90 Hz)
Conexión	Inalámbrica o por cable (link a PC)
Compatibilidad	Amplia gama de dispositivos y juegos de Oculus
Precio	Alrededor de $500 USD

Fuente: (Oculus 2020)

Realizado por: Morales, 2023.

3.3.2. Rodillo ciclo simulador "EAGLE"

A la bicicleta se ha adaptado un rodillo ciclo simulador Eagle el cual se puede observar en la **Ilustración 3-3**, que permitirá ubicar de manera estática la bicicleta. Este rodillo cuenta con un tambor que contiene un imán en forma de disco, que puede ser atraído por un imán de neodimio, el cual, se controla mediante una perilla que se puede ubicar en el volante (Olarte 2017). Sin embargo, se ha descartado el uso del imán por la baja sensación de resistencia al movimiento del usuario, lo cual generaría un notable decremento en la sensación inmersiva en la aplicación. En su lugar se ha optado por usar un regulador mecánico para la velocidad, mismo que será controlado por una perilla en la parte inferior la cual varía su contacto directo con la rueda trasera del vehículo aumentando o disminuyendo la fricción.

Ilustración 3-3: Rodillo ciclo simulador Eagle.

Realizado por: Morales, 2023

En la **Tabla 3-2** se presentan las principales características de este rodillo ciclo simulador.

Tabla 3-2: Características del rodillo ciclo simulador "EAGLE"

Característica /Dispositivo	Rodillo
Material	Acero aleado
Marca	Eagle
Material del cuerpo	Acero inoxidable
Material del marco	Acero aleado
Mecanismo de resistencia	Magnético
Capacidad de carga	140 kg
Dimensiones	77x56x47,5 cm
Peso	10 kg

Fuente: (Olarte 2017)

Realizado por: Morales, 2023.

3.3.3. Motor a pasos NEMA 17 17HS4401

Este motor cumple con el estándar Nema 17 para el montaje dentro de la etapa electrónica. Se trata de un motor bipolar con un ángulo de paso de 1,8 grados, equivalente a 200 pasos por vuelta, como se muestra en la **Ilustración 3-4**. Cada bobina admite una corriente de 1,7 amperios y puede generar un torque de hasta 4 kg/cm. Destaca por su capacidad para lograr un posicionamiento preciso, ofreciendo precisión en la rotación y siendo fácil de controlar. Este motor está configurado con cuatro cables codificados por colores, conectados a dos bobinas. Al activar estas bobinas en una secuencia específica, el motor puede girar en ambas direcciones. El par motor de un NEMA 17 puede variar según el modelo y la corriente suministrada. Dada su dimensión y características técnicas, este tipo de motor se utilizará en el desarrollo del proyecto (Toro 2023).

Ilustración 3-4: Motor NEMA 17HS4401

Realizado por: Morales, 2023.

En la **Tabla 3-3** se detalla las especificaciones técnicas de este tipo de motor paso a paso.

Tabla 3-3: Especificaciones técnicas del motor paso a paso Nema 17.

Especificación Técnica	Descripción
Tipo de motor	Bipolar
Voltaje de operación	12 [V CC]
Corriente	1 [A]
Torque	4000 g/cm
Ángulo del paso	1,8 grados
Pasos por vuelta	200 pasos
Resistencia por fase	1.5 ohms
Diámetro del eje	5 mm
Precio	$18

Fuente: (Toro 2023)

Realizado por: Morales, 2023.

3.3.4. Controlador de motor paso a paso TB6600

Este elemento es presentado en la **Ilustración 3-5**, el cual se destaca por ser compatible con motores bifásicos incluyendo los populares NEMA 17. Este controlador ofrece una alta precisión de la velocidad y la dirección, permitiendo ajustar los micro pasos y la corriente de salida a través de un interruptor DIP de 6 posiciones. Para obtener documentación sobre su funcionamiento, (Figueroa 2023) información detallada sobre su uso.

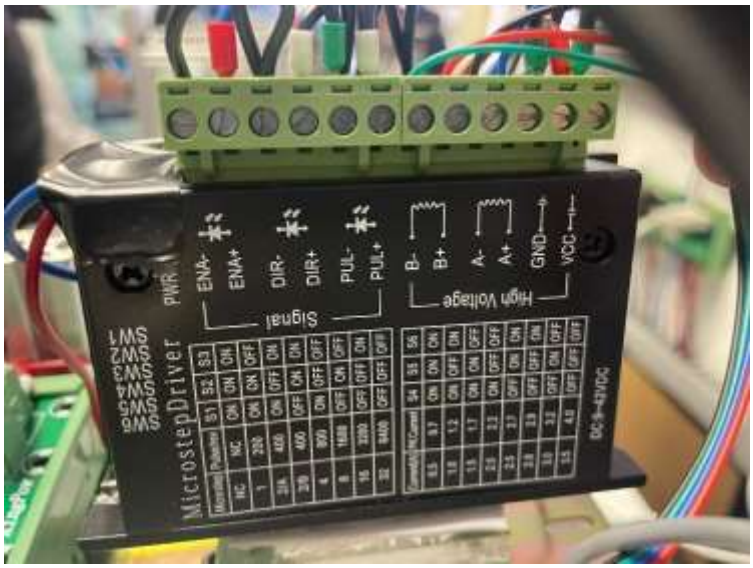


Ilustración 3-5: Controlador TB6600 de motores a paso.

Realizado por: Morales, 2023.

Las principales características de este elemento se detallan en la **Tabla 3-4**.

Tabla 3-4: Especificaciones técnicas del controlador TB6600.

Especificación Técnica	Descripción
Voltaje de operación	9- 42 [V CC]
Corriente de accionamiento	1 [A] - 4 [A]
Interruptor de marcación	6 bits
Motores permitidos	Adecuado para motores con 2 bobinas Nema 17 y Nema 23
Temperatura de funcionamiento	-10 ℃ hasta 45 ℃
Control de corriente media	Combinaciones posibles desde 0,5 [A] hasta 3,5 [A]

Fuente: (Culiáñez 2023)

Realizado por: (Morales, 2023)

3.3.5. Fuente de alimentación

La fuente de alimentación seleccionada se muestra en la **Ilustración 3-6**.

Ilustración 3-6: Fuente de alimentación

Fuente: (TECNEU 2023)

Las especificaciones técnicas de este elemento se presentan en la **Tabla 3-5**.

Tabla 3-5: Especificaciones técnicas de la fuente de alimentación.

Especificación Técnica	Descripción
Voltaje de entrada	110-220 [VAC] ±15%
Voltaje de salida	24 [VDC]
Corriente de salida	5 [A]
Potencia	120 [W]
Eficiencia	81%
Dimensiones	19,8 x 9,8 x 4,19 [cm]

Fuente: (Pérez Pujol 2022)

Realizado por: (Morales, 2023)

3.3.6. Sensor de efecto Hall de 2 hilos

Este sensor es un interruptor magnético que funciona mediante efecto Hall. Consta de 2 hilos tal como se observa en la **Ilustración 3-7**, y es apto para altas frecuencias lo cual es adecuado para este tipo de proyecto (Industrias GSL 2021).

Ilustración 3-7: Sensor magnético por efecto Hall.

Realizado por: Morales, 2023

La **Tabla 3-6** presenta las principales especificaciones técnicas del sensor magnético de efecto Hall usado.

Tabla 3-6: Características del sensor de efecto Hall

Especificación Técnica	Descripción
Tipo de sensor	Efecto Hall
Número de hilos	2 hilos
Tipo de salida	Normalmente abierto
Rango de operación	2 a 10 mV/mT
Dimensión	28.6 x 6.40 x 19.0 mm
Voltaje de operación	2.7 a 24 V
Corriente de funcionamiento	4 a 8 mA
Precisión	$\pm$ 2% a $\pm$ 5%
Temperatura de operación	-40 a 125 $°C$
Salida	Digital

Fuente: (Littlefuse 2022)

Realizado por: (Morales, 2023)

3.3.7. Sensor de campo electromagnético GY-273

Este dispositivo integra el sensor magnetómetro HMC5883L del fabricante ARCELI mostrado en la **Ilustración 3-8**, el cual tiene la capacidad de detectar el campo magnético en tres dimensiones. Se utiliza ampliamente en aplicaciones que requieren la determinación de la orientación respecto

al norte magnético de la tierra. Este módulo viene equipado con la electrónica necesaria para su conexión sencilla a placas como Arduino (Arias 2020).

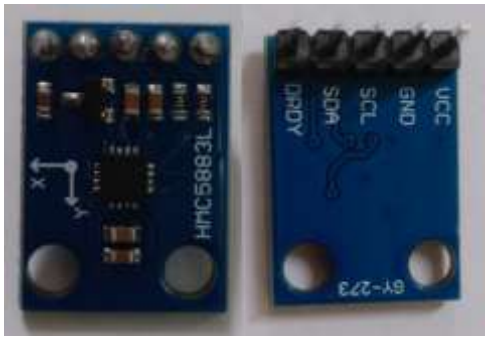


Ilustración 3-8: Sensor de campo electromagnético.

Realizado por: Morales, 2024.

A continuación, en la **Tabla 3-7** se muestran las especificaciones técnicas del sensor GY-273, donde es importante resaltar que para mayor precisión se debe usar el valor de la declinación magnética de cada ubicación.

Tabla 3-7: Especificaciones técnicas del sensor GY-273.

Especificación Técnica	Descripción
Tipo de sensor	Magnetómetro de 3 ejes
Rango de medición	±1.3 a ±8 Gauss
Resolución	5 milli-gauss
Precisión	1° a 2°
Interfaz de comunicación	I2C
Voltaje de alimentación	3 a 5 V
Consumo de corriente	3.5 mA típico
Temperatura de operación	-40°C a +85°C
Dimensiones	13.5 mm x 10.5 mm x 4 mm
Declinación Magnética	-4°12’

Fuente: (Honeywell 2023)

Realizado por: (Morales, 2023)

3.3.8. ESP32

Esta tarjeta de desarrollo ha ganado popularidad gracias a su versatilidad, capacidades avanzadas y su bajo costo. Desarrollado por Espressif Systems, este dispositivo ofrece conexiones WiFi y Bluetooth, lo que lo hace ideal para una amplia variedad de aplicaciones en el ámbito del Internet de las Cosas (IoT), la automatización del hogar y el prototipado. Destacada por su potencia de

procesamiento, sus múltiples interfaces periféricas y su habilidad para manejar aplicaciones complejas. Su modelo físico se visualiza en la **Ilustración 3-9**, la ESP32 es una opción atractiva para proyectos que requieren conectividad inalámbrica y poder de procesamiento en un paquete compacto y asequible (Espressif Systems 2020).

Ilustración 3-9: Placa de desarrollo esp32.

Realizado por: Morales, 2023.

En la **Tabla 3-8** se presenta las principales especificaciones técnicas de la tarjeta de desarrollo ESP32.

Tabla 3-8: Características de la ESP32

Característica /Dispositivo	ESP32
Microcontrolador	ESP32 (Tensilica)
Velocidad	160 MHz (dual core)
Memoria RAM	520 KB
Almacenamiento	Flash integrado
GPIO	36-40 pines
Conectividad inalámbrica	Wi-Fi, Bluetooth
Sistema Operativo	FreeRTOS
Consumo de energía	Variable (bajo)
Precio	Alrededor de $12 USD

Fuente: (Espressif Systems 2020)

Realizado por: Morales, 2023.

En la **Tabla 3-9** se detallan los terminales a usar de la ESP32, junto con su función determinada.

Tabla 3-9: Terminales usados de la ESP32

Terminal	Función
GPIO 14	Terminal DIR+ TB6600

GPIO 21	Pin SDA Sensor campo electromagnético
GPIO 22	Pin SCL Sensor campo electromagnético
GPIO 25	Terminal PUL+ TB6600
GPIO 26	Terminal ENA+ TB6600
GPIO 32	Sensor para el cálculo de velocidad

Realizado por: Morales, 2023.

3.4. Diagrama de conexión electrónica

En la **Ilustración 3-10** se muestra el esquema de conexiones realizado en el software de diseño electrónico Fritzing versión 1.0.1. Con el propósito de comprender el funcionamiento de esta herramienta (Germana 2011) pone a disposición de los usuarios un manual detallado de uso.

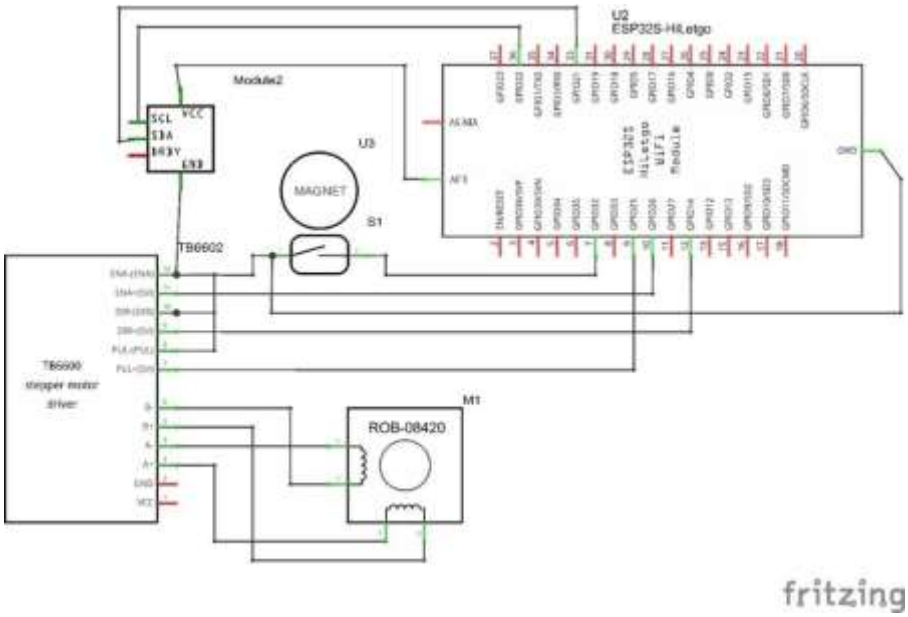


Ilustración 3-10: Esquema de conexiones en el software Fritzing.

Realizado por: Morales, 2023.

Este diseño está conformado por la tarjeta de desarrollo ESP32, el controlador de motores a pasos TB6600, el cual es el encargado de la activación y procesamiento del motor, también se cuenta con el motor a Pasos NEMA 17 donde se puede destacar la configuración de las bobinas del motor las cuales varían dependiendo del modelo, el imán ubicado a lado del sensor para su activación, también se cuenta con el sensor GY-273 el cual es el encargado de detectar el giro del manubrio.

El controlador TB6600 es alimentado por la fuente de poder de 110 [VAC] a 24 [VCC], los pines ENA-, DIR-, PUL- son conectados directamente a tierra, estableciendo de esta forma una configuración de conexión cátodo.

El motor a pasos NEMA 17 posee 4 pines para conectar sus bobinas. Mediante un multímetro se ha podido probar la continuidad entre las bobinas para no causar desperfectos en el motor a la hora del funcionamiento, de lo cual, el primer pin y el tercer pin conforman la bobina B, el segundo pin y el cuarto pin son de la bobina A. Estos pares de bobinas se deben conectar correctamente al controlador TB6600 en los pines B+, B- para la bobina B, y en la bobina A se debe conectar a los pines A+, A-.

El sensor magnético de efecto Hall mediante programación se ha configurado con una resistencia interna PULL-UP para mejorar su eficacia al estar conectado 1 hilo a tierra para utilizar interrupciones con pulsos en bajo. Además, se ha añadido un imán el cual simula la activación de dicho sensor.

El sensor GY-273 funciona a un voltaje de operación de 3.3 [V] con la dirección I2C 0X3C conectado a SDA y SCL.

La **Ilustración 3-11** contiene el diseño PCB del circuito realizado en el software Proteus 8.12, el cual es un destacado software de automatización para el diseño electrónico, este programa facilita a los diseñadores de placas de circuito impreso la integración fluida de diagramas esquemáticos, la disposición de componentes, el enrutamiento de las PCB y el acceso a una amplia biblioteca de contenidos. Además, también se visualiza la implementación física de la placa PCB.

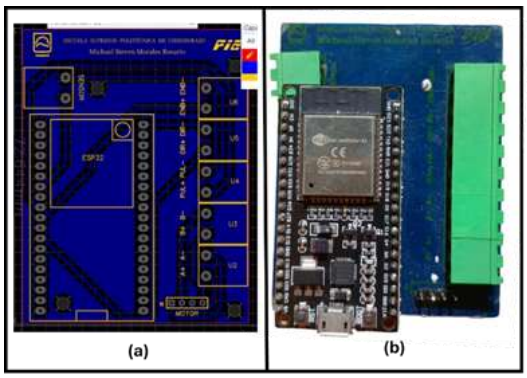


Ilustración 3-11: a) Diseño de la placa PCB. b) Placa PCB física

Realizado por: Morales, 2024.

3.5. Diseños 3D implementados en el prototipo

Con el propósito de mantener a los elementos en posiciones específicas y estables, se ha optado por diseñar estructuras que garanticen estas acciones. Para todos los diseños se empleó el programa de software para modelado Fusion 360 en su versión 2.0.18441 x86_64. (Autodesk 2024) brinda un manual de funcionamiento para que sus usuarios puedan ambientarse con cada apartado de esta herramienta.

La fabricación se ha realizado mediante la impresora marca Creality con modelo Ender-3 Max Neo, (Creality 2022) presenta las diferentes especificaciones técnicas de este producto. El material utilizado ha sido PLA, contando con un 20% de relleno en todas sus estructuras.

3.5.1. Estructuras para el motor

Debido a que el motor no se encuentra estable, al momento de su activación existirán diversos errores, por lo cual se ha diseñado una estructura la cual se visualiza en la **Ilustración 3-12,** esta cuenta con 2 ganchos que se podrán acoplar a la base de el rodillo ciclo simulador, junto con una base en la que se ha considerado un espacio según las dimensiones del motor y una abertura para sus cables.

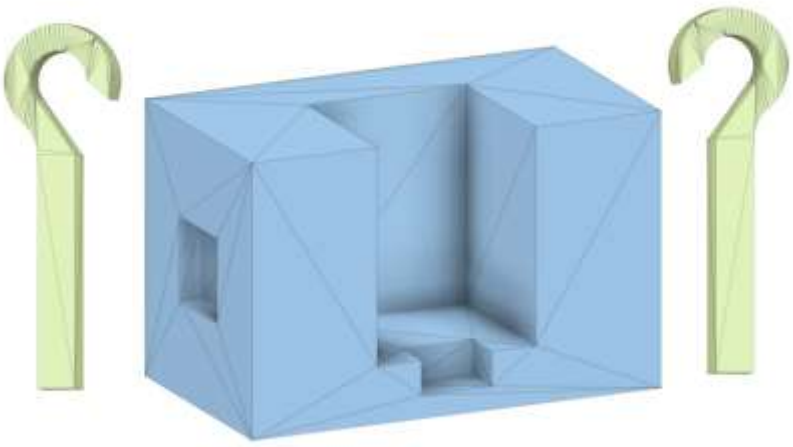


Ilustración 3-12: Diseño de estructura para el acople del motor con el rodillo

Realizado por: Morales, 2024.

En la **Tabla 3-10** se presenta las medidas fundamentales expresadas en milímetros de la base y los ganchos del motor.

Tabla 3-10: Medidas fundamentales de la estructura para el motor

Medida/ Pieza	Base	Gancho
Alto	630 mm	2850 mm
Ancho	1110 mm	300 mm
Largo	710 mm	250 mm

Realizado por: Morales, 2023.

3.5.2. Soporte para el sensor de campo electromagnético

Con el propósito de mantener la estabilidad y posición fija del sensor GY-273 en la bicicleta, se ha optado por diseñar un soporte adaptable al manubrio, este se evidencia en la **Ilustración 3-13**.

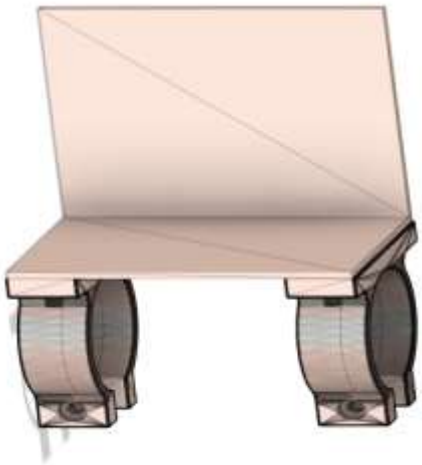


Ilustración 3-13: Diseño de soporte para el sensor de campo electromagnético

Realizado por: Morales, 2024

En la **Tabla 3-11** se presenta las medidas fundamentales expresadas en milímetros del soporte para el sensor GY-273.

Tabla 3-11: Medidas fundamentales del soporte para el sensor GY-273

Medida/Pieza	Base
Alto	1094 mm
Ancho	480 mm
Largo	710 mm

Realizado por: Morales, 2024

3.6. Incorporación de elementos físicos del sistema

El elemento principal alrededor de este proyecto es el vehículo, en este caso una bicicleta modelo GTI aro 27 la cual se muestra en la **Ilustración 3-14**, con la cual, el usuario estará en contacto directo.

Ilustración 3-14: Bicicleta GTI.

Realizado por: Morales, 2024

La correcta implementación entre el rodillo ciclo simulador y la bicicleta se observa en la **Ilustración 3-15.**

Ilustración 3-15: Acople de la bicicleta con el rodillo ciclo simulador

Realizado por: Morales, 2024

3.6.1. Implementación de los sensores

En este apartado se detalla la implementación de los elementos para la adquisición de datos, donde se específica la posición en la que se debe ubicar cada sensor.

3.6.1.1. Implementación de sensor de efecto Hall

La implementación del imán para la detección del sensor magnético por efecto Hall se muestra en la **Ilustración 3-16**. Para la ubicación de este elemento es necesario que el sensor magnético se encuentre en algún punto del contorno del perímetro recorrido por el imán.

Ilustración 3-16: Imán para la detección por efecto Hall.

Realizado por: Morales, 2024

3.6.1.2. Implementación de sensor de campo electromagnético

Este elemente ha sido ubicado junto con su soporte en el manubrio de la bicicleta para poder captar la variación en la inclinación del campo magnético inicial, esto se evidencia en la **Ilustración 3-17**.

Ilustración 3-17: Implementación física del sensor GY-273 con el soporte en la bicicleta

Realizado por: Morales, 2024

3.6.2. Implementación del sistema regulador de fuerza

El control del motor incluye la interacción con el rodillo ciclosimulador, para cumplir el requisito de proporcionar una experiencia inmersiva mediante, una interacción envolvente y realista para el usuario. Se ha realizado un diseño en la herramienta de software Tinkercad para crear una representación conceptual del papel que cumple el motor en el proyecto, (OpenLab 2022) presenta un manual de uso para la comprensión de esta herramienta.

La perilla para acercar o alejar el cilindro que contiene al rodillo ciclosimulador, es usado para generar la fricción con la rueda. El acople propuesto entre el motor NEMA 17 y la perilla del rodillo ciclosimulador se puede se visualizar en la **Ilustración 3-18.**

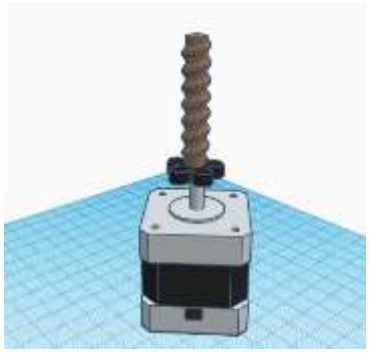


Ilustración 3-18: Acople de motor NEMA 17 con perilla del rodillo

Realizado por: Morales, 2024.

La representación del sistema de regulación de fricción se evidencia en la **Ilustración 3-19**.

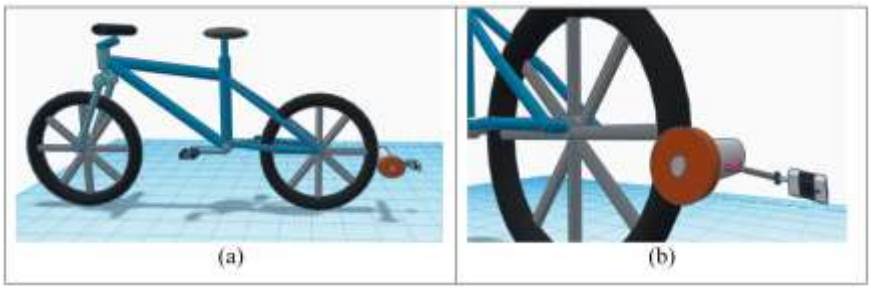


Ilustración 3-19: Implementación del sistema regulador de fricción en Tinkercad. a) Vista general, b) Vista lateral

Realizado por: Morales, 2024.

La implementación física de la estructura que contendrá al motor se muestra en la **Ilustración 3-20.**

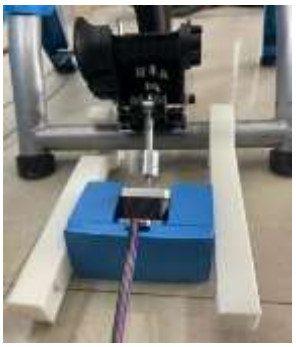


Ilustración 3-20: Implementación física de la estructura para el motor

Realizado por: Morales, 2024.

3.7. Desarrollo del software para el simulador de realidad virtual

En esta sección se detallan los softwares seleccionados para el desarrollo de la programación de los elementos de hardware previamente definidos, así como el proceso de creación del entorno de realidad virtual y la conexión del sistema con diferentes plataformas. También se abordan las consideraciones necesarias para garantizar el correcto funcionamiento del simulador.

3.7.1. Creación del ambiente virtual en Unity

Para cumplir el requerimiento de poder experimentar distintos ambientes como rectas, curvas y pendientes se utiliza el motor gráfico de Unity 2022.3.13f1 LTS, esta versión es seleccionada debido a que al ser una versión LTS cuenta con el soporte necesario para trabajar con sus librerías de manera estable.

Para la creación de un proyecto, la implementación de un terreno y poder conocer las diversas propiedades con se puede llevar a cabo con este motor gráfico, (Unity 2024a) presenta una documentación con instrucciones en su página oficial mediante las cuales se pudo llegar a crear el entorno mostrado en la **Ilustración 3-21**.

Ilustración 3-21: Entorno virtual creado en Unity 2022.3.13f1 LTS

Realizado por: Morales, 2024.

3.7.2. Importación de Assets al proyecto

Los Assets son todos los componentes previamente desarrollados que pueden ser importados al proyecto. Estos pueden ser añadidos de diversas formas, pero Unity al contar con una amplia comunidad es posible encontrarlos en su propia tienda de elementos Unity Asset Store la cual es proporcionada por (Unity 2024b).

Los Assets necesarios para llevar a cabo este proyecto se los visualiza en la **Ilustración 3-22**.

Ilustración 3-22: Assets integrados al proyecto de Unity.

Realizado por: Morales, 2023

En la **Tabla 3-12** se presenta la función que cumple cada uno de estos assets dentro del proyecto de Unity, junto con el enlace para la descarga de estos.

Tabla 3-12: Función de los assets integrados

Asset	Función	Enlace de descarga
EasyRoads3D Free v3	Creación de carreteras, rectas, curvas, colinas	https://lc.cx/6lrx80
Low-Poly Bicycle # 5	Bicicleta virtual	https://lc.cx/6G0Izu
Bézier Path Creathor	Generación de trayectoria para la ruta	https://lc.cx/G4gmXi

Realizado por: Morales, 2024

3.7.3. Configuración para la comunicación entre Unity y la tarjeta de desarrollo

Para que Unity acceda a los puertos de entrada y salida de la comunicación serial con la ESP32, se utiliza la librería *System.IO.Ports* que permite la creación de un elemento que interactúe

directamente con los puertos seriales (Galarza 2023). En la **Tabla 3-13** se presentan las características usadas en la configuración del puerto serial.

Tabla 3-13: Configuración para la comunicación serial

Característica	Configuración
Puerto	COM3
Velocidad de transmisión	115200 baudios
Tiempo de lectura	1 ms

Realizado por: Morales, 2024

3.7.4. Medición de velocidad con datos adquiridos

La ESP32 monitorea el sensor de efecto Hall para detectar cambios de estado y calcular el tiempo transcurrido desde el último cambio. Si el sensor permanece en estado alto durante más de tres segundos, se considera que la llanta ha dejado de moverse y la velocidad se reestablece en cero. Las interrupciones son esenciales para calcular la velocidad, convirtiendo los tiempos de milisegundos a minutos y dividiendo para obtener el conteo por segundos. La **ecuación 1** utilizada es representativa para comprender el proceso paso a paso, donde el valor de 2π representa la circunferencia completa de la llanta (Márquez, Castro y Urquiza 2022).

$$V = \frac{2\pi \cdot Radio\ de\ la\ llanta}{Tiempo\ de\ Interrupción \cdot 60} \cdot 1000 \qquad (1)$$

Para evitar retransmitir este valor de velocidad cero repetidamente, se utiliza una variable de control tipo booleano para verificar si el mensaje ha sido enviado. De esta manera se asegura una única transmisión hasta detectar un nuevo movimiento.

3.7.5. Simulación de pendiente

Para este proyecto se ha usado el valor de la rotación sobre el eje X como referencia de la pendiente, debido a que, la variación sobre este eje está directamente relacionado con la pendiente a la que está expuesto el vehículo en un determinado momento. A continuación, se muestran ejemplos de la variación del eje X y el efecto sobre el vehículo.

En la **Ilustración 3-23** se logra apreciar un valor de 0° en la rotación del eje X, lo cual se puede comprender como estar en una recta.

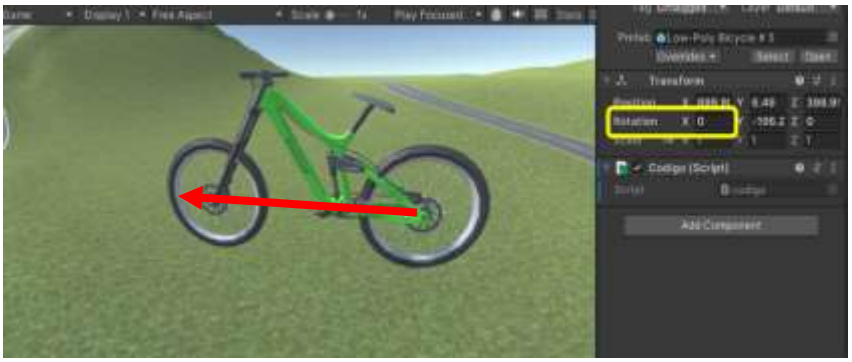


Ilustración 3-23: Rotación de 0 grados sobre el eje X del vehículo.

Realizado por: Morales, 2023

En la **Ilustración 3-24** se observa un valor de -45° en la rotación sobre el eje X del vehículo, mediante este valor de rotación se puede simular una pendiente creciente o el subir una cuesta.

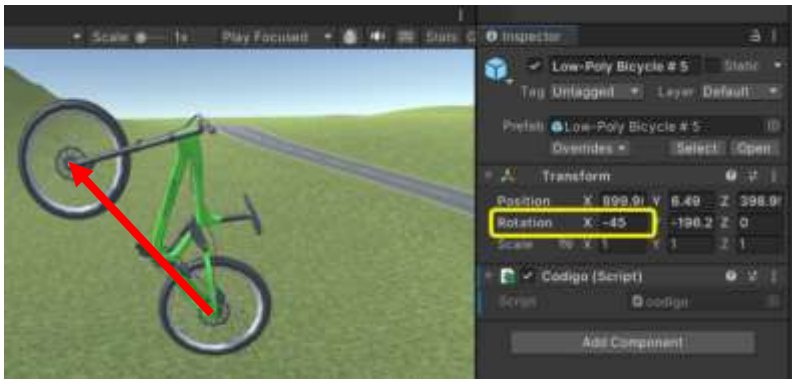


Ilustración 3-24: Rotación de -45 grados sobre el eje X del vehículo.

Realizado por: Morales, 2023

En la **Ilustración 3-25** se observa un valor de 45° en la rotación sobre el eje X del vehículo, mediante este valor de rotación se puede simular una pendiente decreciente o el bajar una cuesta.

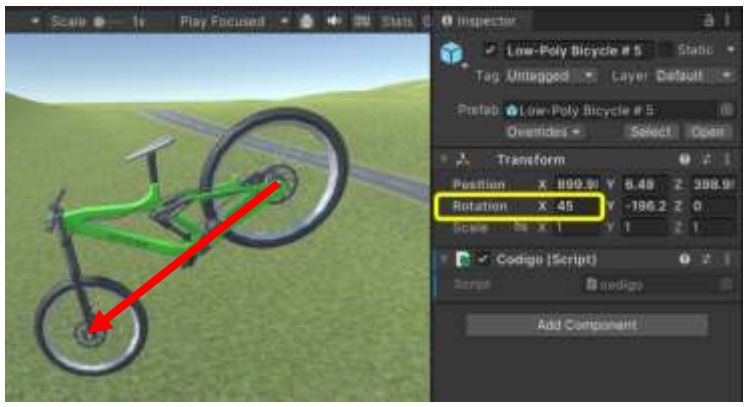


Ilustración 3-25: Rotación de 45 grados sobre el eje X del vehículo.

Realizado por: Morales, 2023

Los intervalo máximos y mínimos de $[-45°, 45°]$ han sido establecidos debido a que en el mundo del ciclismo la inclinación de las pendientes se determina en base a porcentajes, siendo estos de 0% a 100% (Jiménez et al. 2009).
La relación que existe entre los valores de pendiente en escala de ángulos y porcentaje se evidencia en la **ecuación 2.**

$$m(\%) = \tan[m(°)] \cdot 100 \qquad (2)$$

Donde *m(%)* es el valor de la pendiente en porcentaje, el cual es resultado de aplicar la tangente del valor de la pendiente $m(°)$ expresado en grados y multiplicarlo por 100. Se considera que el valor 45 es el máximo, debido que al tener una pendiente de 45° se obtendrá un valor del 100%.
Además, se ha considerado la escala de grados por la mayor comprensión que tienen la mayoría de los usuarios con esta, debido que la escala de porcentajes es más usada por los ciclistas profesionales o amateurs (Cottini 2014).

Una vez considerado este valor para la pendiente se procede a enviar la información de la rotación hacia la tarjeta de desarrollo para el control del motor.

3.7.5.1. Configuración de IDE Arduino para control de TB6600

La configuración de los pines se debe realizar en el IDE Arduino, en la cual la conexión física del controlador TB6600 debe ser especificada. En el IDE de Arduino se han establecido 3 variables esenciales de salida para el controlador del motor paso a paso TB6600 mediante los pines de la tarjeta de desarrollo tal como se muestra en el fragmento de código de la **Ilustración 3-26**.

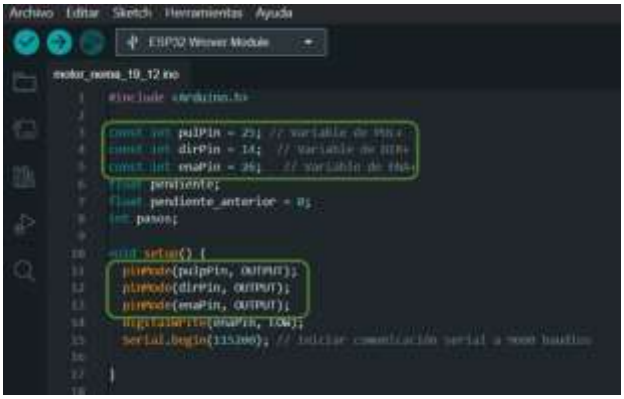


Ilustración 3-26: Fragmento de código de la configuración de pines para la ESP32.

Realizado por: Morales, 2023

Para obtener la relación que existe entre el valor de la pendiente y la respuesta que deberá tener el motor, se ha utilizado la función *map* en el IDE Arduino, la cual es explicada matemáticamente

mediante la **ecuación 3**. Donde se procede a convertir el valor de la diferencia entre la pendiente actual y la pendiente anterior en un número de pasos específicos para el motor

$$map = \frac{(valor - pen\_min) \cdot (mot\_max - mot\_min)}{pen\_max} + mot\_min \qquad (3)$$

En la **Tabla 3-14** se muestra el significado de cada variable denotada en la ecuación anterior.

Tabla 3-14: Configuración para la comunicación serial

Variable	Significado
Valor	Pendiente actual
pen_min	Pendiente mínima establecida
pen_max	Pendiente máxima establecida
mot_min	Paso mínimo establecido
mot_max	Paso máximo establecido

Realizado por: Morales, 2024

3.7.6. Exportación del programa para diferentes plataformas

En este proyecto se ha usado el dispositivo de realidad virtual Oculus Quest 2 debido a su disponibilidad y características. Estas gafas de realidad virtual operan con un sistema interno basado en Android. Para conectar este dispositivo a un ordenador con sistema Windows, es posible utilizar dos métodos: 1) Oculus Link, que implica la conexión mediante cable, o 2) Air Link, que permite la conexión a través de wifi. En ambos casos, es necesario tener instalada la aplicación Oculus en el ordenador (Salas Noain y Delgado del Castillo 2023).

Al momento de realizar ese paso se abrirá la ventana mostrada en la **Ilustración 3-27** donde se puede visualizar el panel que servirá para el cambio de plataforma de Windows, en el cual se encuentra el proyecto. Para poder usar el visor es necesario cambiar a la opción Android marcada en el cuadro verde.

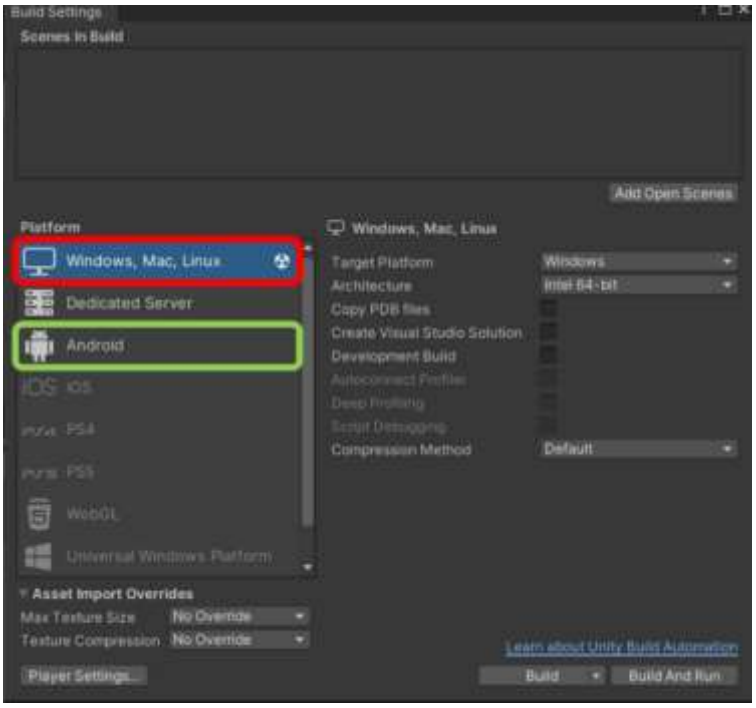


Ilustración 3-27: Panel de Build Settings en Unity.

Realizado por: Morales, 2023

Es importante verificar si se dispone con los requerimientos necesarios para poder construir un juego para Android. Estos requisitos son JDK, SDK, NDK y GRADLE. La descripción de cada uno de estos elementos se muestra en la **Tabla 3-15** junto con la versión usada y el enlace de descarga.

Tabla 3-15: Descripción de JDK, SDK, NDK y Gradle.

Requisito	Descripción	VERSIÓN USADA	Enlace de descarga
JDK	Conjunto de herramientas para desarrollar aplicaciones Java, incluye el compilador para convertir el código fuente en ejecutable.	11	https://lc.cx/9WoAgk
SDK	Necesario para desarrollar el software en una plataforma específica, con bibliotecas, documentación, ejemplos, compiladores y depuradores.	34.0.0	https://lc.cx/N9S4Sz
NDK	Sirve para escribir código C y C++ en aplicaciones Android, permitiendo integrar eventos de Java en el sistema.	26.2.11394342	https://lc.cx/FBHleZ
Gradle	Herramienta de construcción de proyectos para Java y Android, automatizando la compilación, distribución y	7.2	https://lc.cx/ItowVC

	gestión de dependencias de un programa.

Fuente: (Mindiola 2021)

Realizado por: Morales, 2023

3.7.6.1. Herramientas para crear aplicaciones de realidad virtual en Unity

Unity ofrece diversas herramientas para la creación de programas de realidad virtual y aumentada, teniendo acceso a funciones específicas del hardware, lo que facilita la integración de los dispositivos Oculus con este motor gráfico. Estos elementos se detallan en la **Tabla 3-16.**

Tabla 3-16: Herramientas para crear aplicaciones VR en Unity.

Herramienta	Descripción
Open XR	Especificación estándar abierta para aplicaciones de realidad virtual y aumentada que permite funcionar en diferentes plataformas XR sin reescribir el código.
Project Validation	Especificación estándar abierta para aplicaciones de realidad virtual y aumentada que permite funcionar en diferentes plataformas XR sin reescribir el código.
XR Plug-in Management	Permite gestionar los distintos plug-ins y APIs de XR (Realidad Extendida) para diversos dispositivos.
XR Interaction Toolkit	Conjunto de herramientas para simplificar la implementación de interacciones comunes en entornos de realidad virtual, facilitando la creación de experiencias interactivas para dispositivos XR, como Oculus.

Fuente: (Arribére et al. 2022)

Realizado por: Morales, 2023

En el caso de no poder usar la app de Oculus en pc, se deberá utilizar el software SideQuest que permite crear archivos con extensión apk para instalarlos en Oculus. (SideQuest 2024) ofrece una guía detallada sobre las funcionalidades de este software.

3.7.7. Registro y visualización de información

Estos valores son necesarios para que el usuario pueda verificar su rendimiento de la sesión de entretenimiento experimentada.

3.7.7.1. Conexión de la tarjeta de desarrollo con NodeRed

Para poder realizar una conexión exitosa entre la tarjeta de desarrollo y Node-RED se ha usado el protocolo de comunicación MQTT, donde (Triviño 2023) brinda la explicación necesaria para poder conectar la ESP32 con esta plataforma. Hay que considerar que, al trabajar con una red

local, la tarjeta de desarrollo y Node-RED deben mantener las mismas configuraciones, para lograr establecer los protocolos de publicación y suscripción.

Para poder visualizar los datos enviados desde la tarjeta de desarrollo se utiliza el servidor MQTT, configurando el nodo MQTT In de Node-Red mostrado en la **Ilustración 3-28**.

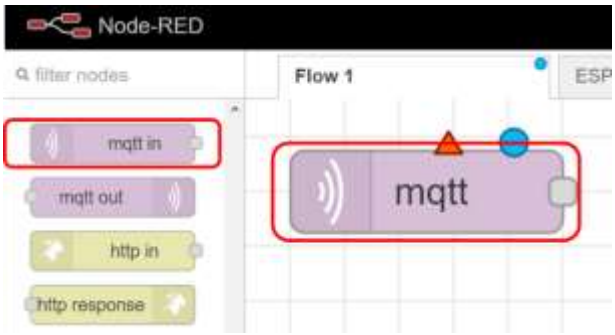


Ilustración 3-28: Nodo MQTT In de Node-RED.

Realizado por: Morales, 2023

Los datos receptados se pueden visualizar en un dashboard mediante el cual se representa la información en diferentes gráficos estadísticos, tal como se muestra en la **Ilustración 3-29**.

Ilustración 3-29: Recepción de datos en Node-RED.

Realizado por: Morales, 2023

3.7.7.2. Conexión de Node-RED con MySQL

Dentro de Node-RED, existe un nodo específico llamado "mysql" mostrado en la **Ilustración 3-30**, que permite el acceso a una base de datos MySQL. Este nodo puede configurarse para realizar distintos tipos de consultas hacia la base de datos, permitiendo la creación y eliminación de tablas. (Cun y Tutiven 2020) detalla el proceso para la configuración del nodo “mysql” en Node-RED,

considerando que, al ser una base de datos local, se deberá ubicar los parámetros del nombre del host, el número del puerto, el nombre de usuario y la contraseña establecida.

Dentro de la base de datos, se creará una tabla por cada usuario, la cual contendrá 3 columnas, siendo estas para los datos de velocidad, pendiente y tiempo registrados durante la sesión de simulación. (Oracle 2024) indica que los límites de columnas en una tabla de MySQL son de 4096, y que el límite de tablas en una base de datos es de 4 mil millones, esto puede variar en base a las características del servidor local.

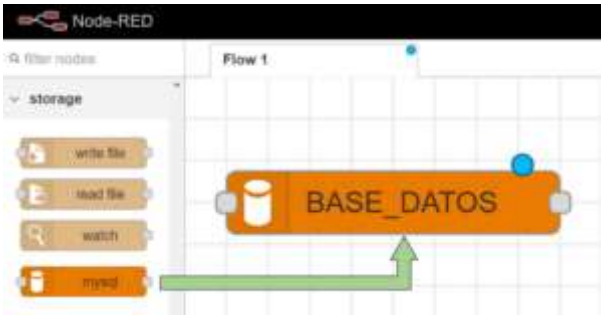


Ilustración 3-30: Nodo mysql en Node-RED.

Realizado por: Morales, 2023

CAPÍTULO IV

4. MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Para evaluar el correcto funcionamiento del proyecto, se llevó a cabo una serie de pruebas que son mostradas en este capítulo, las cuales permitieron demostrar su eficiencia y rendimiento, además del análisis del costo del simulador.

4.1. Prueba MQTT

4.1.1. Análisis de la conexión entre la tarjeta de desarrollo y MQTT

Analizar la latencia que existe entre diversos brókeres MQTT es muy importante debido a que este elemento permite realizar la comunicación inalámbrica entre la tarjeta de desarrollo, los visores Oculus Quest 2 y Node-RED. Se han probado 3 tipos de bróker MQTT en el transcurso de este proyecto como se puede observar en la **Ilustración 4-1** y la **Ilustración 4-2**, de los cuales se ha establecido un análisis en base a la latencia de comunicación que existe entre el microcontrolador y el bróker.

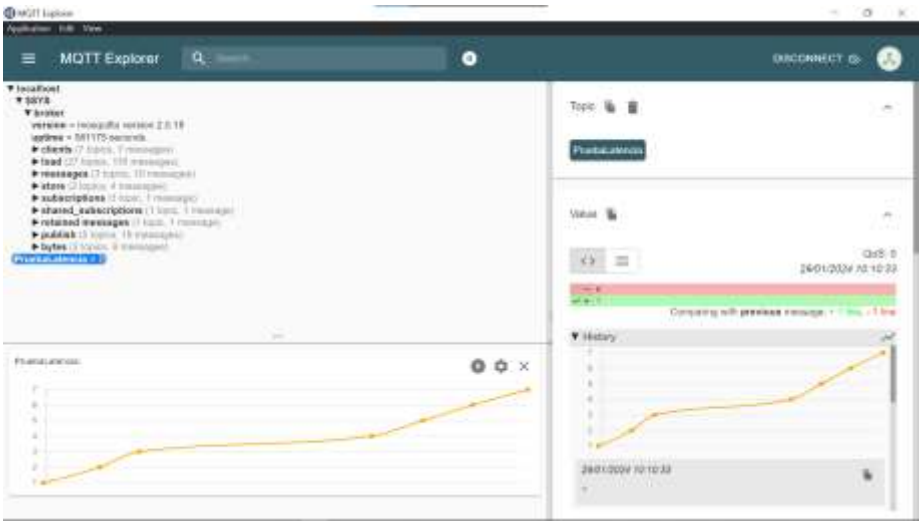


Ilustración 4-1: Visualización de los tópicos en MQTT Explorer.

Realizado por: Morales, 2024.

Para la validación del servidor se han establecido distintos tiempos de retardo para la publicación de mensajes desde la tarjeta de desarrollo. Se han tomado los primeros 30 datos receptados por el tópico al que se ha suscrito el bróker MQTT, esto debido al teorema del límite central, que establece que las distribuciones de las medias de muestras grandes tienden a ser normales.

Esto permite hacer inferencias más sólidas sobre la población en general. Además, con un tamaño de muestra de al menos 30, se mejora la precisión de las estimaciones de la media muestral en comparación con tamaños de muestra más pequeños. Es importante resaltar que estas pruebas han sido evaluadas con una conexión promedio de 4.7 Mbps, lo cual es de los principales elementos encargados de hacer más precisa.

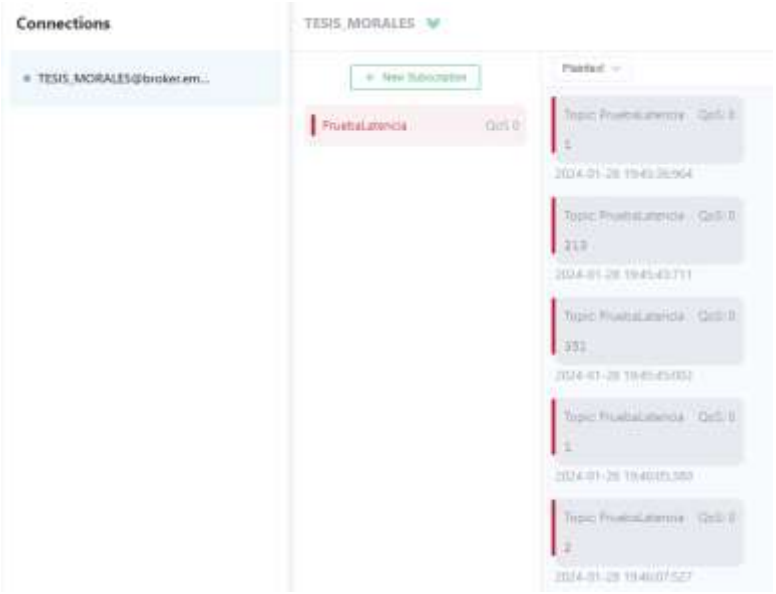


Ilustración 4-2: Visualización de tópicos en EMQX.IO.

Realizado por: Morales, 2024.

En la **Tabla 4-1** se visualizan los primeros 20 datos receptados al aplicar un retardo de 100 milisegundos en su tiempo de latencia. De esta información se puede observar como en los tres tipos de bróker MQTT llegan los datos sin interrupción alguna lo cual representa 0% de perdida de información.

Tabla 4-1: Datos receptados con retraso de 100 ms.

EMQX.IO	TEST MOSQUITTO	LOCAL HOST
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13

14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30

Realizado por: Morales, 2024.

Al disminuir el tiempo de retraso a 50 milisegundos se está procediendo a aumentar la velocidad de envío, lo cual como se puede observar en la **Tabla 4-2. Esto** genera pérdidas de datos en el bróker público de *EMQX.IO* debido a que este bróker es proporcionado por el navegador, la cual es más propenso a ser afectado por los valores del delay.

Tabla 4-2: Datos receptados con retraso de 50 ms.

EMQX.IO	TEST MOSQUITTO	LOCAL HOST
1	1	1
3	2	2
4	3	3
6	4	4
8	5	5
9	6	6
10	7	7
13	8	8
16	9	9
18	10	10
21	11	11
24	12	12
26	13	13
29	14	14
31	15	15
32	16	16
35	17	17
37	18	18
40	19	19
42	20	20
45	21	21
47	22	22
48	23	23
51	24	24
53	25	25
55	26	26
57	27	27
61	28	28
64	29	29
66	30	30

Realizado por: Morales, 2024

En la etapa final de esta prueba se procedió a disminuir el valor del retraso a 10 ms, obteniendo los datos mostrados en la **Tabla 4-3**. Esto causa una notable perdida de información ente los primeros 20 datos obtenidos por los brókeres *EMQX.IO* y *Test Mosquitto*. En el primero los datos llegaron hasta 107 quedando totalmente descartado este bróker, el segundo también cuenta con una cantidad de perdida de información, pero la principal razón por la que se descarta este bróker es debido a que al ser un servidor público varios usuarios del mundo se encuentran activamente enviando diversos mensajes a diferentes tópicos, lo cual puede ocasionar que existan conflictos en la publicación y suscripción de los tópicos usados en este proyecto.

Tabla 4-3: Datos receptados con retraso de 10 ms.

EMQX.IO	TEST MOSQUITTO	LOCAL HOST
1	1	1
6	2	2
10	4	3
15	5	4
21	7	5
27	9	6
31	10	7
36	11	8
42	12	9
49	14	10
53	16	11
58	18	12
64	19	13
71	21	14
76	22	15
79	24	16
85	26	17
92	27	18
99	29	19
107	30	20
113	31	21
119	33	22
125	34	23
131	36	24
135	37	25
139	38	26
144	40	27
148	42	28
153	43	29
157	44	30

Realizado por: Morales, 2024.

Se realizó un análisis estadístico para conocer el comportamiento de los datos receptados a través de los tres diferentes brókers MQTT propuestos anteriormente, considerando la desviación estándar de los intervalos de retraso de 100, 50 y 10 ms tal como se puede ver en la **Tabla 4-4**. Los resultados obtenidos a los 100 ms son los mismos en todos los servidores, en el delay de 50 ms comienza a existir variación en el bróker EMQX.IO, lo cual hace que los otros dos sigan siendo más estables. Al llegar al retraso de los 10 ms, únicamente el Local Host sigue

manteniendo la integridad de los datos. Mediante esto se puede determinar que el bróker Local Host es la mejor elección para no tener inconvenientes a la hora de la simulación.

Tabla 4-4: Desviación estándar entre los brókers MQTT.

Broker	sd 100ms	sd 50 ms	sd 10ms
EMQX.IO	0	0.7647191	1.3045131
TEST_MOSQUITTO	0	0.0000000	0.5129892
LOCAL HOST	0	0.0000000	0.0000000

Realizado por: Morales, 2024.

La **Ilustración 4-3** muestra gráficamente la diferencia obtenida entre las latencias definidas en los brókers MQTT, de esta forma se puede observar como el único servidor que no sufre ningún tipo de perturbación es el Local Host.

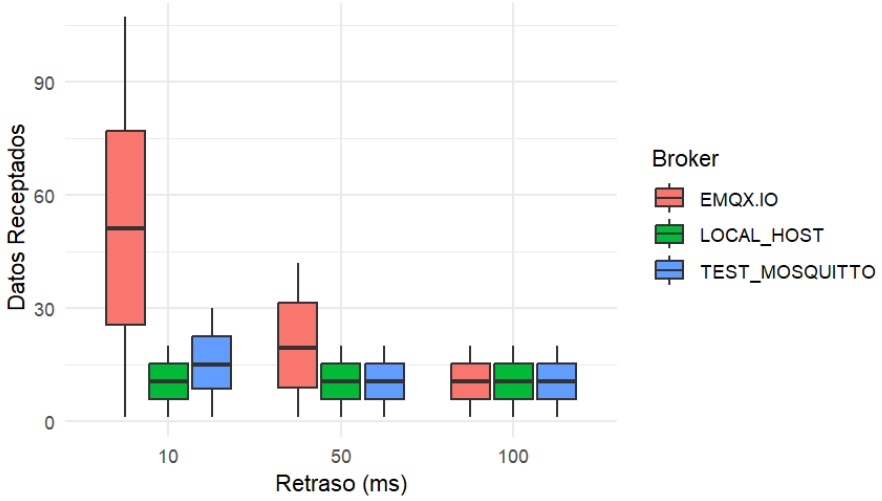


Ilustración 4-3: Retrasos en los brókers MQTT.

Realizado por: Morales, 2024.

El bróker local permite establecer la comunicación MQTT de forma estable y segura en comparación de los otros dos con los que se ha realizado la comparación, logrando publicar y suscribirse a tópicos de manera eficiente en todos los tiempos de retardo.

4.1.2. Prueba de conectividad de Oculus con el bróker MQTT

Se realizó una comprobación de la conexión entre los visores Oculus Quest 2 y el servidor MQTT, para esto se usó la aplicación MyMQTT para realizar los métodos de suscripción y publicación con los tópicos.

En la **Ilustración 4-4** se visualiza una captura del entorno de realidad virtual, en la cual el dispositivo está conectado a una dirección IP específica. Se usa el tópico *"PruebaLatencia"* para

la suscripción de datos enviados desde la computadora, donde se logra garantizar la comunicación entre las diferentes plataformas.

Ilustración 4-4: Suscripción de Oculus Quest 2 al tópico MQTT.

Realizado por: Morales, 2024.

En la **Ilustración 4-5** se muestra la asignación del tópico "*EnviOculus*" para poder enviar datos y comprobar la posibilidad de la publicación de mensajes a tópicos específicos desde Oculus Quest 2.

Ilustración 4-5: Publicación desde Oculus Quest 2 al tópico MQTT.

Realizado por: Morales, 2024.

Para verificar que los datos realmente son enviados por el visor de realidad virtual, se ha usado la aplicación *MQTT Explorer* en el ordenador tal como se muestra en la **Ilustración 4-6**, donde se encuentra suscrito al tópico específico

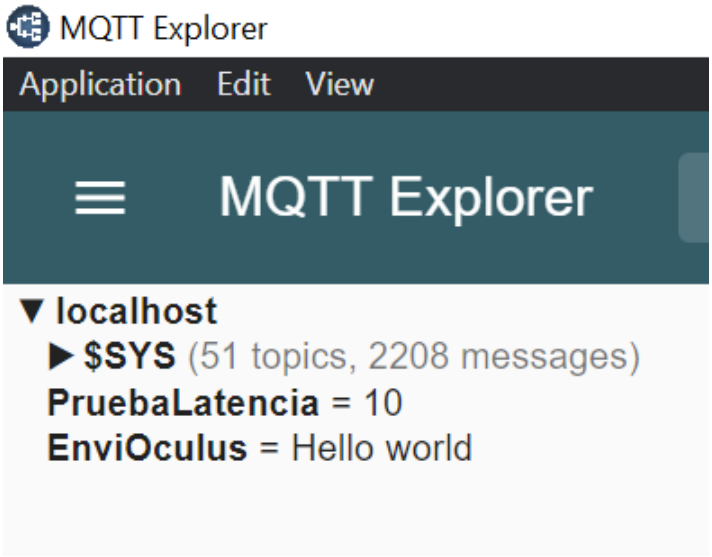


Ilustración 4-6: Suscripción de la computadora al tópico MQTT.

Realizado por: Morales, 2024.

4.2. Prueba de comunicación bidireccional entre ESP 32 y Unity

Esta prueba fue llevada a cabo para verificar los tiempos de envío y recepción de mensajes que existe entre la tarjeta de desarrollo. Para esto se debe considerar que la comunicación establecida se realiza mediante MQTT, debido que al iniciar la ESP32 procede a conectarse a la red WiFi definida previamente en el código. Cada red a la que se conecta cuenta con una dirección IP definida, por lo cual también es necesario establecer la dirección según la red disponible.

4.2.1. Comunicación entre la tarjeta de desarrollo y Unity con MQTT

La primera etapa de esta prueba consta de medir el tiempo de conexión entre la tarjeta de desarrollo con el bróker MQTT, junto con el tiempo de vinculación de Unity con dicho bróker. Para esto, se incluyen códigos tanto para ESP32 y para Unity, en los cuales se toma el tiempo que se demora en establecer la comunicación de los entornos con el bróker.

En la **Tabla 4-5** se presenta el tiempo de conexión que existe entre la tarjeta de desarrollo y Unity para poder vincularse con el bróker MQTT. Se han realizados 10 intentos en los cuales se ha cronometrado dicho tiempo. Al analizar esta tabla se puede observar que existe un tiempo mayor

de enlazamiento del bróker con la ESP32, esto debido a que la tarjeta de desarrollo primero debe conectarse a la red WiFi, lo cual es prioridad para poder conectarse al servidor MQTT, en cambio, Unity al ser una aplicación que se ejecuta en computadoras, celulares o en los visores de Oculus, son dispositivos que previamente se han conectado a internet, por lo cual Unity no utiliza un tiempo de conexión, lo cual hace que su latencia sea inferior.

Tabla 4-5: Tiempo de conexión de ESP32 y Unity con MQTT.

Número de prueba realizada	Tiempo de conexión (ms)	
	ESP32	Unity
1	13.12	7.13
2	12.81	6.78
3	12.97	6.85
4	13.07	6.94
5	12.85	7.05
6	12.92	6.91
7	13.02	7.11
8	13.07	6.92
9	12.96	7.07
10	13.08	7.03
11	12.96	7.06
12	12.91	6.87
13	13.05	6.91
14	12.87	7.11
15	13.06	6.95
16	13.02	7.07
17	13.09	7.03
18	12.88	7.12
19	12.91	6.93
20	13.01	6.98
21	12.80	7.05
22	12.97	6.88
23	13.08	7.06
24	13.03	7.12
25	12.91	6.96
26	12.96	6.94
27	12.94	6.97
28	13.05	7.02
29	12.88	7.09
30	12.93	6.91

Realizado por: Morales, 2024.

Los datos obtenidos en la tabla anterior son usados con el objetivo de comparar las medias en los tiempos de conexión, donde se plantearon las siguientes hipótesis estadísticas:

H0: *Los tiempos de conexión media entre ESP32 y Unity son mayor a 6.1 segundos.*

H1: *Los tiempos de conexión media entre ESP32 y Unity son menores o iguales a 6.1 segundos.*

Los resultados obtenidos se presentan en la **Tabla 4-6**. Al analizar el valor t y el valor p se logra identificar un fuerte rechazo de la hipótesis nula, por lo que la hipótesis alternativa se verifica. Esto significa que hay la evidencia estadística significativa para afirmar que existe una diferencia

en los tiempos de conexión siendo el valor obtenido de 6.008 segundos. Además, el intervalo de confianza al 95% brinda la estimación dónde se encuentra esta diferencia de medias en la población general.

Tabla 4-6: Prueba t de Student sobre el tiempo de conexión entre ESP32 y Unity.

Comparación	Media ESP32	Media Unity	Diferencia de medias	Valor t	Valor p	Intervalo de confianza al 95%
ESP32 - UNITY	12.987	6.979	6.008	121.88	< 2.2e-16	[5.904, 6.112]

Realizado por: Morales, 2024.

La **Ilustración 4-7** muestra la gráficamente la diferencia que existe entre los tiempos de conexión de la ESP32 y Unity con el bróker MQTT, pero, aunque visualmente parezca una brecha considerable es importante tener en cuenta que el tiempo está en la escala de milisegundos.

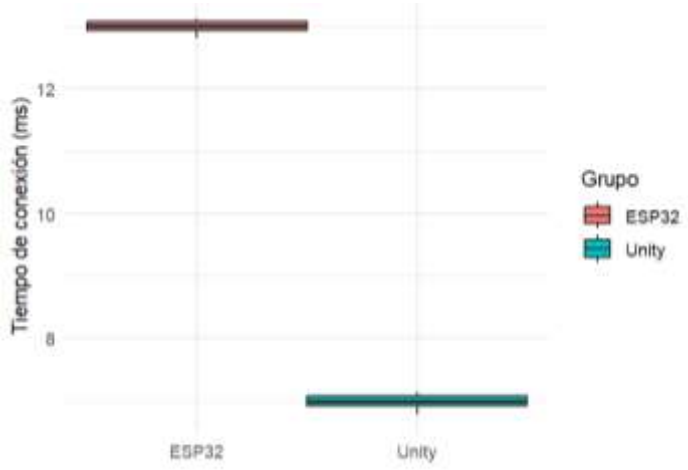


Ilustración 4-7: Tiempo de Conexión de ESP32 y Unity con MQTT.

Realizado por: Morales, 2024.

4.3. Prueba de velocidad

Para determinar la velocidad se realizó una prueba donde se decidió usar un velocímetro para bicicletas SB-318 como equipo patrón, el mismo que se puede observar en la **Ilustración 4-8**. Este permite realizar una comparación entre los datos obtenidos al medir la velocidad que nos muestra la tarjeta de desarrollo ESP32. Para poder llevar a cabo esto se conectaron ambos dispositivos, logrando de esta forma visualizar de forma paralela la velocidad en el monitor serial del IDE Arduino y la pantalla digital del velocímetro.

Ilustración 4-8: Velocímetro SB-318.

Realizado por: Morales, 2024.

En la **Tabla 4-7** se puede observar cómo los valores de velocidad entregados por la tarjeta de desarrollo y el equipo patrón tienden a ser similares en todas las pruebas. De esta forma se puede verificar que las diferencias entre ambos métodos de medición son mínimas, lo cual indica una notable consistencia en la ejecución del proyecto.

Tabla 4-7: Comparación de velocidad calculada entre ESP32 y SB-318.

Número de prueba realizada	Velocidad calculada (m/s)		Diferencia entre ESP32 y SB 31-8
	ESP32	Velocímetro SB-318	
1	0.00	0.00	0.00
2	2.34	2.40	-0.06
3	5.78	5.85	-0.07
4	9.67	9.60	0.07
5	13.45	13.50	-0.05
6	16.98	17.05	-0.07
7	20.01	20.03	-0.02
8	23.56	23.50	0.06
9	26.80	26.75	0.05
10	29.34	29.40	-0.06
11	30.00	30.05	-0.05
12	28.45	28.50	-0.05
13	25.90	25.85	0.05
14	22.33	22.30	0.03
15	18.77	18.80	-0.03
16	15.22	15.20	0.02
17	11.68	11.65	0.03
18	8.13	8.10	0.03
19	5.59	5.55	0.04
20	2.04	2.00	0.04
21	0.49	0.45	0.04
22	1.57	1.60	-0.03
23	4.12	4.10	0.02
24	6.66	6.70	-0.04
25	9.21	9.25	-0.04
26	11.75	11.80	-0.05
27	14.30	14.25	0.05
28	16.84	16.80	0.04

29	19.39	19.35	0.04
30	21.93	21.90	0.03

Realizado por: Morales, 2024.

Se realizó la prueba estadística de correlación por el método de Pearson obteniendo un resultado 0.999988 lo que indica una correlación positiva muy fuerte entre las velocidades calculadas por la ESP32 y las medidas del velocímetro SB-318. En la **Ilustración 4-9** evidencia como los métodos están muy alineados, debido a que una correlación cercana a 1 significa que existe una relación lineal casi perfecta entre las dos variables.

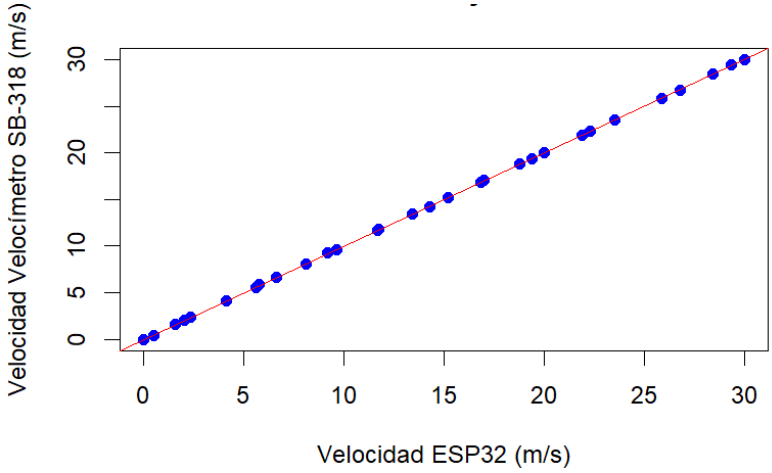


Ilustración 4-9: Prueba de correlación entre los métodos de medición de velocidad.

Realizado por: Morales, 2024.

4.4. Pruebas con el motor

4.4.1. Margen de error del desplazamiento con el motor

El propósito de esta prueba es calcular el margen de error entre la distancia que el motor paso a paso Nema 17 respecto del número específico de pasos. Las pruebas fueron llevadas a cabo en el área de trabajo del prototipo, donde se midió la distancia que se movió el perno conectado al sistema. Se hizo uso de un calibrador debido a que la variación de distancia es pequeña y esto garantiza mayor precisión. Se realizó la prueba con diferentes cantidades de pasos, donde se debe considerar que este motor cuenta con 200 pasos completos individuales por revolución.

En la **Tabla 4-8** se muestra la distancia medida entre los pasos ingresados, donde se ha logrado comprobar que existe una baja variación de distancia entre pasos, lo cual garantiza un menor nivel de error o perturbación.

Tabla 4-8: Valores obtenidos con diversos pasos.

Pasos	Distancia esperada	Distancia registrada	Diferencia obtenida	Cantidad de pasos perdidos o ganados
25	1	1	0	0
25	1	1.1	0.1	2.5
25	1	1	0	0
25	1	1	0	0
25	1	1	0	0
50	2	2	0	0
50	2	2	0	0
50	2	2.1	0.1	2.5
50	2	2	0	0
50	2	2	0	0
75	3	3	0	0
75	3	3	0	0
75	3	3	0	0
75	3	2.98	-0.1	-0.5
75	3	3	0	0
100	4	4	0	0
100	4	4	0	0
100	4	4.1	0.1	2.5
100	4	4	0	0
100	4	4	0	0
125	5	5	0	0
125	5	5	0	0
125	5	5	0	0
125	5	5.1	0.1	2.5
125	5	5	0	0
150	6	6	0	0
150	6	6.1	0.1	2.5
150	6	6	0	0
150	6	6	0	0
150	6	6	0	0

Realizado por: Morales, 2024.

La **Ilustración 4-10** representa la frecuencia de errores obtenidos en las muestras de los pasos del motor, donde en la mayoría de los casos no existe ninguna variación, lo cual garantiza que al momento de la simulación no existirá una perturbación significativa.

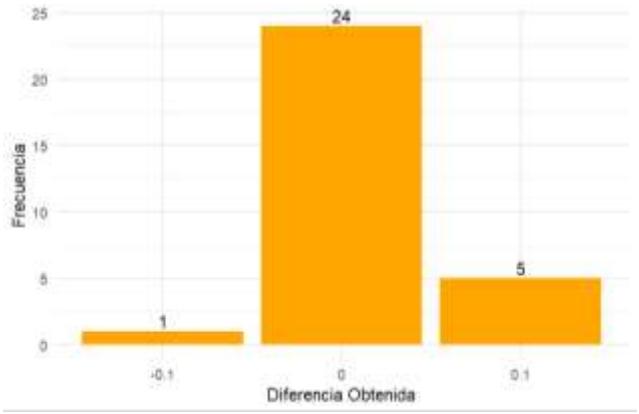


Ilustración 4-10: Frecuencia de errores en las muestras.

Realizado por: Morales, 2024.

4.4.2. Tiempos de accionamiento del motor mediante el ángulo de inclinación

Esta prueba permite analizar los tiempos de respuesta entre el motor y el software de realidad virtual. La rápida respuesta de accionamiento del motor es indispensable para que el operador no pierda la sensación de inmersión.

Para esto, se consideró el tiempo de inicio del movimiento del motor hasta la finalización de este mismo, lo cual da como resultado el tiempo que tardó en ejecutar el movimiento en función de la pendiente. Se han realizado varias pruebas con sesiones en las que se cuenta con variaciones del ángulo de la pendiente receptada.

En la **Ilustración 4-11** se puede observar cómo se ha realizado la prueba para el tiempo de ejecución entre pasos, fue necesario considerar la función *micros ()* debido a que el tiempo entre pasos se lo está realizando en microsegundos.

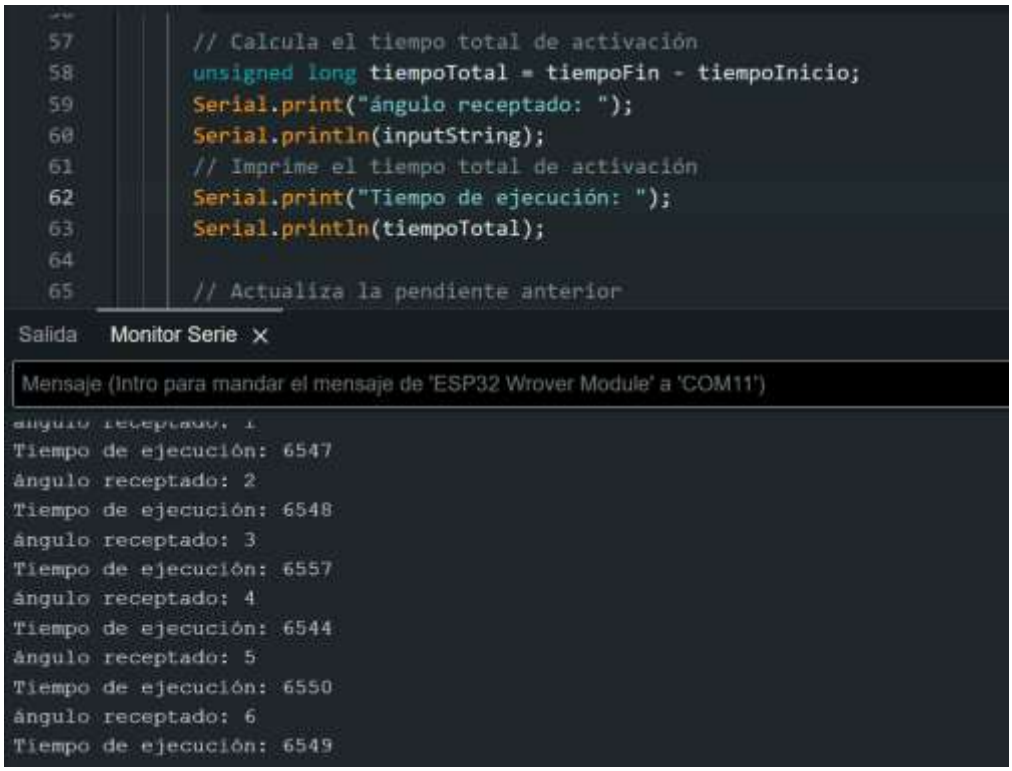


Ilustración 4-11: Tiempo de ejecución entre pasos.

Realizado por: Morales, 2024.

En la **Tabla 4-9** se puede visualizar una rutina en la cual el ángulo de la pendiente va incrementando una unidad secuencialmente, se debe considerar que el punto de partida para todas las rutinas siempre será desde cero.

Tabla 4-9: Primera rutina.

Ángulo de pendiente receptado	Tiempo de accionamiento en us	Diferencia entre pasos
1	6547	1
2	6548	1
3	6557	1
4	6544	1
5	6550	1
6	6549	1
7	6547	1
8	6548	1
9	6544	1
10	6555	1
9	6549	1
8	6549	1
7	6546	1
6	6549	1
5	6551	1
4	6547	1
3	6549	1
2	6554	1
1	6549	1
0	6547	1
-1	6551	1
-2	6553	1
-3	6550	1
-4	6544	1
-5	6544	1
-6	6553	1
-7	6551	1
-8	6543	1
-9	6552	1
-10	6558	1

Realizado por: Morales, 2024.

Para esta rutina se ha duplicado el valor de los ángulos de la pendiente tal como se evidencia en la **Tabla 4-10**. Mediante esto se puede visualizar que se mantienen valores similares, lo cual garantiza una eficiencia en el tiempo de ejecución del motor. Además, se pudo comprobar que cada vez que se recepte un ángulo enviado desde Unity hacia la ESP32 la respuesta será inmediata debido a que su reacción se da en intervalos de microsegundos.

Tabla 4-10: Segunda rutina.

Ángulo de pendiente receptado	Tiempo de accionamiento en us	Diferencia entre pasos
2	13345	2
4	13346	2
6	13343	2
8	13338	2

10	13347	2
12	13352	2
14	13345	2
16	13346	2
18	13350	2
20	13349	2
18	13343	2
16	13354	2
14	13345	2
12	13347	2
10	13344	2
8	13347	2
6	13353	2
4	13344	2
2	13341	2
0	13350	2
-2	13346	2
-4	13349	2
-6	13340	2
-8	13355	2
-10	13348	2
-12	13352	2
-14	13344	2
-16	13348	2
-18	13344	2
-20	13342	2

Realizado por: Morales, 2024.

Para realizar esta prueba estadística se considera que las muestras otorgadas son independientes, esto debido a que se evalúa los tiempos de accionamiento del motor entre dos rutinas diferentes, para lo cual se establecieron las siguientes hipótesis:

> ***H0:*** *Los tiempos de accionamiento del motor entre las dos rutinas tienen una diferencia significativa mayor a 6500 microsegundos.*
> ***H1:*** *Los tiempos de accionamiento del motor entre las dos rutinas tienen una diferencia significativa menor o igual a 6500 microsegundos.*

En la **Tabla 4-11** se resume los resultados de la prueba t de Welch entre las dos rutinas planteadas, en base a que el dato obtenido en el valor de probabilidad es extremadamente bajo, se rechaza la hipótesis nula y la alternativa queda verificada. Por tanto, se verifica que no existen diferencias significativas en los tiempos de accionamiento del motor entre las dos rutinas. Este tiempo de accionamiento no es mayor a 6000 microsegundos. Esto aporta la información suficiente para verificar que cada sesión es única y el motor debe accionar correctamente sin importar los intervalos de pendiente registrados.

Tabla 4-11: Prueba t de Welch sobre el tiempo de accionamiento del motor.

Media de Rutina 1	Media de Rutina 2	df	Valor t	Valor p	Intervalo de confianza al 95%
13346.567	6549.267	57.63	6620	< 2.2e-16	[6799.356, 6795.244]

Realizado por: Morales, 2024.

A través de la **Ilustración 4-12** se puede comprender que, independientemente de la rutina impartida por el usuario, el motor genera una respuesta estable, brindando así una experiencia más inmersiva.

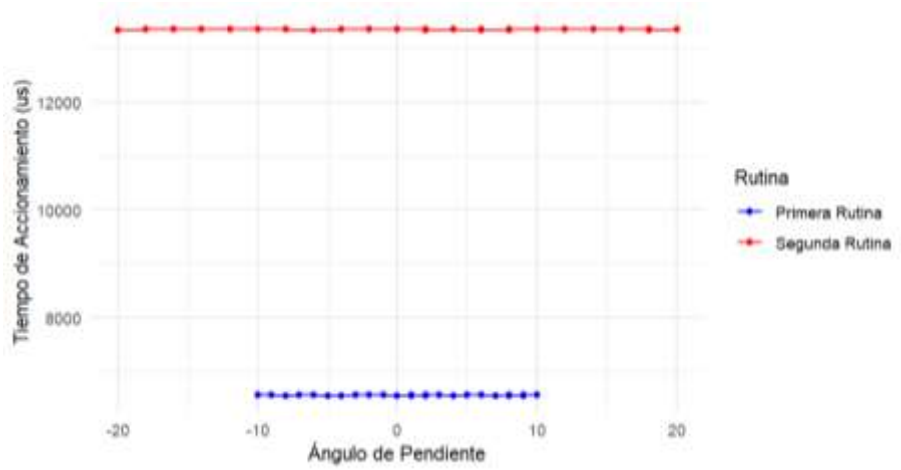


Ilustración 4-12: Comparación de tiempos de accionamiento entre rutinas.

Realizado por: Morales, 2024.

4.5. Prueba de conectividad entre ESP32 con Node-RED y la base de datos

Esta prueba permite garantizar que los datos son enviados y recibidos correctamente desde la tarjeta de desarrollo hacia la base de datos local. Este proceso tiene un paso intermedio por Node-RED, y de esta forma se procede a realizar las pruebas de comunicación entre estas plataformas.

4.5.1. Prueba entre ESP32 y Node-RED

Para llevar a cabo esta prueba se realizó el envío de valores de las variables pendiente, tiempo y velocidad. En la **Ilustración 4-13** se puede visualizar como en Node-RED se añadió el nodo debug para poder visualizar los datos que llegaban desde la tarjeta de desarrollo.

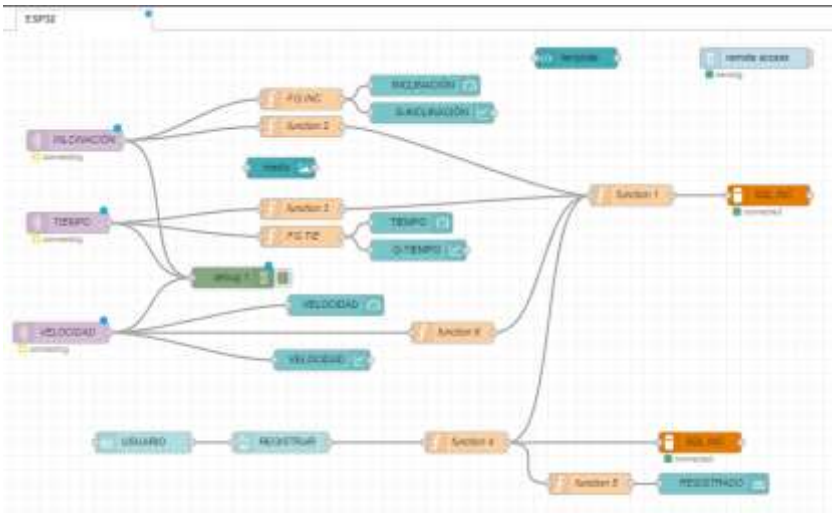


Ilustración 4-13: Adición del nodo debug en Node-RED.

Realizado por: Morales, 2024.

Al haber añadido este nodo se logró tener el tiempo de referencia tal como se muestra en la **Ilustración 4-14.**

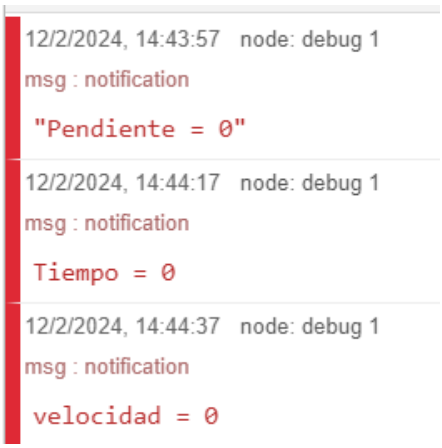


Ilustración 4-14: Recepción de datos enviados desde ESP32.

Realizado por: Morales, 2024.

Todos los datos que llegan a Node-RED se los puede visualizar en la **Tabla 4-12**. Se han tomado los primeros 30 datos recibidos, que deberán ser almacenados en la base de datos.

Tabla 4-12: Datos llegados a Node-RED.

Pendiente (grados)	Tiempo (s)	Velocidad (m/s)
0	0	0
0	1	4.13
0	2	8.27
1	3	12.37
2	4	15.22

3	5	16.41
4	6	17.54
5	7	16.37
6	8	16.45
7	9	16.29
6	10	15.38
5	11	16.42
4	12	16.25
3	13	16.13
2	14	16.48
1	15	15.26
0	16	12.18
-1	17	10.43
-2	18	10.13
-3	19	8.24
-4	20	7.25
-5	21	5.45
-6	22	4.27
-7	23	6.13
-8	24	7.31
-9	25	8.16
-10	26	10.24
-9	27	12.05
-8	28	12.51
-7	29	14.35
-6	30	15.17

Realizado por: Morales, 2024.

4.5.2. Prueba entre Node-RED con la base de datos

Para determinar el proceso final de la comunicación, la información debe ser almacenada en la base de datos. En la **Ilustración 4-15** se puede visualizar la recepción correcta de los datos.

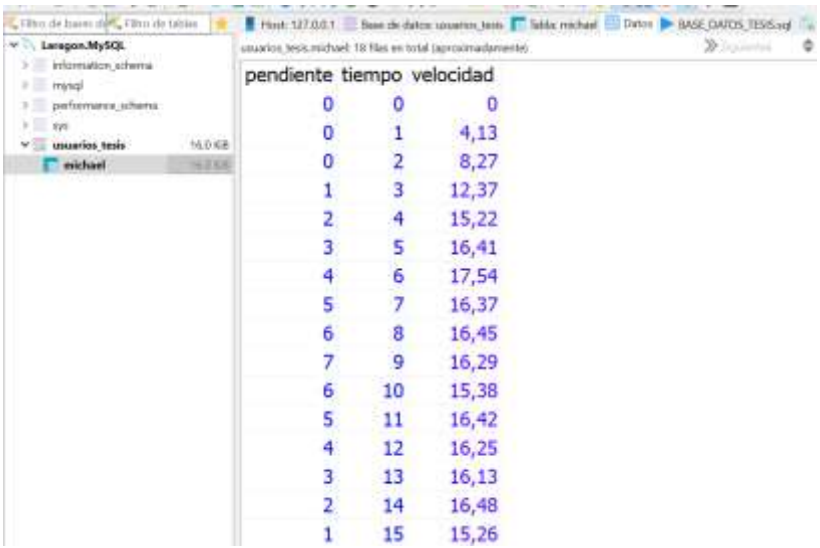


Ilustración 4-15: Recepción de datos en SQL.

Realizado por: Morales, 2024.

En la **Tabla 4-13** se pueden verificar los datos que son enviados a la base de datos, cumpliendo así todo el proceso desde la publicación de la tarjeta de desarrollo al bróker MQTT, luego la suscripción de Node-RED y finalmente el almacenamiento en la SQL.

Tabla 4-13: Datos llegados a la base de datos.

Pendiente (grados)	Tiempo (s)	Velocidad (m/s)
0	0	0
0	1	4.13
0	2	8.27
1	3	12.37
2	4	15.22
3	5	16.41
4	6	17.54
5	7	16.37
6	8	16.45
7	9	16.29
6	10	15.38
5	11	16.42
4	12	16.25
3	13	16.13
2	14	16.48
1	15	15.26
0	16	12.18
-1	17	10.43
-2	18	10.13
-3	19	8.24
-4	20	7.25
-5	21	5.45
-6	22	4.27
-7	23	6.13
-8	24	7.31
-9	25	8.16
-10	26	10.24
-9	27	12.05
-8	28	12.51
-7	29	14.35
-6	30	15.17

Realizado por: Morales, 2024.

En la **Ilustración 4-16** se muestra los gráficos obtenidos al realizar las pruebas de correlación de Pearson para las variables enviadas y recibidas. Esto con el objetivo de evidenciar si existe pérdidas en los datos. El valor de correlación obtenido es igual a 1, lo que significa que hay un envío y recepción correcto de las variables.

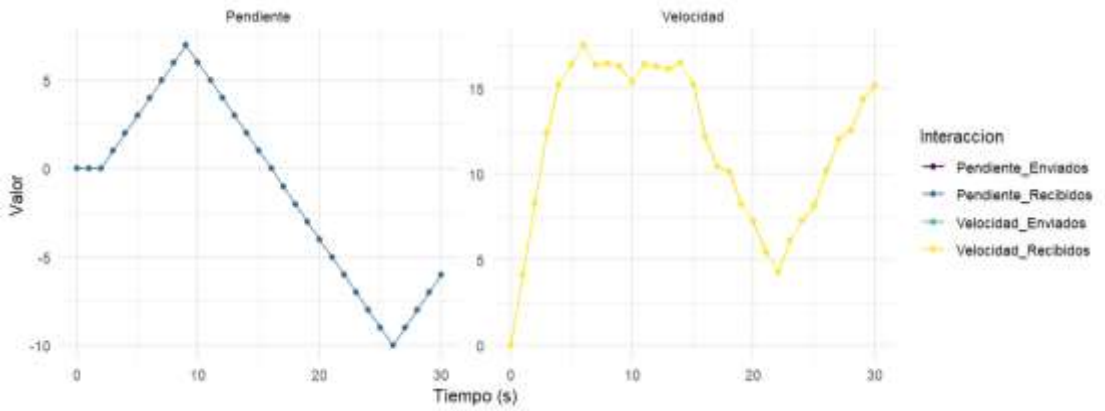


Ilustración 4-16: Comparación de las variables enviadas y recibidas en función del tiempo.

Realizado por: Morales, 2024.

4.6. Análisis de encuestas realizadas a usuarios

El objetivo de esta prueba es comprobar la respuesta que tienen los usuarios que probaron este prototipo. Estas encuestas fueron realizadas a 30 participantes mediante la plataforma Google Forms, permitiendo así un fácil acceso a las personas. Además, esta página nos otorga los diagramas estadísticos de cada pregunta, para las preguntas se planteó la escala de Likert donde se cuenta con 3 intervalos, negativo, neutral y positivo (Canto de Gante et al. 2020).

En la **Ilustración 4-17** se visualiza la primera pregunta realizada a los encuestados, en la cual se puede verificar como el 70% de las personas se encuentran muy satisfechas con la experiencia brindada en el transcurso de la simulación, mientras que el 26.7% mencionó que se sintieron satisfechos, los cual en la escala de Likert nos muestra un margen positivo. La pregunta realizada fue**: "**¿Qué tan satisfecho se encuentra con la experiencia de simulación de ciclismo?**".**

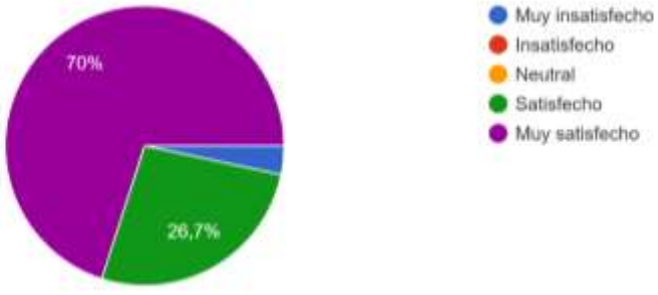


Ilustración 4-17: Porcentajes obtenidos por los usuarios con un 96.7% de satisfacción

Realizado por: Morales, 2024.

La **Ilustración 4-18** demuestra el grado de inmersión que tuvieron los usuarios durante su tiempo en el entorno de realidad virtual, donde se mantuvo un rango del 90% de margen positivo, mismo que logra evidenciar su agrado con la sensación brindada. La pregunta realizada fue: "¿Qué tal le pareció el grado de inmersión del entorno de realidad virtual?".

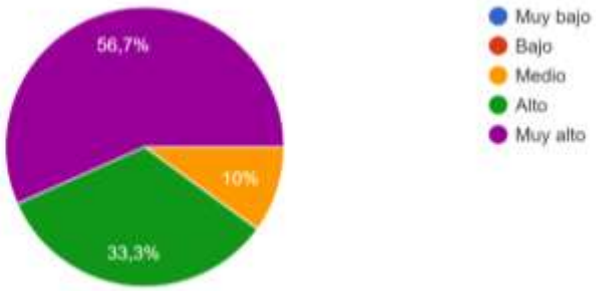


Ilustración 4-18: Porcentajes obtenidos por los usuarios con un 90% de satisfacción

Realizado por: Morales, 2024.

En la **Ilustración 4-19** se visualiza la calificación que los usuarios le dieron al entorno de simulación de realidad, donde el 93% de ellos lo evaluaron en un rango positivo, dando de esta forma a entender que el ambiente creado en Unity les pareció adecuado para la simulación. La pregunta realizada fue: "¿Cómo califica el entorno de la simulación de realidad virtual?".

Ilustración 4-19: Porcentajes obtenidos por los usuarios con un 93.3% de satisfacción

Realizado por: Morales, 2024.

La pregunta realizada en la **Ilustración 4-20** ayudó a poder determinar qué tan entretenidos se encontraron los usuarios, donde se obtuvo que el 100% de los usuarios se divirtieron según la escala de Likert, lo cual es satisfactorio para el proyecto. La pregunta realizada fue: "¿Cómo calificaría la experiencia de entretenimiento mientras usaba el simulador?".

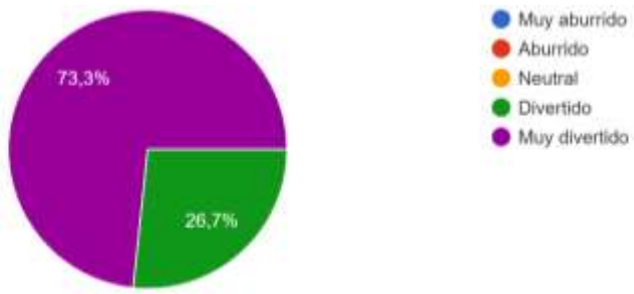


Ilustración 4-20: Porcentajes obtenidos por los usuarios con un 100% de satisfacción

Realizado por: Morales, 2024.

La quinta pregunta mostrada en la **Ilustración 4-21** demuestra cómo el 96.7% de las personas que probaron el simulador de ciclismo se sintieron interesados y mantuvieron su atención durante la sesión que realizaron. Esto garantiza de igual forma el entretenimiento con el prototipo. La pregunta realizada fue: "¿El simulador fue capaz de mantener tu interés y atención durante la sesión de uso?".

Ilustración 4-21: Porcentajes obtenidos por los usuarios con un 96.7% de satisfacción

Realizado por: Morales, 2024.

La sexta pregunta es necesaria para poder conocer el grado de dificultad que los usuarios consideraron que el simulador. Al tener en cuenta que los simuladores de realidad no son muy comunes en la actualidad la gente no se encuentra acostumbrada a este tipo de sistemas, pero mediante los resultados obtenidos en la **Ilustración 4-22** se observa como los usuarios lo han clasificado con un 96.6% de margen positivo, esto determina que el sistema es intuitivo para usar. La pregunta realizada fue: "¿Cómo calificaría el grado de dificultad de uso del sistema de simulación?".

Ilustración 4-22: Porcentajes obtenidos por los usuarios con un 96.6% de satisfacción

Realizado por: Morales, 2024.

Al efectuar esta séptima pregunta se buscó conocer la calificación que los participantes le dan a la estabilidad de la conexión y del sistema al momento de su uso. Para esto fue importante verificar que la conexión de internet donde se encontraba el simulador fuese estable. Esto para no afectar a la experiencia del usuario con latencias en la ejecución. En la **Ilustración 4-23** se evidencia que el 93.3% de los usuarios han dado opiniones positivas referentes a esta pregunta. La pregunta realizada fue: "¿Cómo describirías la conexión y la estabilidad del sistema entre la tarjeta de desarrollo, el software de realidad virtual y el sistema de fricción?".

Ilustración 4-23: Porcentajes obtenidos por los usuarios con un 93.3% de satisfacción

Realizado por: Morales, 2024.

La última pregunta es precisa para saber si al finalizar la sesión de simulación los participantes recomendarían el sistema a otras personas. Esta recomendación ayudó a que más personas se acerquen a probar el sistema. Esto se observa en la **Ilustración 4-24** donde el 100% de los usuarios han dado una respuesta positiva. La pregunta realizada fue: "Recomendaría este simulador a otras personas?".

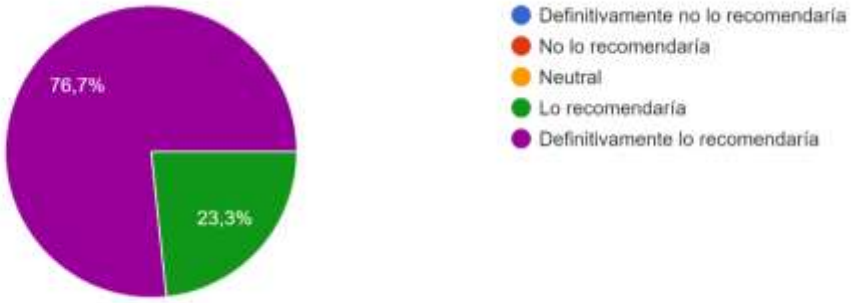


Ilustración 4-24: Porcentajes obtenidos por los usuarios con un 100% de satisfacción

Realizado por: Morales, 2024.

En base a la **Ilustración 4-25** se estableció una valoración satisfactoria al simulador de realidad virtual, donde los usuarios puntuaron los diferentes parámetros evaluados con una respuesta positiva en cada uno de estos, manteniéndose en todas las preguntas con un promedio superior a 6 y de esta manera validando al sistema general.

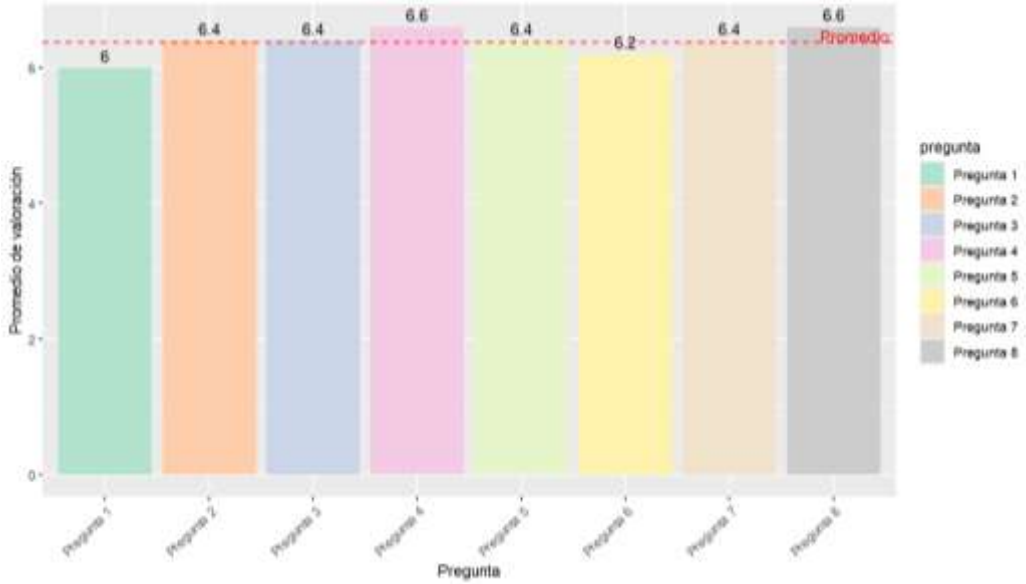


Ilustración 4-25: Valoración general de la encuesta.

Realizado por: Morales, 2024.

4.7. Análisis económico del prototipo

En la **Tabla 4-14** se presenta un análisis de costos en el que se detallan los componentes, la cantidad y sus precios para la construcción del simulador.

Tabla 4-14: Análisis económico para la construcción del prototipo

Cantidad	Componente	Precio Unitario	Precio total
1	Oculus Quest 2	$500.00	$500.00
1	Bicicleta GTI aro 27”	$140.00	$140.00
1	Rodillo ciclosimulador “EAGLE”	$150.00	$150.00
1	ESP32	$12.00	$12.00
1	Controlador TB6600	$18.00	$18.00
1	Motores a pasos NEMA 17 17HS4401	$20.00	$20.00
1	Regulador de voltaje AC/DC 24V/5A	$20.00	$20.00
1	Sensor de campo electromagnético GY-273	$4.00	$4.00
1	Sensor magnético de efecto Hall	$3.00	$3.00
1	Imán acoplable	$1.00	$1.00
	Impresión 3D	$40.00	$40.00
	Estructura MDF	$9.00	$9.00
	Elementos varios	$50.00	$50.00
	Costo de ingeniería 10% del costo total	$96.70	$96.70
	Costo total del simulador		$1063.70

Realizado por: Morales, 2024.

De acuerdo con la tabla anterior, se puede notar que aproximadamente el 60% del presupuesto corresponde al dispositivo Oculus Quest 2 y a la bicicleta GTI utilizada en este prototipo. Estos valores pueden variar en relación con el precio de los elementos que se usen para nuevos desarrollos en base a esta tecnología.

Se ha determinado que el costo total para la implementación del prototipo es de 1063.70 dólares americanos (USD). Con este valor, es posible realizar un análisis de costo-beneficio. Actualmente, no existe en el mercado un dispositivo que ofrezca todas las características desarrolladas para este prototipo. Por lo tanto, se ha optado por seleccionar un dispositivo que brinde la mayor cantidad de características similares al proyecto. El simulador para ciclismo "Zwift" ha sido seleccionado debido a su popularidad en el mercado. Es importante considerar que este simulador requiere elementos adicionales para su uso, como un rodillo inteligente, controladores remotos y una suscripción anual de pago. En la **Tabla 4-15** se detallan los precios de cada uno de estos elementos para establecer el costo total de este simulador.

Tabla 4-15: Costo total del simulador virtual Zwift

Componente	Precio total
Rodillo inteligente Wahoo Kickr	$1400.00
Controladores remotos Zwift Play	$150.00
Suscripción Anual	$150.00
Costo Total del simulador Zwift	$1700.00

Fuente: (Zwift 2024)

Realizado por: Morales, 2024

En la **Tabla 4-16** se presenta la comparativa entre las características de estos dispositivos. Al considerar los costos, se observa una diferencia del 62.57%. Es importante enfocar que cada uno de estos dispositivos está enfocado en un campo específico, por lo que sus características han sido desarrolladas para satisfacer requerimientos de entretenimiento o de entrenamiento según el simulador.

Tabla 4-16: Comparativa entre el prototipo y el simulador de ciclismo Zwift

Características	Prototipo	Zwift
Visualización mediante Realidad Virtual	Si	No
Giro mediante el manubrio	Si	No
Multijugador	No	Si
Modificación física del vehículo	No	Si
Registro de datos	Almacenamiento local	Almacenamiento en la nube
Entorno visual	Personalizable	Mundo virtual fijo
Precio	$1063.70	$1700.00

Fuente: (Zwift 2024)

Realizado por: Morales, 2024

CONCLUSIONES

- Se ha implementado un simulador de realidad virtual para vehículos ligeros manuales enfocado en el entretenimiento, donde es posible que un usuario tenga una experiencia inmersiva a la hora de usar una bicicleta en un sitio fijo mediante un rodillo ciclosimulador, generando satisfactoriamente fricción en función del ángulo de la pendiente en la simulación. Además, ofreciendo la capacidad de almacenar datos específicos de la sesión de entretenimiento.

- El bróker MQTT a través de Local Host mostró ser un protocolo de comunicación estable y seguro, siendo aquel que no contó con perturbaciones en su desviación estándar al momento de la publicación y suscripción de los tópicos, esto teniendo en consideración que la latencia generada se debe a la calidad de conexión con internet.

- Luego de realizar las pruebas de comunicación entre la tarjeta de desarrollo con Unity mediante el protocolo MQTT se obtuvo una diferencia significativa en los tiempos de conexión de 6.008 segundos aplicando la prueba t de Student, lo cual es garantizado al obtener un valor de proporcionalidad demasiado bajo.

- A partir de las pruebas realizadas al proceso de medición de velocidad, se determinó que el de sensado de la velocidad mediante el sensor de efecto Hall y la tarjeta de desarrollo ESP32 garantiza que no existen diferencias significativas en comparación con el velocímetro SB-318, esto mediante un análisis de correlación por el método de Pearson que brindó un valor de 0.999 que representa una relación lineal casi perfecta entre las dos variables.

- Con respecto a la prueba de error del motor se estableció intervalos de error desde -0.10 mm hasta 0.10 mm, existiendo una baja frecuencia en estos límites y una notable permanencia en la distorsión nula.

- En base a la prueba t de Welch se fundamentó que existe una diferencia significativa en los tiempos de accionamiento del motor entre las rutinas planteadas de más de 6000 microsegundos, proporcionando así la verificación de que cada sesión de simulación es única y el motor se accionará correctamente.

- Las pruebas de envío de información desde la tarjeta de desarrollo hacia la base de datos lograron registrar un 100% de recepción de los datos, esto debido a que el valor de correlación obtenido es igual a 1.

- El entorno de realidad virtual diseñado ha demostrado ser una parte valiosa en el proyecto, ofreciendo una mejor experiencia al usuario, en el diseño se establecieron diferentes factores para generar una mayor sensación a los participantes, adicionando diversos elementos tales como sonidos, fauna y flora, los cuales al ser integrados a los visores Oculus Quest 2 brindaron una mejor respuesta con las personas según las encuestas realizadas donde se obtuvo un promedio de 6.375, considerado un margen positivo basado en la escala de Likert.

RECOMENDACIONES

- El sistema de comunicación del simulador de realidad virtual mediante el protocolo MQTT presenta un desafío, ya que tal como se presentó se hace uso de un local host, lo que causa que al momento de estar conectado en una red WiFi diferente a la establecida en el bróker, se produce un cambio en la dirección IP. Para resolver este problema se propone definir un bróker público mediante un servidor, lo que permitirá que el usuario pueda conectarse desde cualquier red de internet y tener acceso a los protocolos de suscripción y publicación.

- Optimizar el diseño electrónico propuesto, esto implica el revisar y mejorar la fiabilidad y facilitar el mantenimiento. De igual forma establecer algún diseño estético y profesional para la visualización de los datos registrados en diversos dispositivos, expandiendo así el seguimiento local de las variables almacenadas.

- Incorporar mejoras en el diseño del entorno virtual, para que de esta forma el ciclo paseo no sea monótono, permitiendo enriquecer la experiencia del usuario. Al diversificar las acciones que el usuario puede efectuar, se aumenta el grado de inmersión. Además, al integrar elementos visuales más detallados y atractivos, así como otros elementos se logrará ampliar la calidad de la experiencia brindada.

- Considerar mayor cantidad de variables físicas del entorno y del usuario, para poder generar una experiencia mucho más personalizada en cuestión de sus características.

BIBLIOGRAFÍA

ARDUINO, 2023. Arduino UNO R3 Features. *Https://Docs.Arduino.Cc* [en línea]. Disponible en: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf.

ARIAS, J.G., 2020. *Robótica aplicada: robot de asistencia para personas con deficiencia motriz* [en línea]. S.l.: UNIVERSIDAD DE LA SALLE. Disponible en: https://ciencia.lasalle.edu.co/ing_automatizacion/790.

ARRIBÉRE, A.S., et. al., 2022. VRBike: Realidad virtual enciclismo indoor. *Instituto Tecnológico Buenos Aires*. S.l.:

AUTODESK, 2024. Fusion Help | Fusion API User's Manual | Autodesk. *Fusion* [en línea]. [consulta: 17 marzo 2024]. Disponible en: https://help.autodesk.com/view/fusion360/ENU/?guid=GUID-C1545D80-D804-4CF3-886D-9B5C54B2D7A2.

BICIO, 2021. Rodillos de transmisión directa. *SPORT* [en línea]. [consulta: 17 noviembre 2023]. Disponible en: https://www.sport.es/bicio/rodillos-de-transmision-directa-informacion-imprescindible/.

BIOT, F.M., 2020. *Imágenes Mediante El Protocolo Mqtt* [en línea]. S.l.: UNIVERSIDAD POLITÉCNICA DE VALENCIA. Disponible en: https://riunet.upv.es/bitstream/handle/10251/152408/Mahedero - Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT..pdf?sequence=1&isAllowed=y.

BRACHO, R. y GUAÑUNA, D., 2023. *Comparación experimental de protocolos de comunicación de IoT en cuanto a la seguridad de transmisión de datos en dispositivos IoT de cobertura WLAN* [en línea]. S.l.: UNIVERSIDAD POLITÉCNICA SALESIANA. Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/24195/1/TTS1134.pdf.

BRISEÑO, J., 2023. *SISTEMA ELECTRÓNICO PARA LA DETECCIÓN DE POSICIÓN ANGULAR Y ERGONOMÍA DE CICLISTAS EMPLEANDO VISIÓN ARTIFICIAL* [en línea]. S.l.: Universidad técnica de Ambato. Disponible en: http://repositorio.uta.edu.ec/bitstream/123456789/38480/1/t2276te.pdf.

CANTO DE GANTE, A., et. al., 2020. Escala de Likert: Una alternativa para elaborar e interpretar un instrumento de percepción social. *Alta Tecnología y Sociedad* [en línea], vol. 38, no. 1, ISSN 1940-2171. Disponible en: https://www.researchgate.net/profile/Alberto-Fernandez-45/publication/361533522_Escala_de_Likert_Una_alternativa_para_elaborar_e_interpretar_un_instrumento_de_percepcion_social/links/62b736d0d49f803365b968

10/Escala-de-Likert-Una-alternativa-para-elaborar-.

CHAMBERS, L., 2021. *Arquitectura orientada a eventos sobre protocolo MQTT* [en línea]. S.l.: UNIVERSIDAD NACIONAL DE LA PLATA. Disponible en: https://sedici.unlp.edu.ar/bitstream/handle/10915/130301/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y.

CHESSA, M., et. al., 2019. La calidad perceptiva del Oculus Rift para la realidad virtual inmersiva. *Human-Computer Interaction*, vol. 34, no. 1, ISSN 07370024. DOI 10.1080/07370024.2016.1243478.

COTTINI, A., 2014. Pavimentos en pendiente. *REVISTA DE LA UNIVERSIDAD DE MENDOZA* [en línea], vol. 8, Disponible en: https://www.um.edu.ar/ojs2019/index.php/RUM/article/view/150.

CREALITY, 2022. Impresora 3D Ender-3 Max Neo - Creality 3D. *creality* [en línea]. [consulta: 17 marzo 2024]. Disponible en: https://www.creality.com/es/products/ender-3-max-neo-3d-printer.

CULIÁÑEZ, A., 2023. *Desarrollo del videojuego educativo PLMan en Godot Engine* [en línea]. S.l.: Universidad de Alicante. Disponible en: https://doc-0s-c0-apps-viewer.googleusercontent.com/viewer/secure/pdf/aftrspj1k6vniqetiv1u41j9b4o783mc/4m8daauv6vci6bfi4jk5jprdn3iv34d4/1699577025000/lantern/17484097836479701688/ACFrOgDD8rWkJFL2HuMKU_K4LbkE3gpZNMrI_tuS0x8x3STywr64btM2qC64IPru2y_shLB1S_V3.

CUN, R. y TUTIVEN, A., 2020. *Automatización y simulación de una línea de producción aplicando SCADA con Node-Red reemplazando las pantallas HMI. PROYECTO* [en línea]. S.l.: ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Facultad. Disponible en: https://www.dspace.espol.edu.ec/bitstream/123456789/57204/1/T-113100 Cun - Tutiven.pdf.

ESPRESSIF SYSTEMS, 2020. ESP32-S2 Series Modules. [en línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

FIGUEROA, J., 2023. *Diseño y Ensamble de un Biorreactor Automatizado para el Proceso de Fermentación del Cacao* [en línea]. S.l.: UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD). Disponible en: https://repository.unad.edu.co/bitstream/handle/10596/56194/jlfigueroasa.pdf?sequence=1&isAllowed=y.

GALARZA, B., 2023. *SISTEMA DE TELEOPERACIÓN CON REALIDAD VIRTUAL PARA LA MANIPULACIÓN DE UN ROBOT MÓVIL.* [en línea]. S.l.: UNIVERSIDAD TÉCNICA DE AMBATO. Disponible en: https://repositorio.uta.edu.ec/bitstream/123456789/39188/1/t2320te.pdf.

GARCÍA, A., 2000. *Realidad virtual*. Madrid: [Universidad Complutense], Servicio de Publicaciones.

GERMANA, O., 2011. Fritzing: Primeros pasos. *Fritzing.org* [en línea]. Venezuela: Disponible en: https://fritzing.org/media/uploads/learning/translations/Fritzing-PrimerosPasos.pdf.

GUTIÉRREZ, A.M.L., 2020. ¿Un mundo nuevo? Realidad virtual, realidad aumentada, inteligencia artificial, humanidad mejorada, Internet de las cosas. *Arbor*, vol. 196, no. 797, ISSN 1988303X. DOI 10.3989/arbor.2020.797n3009.

GUZMAN, E. y SANCHEZ, J., 2020. *DISEÑO DE MÓDULOS PARA LA ADQUISICIÓN DE DATOS Y CARACTERIZACIÓN DE SENSORES DE PROXIMIDAD CON PLC y HMI.* [en línea]. S.l.: UNIVERSIDAD ANTONIO NARIÑO. Disponible en: http://repositorio.uan.edu.co/bitstream/123456789/3099/11/2020JoseDanielSanchezDiaz.pdf.

HERRERA DEL GENER, A.M., 2022. *Exergames: propuesta de un gamepad para sensar movimientos del jugador*. La Plata: Universidad Nacional de la Plata.

HERRERA, M., 2004. Consideraciones para el diseño didáctico de ambientes virtuales de aprendizaje: una propuesta basada en las funciones cognitivas del aprendizaje. *Revista Iberoamericana de Educación (ISSN: 1681-5653)* [en línea], Disponible en: https://rieoei.org/historico/deloslectores/1326Herrera.pdf.

HONEYWELL, 2023. GY-273 Module Magnetic Fiel Sensor. [en línea]. [consulta: 17 marzo 2024]. Disponible en: https://www.mpja.com/download/36668mp.pdf.

IJSSELSTEIJN, W.A., et. al., 2004. Ciclismo Virtual: Efectos de la inmersión y de un entrenador virtual sobre la motivación y la presencia en una aplicación de fitness doméstica. . S.l.:

INDUSTRIAS GSL, 2021. Sensor magnético. *Industrias GSL* [en línea]. [consulta: 11 noviembre 2023]. Disponible en: https://industriasgsl.com/blogs/automatizacion/sensor_magnetico.

JIMÉNEZ, J., et. al., 2009. *Medicina y fisiología del ciclismo* [en línea]. S.l.: s.n. vol. 45. ISBN 9782707156006. Disponible en: https://buleria.unileon.es/bitstream/handle/10612/9306/Garcia-Lopez-2009-Biomecanica-Ciclismo-Libro-FEMEDE.pdf?sequence=1&isAllowed=y.

LANNINGHAM-FOSTER, et. al., 2006. Gasto energético del tiempo de pantalla sedentario comparado con el tiempo de pantalla activo en niños. *Pediatrics*, vol. 118, no. 6, ISSN 00314005. DOI 10.1542/peds.2006-1087.

LIDON, M., 2019. Unity 3D. *MARCOMBO*. S.l.: s.n.,

LITTLEFUSE, 2022. Electronics Catalogs. *Littlefuse* [en línea]. [consulta: 19 marzo 2024]. Disponible en: https://electronicscatalogs.littelfuse.com/Smart-Building-

Application-Guide/20/.

MÁRQUEZ, C., et. al., 2022. Análisis de la aerodinámica de un prototipo de cuadro de bicicleta. *Sociedad Mexicana de Ingeniería Mecánica* [en línea], Disponible en: https://somim.org.mx/memorias/memorias2022/articulos/A4_122.pdf.

MARTÍNEZ, S., 2021. *Motores gráficos en tiempo real aplicados a la arquitectura* [en línea]. S.l.: Universidad de Alicante. Disponible en: https://rua.ua.es/dspace/bitstream/10045/117031/1/MOTORES_GRAFICOS_DE_TIEMPO_REAL_APLICADOS_A_LA_AR_MARTINEZ_MARTINEZ_SANTIAGO.pdf.

MARTINEZ, W., 2021. *DESARROLLO DE UN PROTOTIPO PARA LA MEDICIÓN DE PARTÍCULAS PM2.5 EN LA CIUDAD DE BOGOTÁ* [en línea]. S.l.: Universidad Católica de Colombia. Disponible en: https://repository.ucatolica.edu.co/server/api/core/bitstreams/94fd1cd2-bd0c-48d2-bf27-0b3308ecef5a/content.

MESTRE, D., et. al., 2011. Does Virtual Reality EnhanceExercise Performance, Enjoyment,and Dissociation? An Exploratory Study on a Stationary Bike Apparatus. . Msrseille:

MINDIOLA, M., 2021. *APLICACIÓN DE REALIDAD VIRTUAL PARA LA DESCRIPCIÓN DEL PROCESO AUTOMATIZADO DE PRODUCCIÓN DE BIOETANOL A BASE DE MAÍZ.* [en línea]. S.l.: UNIVERSIDAD DE PAMPLONA. Disponible en: http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/4524/1/Mindiola_2021_TG.pdf.

MORALES, Á., 2023. *Motores Gráficos* [en línea]. S.l.: Universidad de Jaén. Disponible en: https://crea.ujaen.es/bitstream/10953.1/19498/1/MORALES GARCÍA%2C ÁLVARO_INFORMÁTICA_TFM.pdf.

OCULUS, 2020. Características del Oculus Rift. *Oculus* [en línea]. [consulta: 15 noviembre 2023]. Disponible en: https://www.oculus.com/rift-s/features/?locale=es_ES.

OLARTE, I., 2017. *ESTUDIO COMPARATIVO DE LA VENTAJA MECÁNICA DE LOS GRUPOS MUSCULARES ACTUADORES DEL EJERCICIO DE PEDALEO AL USAR UN PLATO CIRCULAR Y UN PLATO O- SYMETRIC.* [en línea]. S.l.: UNIVERSIDAD DE LOS ANDRES. Disponible en: https://repositorio.uniandes.edu.co/server/api/core/bitstreams/9f9bdaa1-013b-4f09-ad26-a13c79fa8a7d/content.

OPENLAB, 2022. Diseño 3D. *Universidad de Huelva* [en línea]. España: Disponible en: https://www.uhu.es/osl/wp-content/uploads/2018/05/Manual-de-TinkerCad..pdf.

ORACLE, 2024. MySQL :: MySQL Documentation. [en línea]. [consulta: 18 marzo 2024].

Disponible en: https://dev.mysql.com/doc/.

ORTEGA, M.,et. al., 2017. Aplicación de la realidad virtual. Agente de neurorecuperador psíquico-físico y deportivo. *International Journal of Developmental and Educational Psychology. Revista INFAD de Psicología.*, vol. 4, no. 1, ISSN 0214-9877. DOI 10.17060/ijodaep.2017.n1.v4.1060.

PALACIOS, J., et. al., 2023. DISPOSITIVOS IMU: UNA COMPARACIÓN CON PERSPECTIVA MECATRÓNICA. *Pistas Educativas*, vol. 45, no. 146,

PÉREZ PUJOL, J., 2022. *Diseño de dirección asistida para bicicleta*. Donostia - San Sebastián: Tecnun Univiersidad de Navarra.

PLAYSTATION, 2020. PlayStation VR. *PlayStation* [en línea]. [consulta: 16 noviembre 2023]. Disponible en: https://www.playstation.com/es-ec/ps-vr/.

QIAOQI WANG, 2018. *Proyecto de control interactivo de exploración VR de Unity y Arduino*. Valencia: Universitat Politecnica de Valencia.

QUINTANA, D., 2023. *DISEÑO Y CONSTRUCCIÓN DE UNA BOBINADORA SEMIAUTOMÁTICA DE TRANSFORMADORES MONOFÁSICOS DE BAJA POTENCIA* [en línea]. S.l.: Universidad Politécnica Salesiana. Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/26339/1/TTS1573.pdf.

RANGEL, Á., 2022. *Desarrollo de un videojuego educativo en Unity: Implementación y almacenamiento de la base de datos.* [en línea]. S.l.: Universidad Politécnica de Madrid. Disponible en: https://oa.upm.es/73253/1/TFG_ALVARO_RANGEL_ARRANZ.pdf.

RASPBERRY PI FOUNDATION, 2018. Raspberry Pi 3 model B. *Raspberry Pi* [en línea], Disponible en: https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf?_gl=1*e07hbo*_ga*MTQ1MDYxNDk4LjE2OTMzMjQxNjM.*_ga_22FD70LWDS*MTY5MzQyODk5Mi4yLjEuMTY5MzQyOTAyOC4wLjAuMA.

RUTASENBICI, 2023. Rodillos para Bicicleta. *Rutasenbici* [en línea]. [consulta: 16 noviembre 2023]. Disponible en: https://rutasenbici.net/rodillos-para-bicicleta-comparativa/.

SALAS NOAIN, D. y DELGADO DEL CASTILLO, M.A., 2023. *Videojuego de realidad virtual para realizar ejercicios en bicicleta estacionaria mediante el uso de un sistema de detección de movimiento y visor Google Cardboard* [en línea]. Lima: UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS. Disponible en: https://academica-e.unavarra.es/bitstream/handle/2454/45183/MEMORIA_TFM_LMR.pdf?sequence=1&isAllowed=y.

SÁNCHEZ, M., 2020. *SISTEMA GRUPAL DE ENTRENAMIENTO ESTÁTICO AL AIRE*

LIBRE PARA AFICIONADOS AL CICLISMO. S.l.: Universidad de Bógota Jorge Tadeo Lozano.

SIDEQUEST, 2024. SideQuest. [en línea]. [consulta: 18 marzo 2024]. Disponible en: https://sidequestvr.com/setup-howto.

STAIANO, A.E. y CALVERT, S.L., 2011. Exergames para cursos de educación física: Beneficios físicos, sociales y cognitivos. *Child Development Perspectives*, vol. 5,

TECNEU, 2023. Fuente Conmutada De Alimentación 24v 5a 120w, 110/220vca (S-120-24). *TECNEU* [en línea]. [consulta: 14 marzo 2024]. Disponible en: https://www.tecneu.com/products/fuente-conmutada-de-alimentacion-24v-5a-120w-110-220vca.

TORO, J., 2023. *DESARROLLO DE UN BRAZO ROBÓTICO DE TIPO SCARA CON ESTRUCTURA DE FIBRA DE CARBONO Y SISTEMA DE MODELADO POR DEPOSICIÓN FUNDIDA* [en línea]. S.l.: UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL CARRERA DE MECATRÓNICA. Disponible en: https://dspace.ups.edu.ec/handle/123456789/25008.

TRIVIÑO, L., 2023. *Conectar cosas a internet con esp32 y raspberry pi* [en línea]. USTA. Colombia: USTA. ISBN 978-958-782-646-3. Disponible en: https://repository.usta.edu.co/bitstream/handle/11634/53071/Obracompleta.Coleccionmodular.2023Trivinoluis.pdf?sequence=1&isAllowed=y.

UNITY, 2024a. Unity - Manual: Unity User Manual 2022.3 (LTS). *Unity3D* [en línea]. [consulta: 17 marzo 2024]. Disponible en: https://docs.unity3d.com/Manual/index.html.

UNITY, 2024b. Unity Asset Store - The Best Assets for Game Making. [en línea]. [consulta: 17 marzo 2024]. Disponible en: https://assetstore.unity.com/.

VALAREZO-GUZMÁN, G.E., et. al., 2023. Simulación y realidad virtual aplicadas a la educación. *RECIMUNDO* [en línea], vol. 7, no. 1, ISSN 2588073X. DOI 10.26820/recimundo/7.(1).enero.2023.432-444. Disponible en: https://recimundo.com/index.php/es/article/view/1967.

VALVERDE, V., et. al., 2019. Análisis descriptivo de base de datos relacional y no relacional. *Atlante Cuadernos de Educación y Desarrollo* [en línea], no. 108, ISSN 1989-4155. Disponible en: https://www.eumed.net/rev/atlante/2019/06/base-datos-relacional.html.

VELÁSQUEZ, Y., et. al., 2021. Detección de la adulteración de la leche mediante sensor óptico. *Revista de Iniciación Científica* [en línea], vol. 7, ISSN 2412-0464. DOI 10.33412/rev-ric.v7.0.3248. Disponible en: https://revistas.utp.ac.pa/index.php/ric/article/view/3248/3943.

VELASTEGUÍ LÓPEZ, E., 2019. El proceso de enseñanza aprendizaje mediante el uso

de realidad virtual. *Explorador Digital*, vol. 2, no. 2, ISSN 2661-6831. DOI 10.33262/exploradordigital.v2i2.328.

VÉLIZ VEGA, A., et. al., 2021. Aprendizaje adaptativo basado en Simuladores de Realidad Virtual Adaptive learning based on Virtual Reality Simulators. *Revista Cubana de Ciencias Informáticas*, vol. 15, no. 2, ISSN 2227-1899.

YAGUANA, C., 2021. *Análisis del sistema de realidad virtual HTC VIVE, dentro de entornos lúdicos y parasociales, aplicado a un grupo de adultos jóvenes* [en línea]. S.l.: Universidad Andina Simón Bolívar. Disponible en: https://repositorio.uasb.edu.ec/bitstream/10644/8425/1/T3678-MEC-Yaguana-Comunicacion.pdf.

ZWIFT, 2024. Wahoo KICKR y suscripción de 1 año a Zwift | Rodillo. [en línea]. [consulta: 23 marzo 2024]. Disponible en: https://eu.zwift.com/es/collections/all/products/wahoo-kickr?variant=42781153722619.

ANEXOS

ANEXO A: HOJA DE DATOS DE LA ESP32

5. Electrical Characteristics

5.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in Table 4 below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the recommended operating conditions.

Table 4: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	-0.3	3.6	V
I_{output} [1]	Cumulative IO output current	-	1,100	mA
T_{store}	Storage temperature	-40	150	°C

1. The module worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground. Please note that pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.
2. Please see Appendix IO_MUX of ESP32 Datasheet for IO's power domain.

5.2 Recommended Operating Conditions

Table 5: Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
VDD33	Power supply voltage	3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T	Operating temperature	-40	-	85	°C

5.3 DC Characteristics (3.3 V, 25 °C)

Table 6: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter		Min	Typ	Max	Unit
C_{IN}	Pin capacitance		-	2	-	pF
V_{IH}	High-level input voltage		0.75×VDD[1]	-	VDD[1]+0.3	V
V_{IL}	Low-level input voltage		-0.3	-	0.25×VDD[1]	V
I_{IH}	High-level input current		-	-	50	nA
I_{IL}	Low-level input current		-	-	50	nA
V_{OH}	High-level output voltage		0.8×VDD[1]	-	-	V
V_{OL}	Low-level output voltage		-	-	0.1×VDD[1]	V
I_{VDD}	High-level source current (VDD[1] = 3.3 V, V_{OH} >= 2.64 V, output drive strength set to the maximum)	VDD3P3_CPU power domain [1, 2]	-	40	-	mA
		VDD3P3_RTC power domain [1, 2]	-	40	-	mA
		VDD_SDIO power domain [1, 2]	-	20	-	mA

Symbol	Parameter	Min	Typ	Max	Unit
I_{OL}	Low-level sink current (VDD[1] = 3.3 V, V_{OL} = 0.495 V, output drive strength set to the maximum)	-	28	-	mA
R_{PU}	Resistance of internal pull-up resistor	-	45	-	kΩ
R_{PD}	Resistance of internal pull-down resistor	-	45	-	kΩ
V_{IL_nRST}	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V

Notes:

1. Please see Appendix IO_MUX of ESP32 Datasheet for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3_CPU and VDD3P3_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 20 mA. V_{OH}>=2.64 V, as the number of current source pins increases.
3. Pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.

5.4 Wi-Fi Radio

Table 7: Wi-Fi Radio Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Operating frequency range [1]	-	2412	-	2484	MHz
Output impedance [2]	-	-	50(nom.)	-	Ω
TX power [3]	11n, MCS7	12	13	14	dBm
	11b mode	17.5	18.5	20	dBm
Sensitivity	11b, 1 Mbps	-	-98	-	dBm
	11b, 11 Mbps	-	-89	-	dBm
	11g, 6 Mbps	-	-92	-	dBm
	11g, 54 Mbps	-	-74	-	dBm
	11n, HT20, MCS0	-	-91	-	dBm
	11n, HT20, MCS7	-	-71	-	dBm
	11n, HT40, MCS0	-	-89	-	dBm
	11n, HT40, MCS7	-	-69	-	dBm
Adjacent channel rejection	11g, 6 Mbps	-	31	-	dB
	11g, 54 Mbps	-	14	-	dB
	11n, HT20, MCS0	-	31	-	dB
	11n, HT20, MCS7	-	13	-	dB

1. Device should operate in the frequency range allocated by regional regulatory authorities. Target operating frequency range is configurable by software.
2. For the modules that use IPEX antennas, the output impedance is 50 Ω. For other modules without IPEX antennas, users do not need to concern about the output impedance.
3. Target TX power is configurable based on device or certification requirements.

6. Schematics

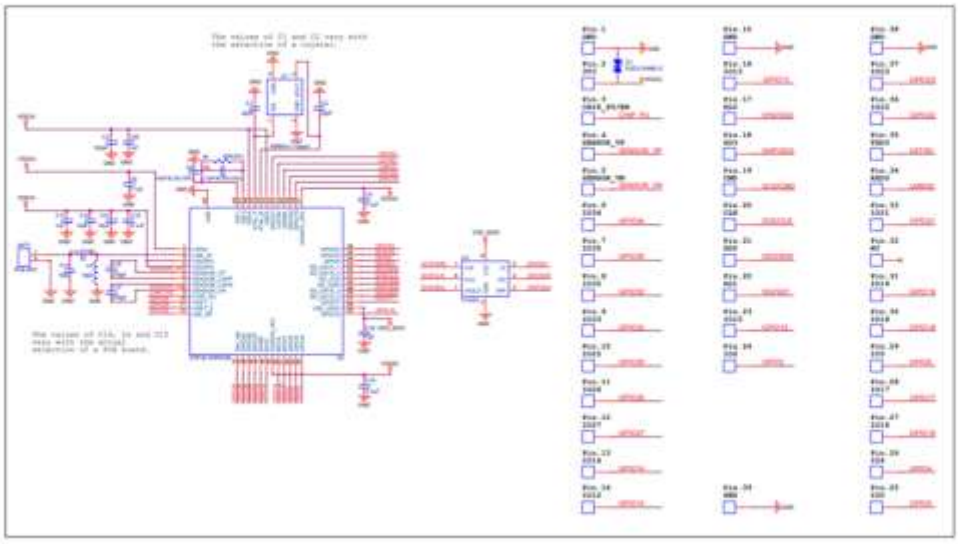


Figure 3: ESP32-WROOM-32 Schematics

ANEXO B: HOJA DE DATOS CONTROLADOR TB6600

1. Introduction

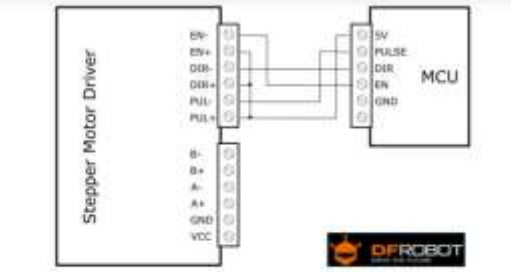
This is a professional two-phase stepper motor driver. It supports speed and direction control. You can set its micro step and output current with 6 DIP switch. There are 7 kinds of micro steps (1, 2 / A, 2 / B, 4, 8, 16, 32) and 8 kinds of current control (0.5A, 1A, 1.5A, 2A, 2.5A, 2.8A, 3.0A, 3.5A) in all. And all signal terminals adopt high-speed optocoupler isolation, enhancing its anti-high-frequency interference ability.

Features:

- Support 8 kinds of current control
- Support 7 kinds of micro steps adjustable
- The interfaces adopt high-speed optocoupler isolation
- Automatic semi-flow to reduce heat
- Large area heat sink
- Anti-high-frequency interference ability
- Input anti-reverse protection
- Overheat, over current and short circuit protection

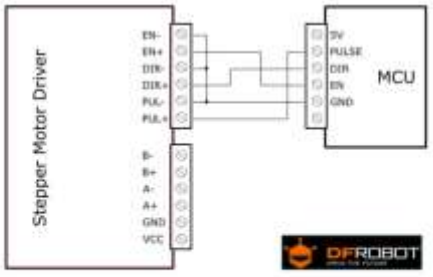
Electrical Specification:

Input Current	0~5.0A
Output Current	0.5-4.0A
Power (MAX)	160W
Micro Step	1, 2/A, 2/B, 4, 8, 16, 32
Temperature	-10~45℃
Humidity	No Condensation
Weight	0.2 kg
Dimension	96*56*33 mm



Common-Cathode Connection:

Connect PUL -, DIR - and EN - to the ground terminal of the control system. Pulse signal connects to PUL-; direction signal connects to Dir- ; Enable signal connects to EN-. As shown below:



Note: When "EN" is in the valid state, the motor is in a free states (Off-line mode). In this mode, you can adjust the motor shaft position manually. When "EN" is in the invalid state, the motor will be in an automatic control mode.

INPUT & OUTPUT:

- **Signal Input:**

PUL+	Pulse +
PUL-	Pulse -
DIR+	Direction +
DIR-	Direction -
EN+	Off-line Control Enable +
EN-	Off-line Control Enable -

- **Motor Machine Winding:**

A+	Stepper motor A+
A-	Stepper motor A-
B+	Stepper motor B+
B-	Stepper motor B-

- **Power Supply:**

VCC	VCC (DC9-42V)
GND	GND

- **Wiring instructions**

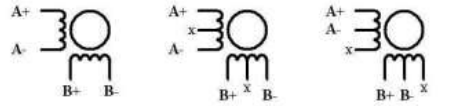
There are three input signals in all: ① Step pulse signal PUL +, PUL-; ② Direction signal DIR +, DIR-; ③ off-line signal EN +, EN-. The driver supports common-cathode and common-anode circuit, you can select one according to your demand.

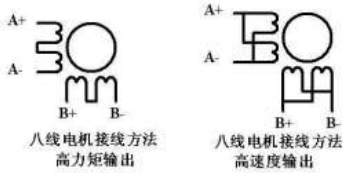
Common-Anode Connection:

Connect PUL +, DIR + and EN + to the power supply of the control system. If the power supply is + 5V, it can be directly connected. If the power supply is more than + 5V, the current limiting resistor R must be added externally. To ensure that the controller pin can output 8 ~ 15mA current to drive the internal optocoupler chip. Pulse signal connects to PUL-; direction signal connects to Dir- ; Enable signal connects to EN-. As shown below:

2. Stepper Motor Wiring:

Two-phase 4-wire, 6-wire, 8-wire motor wiring, as shown below:





ANEXO C: HOJA DE DATOS DEL MOTOR A PASOS NEMA 17HS4401

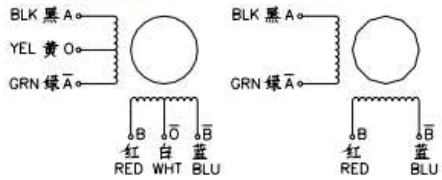
MotionKing

HB Stepper Motor Catalog

MotionKing (China) Motor Industry Co., Ltd.

2 Phase Hybrid Stepper Motor
17H2A series-Size 42mm(1.8 degree)

Wiring Diagram:



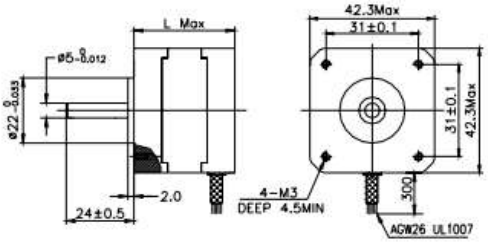
UNI-POLAR(6 LEADS) BI-POLAR(4LEADS)

Electrical Specifications:

Series Model	Old P/N	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm²)	Lead Wire (No.)	Motor Weight (g)
17H2A2406	17HS2408	28	0.6	8	10	12	1.6	34	4	150
17H2A3413	17HS3401	34	1.3	2.4	2.8	28	1.6	34	4	220
17H2A3417	17HS3410	34	1.7	1.2	1.8	28	1.6	34	4	220
17H2A3404	17HS3430	34	0.4	30	35	28	1.6	34	4	220
17H2A3604	17HS3630	34	0.4	30	18	21	1.6	34	6	220
17H2A4417	17HS4401	40	1.7	1.5	2.8	40	2.2	54	4	280
17H2A4413	17HS4402	40	1.3	2.5	5.0	40	2.2	54	4	280
17H2A4612	17HS4612	40	1.2	4.1	3.5	28	2.2	54	6	280
17H2A4604	17HS4630	40	0.4	30	28	28	2.2	54	6	280
17H2A8417	17HS8401	48	1.7	1.8	3.2	52	2.6	68	4	350
17H2A8413	17HS8402	48	1.3	3.2	5.5	52	2.6	68	4	350
17H2A8423	17HS8403	48	2.3	1.2	1.6	52	2.6	68	4	350
17H2A8604	17HS8630	48	0.4	30	38	34	2.6	68	6	350
17H2A9612	17HS9612	58	1.2	4.5	5.5	54	3.2	75	6	450
17H2A9425	17HS9425	58	2.5	1.7	2.7	76	3.2	75	4	450

*Note: We can manufacture products according to customer's requirements.

Dimensions: unit=mm



Motor Length:

Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
17HS4XXX	40 mm
17HS8XXX	48 mm

ANEXO D: HOJA DE DATOS DEL SENSOR GY-273

Compass Module 3-Axis HMC5883L

The Compass Module 3-Axis HMC5883L is a low-field magnetic sensing device with a digital interface. The compass module converts any magnetic field to a differential voltage output on 3 axes. This voltage shift is the raw digital output value, which can then be used to calculate headings or sense magnetic fields coming from different directions. The module is designed for use with a large variety of microcontrollers with different voltage requirements.

Features

- 3-Axis magneto-resistive sensor
- 1 to 2 degree compass heading accuracy
- Wide magnetic field range (+/-8 gauss)
- Fast 160 Hz maximum output rate
- Precision in-axis sensitivity and linearity
- Measures Earth's magnetic field, from milli-gauss to 8 gauss

Key Specifications

- Power Requirements: 2.7 to 6.5 VDC
- Communication Interface: I²C (up to 400 kHz)
- Operating temperature: -22 to +185 °F (-30 to +85 °C)
- Dimensions: 0.73 x .65 in (1.8 x 1.7 cm)

Application Ideas

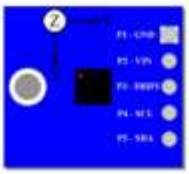
- Auto and personal navigation
- UAV systems
- Robotic navigation
- Location-based services (LBS)

Resources and Downloads

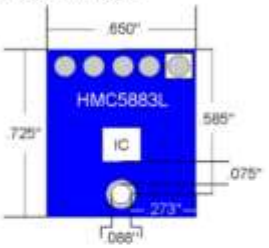
Check for the latest version of this document, free software, and example programs from the Compass Module 3-Axis HMC5883L product page. Go to www.parallax.com and search for part number 29133.

Pin Descriptions

Pin	Name	Type	Function
1	GND	G	Ground
2	VIN	P	Supply Voltage – 2.7 to 6.5 VDC (module is regulated to 2.5VDC)
3	DRDY	I	Data Ready, interrupt. (internally pulled high, see datasheet for details)
4	SCL	I	I²C Clock (Pulled high on module, Clock 160Hz Default)
5	SDA	IO	I²C Data (Pulled high on module)



Module Dimensions



Device Information

The following information is taken from the HMC5883L datasheet

Specifications

Characteristics	Conditions	Min	Typ	Max	Units
Average Current Draw	Idle Mode Measurement Mode	- -	2 100	- -	µA
Field Range	Full Scale (FS) – Total applied field (Typical)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Resolution	VDD=3.0V, GN=2		5		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		Ppm
Cross-Axis Sensitivity	Test Conditions: Cross field +0.5 gauss, Happied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurement Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data- ready		6		ms
Turn-on Time	Ready for I²C commands		200		µs
Gain Tolerance	All gain/dynamic range settings		±5		%
I²C Address	7-bit address 8-bit read address 8-bit write address		0X1E 0X3D 0X3C		hex hex hex
I²C Rate	Controlled by I²C Master			400	kHz
I²C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL & SDA – Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		 0.2*VDDIO 0.8*VDDIO		 Volts Volts

Communication Protocol

The compass module communicates via a two-wire I²C bus system as a slave device. It supports standard and fast modes, 100 kHz and 400 kHz, but does not support a high-speed mode. No external pull-up resistors are required to support these standard and fast speed modes (4.7 kΩ pull-up resistors are built into module). *See page 8 in the datasheet for I²C communication details.

Modes of Operation

The Compass Module has several modes that can be accessed from the I²C bus, whose primary purpose is power management. All the modes are controlled by the Mode Register. The following section describing modes, was taken from the Honeywell HMC5883L datasheet.
*See page 9 in the datasheet for more details.

Continuous-Measurement Mode
During continuous-measurement mode, the device continuously makes measurements, at user selectable rate, and places measured data in data output registers. All registers maintain values while in continuous-measurement mode and I²C bus is enabled for use by other devices on the network.

Single-Measurement Mode
This is the default power-up mode. During single-measurement mode, the device makes a single measurement and places the measured data in data output registers. All registers maintain values while in single-measurement mode. The I²C bus is enabled for use by other devices on the network while in single-measurement mode.

Idle Mode
During this mode the device is accessible through the I²C bus, but major sources of power consumption are disabled, such as, but not limited to, the ADC, the amplifier, and the sensor bias current. All registers maintain values while in idle mode. The I²C bus is enabled for use by other devices on the network while in idle mode.

Register List

This device is controlled and configured via a number of on-chip registers. The table below lists the registers and their access. All address locations are 8 bits. For a detailed description of these registers please access the datasheet on our Product Page.

Address Location	Register Name	Access
00	Configuration A	Read/Write
01	Configuration B	Read/Write
02	Mode	Read/Write
03	Data Output X MSB	Read
04	Data Output X LSB	Read
05	Data Output Z MSB	Read
06	Data Output Z LSB	Read
07	Data Output Y MSB	Read
08	Data Output Y LSB	Read
09	Status	Read
10	Identification A	Read
11	Identification B	Read
12	Identification C	Read

ANEXO E: PRUEBAS DE LOS USUARIOS CON EL SISTEMA

ANEXO F: CÓDIGO ESP32

```
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <MechaQMC5883.h>
WiFiClient esp32Client;
PubSubClient mqttClient(esp32Client);
const char* ssid = "SSID ACTUAL";
const char* password = "CONTRASEÑA ACTUAL";
const char* server = "IP DEL SERVIDOR";
int port = 1883;
const unsigned long TIMEOUT = 2500;
unsigned long startTime = 0;
unsigned long lastTimeSent = 0;
unsigned long lastDataReceivedTime = 0;
bool contadorIniciado = false;
int ctiempo = 0;
const int sensorPin = 32;
const float radioLlanta = 0.21;
const float pi = 3.1416;
int previousState = HIGH;
unsigned long previousTime = 0;
int cont = 0;
float velocidadAngular = 0;
unsigned long tiempoOFF = 0;
bool velocidadCeroEnviada = false;
int data;
const int pulPin = 25;
const int dirPin = 14;
const int enPin = 26;
String pendiente;
int pendiente;
int pendiente_anterior = 0;
int pasos;
MechaQMC5883 qmc;
float acimutReferencia = -1;
void wifiInit() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
}
void callback(char* topic, byte* payload, unsigned int length) {
  String messageTemp;
  for (int i = 0; i < length; i++) {
    messageTemp += (char)payload[i];
  }
  if (String(topic) == "inclinación") {
    float pendiente = messageTemp.toFloat() * -1;
    if (pendiente < -30 || pendiente > 45) {
    } else {
      if (pendiente != pendiente_anterior) {
        if (pendiente > pendiente_anterior) {
          digitalWrite(dirPin, HIGH);
        } else {
          digitalWrite(dirPin, LOW);
        }
        pasos = map(abs(pendiente - pendiente_anterior), 0, 30, 0, 2000);
        for (int x = 0; x < pasos; x++) {
          digitalWrite(pulPin, HIGH);
          delayMicroseconds(125);
          digitalWrite(pulPin, LOW);
          delayMicroseconds(125);
        }
        pendiente_anterior = pendiente;
        String datos = String(pendiente);
        mqttClient.publish("pendiente", datos.c_str());
      }
    }
  }
}
void reconnect() {
  while (!mqttClient.connected()) {
    if (mqttClient.connect("arduinoClient")) {
      mqttClient.subscribe("inclinación");
    } else {
      delay(5000);
    }
  }
}
void setup() {
  pinMode(sensorPin, INPUT_PULLUP);
  pinMode(pulPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enPin, OUTPUT);
  digitalWrite(enPin, LOW);
  Serial.begin(115200);
  wifiInit();
  mqttClient.setServer(server, port);
  mqttClient.setCallback(callback);
  mqttClient.subscribe("inclinación");
  Wire.begin();
  qmc.init();
  int x, y, z;
  qmc.read(&x, &y, &z, &acimutReferencia);
}
void loop() {
  int x, y, z;  float acimutActual;
  qmc.read(&x, &y, &z, &acimutActual);
  float anguloGiro = acimutActual - acimutReferencia;
  if (anguloGiro > 180) {
    anguloGiro -= 360;
  } else if (anguloGiro < -180) {
    anguloGiro += 360;
  }
  Serial.println(int(anguloGiro));  // muestra valor de angulo de giro sin decimales
  mqttClient.publish("giro", String(int(anguloGiro * -1)).c_str());
  if (!mqttClient.connected()) {
    reconnect();
  }
  mqttClient.loop();
  int sensorState = digitalRead(sensorPin);
  unsigned long currentTime = millis();
  if (sensorState != previousState) {
    if (sensorState == LOW) {
      cont++;
      unsigned long timeDiff = currentTime - previousTime;
      velocidadAngular = 60000.0 / timeDiff / 68.0 * 2 * pi;
      if (velocidadAngular <= 50) {
        Serial.println(velocidadAngular);
        mqttClient.publish("velocidad", String(velocidadAngular).c_str());
        lastTimeSent = currentTime;
        velocidadCeroEnviada = false;
      }
    }
    previousTime = currentTime;
  }
  if (sensorState == HIGH && !velocidadCeroEnviada && currentTime - lastTimeSent > TIMEOUT) {
    Serial.println(0);
    mqttClient.publish("velocidad", "0");
    velocidadCeroEnviada = true;
  }
  previousState = sensorState;delay(10);
}
```

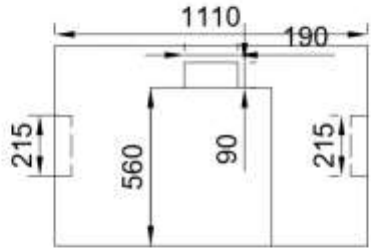
ANEXO G: CÓDIGO UNITY

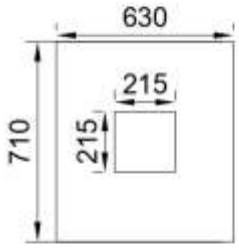
```
using System.Collections;
using UnityEngine;

public class mqttController : MonoBehaviour
{
    public string nameController = "Controller 1";
    public string tagOfTheMQTTReceiver = "MQTT_ENTRADA";
    public mqttReceiver _eventSender;
    public float moveSpeed = 0.0025f; // Velocidad base de movimiento del
objeto
    private float distanceTravelled;
    private float currentSpeed = 0.0f; // Velocidad actual del objeto
    private float targetSpeed = 0.0f; // Velocidad objetivo que se recibe
de MQTT
    private float currentSteeringAngle = 0.0f; // Ángulo de giro actual
del objeto
    private float targetSteeringAngle = 0.0f; // Ángulo de giro objetivo
que se recibe de MQTT
    public float smoothTime = 2f; // Tiempo de suavizado para la
interpolación de la velocidad
    private int lastInclinationPublished;
    public AudioSource soundbici;

    void Start()
    {
        GameObject[] receivers =
GameObject.FindGameObjectsWithTag(tagOfTheMQTTReceiver);
        if (receivers.Length > 0)
        {
            _eventSender = receivers[0].GetComponent<mqttReceiver>();
            if (_eventSender != null)
            {
                _eventSender.OnVelocityMessageArrived += (msg) =>
OnMessageArrivedHandler("velocidad", msg);
                _eventSender.OnSteeringMessageArrived += (msg) =>
OnMessageArrivedHandler("giro", msg);
                _eventSender.OnStartMessageArrived    += (msg) =>
OnMessageArrivedHandler("start", msg);
            }
            else
            {
                Debug.LogError("No se encontró el componente mqttReceiver
en el objeto con tag '" + tagOfTheMQTTReceiver + "'.");
            }
        }
        else
        {
            Debug.LogError("No se encontró ningún objeto con el tag '" +
tagOfTheMQTTReceiver + "'.");
```

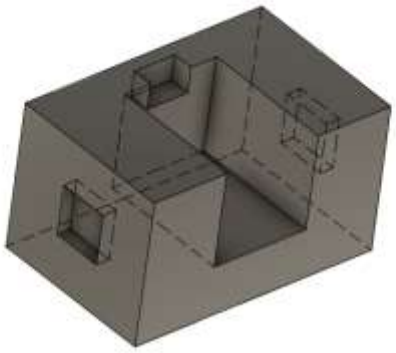
```
        }
    }

    void Update()
    {
        // Suavizar la transición de la velocidad actual a la velocidad
objetivo
        currentSpeed = Mathf.Lerp(currentSpeed, targetSpeed, smoothTime *
Time.deltaTime);

        // Mover la bicicleta hacia adelante según la velocidad actual
        transform.Translate(Vector3.forward * currentSpeed *
Time.deltaTime);

        // Suavizar la transición del ángulo de giro actual al ángulo de
giro objetivo
        currentSteeringAngle = Mathf.Lerp(currentSteeringAngle,
targetSteeringAngle, smoothTime * Time.deltaTime);

        // Girar la bicicleta según el ángulo de giro actual
        transform.Rotate(Vector3.up, currentSteeringAngle *
Time.deltaTime*-2);

        // Publicar la rotación en el eje X al tópico "inclinación"
        PublishInclination();
    }

    private void OnMessageArrivedHandler(string topic, string message)
    {
        Debug.Log("Event Fired. The message, from Object " +
nameController + " is = " + message);
        if (topic == "velocidad")
        {
            // Intenta convertir el mensaje en un valor float para la
velocidad
            if (float.TryParse(message, out float speedValue))
            {
                // Actualiza la velocidad objetivo con el valor recibido
                targetSpeed = speedValue * moveSpeed;
                if(targetSpeed>0){
                    soundbici.Play();

                }
                else if(targetSpeed<=0){
                    soundbici.Stop();
                }
            }
            else
            {
```

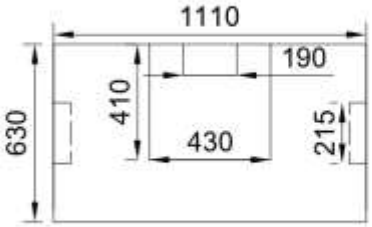
```
                Debug.LogWarning("El mensaje recibido no se puede
convertir en un valor float para la velocidad: " + message);
            }
        }
        else if (topic == "giro")
        {
            // Intenta convertir el mensaje en un valor float para el
giro
            if (float.TryParse(message, out float steeringValue))
            {
                // Actualiza el ángulo de giro objetivo con el valor
recibido
                targetSteeringAngle = steeringValue;
            }
            else
            {
                Debug.LogWarning("El mensaje recibido no se puede
convertir en un valor float para el giro: " + message);
            }
        }

        if (topic == "start" && message == "0")
        {
            Debug.Log("Finalizado: " + message);
            #if UNITY_EDITOR
            UnityEditor.EditorApplication.isPlaying = false;
            #else
            Application.Quit();
            #endif
        }

    }

    void PublishInclination()
    {
        if (_eventSender != null)
        {
            float inclination = transform.rotation.eulerAngles.x;
            // Corregir el valor de la inclinación si es mayor a 180
            if (inclination > 180)
            {
                inclination -= 360;
            }
            // Convertir a entero redondeado
            int roundedInclination = Mathf.RoundToInt(-inclination);
            // Verificar si es diferente al último valor
            if (roundedInclination != lastInclinationPublished)
            {
                // Publicar como entero
                _eventSender.PublishInclination("inclinación",
roundedInclination);
                // Guardar el último valor publicado
                lastInclinationPublished = roundedInclination;
            }
        }
    }
}
```

ANEXO H: PLANOS DE LA BASE DEL MOTOR



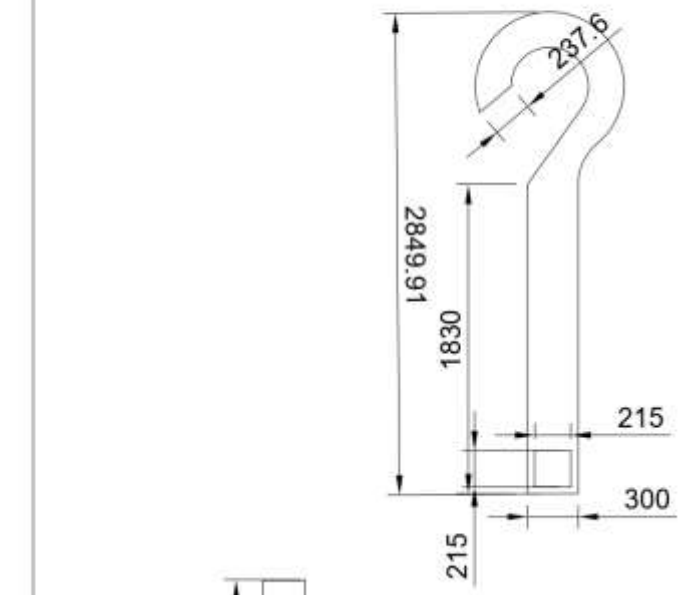


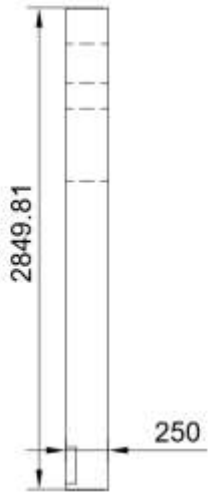


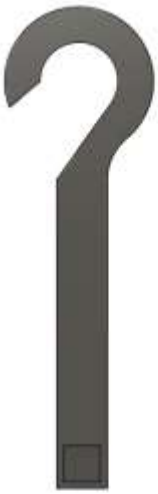


Dept.	Technical reference	Created by Michael Morales	Approved by
		Document type	Document status
		Title Base de motor	DWG No.
			Rev. / Date of issue / Sheet 1/1

ANEXO I: PLANOS DE LOS GANCHOS PARA LA BASE DEL MOTOR

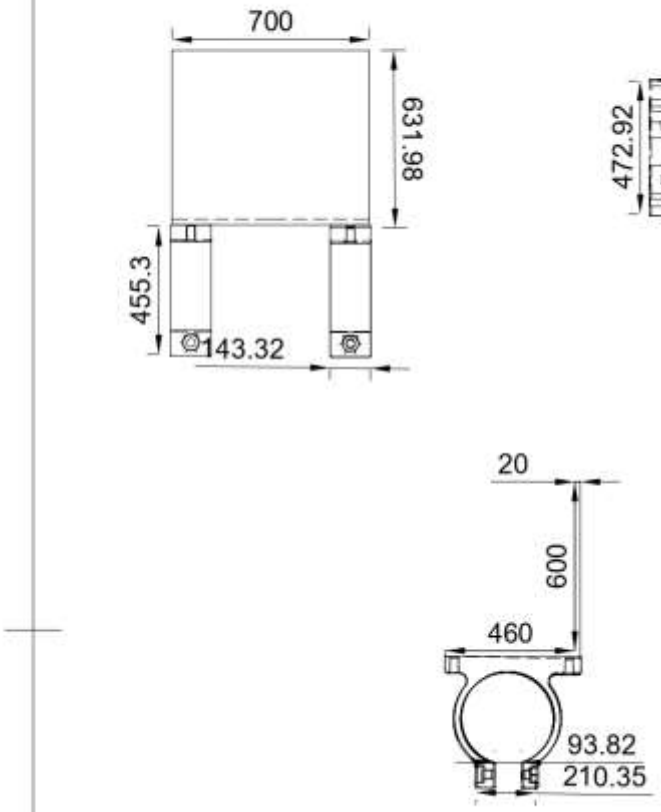






Dept.	Technical reference	Created by: Michael Morales	Approved by
		Document type	Document status
		Title: Gancho Tesis	DWG No.
			Rev. / Date of issue / Sheet: 1/1

ANEXO J: PLANOS DEL SOPORTE PARA EL SENSOR ELECTROMAGNÉTICO



Dept.	Technical reference	Created by Michael Morales	Approved by
		Document type	Document status
		Title Soporte Sensor	DWG No.
			Rev. / Date of issue / Sheet 1/1

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICADO DE CUMPLIMIENTO DE LA GUÍA PARA

NORMALIZACIÓN DE TRABAJOS DE FIN DE GRADO

Fecha de entrega: 20 / 03 / 2024

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: Michael Steven Morales Rosario
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Electrónica y Automatización
Título a optar: Ingeniero en Electrónica y Automatización
 **f. Director del Trabajo de Titulación:** Ing. Ramiro Fernando Isa Jara. PhD  **f. Asesor del Trabajo de Titulación:** Ing. Jorge Luis Hernández Ambato. PhD