

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE WNCs, MEDIANTE HERRAMIENTAS DE
PROGRAMACIÓN NUMÉRICA, REALIDAD VIRTUAL Y AUMENTADA PARA
EL CONTROL DE VELOCIDAD DE MOTORES DC**

**AUTORES:
RODRIGO JAVIER BAUTISTA TOAPANTA
BRYAN ANDRÉS PILLAJO SALAZAR**

**TUTOR:
PILLAJO ANGOS CARLOS GERMÁN**

Quito, Septiembre 2021

CESION DE DERECHOS DE AUTOR

Nosotros, Rodrigo Javier Bautista Toapanta y Bryan Andrés Pillajo Salazar , con documentos de identificación N° 1719992628 y N° 1726793241 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **DESARROLLO DE WNCS, MEDIANTE HERRAMIENTAS DE PROGRAMACIÓN NUMÉRICA, REALIDAD AUMENTADA E IOT PARA EL CONTROL DE VELOCIDAD DE MOTORES DC** , mismo que ha sido desarrollado para optar por el título de: INGENIEROS ELECTRÓNICOS , en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Rodrigo Javier Bautista Toapanta
C.I. No. 1719992628



.....
Bryan Andrés Pillajo Salazar
C.I. No. 1726793241

Quito, Septiembre 2021

DECLARACIÓN DE COUTORIA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DESARROLLO DE WNCS, MEDIANTE HERRAMIENTAS DE PROGRAMACIÓN NUMÉRICA, REALIDAD AUMENTADA E IOT PARA EL CONTROL DE VELOCIDAD DE MOTORES DC, realizado, Bautista Toapanta Rodrigo Javier y Pillajo Salazar Bryan Andrés, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, Septiembre 2021

A handwritten signature in blue ink, consisting of several overlapping loops and lines, positioned above a dotted line.

.....
Carlos Germán Pillajo Angos

C.I. No. 1709255119

DEDICATORIA

Este trabajo es dedicado a mi madre María Adela Salazar Farinango quien con su arduo trabajo, esfuerzo y apoyo incondicional me ha permitido alcanzar una de las metas que me he propuesto, donde gracias a sus valores como la perseverancia y tenacidad que me enseñó desde pequeño se ve reflejado en lo que soy ahora.

Bryan

Dedico este trabajo de titulación a mis padres Manuel Bautista y Martha Toapanta quienes con su apoyo incondicional cariño y dedicación han hecho posible alcanzar esta meta, siendo ellos pilares fundamentales para mi desarrollo personal como profesional.

Rodrigo

AGRADECIMIENTOS

Agradezco a mi padre Luis Antonio Pillajo ya que gracias a los aportes realizados en mi vida profesional y cotidiana ha permitido culminar una etapa más de mi vida con éxito y como no tener presente a toda mi familia quien con su apoyo total me han encaminado a perseverar en cada uno de los obstáculos que se ha presentado a lo largo de mi vida.

Bryan

Quiero agradecer a Dios por permitirme tener y disfrutar de mi familia, gracias a mi familia por todo el apoyo dado en el transcurso de mi carrera y a todos quienes han hecho posible de una u otra manera que hoy en día me encuentre cumpliendo con una meta más en mi vida.

Rodrigo

Agradecemos a la Universidad Politécnica Salesiana, a los docentes y autoridades ya que han sido para nosotros fuente de conocimiento y una guía esencial para la culminación de este proceso, siendo ellos participes activos día a día en la formación ética y profesional para la obtención de nuestro objetivo convirtiéndonos en profesionales de excelencia.

Además, agradecemos a nuestro tutor Ing. Carlos Pillajo quien con su conocimiento y sabiduría nos supo guiar en todo este trayecto motivándonos constantemente dándonos los recursos para que este proyecto se haga realidad.

Los autores

ÍNDICE

DEDICATORIA	iv
AGRADECIMIENTOS	v
ÍNDICE	vi
RESUMEN	xii
ABSTRACT	xiii
1. CAPÍTULO 1	1
ANTECEDENTES	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 JUSTIFICACIÓN	2
1.3 OBJETIVOS	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	3
1.4 METODOLOGÍA	4
2. CAPÍTULO 2	5
FUNDAMENTACIÓN TEORICA	5
2.1 SISTEMAS WNCS	5
2.1.1 NODE RED	6
2.1.2 PLACA DE DESARROLLO NODE MCU ESP8266	6
2.1.3 PROTOCOLO MQTT	7
2.2 MUNDO VIRTUAL	7
2.2.1 SIMULINK	8
2.2.2 VIRTUAL REALITY TOOLBOX	8
2.3 REALIDAD AUMENTADA	8
2.3.1 UNITY	9
2.3.2 VUFORIA	10
2.3.3 ANDROID	10

2.3.4 VISUAL STUDIO	11
2.4 CONTROLADOR PID	11
2.4.1 PID TUNER.....	12
3. CAPÍTULO 3.....	13
FUNDAMENTACIÓN TEORICA.....	13
3.1 RED INALAMBRICA DE CONTROL WNCS.....	13
3.2 PLANTA DE CONTROL DE VELOCIDAD.....	14
3.3 COMUNICACION NODE MCU ESP8266 Y NODE-RED.....	15
3.3.1 INSTALACION DE NODE-RED.....	17
3.3.2 DIAGRAMA DE BLOQUES Y COMUNICACIÓN.....	22
3.3.3 IMPLEMENTACION DE LA COMUNICACION MQTT EN LA WNCS	23
3.4 REALIDAD AUMENTADA	24
3.4.1 INSTALACIÓN Y CONFIGURACIÓN DE UNITY	26
3.4.2 IMPORTACION DE ASSETS DE VUFORIA	31
3.4.3 DISEÑO DE INTERFAZ GRAFICA EN UNITY	34
3.4.4 CONEXION UNITY CON NODE RED.....	36
3.4.5 EXPORTACION DEL APK EN ANDROID	37
3.5 REALIDAD VIRTUAL.....	38
3.5.1 MODELADO DE MOTOR DC.....	38
3.5.2 DIAGRAMA DE BLOQUES DEL MOTOR DC	40
3.5.3 DISEÑO PID PARA PLANTA	41
3.5.4 INSTALACION DE LIBRERÍA TRUE TIME	42
3.5.5 DISEÑO DE MOTOR 3D	45
3.5.6 CONEXIÓN SIMULINK CON VR REALM BUILDER	48
4. CAPÍTULO 4.....	49
PRUEBAS Y RESULTADOS	49

4.1	FUNCIÓN DE TRANSFERENCIA DE LA PLANTA.....	49
4.2	PID CONTROL DE VELOCIDAD.....	51
4.3	REALIDAD AUMENTADA	55
4.4	REALIDAD VIRTUAL	59
5.	CAPÍTULO 5.....	60
	CONCLUSIONES.....	60
	RECOMENDACIONES.....	62
	REFERENCIAS	63
	ANEXOS.....	65

INDICE DE FIGURAS

Figura 2.1. Ventana de trabajo Unity 2018.3.11f1.	9
Figura 2.2. Modelo 3D de árbol Abdul.....	10
Figura 2.3. Interfaz de Visual Studio	11
Figura 3.1. Diagrama didáctico de WNCS.....	13
Figura 3.2. Esquemático planta Velocidad	15
Figura 3.3. Planta de control de velocidad.....	15
Figura 3.4. Instalación de pluggins del NodeMCU	16
Figura 3.5. Instalación de la librería PubSubClient	17
Figura 3.6. Página oficial de NodeJS.....	18
Figura 3.7. Ayudante de instalación de NodeJS	18
Figura 3.8. Configuración como servicio de Windows	19
Figura 3.9. Error VCRUNTIME140.dll.....	20
Figura 3.10. Instalación de librerías de Open ssl	20
Figura 3.11. Aplicación Services Windows.....	21
Figura 3.12. Activación del servicio.....	21
Figura 3.13. Comprobación de puerto TCP	22
Figura 3.14. Diagrama de Bloques para la comunicación del sistema	22
Figura 3.15. Conectividad en Node Red.....	24
Figura 3.16. Reconocimiento de imagen en Unity	25
Figura 3.18. Reconocimiento de imagen en Unity	27
Figura 3.19. Instalación de módulos.....	28
Figura 3.20. Elección de plataforma para la aplicación de AR	29
Figura 3.21. Configuración de los Player Settings en Android.....	30
Figura 3.22. Configuración de los Player Settings en iOS.....	31
Figura 3.23. Configuración de los Player Settings en UWP.....	31
Figura 3.24. Incorporación de elementos de Vuforia	32
Figura 3.25. Generar llave Vuforia	32
Figura 3.26. Creación de una base de datos.....	33
Figura 3.27. Activación de AR Camera	34
Figura 3.28. Jerarquía de las funciones de AR.....	35
Figura 3.29. Escena Final en Unity.....	35
Figura 3.30. Diagrama didáctico de conexión de la realidad aumentada	36

Figura 3.31. Exportación APK a plataforma Android	38
Figura 3.32. Sistema para modelado de Motor DC.....	38
Figura 3.33. Parámetros de diseño de Dinámica del motor	40
Figura 3.34. Diseño en Simulink de la Dinámica del motor	41
Figura 3.35. Diseño de la planta en simulink con PID.....	42
Figura 3.36. Librería Truetime instalada en simulink.	43
Figura 3.37. Diseño de la planta en simulink con PID y conexión con TRUETIME.	44
Figura 3.38. Generador de funciones	44
Figura 3.39. librería simulink 3D.	45
Figura 3.40. Asistente de VR Sink.	45
Figura 3.41. Ventana New Word en 3D.	46
Figura 3.42. Figuras básicas en V Realm Builder 2.0.	47
Figura 3.43. Diseño del motor DC en 3D.	47
Figura 3.44. Exporta variable de rotación para el eje del motor DC.....	48
Figura 3.45. Bloque VR sink con la variable de entrada a controlar.....	48
Figura 4.1. Datos y señal de entrada	50
Figura 4.2. Funciones de transferencia Motor DC.....	51
Figura 4.3. PID Tunner Matlab para Planta de Velocidad (Motor DC)	52
Figura 4.4. Nodo de algoritmo de control PID	53
Figura 4.5. Gráfica de comportamiento inestable.....	54
Figura 4.6. Gráfica estabilizada y sintonizada.	54
Figura 4.7. Set point de 5874[rpm] y 5031[rpm].	55
Figura 4.8. Reconocimiento de la planta por la realidad aumentada en una PC	56
Figura 4.9. Interfaz en el dispositivo Samsung A50.....	57
Figura 4.10. Comparación entre AR y Node-red con Set point = 5874.....	58
Figura 4.11. Comparación entre AR y Node-red con Set point = 5031.....	58
Figura 4.12. Comportamiento de la planta, Realidad Virtual vs Mundo real. ..	59

INDICE DE TABLAS

Tabla 1.1. Características del Microcontrolador ESP8266.....	6
Tabla 3.1. Características del celular Samsung A50.....	25

RESUMEN

Desarrollo de un sistema inalámbrico de control cerrado, el cual permite el control de velocidad de motores DC a través de un algoritmo para este caso PID que está configurado en la herramienta de programación de la red NODE-RED permitiendo de forma visual mediante un motor de flujos enfocado hacia IOT conectar varios dispositivos en una red inalámbrica, la planta se encuentra controlada por la tarjeta embebida ESP8266 que contiene un módulo de comunicación inalámbrica wifi además de entradas y salidas digitales como analógicas que puede ser programada a través del IDE de Arduino junto con librerías que permiten la interconexión con módulos electrónicos, tal como sensores, actuadores, etc. A esto sumado la realidad aumentada (AR) que mediante la identificación de una escena previamente establecida, sobrepone información digitalmente sobre la imagen para observar los datos de velocidad de la planta en tiempo real, esta aplicación puede ser visualizada desde un dispositivo móvil, comunicada por el protocolo MQTT hacia el bróker, en este caso la computadora central mediante una dashboard se puede confirmar los valores de velocidad, se utiliza Realidad Virtual (VR) para la creación de una interfaz gráfica que permita una simulación de una escena 3D en la que a través de herramientas de programación en este caso Simulink nos permite desarrollar el modelado del sistema que cuenta con diversas librerías para simular tanto el mundo virtual como la red inalámbrica y así observar el comportamiento dinámico del mismo de manera ideal para proceder a compararlo con el mundo real.

ABSTRACT

Development of a wireless closed control system, which allows the speed control of DC motors through an algorithm for this case PID that is configured in the programming tool of the NODE-RED network allowing visually through a flow engine focused on IOT connect various devices in a wireless network, the plant is controlled by the ESP8266 embedded card that contains a wireless wifi communication module in addition to digital and analog inputs and outputs that can be programmed through the Arduino IDE along with libraries that allow interconnection with electronic modules, such as sensors, actuators, etc.. In addition to this, augmented reality (AR) is used, which by identifying a previously established scene, digitally superimposes information on the image to observe the speed data of the plant in real time, this application can be viewed from a mobile device, communicated by the MQTT protocol to the broker, in this case the central computer through a dashboard can confirm the speed values, Virtual Reality (VR) is used for the creation of a graphical interface that allows a simulation of a 3D scene in which through programming tools in this case Simulink allows us to develop the modeling of the system that has several libraries to simulate both the virtual world and the wireless network and thus observe the dynamic behavior of the same in an ideal way to proceed to compare it with the real world.

INTRODUCCIÓN

El proyecto presentado a continuación se realizó con fines investigativos como también académicos, debido a que el sistema se encuentra segmentado en dos partes, por un lado, el mundo real en el cual todo el sistema se encuentra conectado de forma inalámbrica a la planta y a su controlador. Este sistema permite observar los datos de velocidad en tiempo real, por otro lado, tenemos la simulación en 3D del sistema que se comporta de forma ideal, por lo que el usuario podrá comparar el comportamiento tanto de manera real como virtual, a continuación, se detalla paso a paso el desarrollo del proceso.

El primer capítulo muestra el planteamiento del problema y los respectivos objetivos que se deben desarrollar en este proyecto, según las características necesarias de un WNCS, mediante herramientas de programación numérica y realidad aumentada e IoT para el control de velocidad de motores DC.

El capítulo 2 presenta rápidamente los conceptos que se involucraran el desarrollo de la red inalámbrica de control tales como: lazos de control cerrado, comunicación inalámbrica con su respectivo protocolo de comunicación MQTT, realidad aumentada, simulación 3D, que permita el desarrollo del sistema.

El capítulo 3 detalla el procedimiento para diseñar la placa electrónica o planta, implementación del protocolo de comunicación MQTT y programación tanto de la realidad aumentada como de la realidad virtual, el desarrollo de la interfaz gráfica de planta de control, con el objetivo de realizar diferentes pruebas a su correcto funcionamiento.

El capítulo 4 indica el desarrollo de los ensayos en la planta con su respectivo algoritmo de control, asociada con la realidad virtual para observar y comparar los resultados adquiridos de manera real e ideal.

En el capítulo 5 se redactan las recomendaciones y conclusiones basado en las pruebas realizadas previamente y los resultados que se obtuvieron en este proyecto poniéndolo

a prueba con variaciones en los Setpoint y causando perturbaciones en el actuador, antes y después de la implementación del sistema.

CAPÍTULO 1

ANTECEDENTES

1.1 PLANTEAMIENTO DEL PROBLEMA

El impacto de nuevas tecnologías tanto en la rama información como de la comunicación traen diversos enfoques para el desarrollo de procesos de control ya que permiten la formación de conceptos innovadores y el desarrollo de conocimientos, en el Ecuador la industria 4.0 está tomando gran participación porque genera nuevas formas de procesamiento en la fabricación dejando obsoleto la manera tradicional y brindando comunicación entre la industria, los proveedores y los clientes, sin embargo al momento de realizar un sistema de control de forma confiable se recurre generalmente a simulaciones de un proceso para obtener una predicción ideal, no obstante al momento de trasladar esto al mundo real ya sea por medio de controladores u otros, hay necesidad de adecuar los datos y algoritmos de control dependiendo del software a utilizar, ya que no existe una comunicación directa entre el proceso de simulado y real. El gran problema que surge en la industrial al momento de adquirir la información de un proceso ya sea características o datos importantes generados en tiempo real, no existe una accesibilidad inmediata a tal información una de las herramientas que se puede brindar una solución es la Realidad Aumentada.

En cualquier tipo de industria para controlar alguna variable física sea temperatura, humedad o ventilación, es implementado un sistema de control a lazo cerrado PID o varios otros tipos de control industrial, ya que se tiene que evaluar y comparar la salida con la entrada del sistema para obtener seguridad y eficacia dentro de la planta. Sin embargo, esto requiere de dispositivos acordes con el ambiente industrial para este caso se utiliza elementos con las mismas características, pero en un entorno didáctico. El proceso evolutivo de las diferentes tecnologías provenientes de la red tiende a agregar servicios web, el uso de redes sociales, redes de sensores que al aplicar en un proceso permiten obtener información de una variable física o de otro tipo y además, permite el control actuadores. (Syafiq and Mohd 2019)

Estos sistemas de control pueden llevar consigo una compleja estructura que ocupara grandes espacios físicos ya que está conformado en sí mismo por sensores, actuadores, controladores, buses de datos y redes de alimentación de energía, que se dispersan por el medio físico donde está instalada la planta, este tipo de estructura presenta varios problemas que obstaculizan la producción de toda una empresa. (Tabaa et al. 2018)

Es necesario estar a la vanguardia con respecto a sistemas de monitoreo remoto como la aplicación de realidad aumentada para la visualización de cambios en las variables controladas, las simulaciones en entornos virtuales las cuales permitirán entender el comportamiento del sistema antes de ser adaptado al mundo real y a base de tecnologías Iot observar y comparar de manera inalámbrica los dispositivos que conforman el sistema a controlar.

Por lo expuesto anteriormente, se plantea la siguiente pregunta: ¿Existe un sistema que permite comparar una variable tanto en el mundo real como en la realidad virtual, por un lado, utilizando herramientas IoT y por otro software de programación numérica?

1.2 JUSTIFICACIÓN

Analizando los distintos problemas que se encuentran en las industrias con respecto a los altos costos de los buses de datos, cableado eléctrico, el desperdicio de espacios que podrían ser aprovechados de una manera más útil en el sistema y la falta de movilidad para el control de la planta. Este proyecto brinda una solución para estos problemas, el cual puede implementarse en las empresas de manera inmediata y a bajo costo, generando un beneficio económico y comodidad en el proceso de la planta. En la actualidad el uso de herramientas de software como simuladores entre otros ayudan tanto a la comprensión como a la predicción del comportamiento de un sistema para su posterior implementación teniendo en cuenta los valores óptimos para lo cual contar con esta herramienta es esencial para minimizar los errores al instante de implementar un sistema de control, para observar el comportamiento de la planta en el mundo real es necesario utilizar herramientas que sean accesibles al usuario, permitan obtener los valores y características las variables en tiempo real para lo cual la Realidad Aumentada es una buena opción.

El proyecto tiene como alcance obtener una planta la cual pueda ser observada y monitoreada tanto de manera física como de manera virtual a través de herramientas de programación numérica para elegir el sistema de control más adecuado y esto sumado a una comunicación inalámbrica la cual permitirá enlazar diferentes procesos en tiempo real ayudando tanto a la comprensión técnica y matemática del sistema a controlar.

1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar un WNCS, mediante herramientas de programación numérica, realidad virtual y aumentada para el control de velocidad de Motores DC.

1.3.2 Objetivos Específicos

- Estudiar la base teórica, utilizando las diferentes herramientas de programación para la implementación de los algoritmos de control.
- Implementar un WNCS (Wireless Network Control System) mediante protocolo MQTT para el control de velocidad de un motor DC.
- Diseñar el entorno virtual a través de una herramienta de simulación 3D (Virtual reality toolbox) para el análisis del comportamiento dinámico ideal del sistema.
- Desarrollar un algoritmo mediante software de realidad aumentada para la visualización de la velocidad del motor DC en tiempo real.
- Realizar pruebas del funcionamiento, mediante la comprobación de las variables para la comunicación entre los elementos que conforman el sistema.

1.4 METODOLOGÍA

- Para poner en marcha el presente proyecto se procede al uso de la metodología de investigación, el cual se detalla paso por paso a continuación:
- Mediante el método inductivo se podrá relacionar el presente proyecto con investigaciones previas, lo cual permitirá basarse en conocimientos ya establecidos, sirviendo de apoyo para el desarrollo de diseños de control inalámbricos.
- Se aplica el método experimental, ya que una vez estudiada las bases teóricas se procede con el diseño el sistema inalámbrico y de procesamiento de las variables controladas para realizar el montaje de la red, mediante la obtención de resultados a través de la experimentación.
- Se realiza el método de análisis ya que permitirá identificar los parámetros específicos, analizar las diferentes variables para adecuar los algoritmos de control que se va a utilizar en el sistema inalámbrico.
- Con el método explorativo se podrá observar los resultados obtenidos con los diferentes algoritmos de control vistos tanto en simuladores, como de forma física así poder determinar cuáles serán los parámetros efectivos para el correcto diseño de control.

CAPÍTULO 2

FUNDAMENTACIÓN TEORICA

En este capítulo se estudia de manera teórica la fundamentación necesaria para la comprensión de un sistema WNCS y el control de un motor DC tanto de forma real o física y de manera virtual, para lo cual se estudia la manera de realizar una conexión inalámbrica que permita enlazar varios dispositivos en la red, además se investiga sobre la comunicación entre una aplicación de realidad aumentada y la red, así como los conceptos para realizar una simulación en 3D desde Matlab mediante Simulink.

2.1 SISTEMAS WNCS

Para el sistema a implementar se tiene control cerrado, el cual comunica la información generada por el sensor y enviada hasta el controlador, de igual manera la información que es emitida por el controlador es enviada hacia el actuador. Este tipo de sistemas de control cerrado que intercambian datos a través de redes digitales han sido implementados desde hace tiempo atrás, pero con el pasar del tiempo existe un creciente interés en la rama de los sistemas de control en red se debe a que en actualmente existe el desarrollo de tecnologías y redes digitalizadas que están a la disponibilidad del público y se la puede encontrar en cualquier parte, las cuales pueden ser implementadas con el fin de realizar un lazo realimentado de control con un costo muy bajo. La conectividad de forma inalámbrica brinda facilidad al momento de buscar la extensión más alta del área en la aplicación de los sistemas de control cerrado hasta los objetos móviles de la red, pues los valores de las variables controladas pueden ser transmitidas desde todos los lugares que contiene la planta.

Un WNCS es un sistema de control de lazo cerrado que se debe considerar en los sistemas de redes inalámbricas para poder transmitir los datos, estas redes hoy en día son una parte fundamental puesto que generan ahorro en el cableado permitiendo un mejor aprovechamiento del espacio físico sin embargo puede existir varios problemas de perturbación en el envío y recepción de datos por esto se debe analizar en qué lugar puede funcionar sin generar problemas al momento de operar ,los sistemas inalámbricos al momento abarcan un área importante de investigación. (A and Hincapié 2018)

2.1.1 NODE RED

Es una herramienta de software en la cual se puede conectar varios dispositivos de hardware con API y servicios en línea. Por lo tanto, los desarrolladores no necesitan escribir ellos mismos las API del lado del servidor. Además, Node-RED tiene un enfoque hacia el IoT permitiendo gráficamente conectar los elementos del sistema con más facilidad por medio de una interface gráfica muy amigable a base de nodos que sirven para recibir, enviar, procesar, transformar y almacenar datos. Por lo tanto, hace que los desarrolladores sean más rápidos al momento de realizar programas de IoT. (Rattanapoka et al. 2019)

Los elementos se pueden guardar para su posterior utilización, su tiempo de ejecución está basado en Node.js y almacenados mediante JSON. (Chanthakit 2018)

2.1.2 PLACA DE DESARROLLO NODE MCU ESP8266

Es un micro controlador es una gran ayuda para sistemas inalámbricos brindando una solución al momento de trabajar con wifi, es ideal para aplicaciones IoT ya que su costo es accesible y sus aplicaciones pueden ser muchas permitiendo el envío y recepción de datos por internet a continuación se detalla las características de este módulo. tabla 1(Ramos-Escamilla 2011)

Tabla 1.1. Características del Microcontrolador ESP8266

Voltaje de funcionamiento	3.3v
Consumo de corriente	10uA-170mA
Memoria Flash	16Mb máx. (512 k normal)
Tipo de Procesador	Tensilica L106 32bit
Velocidad del procesador	80-160Mhz
GPIOs	17
Analógico a digital	1 entrada con 10 bit (1024 valores)
Soporte 802.11	b/g/n/d/e/i/k/r
Máximas conexiones al mismo tiempo	5
Tipo de conexión	Wifi

Características para la selección ideal con respecto a la conexión inalámbrica, Fuente: (Ramos-Escamilla 2011)

2.1.3 PROTOCOLO MQTT

Está basado en un protocolo de publicación y suscripción que funciona con diferentes protocolos como TCP/IP entre otros. Es un protocolo ligero diseñado para maximizar el tráfico de red y brindar de forma rápida y sencilla la confiabilidad de las aplicaciones de mensajería escritas para estos entornos. MQTT se puede utilizar para aplicaciones móviles y también para aplicaciones de telemetría y sensor de máquina a máquina. El patrón de mensajería de suscripción pública requiere un intermediario de mensajes. Así, en MQTT, el intermediario actúa para reenviar el mensaje recibido de los editores a los suscriptores de acuerdo con el tema del mensaje. (Rattanapoka et al. 2019)

Se puede conectar varios clientes a la vez a temas de interés para hacer uso de la información teniendo una interfaz común, los temas siempre se utilizan con jerarquías haciendo que los clientes puedan enviar y recibir mensajes. (Chanthakit 2018)

El protocolo MQTT generalmente se usa en sistemas embebidos porque puede hacer que los sensores se comuniquen hasta el sistema, de manera liviana a través del cable siendo fácil de implementar. (Eridani and Widiyanto 2018)

2.2 MUNDO VIRTUAL

La realidad virtual para el caso está representada en una simulación en 3D que simula cualquier planta o sistema a controlar brindando un ambiente en el que se puede experimentar sin manera de correr riesgos para una posterior implementación, desde un punto de vista más técnico la Realidad Virtual es un modo de manipular e interactuar mediante las computadoras datos sumamente complejos el usuario puede manipular a conveniencia este ambiente siendo una

herramienta muy importante tanto para aprendizaje como para experimentación teniendo en cuenta que este será un mundo ideal. (Escartín, Superior, and José 1998).

2.2.1 SIMULINK

Es una herramienta de alto nivel interpretado por el software Matlab este entorno permite mediante diagrama de bloques obtener simulaciones siendo valores precisos que son generados de forma automática basado en modelos. Permite diseñar sistemas muy complejos mediante su interface, también posee múltiples librerías permitiendo simulaciones en tiempo real de cualquier clase de sistema dinámico con un alto nivel de abstracción. (Sun et al. 2009)

2.2.2 VIRTUAL REALITY TOOLBOX

Es una herramienta que permite realizar simulaciones de cualquier tipo de sistema mecánico en una escena 3D mediante una interface sencilla y practica a base de figuras geométricas básicas se puede enlazar una variable utilizando señales para controlar diversos parámetros como la posición traslación u orientación de los objetos generados en este entorno, este editor permite visualizar el cualquier tipo de planta previamente editada mediante una interface de funciones generando una animación .(Sandino, Bejar, and Ollero 2010)

2.3 REALIDAD AUMENTADA

Consta de tres fases principales como el reconocimiento, seguimiento y alineado de la información para esto la imagen previamente cargada debe estar correctamente alineada con la imagen del mundo real por una parte debe existir un elemento que capture las imágenes para este caso un smartphome, mediante el uso de la cámara va captar un patrón previamente cargado para reconocer y obtener una imagen sintetizada sobre el mismo para este caso los datos del motor y específicamente la velocidad en revoluciones por minuto la RA es capaz de interpretar la información obtenida en el

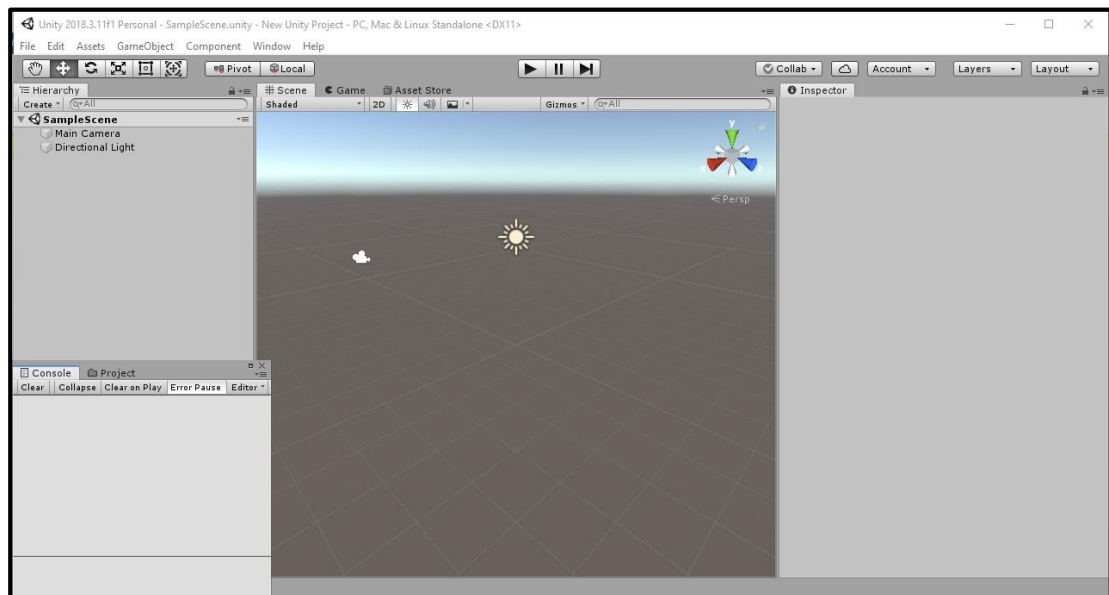
mundo real para generar una imagen o diseño elaborado sobre la misma .(Lacueva Perez F.J., Gracia Bandrés M.A. Sanagustín Grasa L.M., Gonzalez Muñoz C. 20015)

2.3.1 UNITY

El programa Unity es un motor de desarrollo para videojuegos el cual contiene una gran variedad de rutinas y funciones de programación para la creación de un entorno visual que pueden ser desarrolladas para diferentes plataformas como PC, Smartphone y demás. Las características más importantes que contiene Unity y que además facilita el desarrollo de una buena aplicación son los siguientes: motor gráfico y motor físico.

Según Sung Lae Kim (2014) el conjunto de herramientas completo, el espacio de trabajo intuitivo, la función de prueba y edición de juego sobre la marcha de Unity hacen que los desarrolladores ahorren tiempo y esfuerzo. La extensión de realidad aumentada (AR) de Vuforia para Unity permite la detección de características dentro de una imagen, así como el seguimiento dentro de una aplicación de Unity, el entorno de desarrollo se puede mostrar en la figura 2.1.

Figura 2.1. Ventana de trabajo Unity 2018.3.11f1.



Fuente: Bryan Pillajo y Javier Bautista

2.3.2 VUFORIA

Es un kit de desarrollo gratuito y es compatible con múltiples plataformas como Android, iOS o Unity. Para que funcione, los usuarios deben registrarse y descargar la clave de licencia del sitio web de Vuforia. La funcionalidad más utilizada de este paquete es la capacidad de rastrear imágenes en tiempo real. Cada objeto en el mundo real se etiquetará con una imagen, y la aplicación puede reconocer esta imagen y luego colocar el modelo 3D correspondiente sobre el objeto del mundo real, una muestra de la realidad aumentada se puede observar en la figura 2.2. (Nguyen and Dang 2017)

Figura 2.2. Modelo 3D de árbol Abdul



Fuente:(Nguyen and Dang 2017)

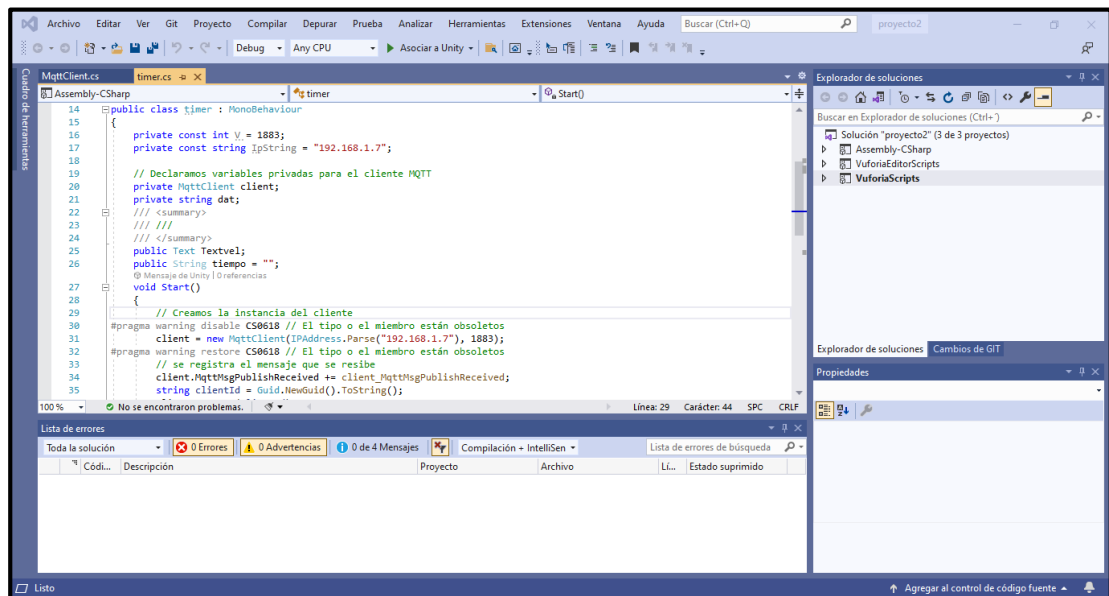
2.3.3 ANDROID

Como dice Báez, Borrego y Cordero (2014) es un tipo de plataforma basado en Linux para celulares, este sistema operativo trabaja por medio de framework de Java a diferencia de otros cualquier persona puede programar y crear aplicaciones siendo un sistema de código abierto Las características de Android permiten que Unity pueda emparejar a su SDK para el desarrollo de aplicaciones para teléfonos inteligentes, utilizando esta función, Unity puede implementar reconocimiento imágenes pregrabadas, inteligencia artificial y seguimiento dentro de la aplicación, así como también utiliza la conectividad inalámbrica del teléfono para el intercambio de datos con un servidor.

2.3.4 VISUAL STUDIO

Como dice Ritchie (2016), los IDE suelen proporcionar un entorno para editar, refactorizar y compilar código; editar interfaces de usuario; lógica de diagrama; construir aplicaciones; y realizar algún nivel de análisis de código, pruebas e integración de control de código fuente. Los IDE comenzaron cuando solo había terminales de consola, por lo que originalmente comenzaron como aplicaciones basadas en consola. Los primeros IDE solo realizaban la edición, la gestión de archivos, la compilación y la depuración / ejecución. Estos entornos funcionaban solo con aplicaciones basadas en consola y no necesitaban editar visualmente la interfaz de usuario. Más tarde, cuando las interfaces gráficas de usuario estuvieron disponibles, los IDE comenzaron a admitir la edición de interfaces de usuario. Aunque cada interfaz de usuario se basa en una especificación que es de naturaleza textual, es raro que las interfaces de usuario se editen textualmente, el entorno de desarrollo de Visual Studio se lo indica en la figura 2.3.

Figura 2.3. Interfaz de Visual Studio



Fuente: Bryan Pillajo y Javier Bautista

2.4 CONTROLADOR PID

El PID es un algoritmo de control que combina su acción proporcional derivativa e integral para ayudar a reducir el sistema hasta llegar a valores requeridos, también logra hacer correcciones para compensar posibles perturbaciones y mantener la

variable de manera controlada aumentando la estabilidad del sistema anticipativo. (Álamo 2007)

2.4.1 PID TUNER

La herramienta de sintonización PID Tuner como dijo Paganelli (2019). Es una aplicación que diseña automáticamente diferentes controladores como P, PI, PD, PID y PID con filtro, etc. El controlador PID está disponible en forma paralela o estándar. Para lograr el rendimiento deseado, puede ajustar interactivamente las ganancias del controlador después del diseño. Los objetivos del ajuste PID con el sintonizador PID son un rendimiento aceptable con una estabilidad robusta que es un margen de ganancia de 60 db con un margen de fase infinito. El algoritmo del sintonizador PID logra los objetivos de diseño mediante un equilibrio entre robustez y rendimiento.

El controlador PID es un controlador de lazo cerrado, es aquel que compara las entradas con las salidas del sistema calculando un error y corrigiendo la salida hasta lograr llegar a un Set point previamente establecido por el operador.

CAPÍTULO 3

FUNDAMENTACIÓN TEORICA

En el presente capítulo detalla el diseño y desarrollo de la WNCS para el control de velocidad de un motor DC tanto con elementos de IoT y realidad aumentada al mismo tiempo acompañada por un diseño en 3D desarrollada con herramientas de programación numérica. Para poder llevar a cabo el objetivo planteado es necesario revisar las características que conforman el sistema para de esta manera cumplir con éxito el proyecto técnico.

3.1 RED INALAMBRICA DE CONTROL WNCS

La red inalámbrica está compuesta por una planta de control de velocidad de un motor dc enlazada por un módulo NODEMCU ESP8266 el cual permite la conexión vía WiFi hasta el bróker de la red compartiendo así datos de entrada y salida, adicional a eso dependiendo de los valores de entrada se modificará el valor de PWM que controla la velocidad del actuador, ya que en la computadora principal se encuentra programado un control PID en el software NODE-RED. A continuación, se puede observar en la figura 3.1 el diagrama didáctico de la red inalámbrica.

Figura 3. 1 Diagrama didáctico de WNCS



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Esta red inalámbrica de control cuenta con un servidor que se encuentra alojado en la computadora principal junto con NODE-RED, que a su vez este software puede utilizar el protocolo MQTT para el enlace de dispositivos y genera una interfaz gráfica o dashboard. Con la ayuda de nodos de funcionalidad gráfica se puede ilustrar la

recepción de los datos de velocidad que además de permitir la visualización se puede modificar el set point del algoritmo PID.

3.2 PLANTA DE CONTROL DE VELOCIDAD

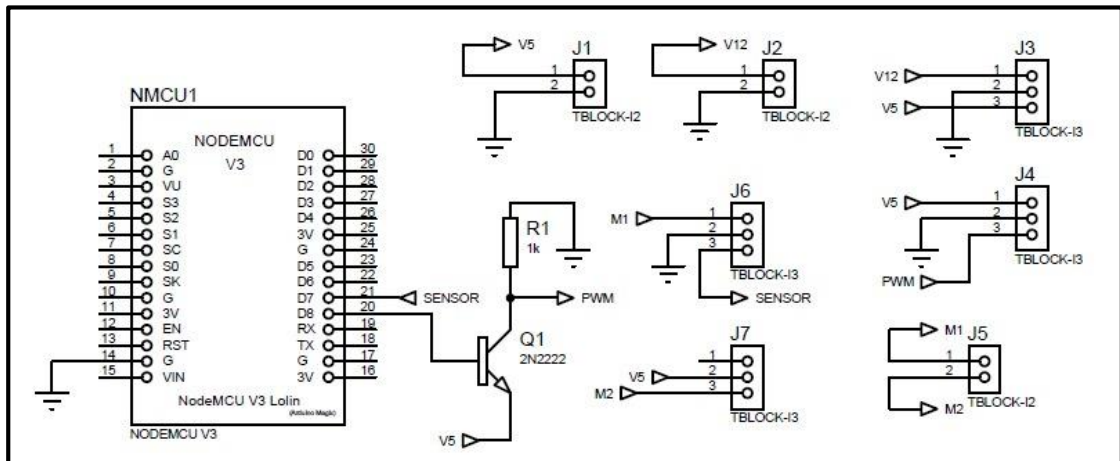
Esta planta está conformada por un controlador ESP8266 enlazado a un motor de velocidad POLOLU de voltaje máximo 9v y velocidad máxima 6500 rpm, en este motor viene incluido un encoder que genera una señal de pulsos el cual permite escalar la velocidad en revoluciones por minuto mediante codificación en el IDE de Arduino. El control de la planta depende del algoritmo configurado en NODE-RED por lo tanto la velocidad de la planta puede ser observada y escrita a través de una Dashboard permitiéndole al operador modificar estos parámetros con la finalidad de analizar la curva generada por la velocidad a través del tiempo.

Elementos:

- Q1 2N2222A Transistor para escalar el pwm de 3.3v a 5v
- R1 1k Ω resistor reductor de corriente
- L298N activación del motor Pololu

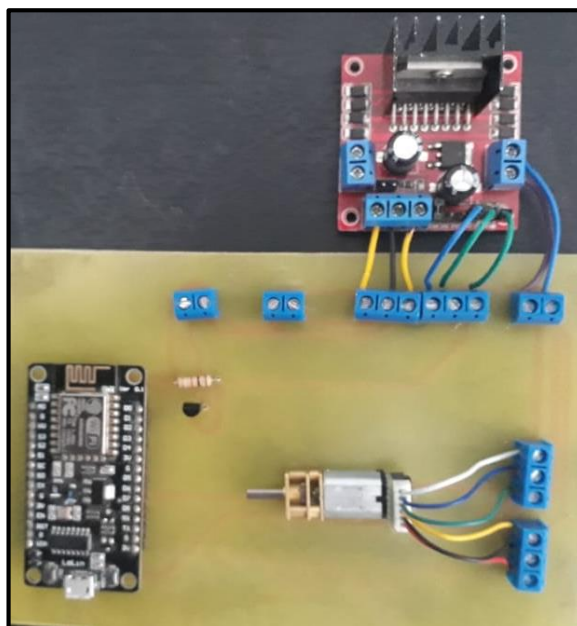
En la figura 3.1 se puede observar el esquemático de la planta diseñada en el Software Proteus que por sus funciones permite la elaboración de una placa electrónica donde todos los componentes serán interconectados tal como se puede observar en la figura 3.2. y el resultado final se observa en la figura 3.3.

Figura 3.2. Esquemático planta Velocidad



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Figura 3.3. Planta de control de velocidad



Elaborado por: Bryan Pillajo y Rodrigo Bautista

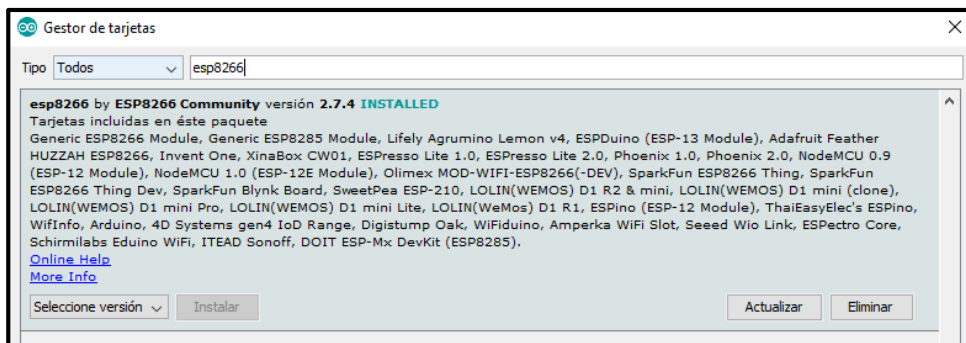
3.3 COMUNICACION NODE MCU ESP8266 Y NODE-RED

La conexión entre la tarjeta embebida ESP8266 depende de las librerías, pluggins y drivers que se instalan en el IDE de Arduino con el fin de utilizar y programar el NodeMCU, por lo tanto es importante realizar el proceso que esta a continuación:

- Es indispensable la instalacion previa de la version 1.6.4 o superior del IDE de Arduino.
- Ir a preferencia y seleccionar la opción de Gestor de URLs Adicionales de Tarjetas, ingresar:

http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Seleccione OK
- Ingrese a la opcion de menú Herramientas > Placa > Gestor de tarjetas se busca la opción ESP8266 Comunity, seleccionar la última versión e instalar, tal como se muestra en la figura 3.4.

Figura 3.4. Instalación de pluggins del NodeMCU



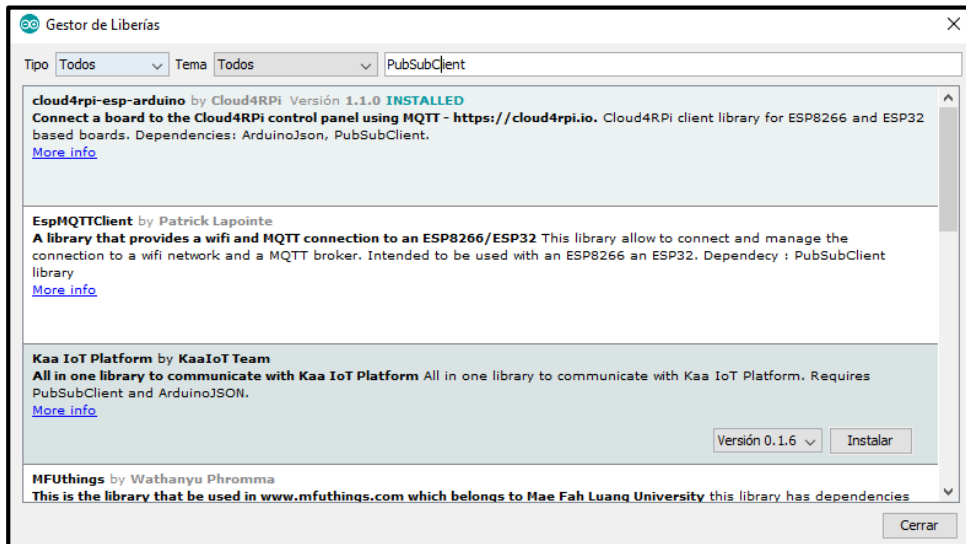
Elaborado por: Bryan Pillajo y Rodrigo Bautista

- Para la programación del NodeMCU se utiliza la conexión con el puerto serial de la computadora y la tarjeta con una velocidad igual a 115.200.

Para configurar la conectividad entre el NodeMCU y NODE-RE a través del protocolo MQTT es necesario la instalación de librerías exclusivas para este medio de programación, la librería necesaria es PubSubClient la cual hace que el módulo trabaje como cliente bajo la suscripción de topics para el intercambio de mensajes.

- Ingresar a la sección de Incluir librería > Gestor de librerías y descargar PubSubClient. Como se puede observar en la figura 3.5.

Figura 3.5. Instalación de la librería PubSubClient



Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.3.1 INSTALACION DE NODE-RED

Node-RED es un software que está creado en base de NodeJS y junto con la librería de JavaScriptD3.js. Básicamente se trata de un software destinado a conectar de manera más simple y gráficamente hardware variado con servicios diversos.

El procedimiento para instalación de la node-red empieza con la descarga del instalador figura 3.6.

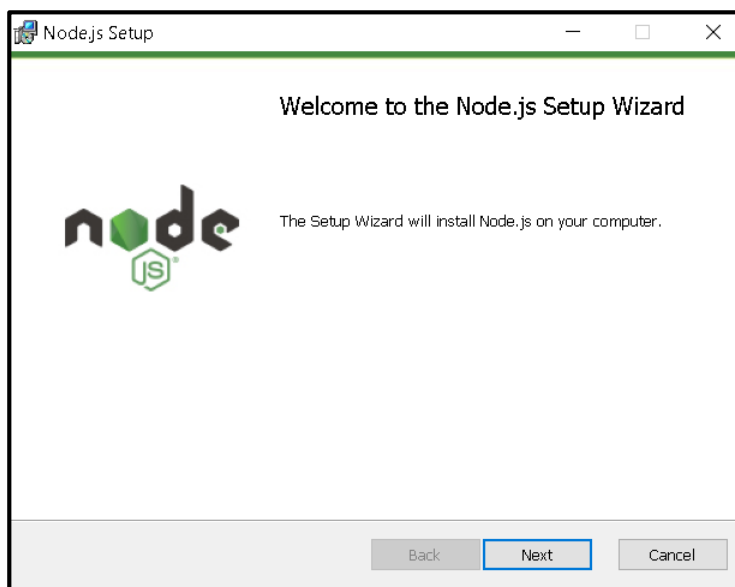
Figura 3.6. Página oficial de NodeJS



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Se despliega un ayudante de instalación de Node.js donde es necesario seguir un procedimiento para la instalación de Node red. El ayudante de instalación se puede visualizar en la figura 3.7.

Figura 3.7. Ayudante de instalación de NodeJS



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Para proceder con la instalación y la descarga de los paquetes npm asociados a node-red ejecutar en el cmd el siguiente comando:

```
npm install -g --unsafe-perm node-red
```

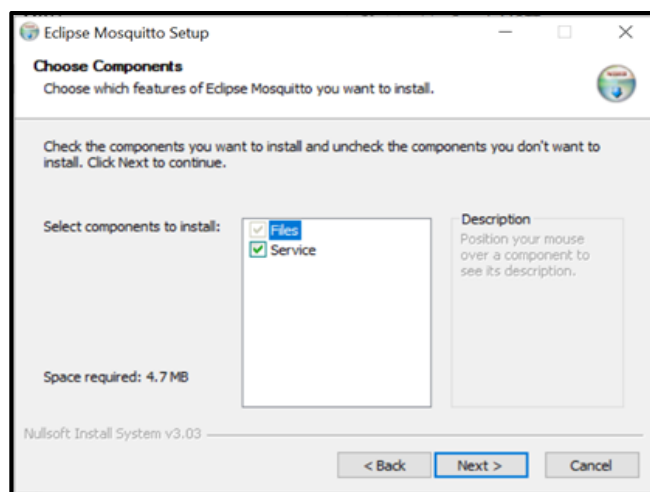
En el terminal de comandos (cmd) escribir el comando `node-red` y se iniciara a correr, los flujos, los nodos y las interfaces, la dirección IP local del equipo está asociada a la siguiente dirección `127.0.0.1:1880` o `localhost:1880`.

-Se tiene que instalar el broker mosquitto.

-En este caso se descarga: `mosquitto-1.6.4-install-windows-x64.exe`

-Se instalará como un servicio de Windows, tal como se muestra en la figura 3.8.

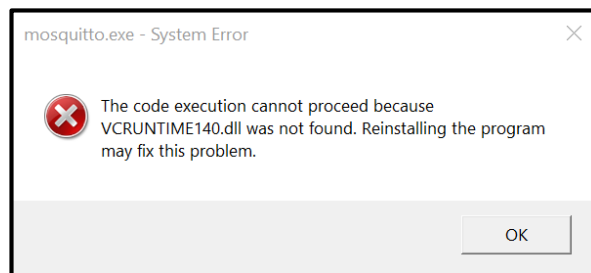
Figura 3.8. Configuración como servicio de Windows



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Durante la instalación de Node Red es posible que se despliegue el siguiente error `VCRUNTIME140.dll` el cual se lo puede observar en la figura 3.9.

Figura 3.9. Error VCRUNTIME140.dll

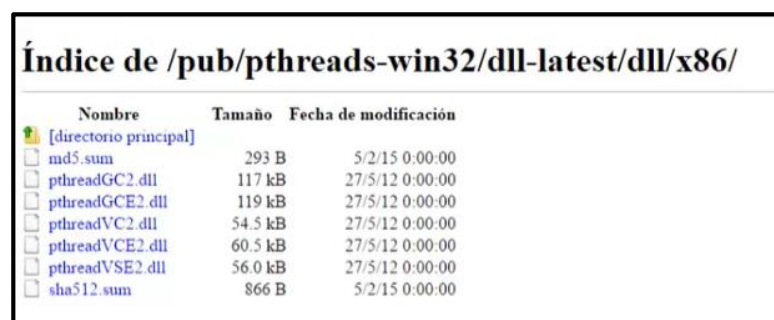


Elaborado por: Bryan Pillajo y Rodrigo Bautista

Es un problema de la librería VCRUNTIME140.dll de Microsoft Visual Studio (VC++) y es necesaria para ejecutar los programas compilados con Microsoft VC++ / Visual Studio. Normalmente los instaladores de software incluyen esta librería en su versión correcta. En cualquier caso si esto último no funciona o no se puede ejecutar la reparación se tendrá que realizar una reinstalación de Microsoft Visual C++.

Muchas veces el bróker necesita ciertas dependencias es decir pequeñas librerías para su correcto funcionamiento uno de ellos es Open ssl para esto debemos descargarlo teniendo en cuenta si el sistema operativo funciona en 64 o 32 bits. Se debe instalar tanto Open ssl y las librerías vistas en la figura 3.10.

Figura 3.10. Instalación de librerías de Open ssl

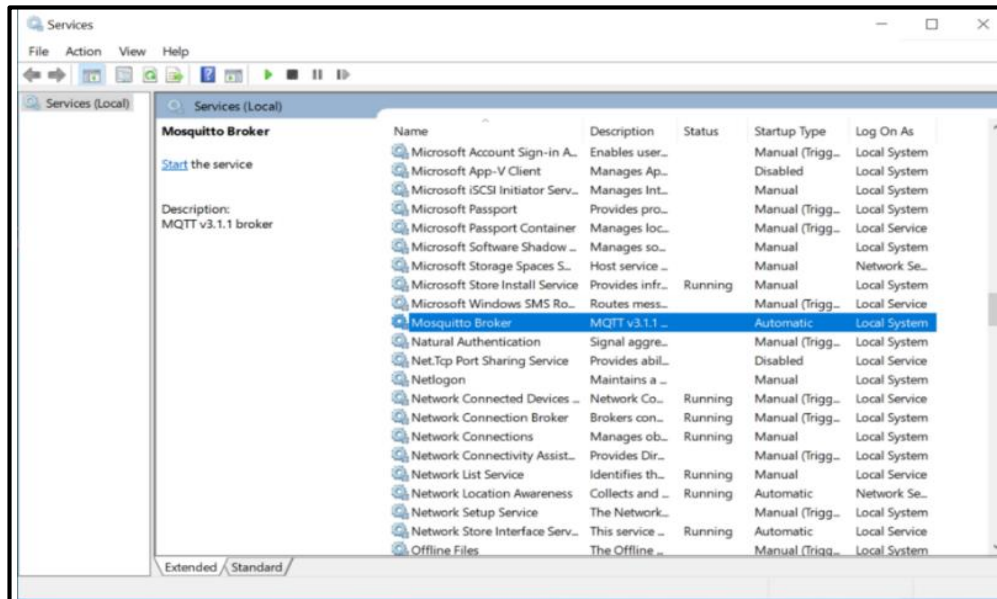


Nombre	Tamaño	Fecha de modificación
[directorio principal]		
md5.sum	293 B	5/2/15 0:00:00
pthreadGC2.dll	117 kB	27/5/12 0:00:00
pthreadGCE2.dll	119 kB	27/5/12 0:00:00
pthreadVC2.dll	54.5 kB	27/5/12 0:00:00
pthreadVCE2.dll	60.5 kB	27/5/12 0:00:00
pthreadVSE2.dll	56.0 kB	27/5/12 0:00:00
sha512.sum	866 B	5/2/15 0:00:00

Elaborado por: Bryan Pillajo y Rodrigo Bautista

Para poder arrancar el servicio abrimos la aplicación Services de Windows.

Figura 3.11. Aplicación Services Windows



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Ya aparecerá running:

Figura 3.12. Activación del servicio

Microsoft Windows SMS Ro...	Routes mess...		Manual (Trigg...	Local Service
Mosquitto Broker	MQTT v3.1.1 ...	Running	Automatic	Local System
Natural Authentication	Signal aggre...		Manual (Trigg...	Local System
Net.Tcp Port Sharing Service	Provides abil...		Disabled	Local Service

Elaborado por: Bryan Pillajo y Rodrigo Bautista

Para comprobar si se está se ejecutará el siguiente comando:

```
netstat -an
```

En esta lista aparecerán las características de escucha del puerto TCP 1883 como se muestra en la figura 3.13.

Figura 3.13. Comprobación de puerto TCP

```
Select Command Prompt
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\admin>netstat -an

Active Connections

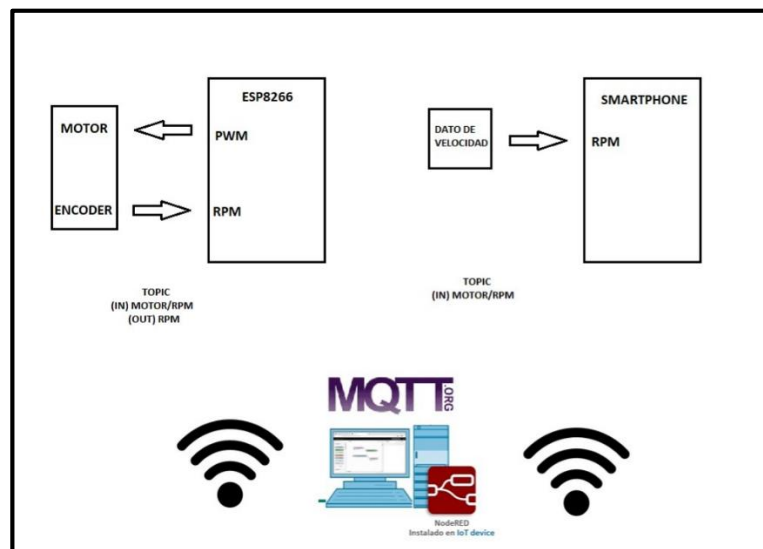
Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135              0.0.0.0:0               LISTENING
TCP   0.0.0.0:445              0.0.0.0:0               LISTENING
TCP   0.0.0.0:1883            0.0.0.0:0               LISTENING
TCP   0.0.0.0:5040            0.0.0.0:0               LISTENING
TCP   0.0.0.0:5357            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49664           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49665           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49666           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49667           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49668           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49669           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49670           0.0.0.0:0               LISTENING
TCP   192.168.10.224:139     0.0.0.0:0               LISTENING
```

Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.3.2 DIAGRAMA DE BLOQUES Y COMUNICACIÓN

La comunicación entre el Bróker o servidor esta generado desde el PC hasta una tarjeta embebida NodeMCU ESP8266 y un Smartphone para lo cual se tiene un respectivo Topic tanto de salida como entrada para la conexión desde Node red como se muestra en la figura 3.14.

Figura 3.14. Diagrama de Bloques para la comunicación del sistema



Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.3.3 IMPLEMENTACION DE LA COMUNICACION MQTT EN LA WNCS

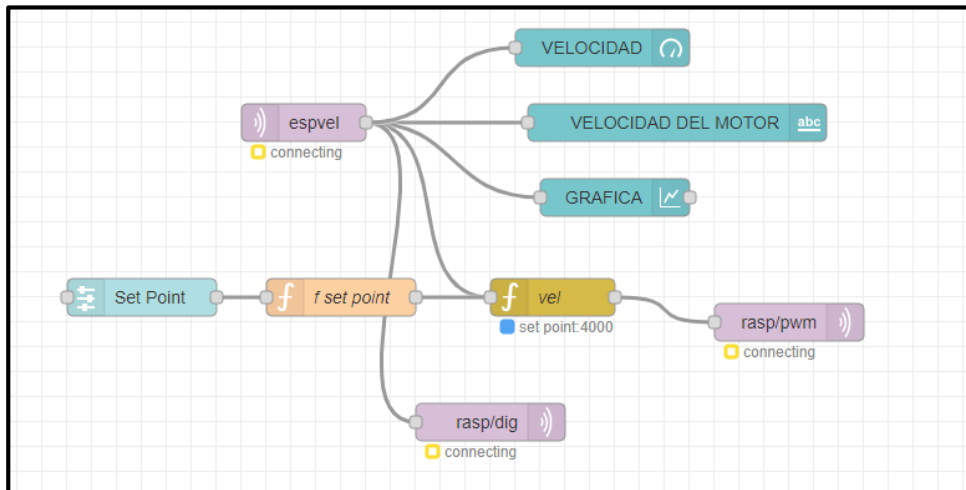
La implementación del protocolo de comunicación MQTT en la red depende de la programación del software Iot (NODE-RED) el cual al ser un editor gráfico de flujo basado en el entorno del navegador se lo puede llamar escribiendo la ip 127.0.0.1:1880 y se desplegará una ventana donde contienen todos los nodos que se encuentran incluidos en esa versión.

Para la interconectividad se utilizará los nodos con las características del protocolo mqtt in que permite escuchar a través de la red el mensaje y mqtt out que su función es enviar a la red a través de la red inalámbrica, estos nodos tienen que ser identificados con el nombre del topic al que representan, en este caso se le nombro como espvel y los dos nodos de salida se les nombro como: rasp/pwm y rasp/dig.

Estos nodos de comunicación inalámbrica permiten enlazar dispositivos (Iot) ya que envían y reciben datos dependiendo del topic al cual se susciben, por lo tanto la salida rasp/pwm enviará datos al ESP8266 que interpretará estos valores y los transformará en una señal PWM. La salida rasp/dig permite enviar los datos de la velocidad hacia el teléfono móvil con la aplicación de realidad aumentada alojada en él. La entrada espvel recibe la señal enviada desde la tarjeta controladora NodeMCU, esta señal es la velocidad del motor en rpm.

El desarrollo del control PID se implementa en NODE-RED gracias a los nodos que tienen la característica de crear funciones, estos nodos permiten ingresar valores y procesarlos antes de ingresar a un nodo con una función prediseñada o que sea enviada a la red, además se importó una librería que viene preconfigurado el controlador PID y que permite el cambio de constantes Kp, Ki, Kd. Finalmente se importó un paquete de nodos que permite la creación de una dashboard que viene con paneles de escritura y lectura de datos los cuales son muy interactivos con el usuario, la programación del algoritmo de control y la conectividad se puede observar en la figura 3.15.

Figura 3.15. Conectividad en Node Red

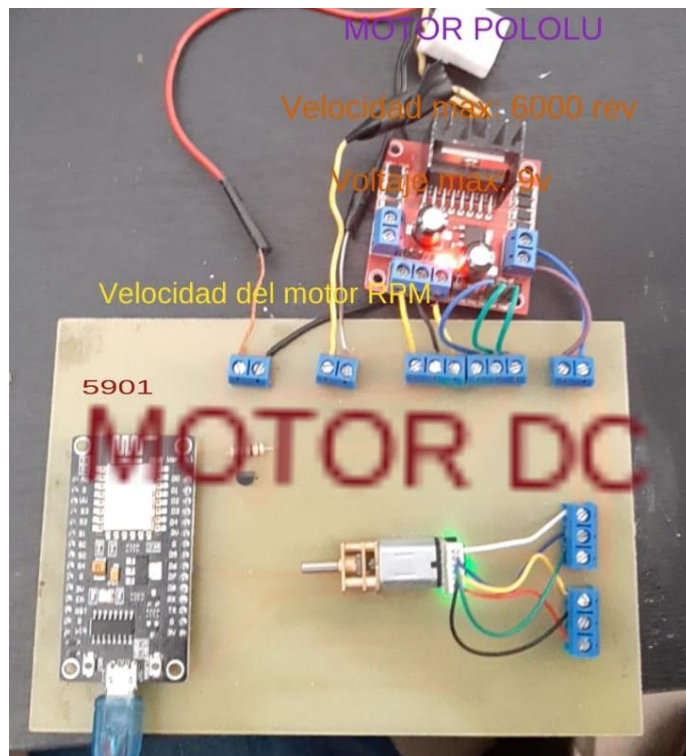


Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.4 REALIDAD AUMENTADA

La realidad aumentada en el presente proyecto tiene la función de una interfaz gráfica que permite al usuario visualizar los datos de velocidad provenientes desde la planta. Estos datos ingresan a la red a través del broker MQTT y es reenviado hacia el dispositivo móvil con un sistema operativo Android. Para la implementación de la realidad aumentada es necesario tomar en cuenta ciertos parámetros que se deben cumplir, primeramente, el reconocimiento de imagen no detecta figuras o imágenes simétricas o muy simples ya que podría generar confusión en el algoritmo y provocaría fallos en la aplicación, el reconocimiento de imagen se lo observa en la figura 3.16, segundo, es indispensable la instalación previa de Android Studio para que el programa Unity pueda hacer uso de los Kits de Desarrollo de Software (SDK) y tercero, Unity es un motor de imagen basado en Scripts que generan una función a los objetos colocados en la interfaz, por lo tanto en este caso se instaló el IDE de Visual Studio, a continuación

Figura 3.16. Reconocimiento de imagen en Unity



Reconocimiento de un patron asimetrico dentro de la imagen. Elaborado por: Bryan Pillajo y Rodrigo Bautista

Unity es un software que permite centralizar todas las herramientas para el diseño y creación de aplicaciones móviles en diferentes plataformas, las cuales deben ser compatibles con los recursos que tiene el dispositivo donde van a ser ejecutadas, por lo tanto, se eligió como dispositivo anfitrión de la APK a un celular con propiedades avanzadas y con suficiente capacidad de procesamiento, en este caso se utilizó un Samsung A50, las características se pueden observar en la tabla 3.1.

Tabla 3.1. Características del celular Samsung A50

Sistema operativo	Android 9.0 Pie
Procesador / MODEM	Exynos 6910 8x
Memoria RAM	4/6
Almacenamiento I/Ex	64/128
Cámara principal	Triple. 25 MP, f / 1.7 + 8 MP, f / 2.2, 12 mm + 5 MP, f / 2.2
WIFI	WI-FI 802.11 a / b / g / n / ac. DB, Direct

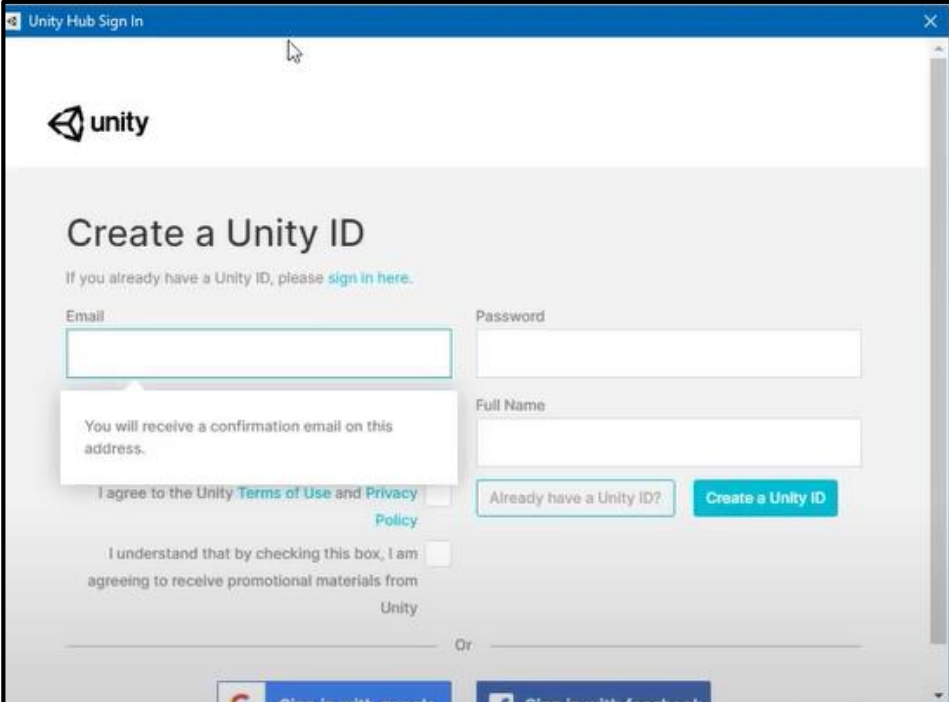
Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.4.1 INSTALACIÓN Y CONFIGURACIÓN DE UNITY

Para la instalación de Unity es necesario seguir los pasos detallados a continuación:

- Ingresar a la página oficial de descargas de Unity, aceptar los términos y descargar el paquete Unity Hub e instalarlo.
- Ingresar a Unity Hub y registrar una cuenta personal que permitirá obtener todos los beneficios que ofrece Unity. Esta cuenta se obtiene llenando todos los datos que se muestran en la figura 3.17. y aceptando los términos y condiciones.

Figura 3.17. Reconocimiento de imagen en Unity

The image shows a screenshot of the Unity Hub Sign In page. The page has a blue header with the Unity logo and the text 'Create a Unity ID'. Below the header, there is a form with several input fields: 'Email', 'Password', and 'Full Name'. There are also checkboxes for 'I agree to the Unity Terms of Use and Privacy Policy' and 'I understand that by checking this box, I am agreeing to receive promotional materials from Unity'. At the bottom of the form, there are two buttons: 'Already have a Unity ID?' and 'Create a Unity ID'. The page is displayed in a browser window titled 'Unity Hub Sign In'.

Registro de cuenta Unity. Elaborado por: Bryan Pillaño y Rodrigo Bautista

- Realizar la activación de licencias de Unity Hub ingresando a manage license y registrándolo para utilización personal y no en una capacidad profesional, tal como se detalla en la figura 3.18.

Figura 3. 18 Reconocimiento de imagen en Unity

New License Activation [X]

License Agreement

Please select one of the options below:

- Unity Personal
- The company or organization I represent earned **less than \$100,000** in gross revenue in the previous fiscal year.
- I don't use Unity in a professional capacity.
- Unity Plus or Pro

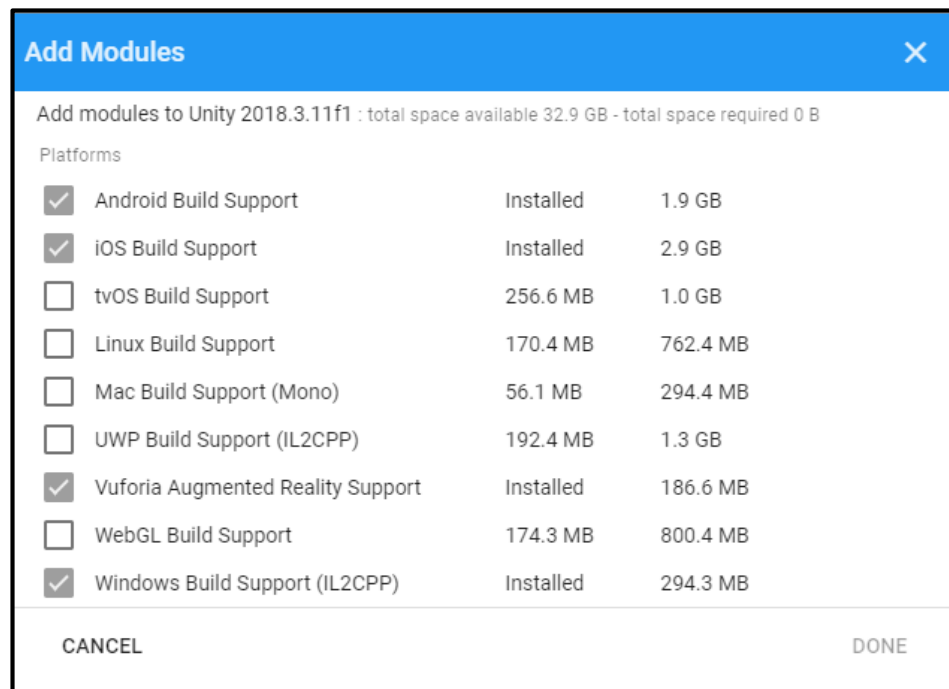
[Buy Professional Edition](#) - [Help](#) - [FAQ](#)

DONE

Activacion de licencias de Unity. Elaborado por: Bryan Pillajo y Rodrigo Bautista

- En la pestaña de installs se elige la versión de Unity con la cual se va a trabajar, por esta ocasión se selecciona la versión 2018.3.11f1 ya que coincide con las características que soporta la instalación de assets desde VUFORIA.
- Para que la funcionalidad del software sea correcta, es necesario la instalación de los módulos que soportaran el desarrollo en diferentes plataformas figura 3.19, caso contrario Unity no permitirá el desarrollo de ningún entorno gráfico, estos módulos son: Android Build Support, iOS Build Support, Vuforia Augmented Reality Support y Windows Build Support (IL2CPP).

Figura 3.29. Instalación de módulos

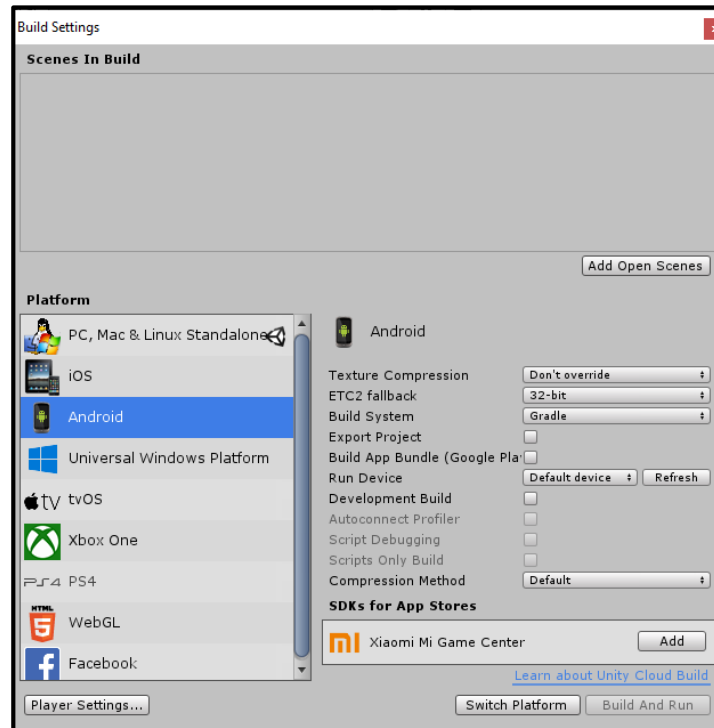


Elaborado por: Bryan Pillajo y Rodrigo Bautista

Concluida la instalación del software Unity es importante realizar ciertas configuraciones internas que permitirán activar las propiedades de los módulos instalados previamente y por consiguiente soportará la importación de las librerías de Vuforia, los pasos para la configuración se detallan en los siguientes ítems.

- Mediante la pestaña File ingresar a la ventana Build Settings la cual permitirá escoger en que plataforma se desarrollara la aplicación, tal como se detalla en la figura 3.20.

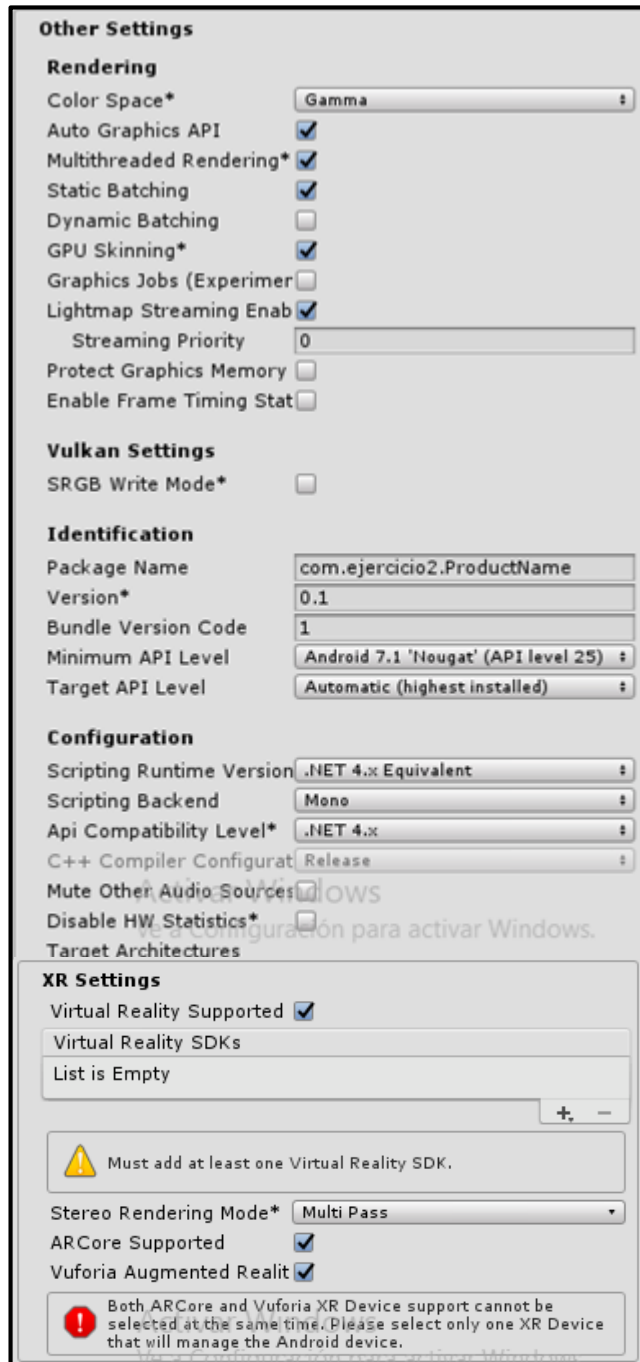
Figura 3.20. Elección de plataforma para la aplicación de AR



Elaborado por: Bryan Pillajo y Rodrigo Bautista

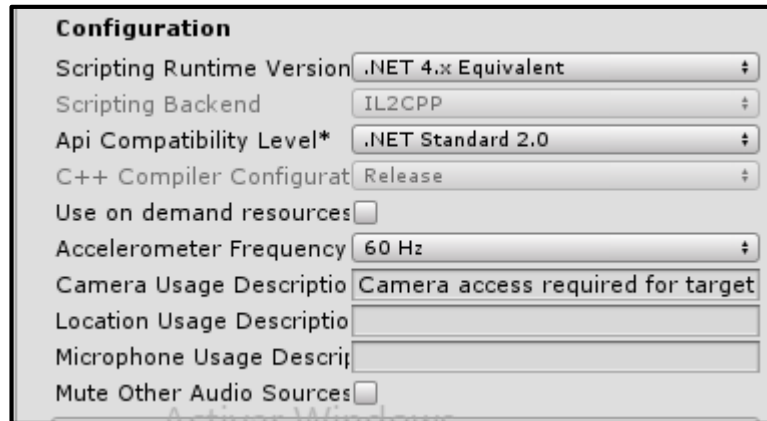
- Como siguiente paso se procede a la configuración de los Player Settings el cual se empezará por la plataforma de Android, iOS y Universal Windows Platform, tal como se observa en la figura 3.21, figura 3.22. y figura 3.23.

Figura 3.21. Configuración de los Player Settings en Android



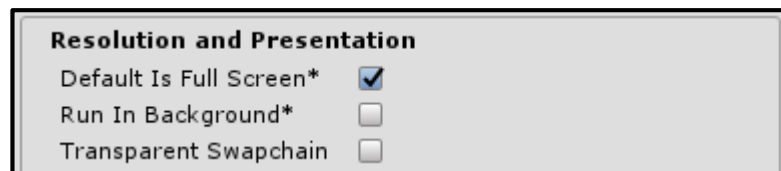
Elaborado por: Bryan Pillajo y Rodrigo Bautista

Figura 3.22. Configuración de los Player Settings en iOS



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Figura 3.23. Configuración de los Player Settings en UWP



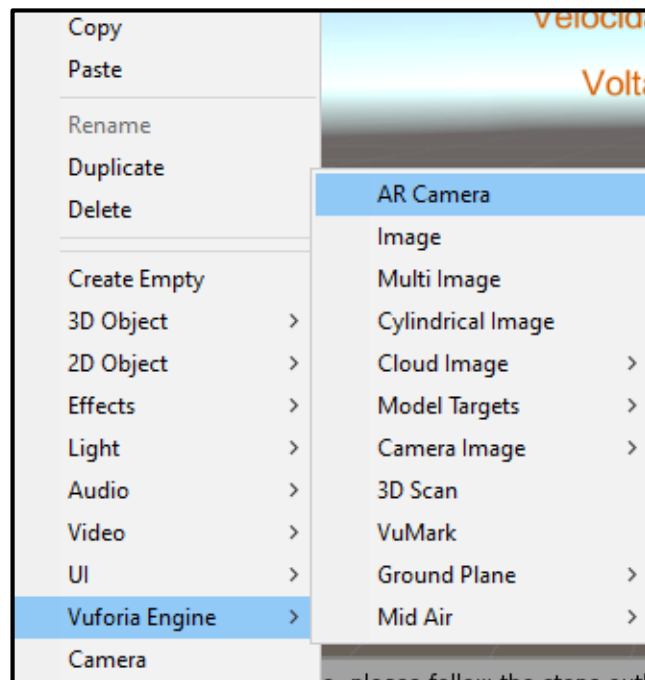
Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.4.2 IMPORTACION DE ASSETS DE VUFORIA

Para que Unity adquiriera la capacidad de desarrollar AR en su plataforma es indispensable la importación de los assets de Vuforia, estos cumplen con la función de una librería y permiten la importación de una imagen que será reconocida durante la ejecución de la aplicación. La instalación de los assets de Vuforia se detalla a continuación:

- En los elementos de la escena de Unity se agrega los elementos de vuforia como AR Camera e Image, tal como se muestra en la figura 3.24.

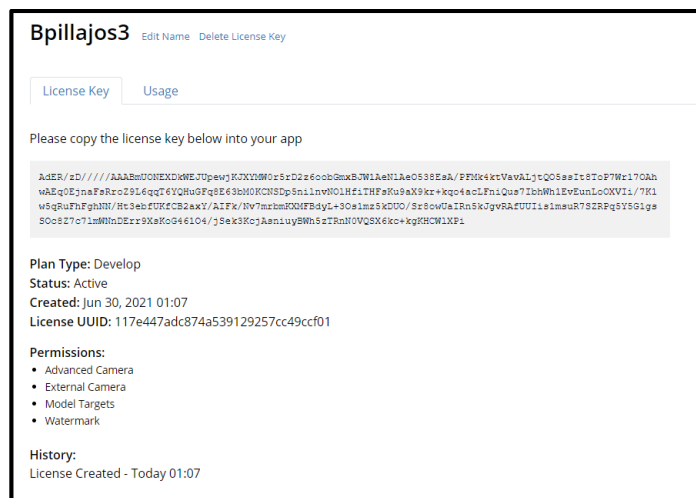
Figura 3. 24 Incorporación de elementos de Vuforia



Selección de elementos como AR Camera e Image Elaborado por: Bryan Pillajo y Rodrigo Bautista

- Como siguiente paso se requiere de la creación de una cuenta en la página oficial de Vuforia para generar una llave encriptada por cada creación que se haga en la página, para lo cual se selecciona la opción Get Development Key para posteriormente llenar los siguientes datos y aceptar los acuerdos y condiciones tal como se detalla en la figura 3.25.

Figura 3.25. Generar llave Vuforia



Elaborado por: Bryan Pillajo y Rodrigo Bautista

- Una vez generada la llave se procede a ingresar en la opción Target Manager el cual permitira subir una imagen a Vuforia y que posteriormente descargaremos el asset generado para el reconocimiento dentro de la aplicación, en este caso se subio una imagen de la tergeta que contiene la plata del sistema y se la puede observar en la figura 3.26.

Figura 3.26. Creación de una base de datos

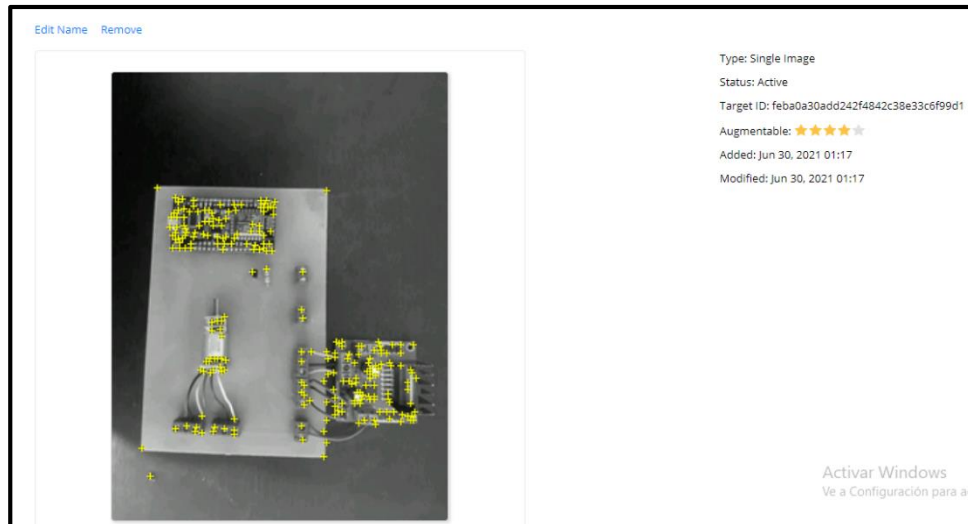
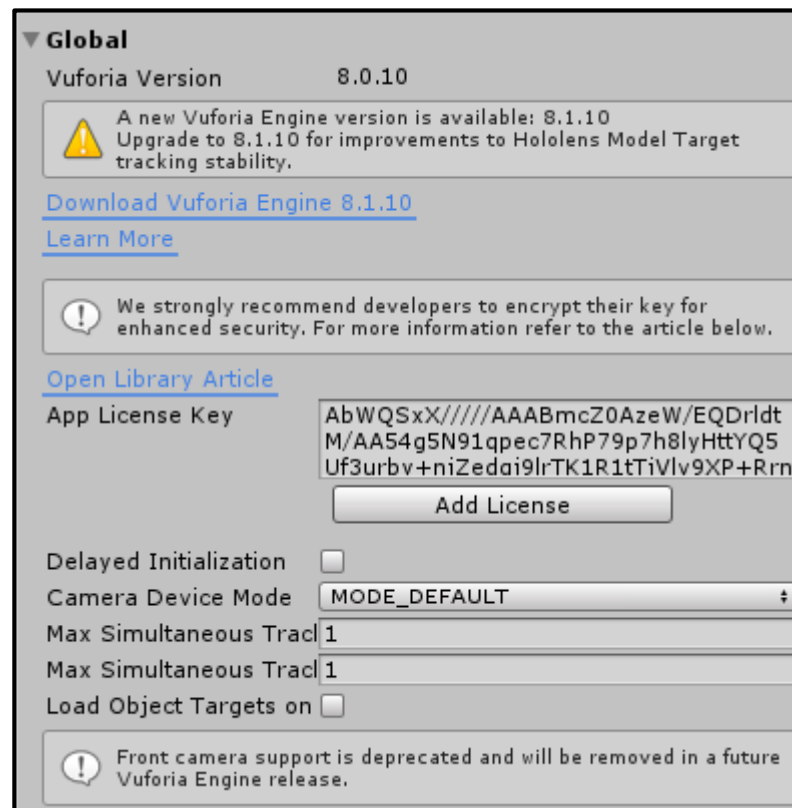


Imagen cargada a la base datos con los patrones de reconocimiento Elaborado por: Bryan Pillajo y Rodrigo Bautista

- Finalmente se descarga la librería que contiene la Image Target con la opción Download Database y se importa desde Unity. La llave anteriormente generada tiene que ser copiada y pegada en las configuraciones de AR Camera dentro de la opción Open Vuforia Engine configuration tal como se observa en la figura 3.27.

Figura 3.27. Activación de AR Camera

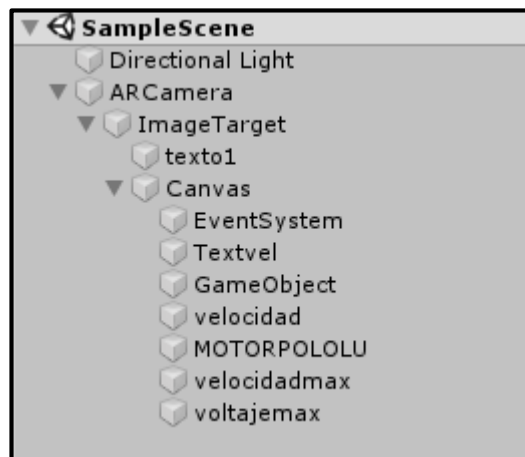


Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.4.3 DISEÑO DE INTERFAZ GRAFICA EN UNITY

Para diseñar la interfaz grafica desde Unity, es importante desarrollar el sentido de orientacion dentro de la escena 3D que nos ofrece el software, para asi modificar las posiciones de Image Target y de AR Camera ya que la camara debe estar enfocando la imagen que se quiere reconocer, la funcion de image target debe estar dentro de AR Camera junto con un texto que identifica el motor como MOTOR 1, tal como se muestra en la figura 3.28.

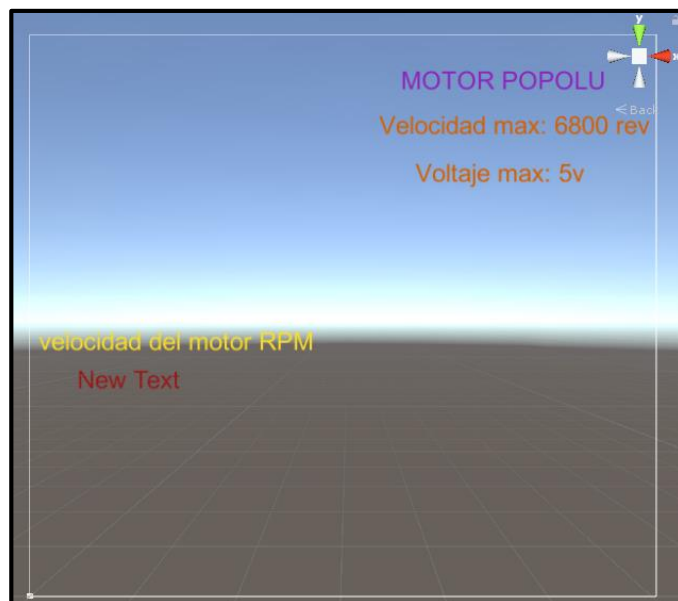
Figura 3.28. Jerarquía de las funciones de AR



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Para la visualización de los datos de velocidad y otro tipo de información adicional sobre el motor se configuró los textos dentro de variables del tipo String que serán activados si y solo si la cámara detecta los patrones memorizados de la imagen pregrabada dentro de Image Target, todo posicionado estratégicamente dentro de un Canvas que simula el espacio físico de la pantalla del dispositivo móvil, el resultado final dentro de la escena de Unity se puede observar en la figura 3.29.

Figura 3.29. Escena Final en Unity

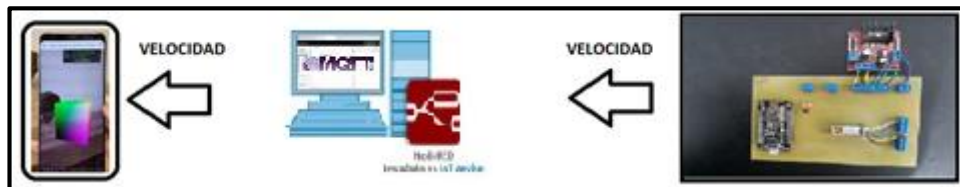


Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.4.4 CONEXION UNITY CON NODE RED

Mediante una conexión inalámbrica se compartirá datos de entrada hacia el dispositivo móvil dentro de la aplicación de realidad aumentada utilizando el protocolo MQTT el cual mediante la programación en lenguaje C# dentro de un Script puede suscribirse y publicar dentro de la red y así en el papel de cliente puede enviar un mensaje utilizando topics o asuntos. El diagrama didáctico de la conexión de la realidad aumentada se la observa en la figura 3.30.

Figura 3.30. Diagrama didáctico de conexión de la realidad aumentada



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Para que la aplicación empiece buscando el broker de la red, es necesario ingresar la ip de la computadora principal y el puerto que se va a utilizar para la conexión, en este caso es el puerto 1880. Estas configuraciones serán ingresadas en el Script que está asociado a una variable del tipo String la cual desplegará el valor en tiempo real de la velocidad.

El Script está conformado por varias partes que son esenciales para la conectividad en el protocolo MQTT, estos parámetros lo describiremos a continuación:

```
client = new MqttClient(IPAddress.Parse("192.168.1.7"), 1883);
```

Crear una instancia del cliente la cual permite enlazarse con la computadora principal o Broker, esta conformada por la ip y el puerto de enlace.

Registro de mensajes recibidos en una variable que cumple la función de acumulador y transforma de formato entero a String e inicia la conexión a la red.

```
client.MqttMsgPublishReceived += client_MqttMsgPublishReceived;
string clientId = Guid.NewGuid().ToString();
client.Connect(clientId);
```

El dispositivo se suscribe al topico rasp/dig el cual nos ayudara a obtener los datos de entrada.

```
client.Subscribe(new string[] { "rasp/dig" }, new byte[] {
    MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE }); }
```

Resepcion de mensajes desde el puerto y conversión de formato a String

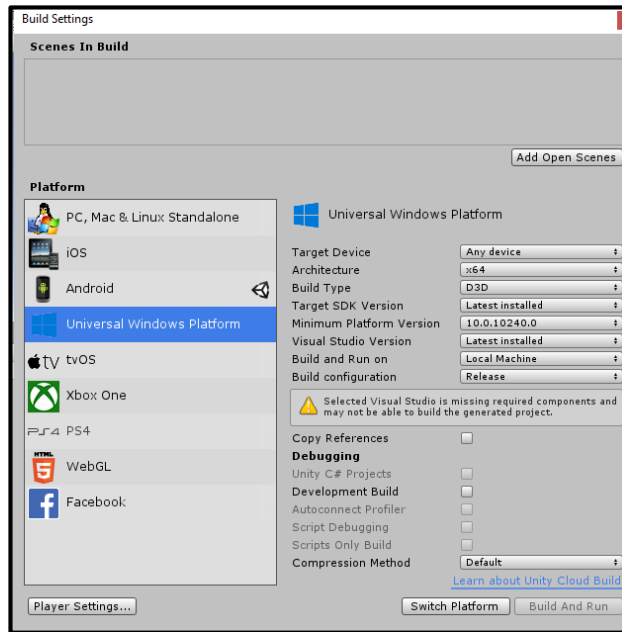
```
void client_MqttMsgPublishReceived(object sender, MqttMsgPublishEventArgs e)
{Debug.Log("Received: " + System.Text.Encoding.UTF8.GetString(e.Message));
dat = System.Text.Encoding.UTF8.GetString(e.Message);
tiempo = dat; }
```

3.4.5 EXPORTACION DEL APK EN ANDROID

Finalmente para la exportacion desde Unity hacia el sistema operativo android es necesario seguir los pasos que se van a detallar a continuación:

- Ingresar a la ventana Build Settings a traves de la pestaña de file y cambiar la plataforma a Android, mediante la opcion Switch Plataforma, tal como se observa en la figura 3.31.

Figura 3.33. Exportación APK a plataforma Android



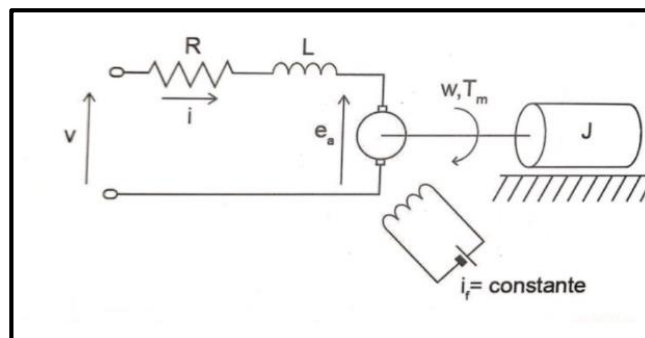
Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.5 REALIDAD VIRTUAL

3.5.1 MODELADO DE MOTOR DC

Para el modelado del motor DC se debe tener en cuenta los elementos que intervienen en el sistema tal como se muestra en la figura 3.32, se modela teniendo en cuenta valores como la R puesta en serie con la inductancia de la bobina a la vez tenemos valores constantes como la tensión.

Figura 3.32. Sistema para modelado de Motor DC



Elaborado por: Bryan Pillajo y Rodrigo Bautista

De este modo se puede hallar las siguientes ecuaciones por medio de un análisis de malla del presente circuito electrónico.

$$v(t) = Ri(t) + l \frac{di(t)}{dt} + E(t)$$

$$L \frac{di(t)}{dt} = v(t) - Ri(t) - E(t) \quad \text{Ec(1)}$$

Se obtiene las demás ecuaciones por la carga mecánica que puede tener el eje del motor al ser manipulado, la resistencia de la parte inductiva da paso a la pérdida de potencia junto con la fricción, histéresis o pérdidas por corrientes de Foucault en el entre hierro del rotor. La ecuación por parte de la mecánica del sistema es la siguiente.

$$T_m(t) = J \frac{dw(t)}{dt} + Bw(t)$$

$$J \frac{dw(t)}{dt} = T_m(t) - Bw(t) \quad \text{Ec(2)}$$

Donde

$T_m(t)$: Torque del motor DC

B: Coeficiente de fricción equivalente al motor de CD (corriente continua) y la carga montada sobre el eje del motor.

J: Momento de inercia total del rotor y de la carga con relación al eje del motor.

$w(t)$: Velocidad angular del motor.

$\frac{dw(t)}{dt}$: Aceleración angular.

Se asume una relación directamente proporcional K_a entre el voltaje del inducido en la armadura y la velocidad angular del motor

$$E(t) = K_a \cdot w(t) \quad \text{Ec(3)}$$

La relación electromecánica que establece que el torque es directamente proporcional a la intensidad de corriente eléctrica del motor DC.

$$T_m(t) = K_m \cdot i(t) \quad \text{Ec(4)}$$

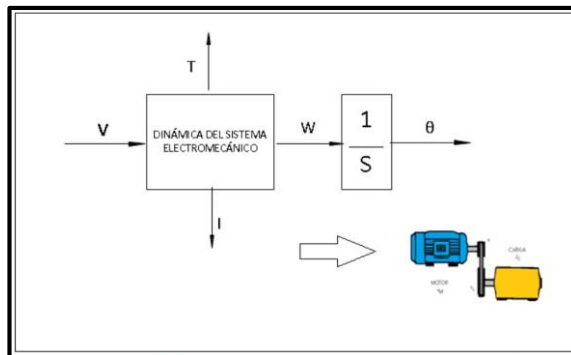
Con todas las ecuaciones anteriormente dichas se aplica la transformada de Laplace y mediante sustitución y cálculos se procede a obtener la función de transferencia la cual permite relacionar el torque del motor con el voltaje de entrada.

$$\frac{T_m(s)}{v(s)} = \frac{K_m(Js + B)}{LJs^2 + (RJ + LB)s + RB + K_m \cdot K_a} \quad \text{Ec(5)}$$

3.5.2 DIAGRAMA DE BLOQUES DEL MOTOR DC

Se procede al diseño en Simulink del motor DC mediante las ecuaciones diferenciales antes vistas para esto se utiliza varios bloques propios de simulink teniendo en cuenta las constantes, las entradas y salidas. Para este caso se tiene como entrada del sistema el voltaje y como salida la corriente, la velocidad angular y la posición angular. El esquema didáctico de como el comportamiento de un motor DC puede ser bien representado en un modelo matemático se encuentra en la figura 3.33.

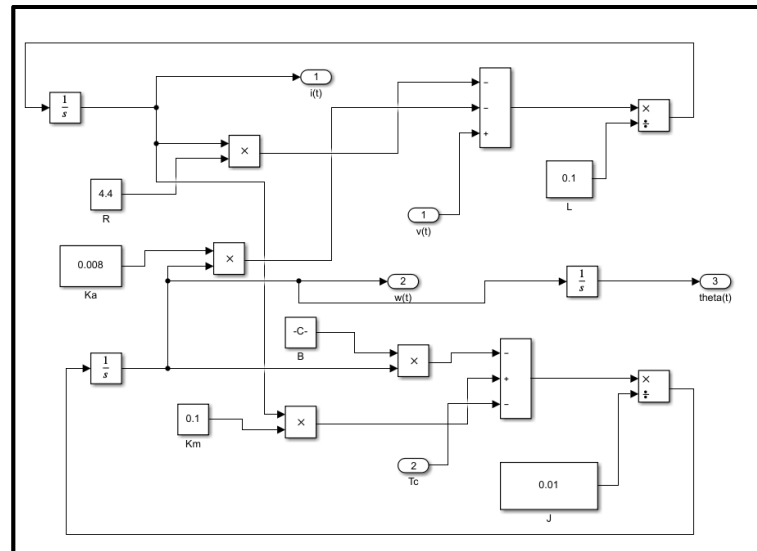
Figura 3.33. Parámetros de diseño de Dinámica del motor



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Siendo este un modelo matemático para un motor DC se puede variar sus parámetros adaptándose a cualquier tipo de motor, tal como se muestra en la figura 3.34.

Figura 3.34. Diseño en Simulink de la Dinámica del motor



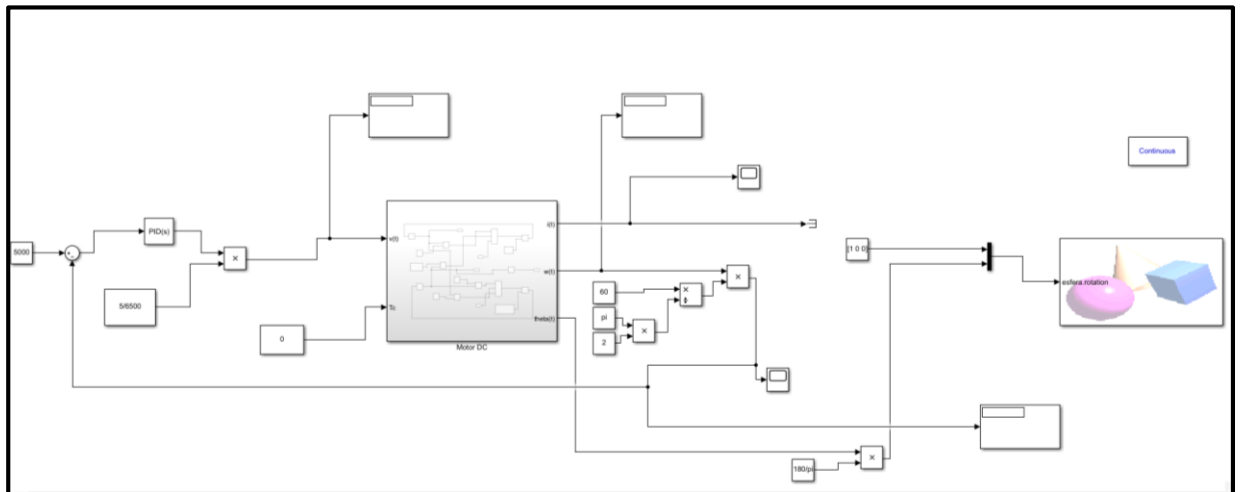
Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.5.3 DISEÑO PID PARA PLANTA

El sistema consta del bloque Motor DC el cual es un subsistema, el bloque PID con las constantes K_p , K_i , K_d previamente establecidas con los valores el sistema esta realimentado para cumplir con el control. Se tiene un setpoint para marcar la velocidad que se quiera obtener.

La salida de velocidad que se obtiene en el sistema está dada en rad/s por lo cual se debe ajustar a rpm. Se tiene el bloque VR SKIN el cual tiene el bloque creado del motor DC en 3D como se muestra en la figura 3.35.

Figura 3.35. Diseño de la planta en simulink con PID.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.5.4 INSTALACION DE LIBRERÍA TRUE TIME

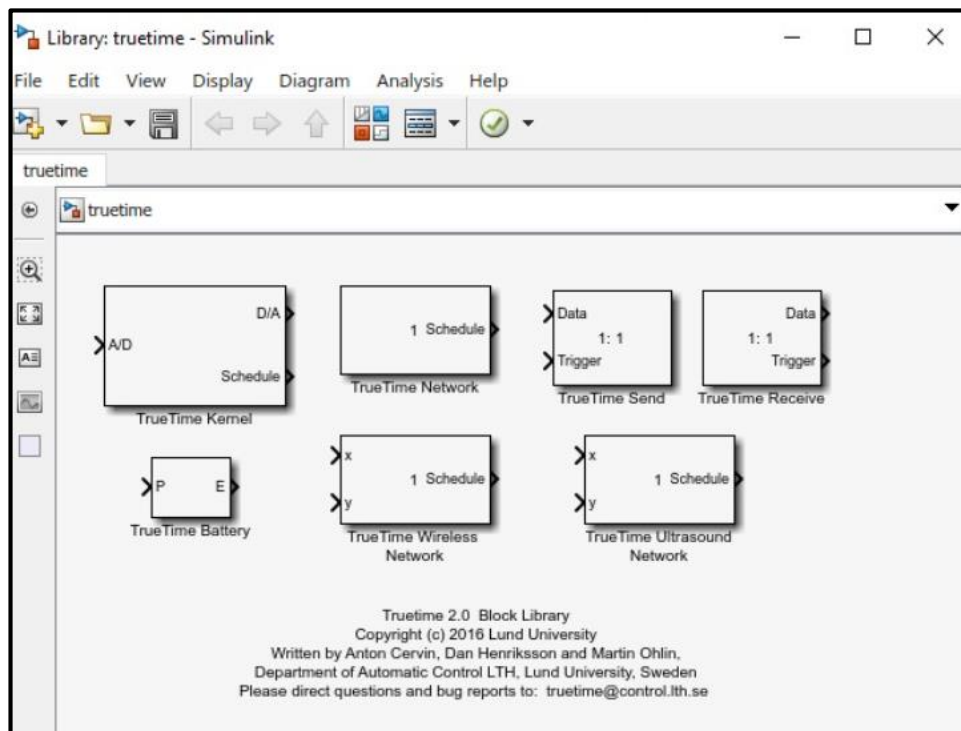
La librería true time es un simulador basado en Matlab para sistemas de control en tiempo real facilitando la ejecución de tareas de controlador, transmisión de la red y dinámica de la planta en forma continua, consta de simulación temporizada, bloques de red, Dispositivos a batería, escalado dinámico de voltaje y relojes locales.

Para proceder con la instalación se debe descargar la librería TRUE TIME 2.0 luego se procede abrir Matlab para inicializar la librería con el comando:

```
Init_truetime
Truetime
```

Al completar este proceso la librería es desplegada en los bloques de Simulink figura 3.36. El cual contiene muchas funciones las cuales ayuda a la simulación de una red con todas sus características incluido los nodos de entrada y salida.

Figura 3.36. Librería Truetime instalada en simulink.



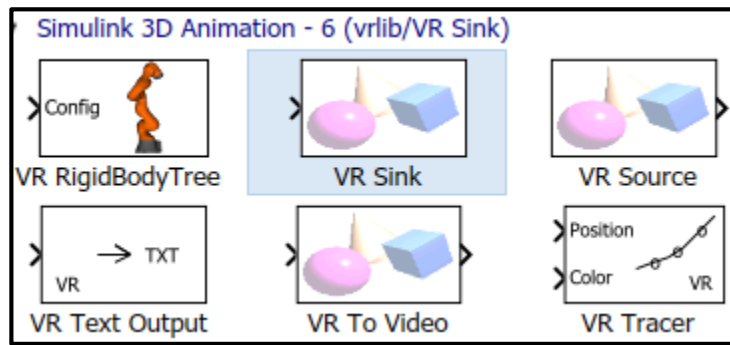
Elaborado por: Bryan Pillajo y Rodrigo Bautista

El sistema se encuentra conformado por dos partes, una de control y otra por el modelo matemático del motor DC, las cuales se conectarán por medio de los nodos TrueTime Send y TrueTime Receive, asociados a una red de que será identificada por el bloque TrueTime Network configurado con una identificación única en el sistema, el programa de simulación se encuentra en la figura 3.37.

3.5.5 DISEÑO DE MOTOR 3D

En las librerías de Simulink se encuentran los bloques Simulink 3D Animation como se muestra en la figura 3.39.

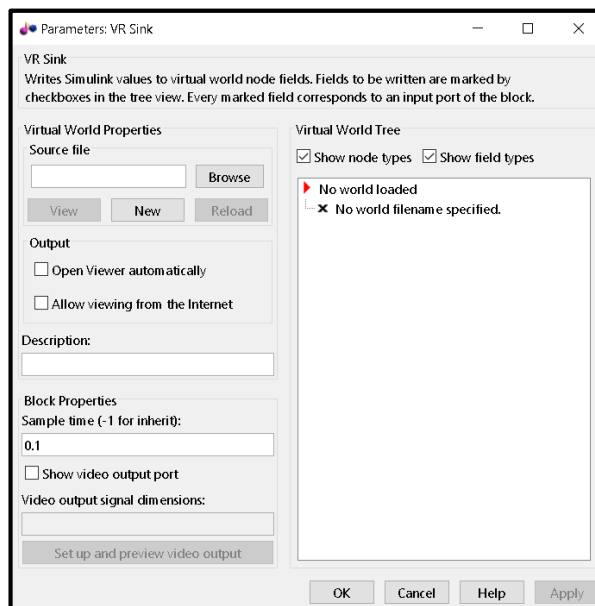
Figura 3.39. Librería simulink 3D.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

El bloque Vr Sink es el que permite acceder a el programa V-realm builder 2.0 para lo cual se procede a crear un proyecto nuevo.

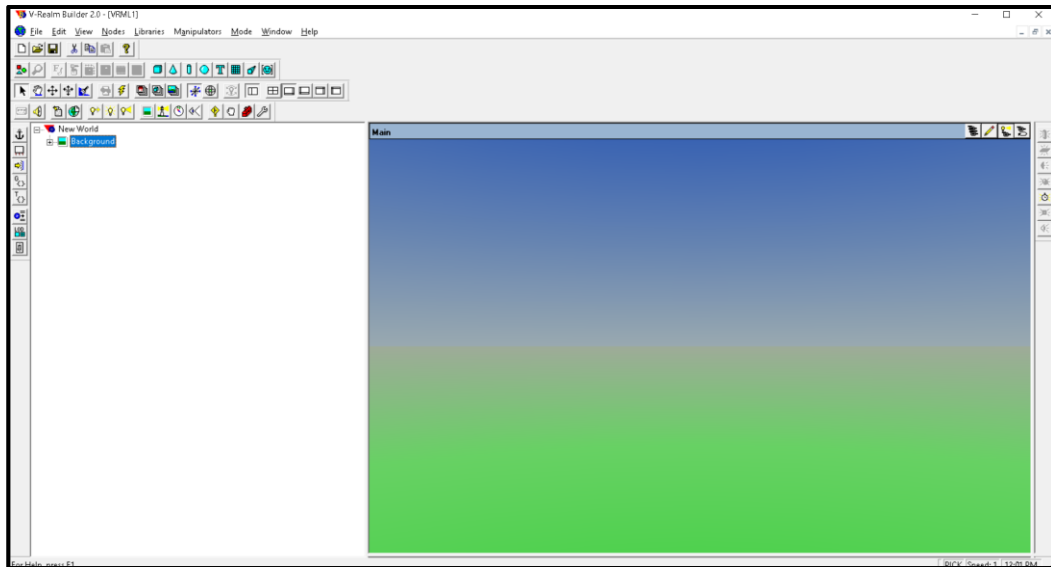
Figura 3.40. Asistente de VR Sink.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Al iniciar un nuevo proyecto se va desplegar una ventana como se observa en la figura 3.41. El cual contiene todas las herramientas de diseño 3D.

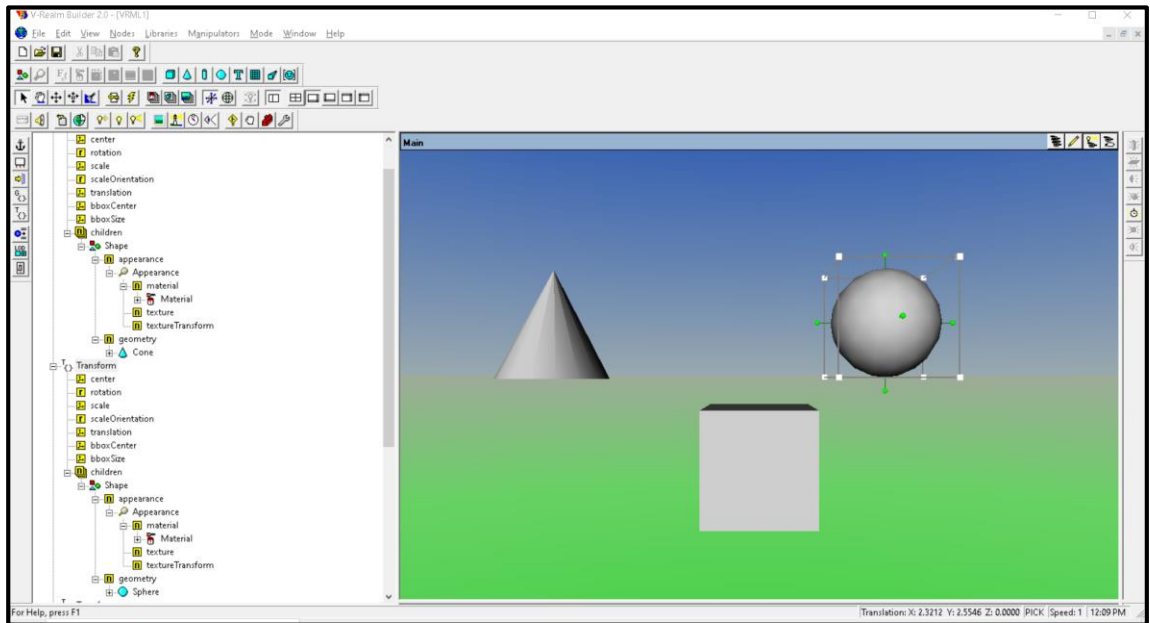
Figura 3.41. Ventana New Word en 3D.



Elaborado por: Bryan Pillajo y Rodrigo Bautista.

El programa V-Realm Builder es una herramienta que permite visualizar sistemas dinámicos por medio de una escena en 3D mediante una interfaz gráfica la cual nos permite manejar de manera sencilla sus características y utilizando generadores de señales de simulink es posible hacer el control de varios parámetros como la posición dimensiones y orientación de los bloques generados así pudiendo diseñar una animación de cualquier sistema que se desee implementar en este caso un motor DC. Como se puede observar en la figura 3.42. Cada elemento posee sus características y parámetros para ser modificado así como texturas, colores y más.

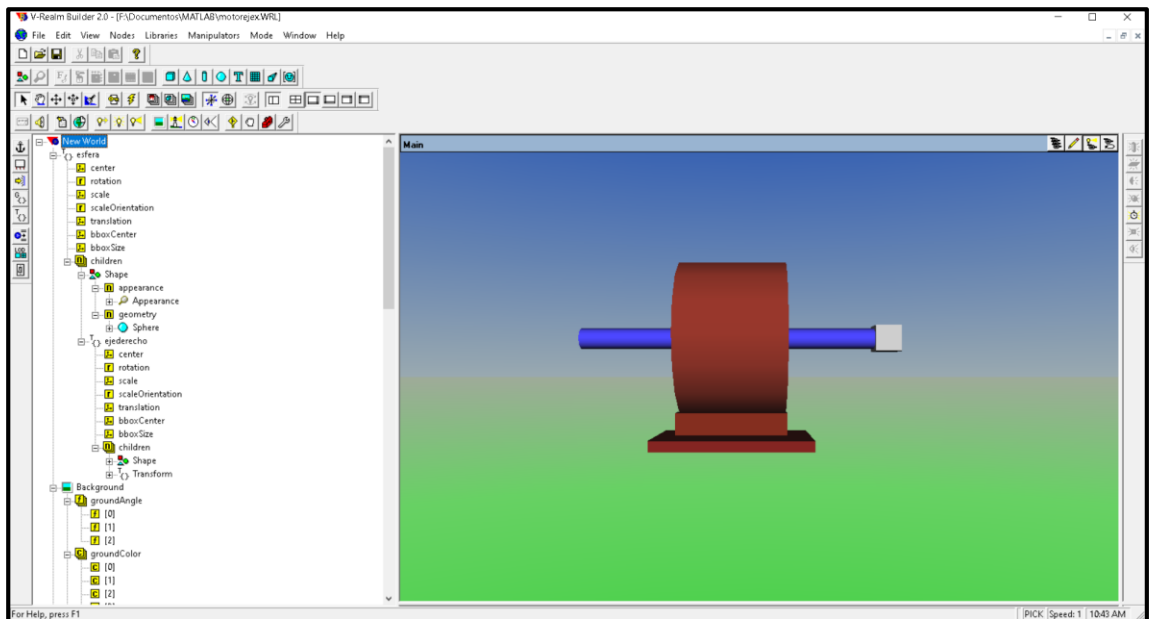
Figura 3.42. Figuras básicas en V Realm Builder 2.0.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

El motor en 3D va ser construido con figuras básicas como se puede observar siendo los principales elementos esferas, cubos, cilindros y más para que en conjunción formen un cuerpo similar al de un motor figura 3.43.

Figura 3.43. Diseño del motor DC en 3D.

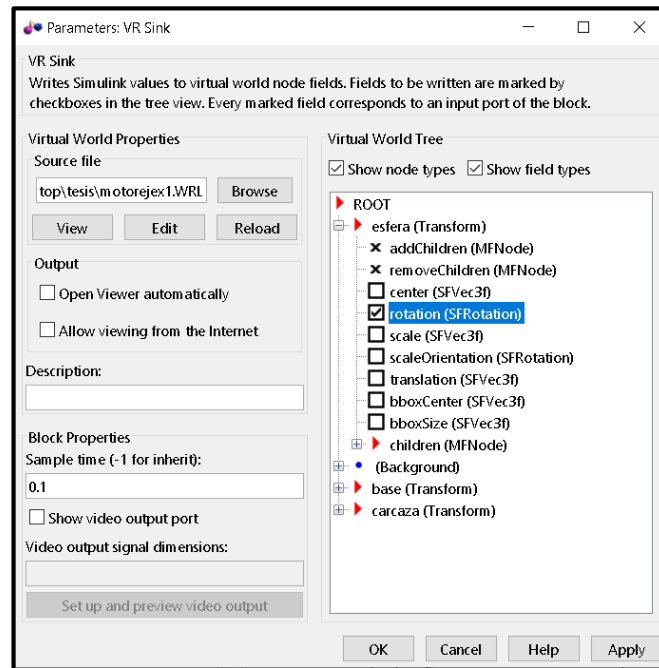


Elaborado por: Bryan Pillajo y Rodrigo Bautista

3.5.6 CONEXIÓN SIMULINK CON VR REALM BUILDER

Al tener el cuerpo del motor ya construido se procede a comunicar con una variable desde simulink como se ve en la figura 3.44. Para este caso la variable a controlar será la rotación del eje del motor.

Figura 3.44. Exporta variable de rotación para el eje del motor DC.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

De esta manera se obtiene el bloque VR sink ya con el parámetro a controlar como se puede observar en la figura 3.45. Se procede a conectar el sistema con el bloque de la animación 3D para su correcto funcionamiento.

Figura 3.45. Bloque VR sink con la variable de entrada a controlar.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

CAPÍTULO 4

PRUEBAS Y RESULTADOS

El capítulo presente se detalla todas las pruebas que de manera sistemática se realizaron para obtener los datos de cada parte del proyecto, las cuales consisten en el control PID de velocidad de un motor DC, comunicada inalámbricamente por medio de Wifi hacia los otros dispositivos Iot como el bróker de la red y un dispositivo móvil para la visualización de datos mediante realidad aumentada enlazados por el protocolo de comunicación MQTT.

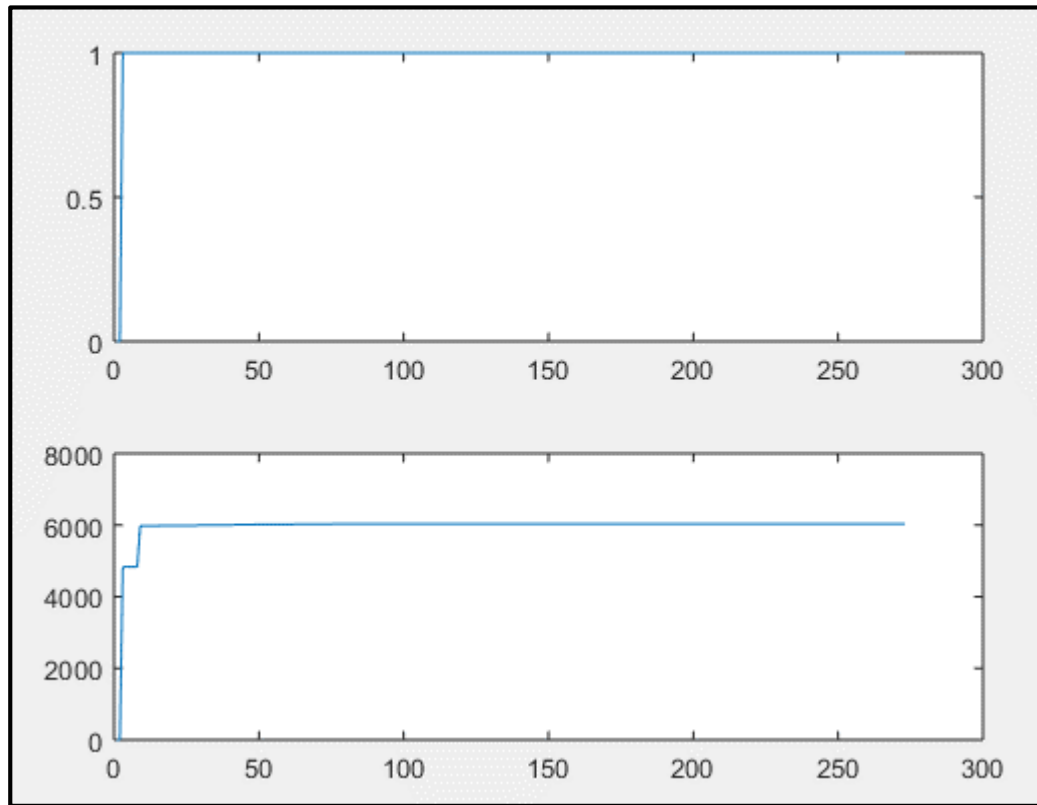
4.1 FUNCIÓN DE TRANSFERENCIA DE LA PLANTA

En este proyecto se obtiene de manera experimental a través de pruebas de funcionamiento el comportamiento de la planta que contiene un motor DC de 9[V] que es controlado por una señal PWM proveniente del controlador ESP8266, con la finalidad de adquirir los datos de velocidad a lo largo del tiempo, además, a través de herramientas de software matemático se puede generar una función de transferencia, el cual proporciona la respuesta de la planta ante distintas variaciones de entrada.

Para que este sistema se ponga en marcha y empiece la adquisición de datos, es necesario la conectividad del respectivo puerto serial en el IDE de Arduino y utilizando la planta, accionar el motor con una señal PWM de 255 adicionalmente por medio de un sensor de velocidad (encoder) se obtiene una curva de salida que caracteriza el comportamiento de la planta en revoluciones por minuto [rpm].

Para adquirir la curva característica de la señal de velocidad, se obtienen los datos reflejados en el monitor serie cada 40 [ms], estos datos son guardados en una hoja de cálculo en Excel y se importa al software matemático MATLAB, este programa posee varias herramientas que permiten realizar el tratamiento de la señal y para este caso se utiliza la herramienta IDENT que admite la curva del comportamiento de velocidad junto con una señal escalón, es decir una señal de 0 a 1 lógico, para obtener la función de transferencia. Las curvas de datos y señal escalón se puede observar en la figura 4.1.

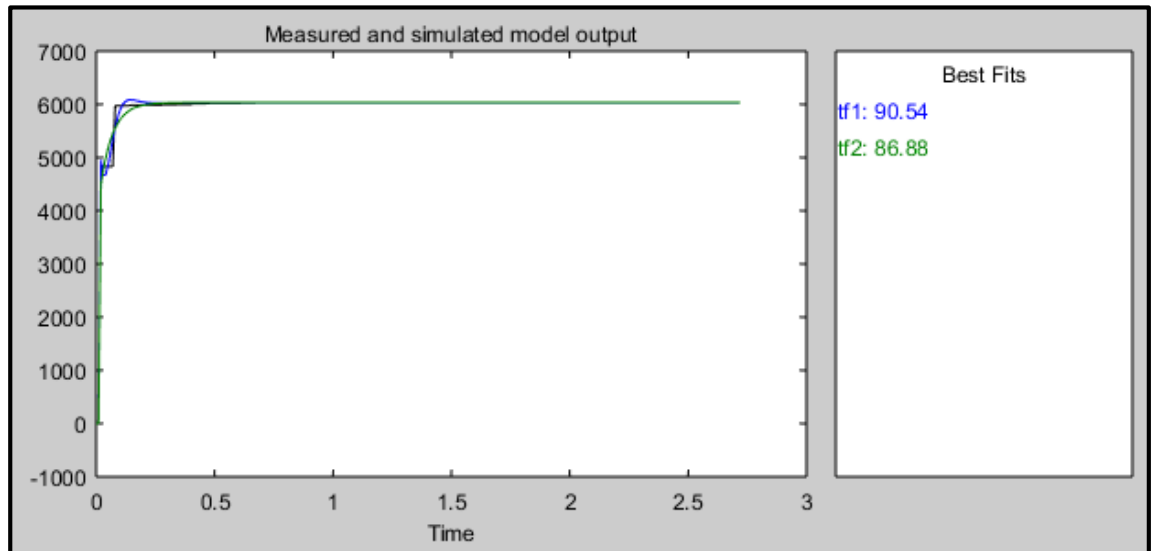
Figura 4.1. Datos y señal de entrada



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Utilizando la herramienta IDENT de Matlab al ser importados los valores de velocidad en el dominio del tiempo y estimando la curva con la opción “Transfer Function Models” se generó dos funciones de transferencia (tf1) y (tf2) utilizando en la primera dos polos y dos ceros, dando como resultado 90.54% de similitud con los datos obtenidos, por lo contrario en la función (tf2) se utilizó 1 polo y 1 cero, dando como resultado 86.88% de similitud con los datos de la planta tal como se observa en la figura 4.2.

Figura 4.2. Funciones de transferencia Motor DC



Elaborado por: Bryan Pillajo y Rodrigo Bautista

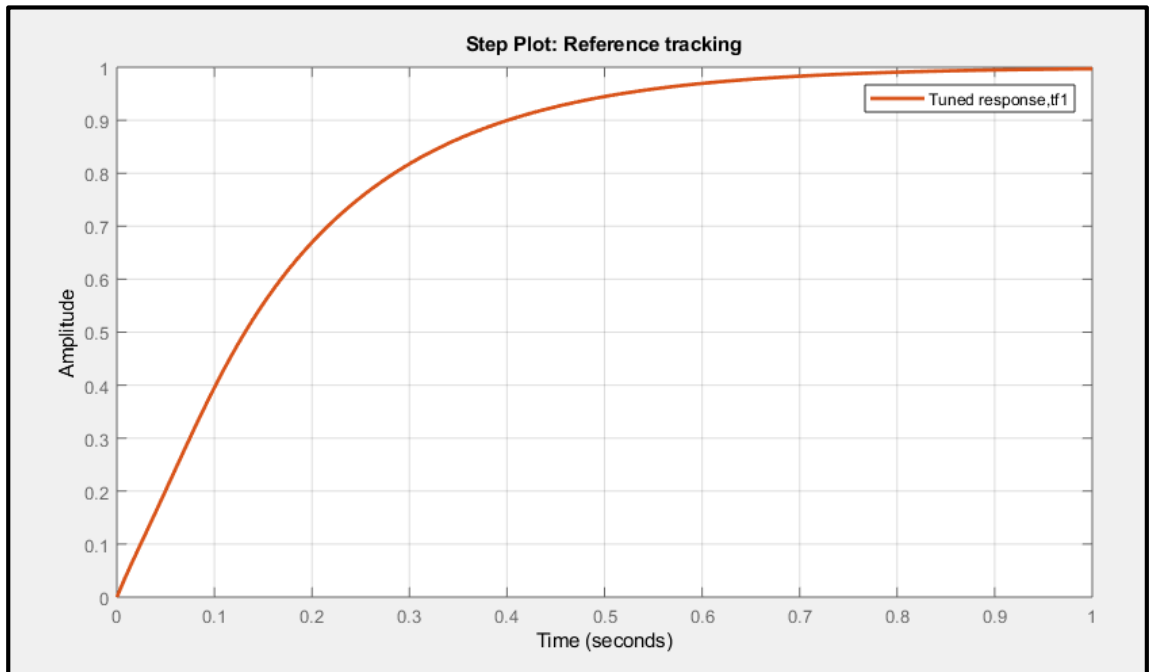
$$tf1 = \frac{4956s^2 + 2.364e^5s + 9.858e^6}{s^2 + 57.46s + 1635} \quad Ec(6)$$

$$tf2 = \frac{4499s + 1.165e^5}{s + 19.31} \quad Ec(7)$$

4.2 PID CONTROL DE VELOCIDAD

Para realizar el control PID se generó los valores característicos de este algoritmo de control utilizando la función de transferencia a través del Toolbox de Matlab PIDTOOL, el cual permite ajustar los valores de K_p , K_i y K_d de acuerdo a las necesidades del usuario tal como se muestra en la figura 4.3.

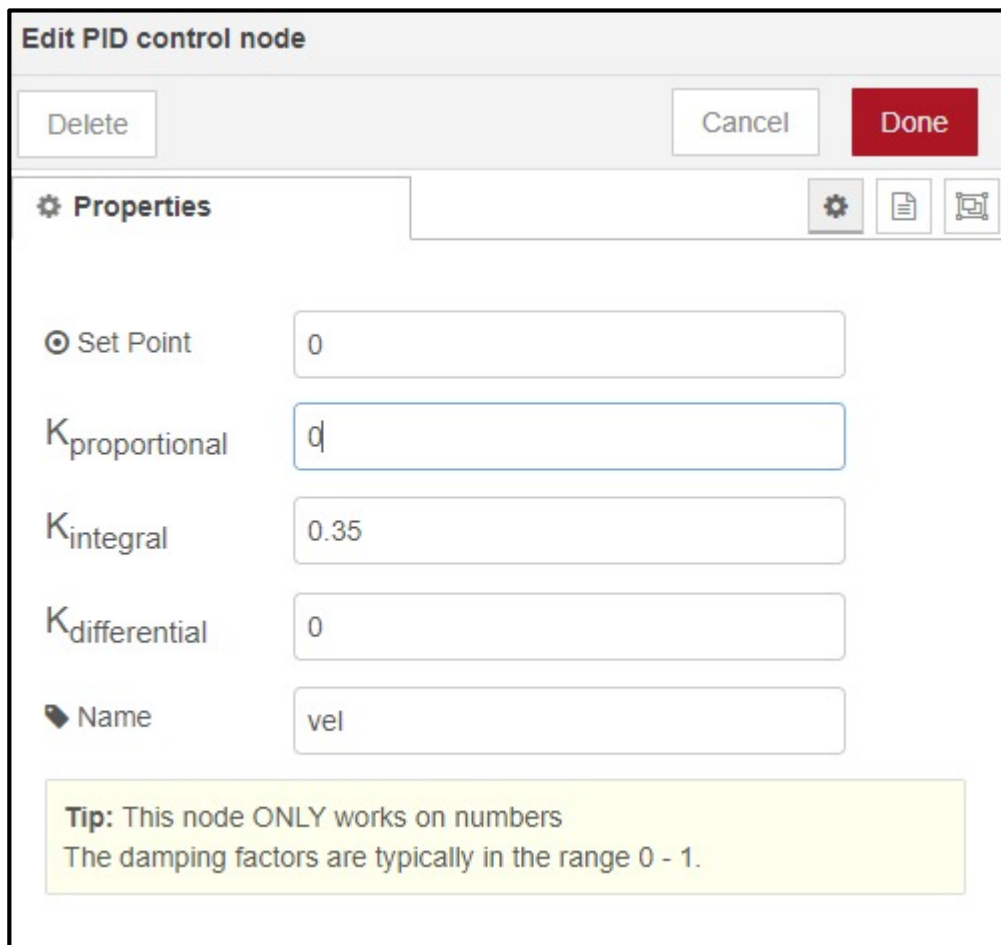
Figura 4.1. PID Tunner Matlab para Planta de Velocidad (Motor DC)



Elaborado por: Bryan Pillajo y Rodrigo Bautista

El algoritmo de control PID de la planta será integrado a la red utilizando una librería de NODE-RED que se llama “node-red-node-pidcontrol” que será importado a la programación visual como un nodo con una entrada y una salida, tal como se muestra en la figura 4.4. Este nodo recibe los datos de velocidad enviados desde la planta y adquiridos por el nodo mqtt con el topic “espvel”, procediendo así con el envío de una respuesta adecuada hacia el ESP8266 enlazado con el nodo mqtt con el topic “rasp/pwm”.

Figura 4.4. Nodo de algoritmo de control PID



Edit PID control node

Delete Cancel Done

⚙️ Properties

Ⓐ Set Point 0

$K_{\text{proportional}}$ 0

K_{integral} 0.35

$K_{\text{differential}}$ 0

🔑 Name vel

Tip: This node ONLY works on numbers
The damping factors are typically in the range 0 - 1.

Elaborado por: Bryan Pillajo y Rodrigo Bautista

En el control de velocidad PID se establecio los valores predeterminados por la herramienta de matlab $K_p= 0$, $K_i= 0.35$, $K_d= 0$. Los cuales al ser incorporados al algoritmo se observo que el comportamiento de la planta es inestable (figura 4.5.) por lo que despues de una etapa de experimentación se logra estabilizar el sistema utilizando los valores $K_p= 0.45$, $K_i=0.0130$, $K_d=0.075$ como se observa en la figura 4.6. utilizando un setpoint de 5400 [rpm].

Figura 4.5. Gráfica de comportamiento inestable.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

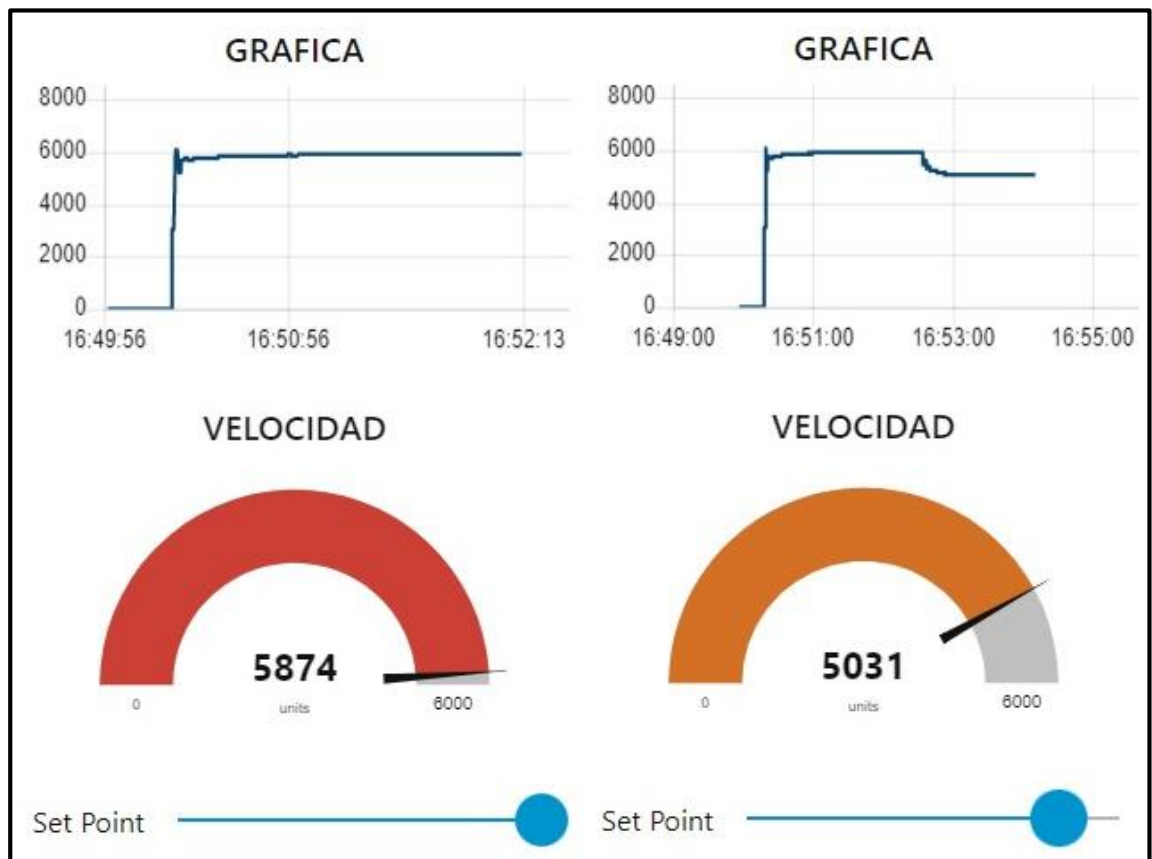
Figura 4.6. Gráfica estabilizada y sintonizada.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Se prueba el funcionamiento eficiente del control PID, colocando dos set point diferentes para visualizar el comportamiento del cambio de valor de entrada, el primer set point es 5874[rpm] y el segundo en ingresar despues de alcanzar la sintonización es 5031[rpm], el cual, como se puede notar en la figura 4.7, la curva de respuesta de la velocidad cambia de estado.

Figura 4.7. Set point de 5874[rpm] y 5031[rpm].



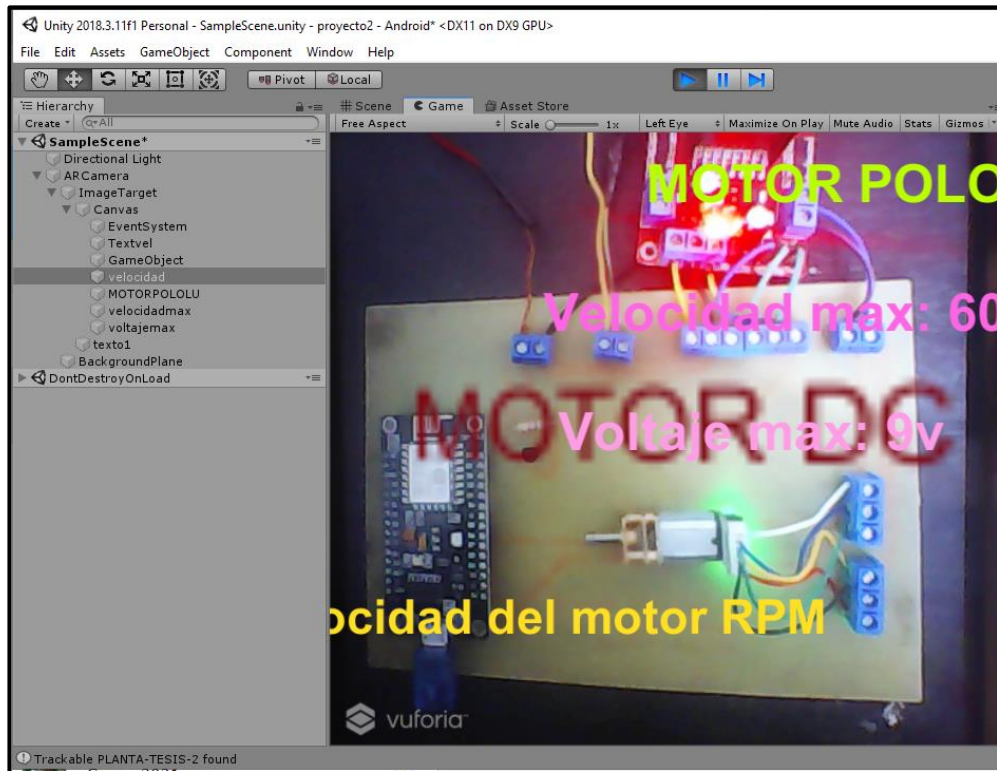
Elaborado por: Bryan Pillajo y Rodrigo Bautista

4.3 REALIDAD AUMENTADA

En la parte de realidad aumentada cumpliendo con el objetivo de realizar una interfaz que permita la visualización de los datos de velocidad enviados desde el bróker de la red (Mosquito), se empieza con el diseño de la dashboard ubicando los textos que serán visualizados en la pantalla del dispositivo Smart, los cuales son la marca del motor, la velocidad máxima, el voltaje máximo, y la velocidad en tiempo real en [rpm].

Para realizar las pruebas de comportamiento de la realidad aumentada se probó la escena dentro del software Unity teniendo como anfitrión una computadora con cámara, la cual tiene la suficiente resolución para reconocer los patrones de la planta y desplegar la información previamente establecida junto con la velocidad del motor en tiempo real, tal como se observa en la figura 4.8.

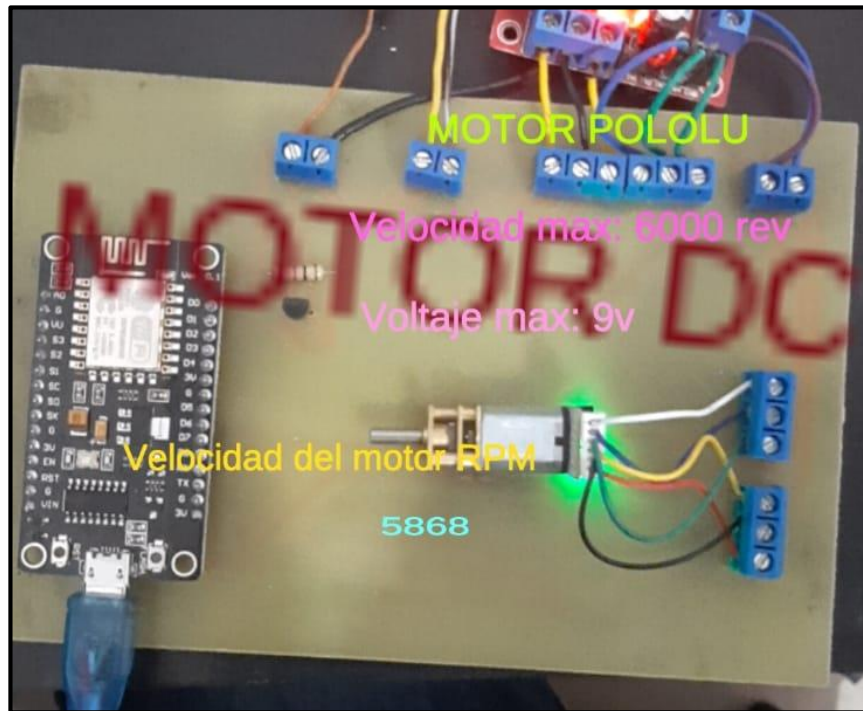
Figura 4.8. Reconocimiento de la planta por la realidad aumentada en una PC



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Como se observa, la aplicación cumple con la función de desplegar los datos del motor solamente si la cámara reconoce los patrones de la planta de velocidad. Por diferencia de dimensiones de pantalla entre la PC y el celular Samsung A50, se ajusta la escala de los mensajes a base de experimentación hasta que la interfaz sea completamente visible para el usuario. (Figura 4.9.)

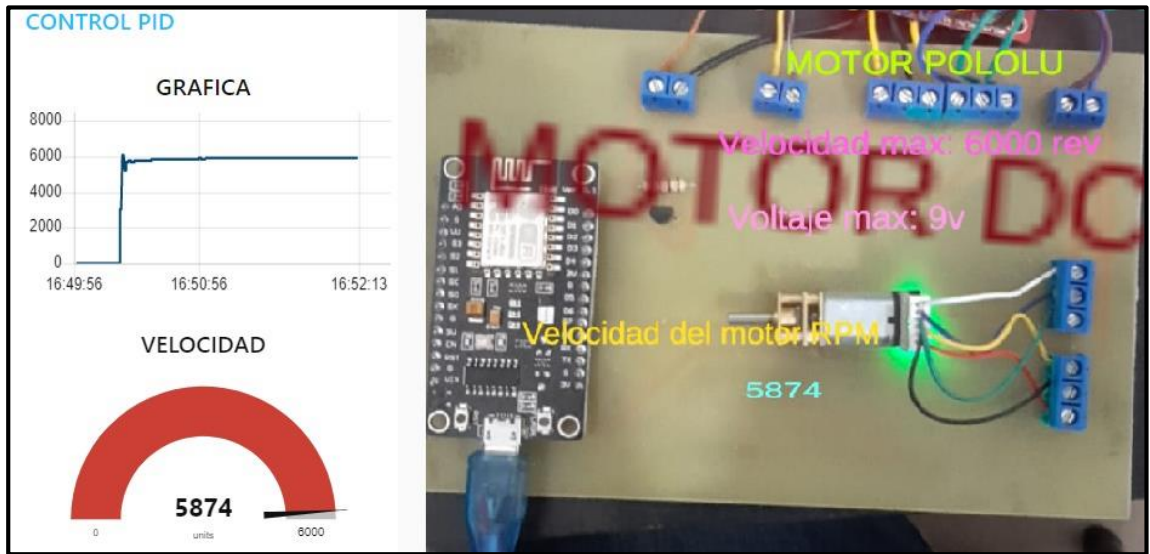
Figura 4.9. Interfaz en el dispositivo Samsung A50



Elaborado por: Bryan Pillajo y Rodrigo Bautista

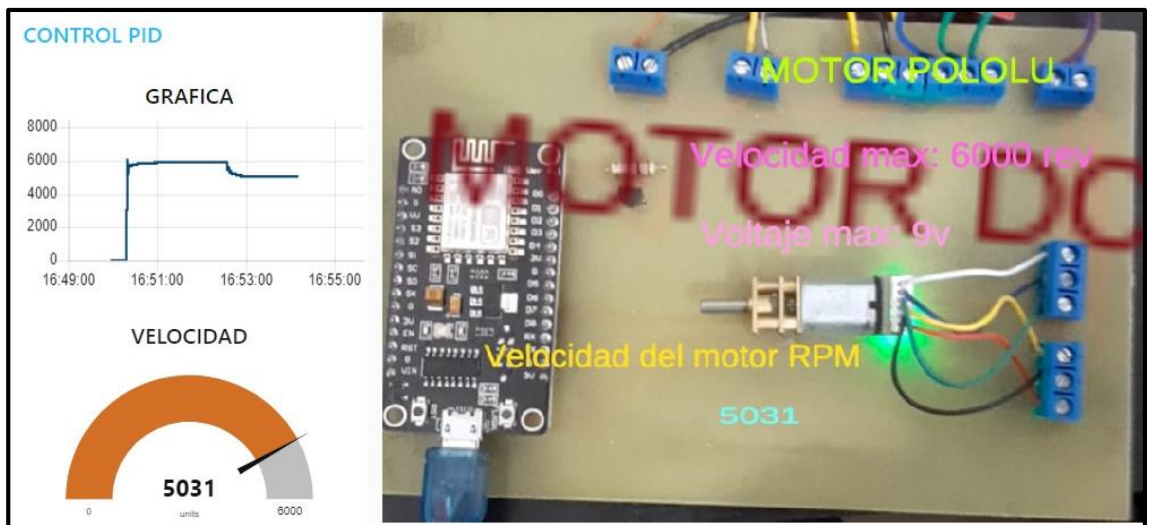
Como prueba de funcionamiento se realiza el cambio de Set point para comparar los valores de salida entre la Dashboard de Node-red y la aplicación de realidad aumentada. Como se observa en la figura 4.10. y figura 4.11, los valores que se pueden visualizar en la interfaz de la computadora son exactamente iguales a los desplegados en la pantalla del dispositivo Smart. Por lo tanto, se ha cumplido con el objetivo planteado anteriormente con respecto a la visualización de los datos de velocidad en tiempo real mediante realidad aumentada.

Figura 4.10. Comparación entre AR y Node-red con Set point = 5874.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

Figura 4.11. Comparación entre AR y Node-red con Set point = 5031.

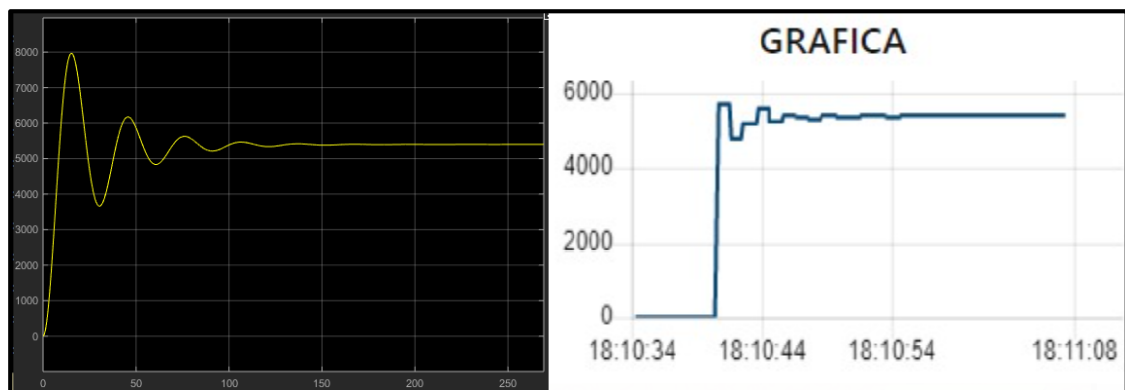


Elaborado por: Bryan Pillajo y Rodrigo Bautista

4.4 REALIDAD VIRTUAL

La realidad virtual en el presente proyecto representa el comportamiento de la red dentro de una simulación en Simulink, en el cual se puede representar la red inalámbrica a través de la librería truetime, el modelo matemático del motor DC y la representación 3D del movimiento del motor. Estas partes se integran formando el sistema WNCS controlado por el algoritmo PID, por lo tanto permite observar virtualmente el comportamiento del sistema para posteriormente comparar los resultados con el mundo real. Tal como se observa en la figura 4.12.

Figura 4.12. Comportamiento de la planta, Realidad Virtual vs Mundo real.



Elaborado por: Bryan Pillajo y Rodrigo Bautista

El motor diseñado en 3D simula el movimiento angular a través de su vástago enlazado a una variable que incrementa su ángulo a medida que pasa el tiempo, al momento de poner en marcha la simulación se puede observar virtualmente el comportamiento del motor.

Partiendo de la experimentación se regula los parámetros de la planta cambiando los valores característicos del modelo matemático del motor y enlazándolos con la parte de control, a través de los bloques Truetime que simulan una comunicación Ethernet, por lo tanto, el resultado de la curva de velocidad vs tiempo de la realidad virtual es similar a la curva obtenida en el mundo real.

CAPÍTULO 5

CONCLUSIONES

La red inalámbrica WNCS contiene una planta de control de velocidad de un motor DC basado en varios procesos que pueden ser aplicados en la industria para el área de control electrónico y en otras ramas como la educación, esta red permite el intercambio de datos a través de Wifi, permitiendo comunicar a los diferentes dispositivos Iot que conforman el sistema de manera inalámbrica y externa. Este sistema de control ocupa pocos espacios físicos para la comunicación, generando al usuario un beneficio económico, comodidad y movilidad en el proceso de la planta para esto se recopiló información acerca de varios procesos que en la actualidad son usados en la industria 4.0.

En la comunicación inalámbrica del sistema se utiliza el protocolo MQTT entre las diferentes partes del sistema: NodeMCU ESP8266, PC y Samsung A50. Este protocolo es utilizado por su sencillez al momento de ser configurado en los dispositivos Iot, por el bajo consumo de energía y porque trabaja con el mínimo de ancho de banda. Es por esto que es adecuado para utilizarlo en sistemas Iot pudiendo adaptarse a cualquier sistema de control con un amplio potencial para su desarrollo.

La Realidad Virtual cumple con la función de simular de manera interactiva el proceso implementado en el mundo real, ya que está conformado con las características que tiene la red WNCS el cual tiene la parte de control PID alojado fuera de planta de velocidad. Esto permite que el usuario realice cambios en las constantes del algoritmo de control y el Set point para su posterior análisis y estudio, teniendo así una herramienta la cual simule cualquier tipo de planta de forma ideal.

La Realidad Aumentada fue desarrollada en Unity por todas las facilidades que brinda al momento que da funciones a los elementos en el entorno gráfico mediante programación a través de Scripts. El software permite exportar el proyecto para diferentes plataformas, como Android, Windows, Linux, lo que permite portabilidad en la aplicación logrando instalarla en varios dispositivos con diferentes sistemas operativos. La APK diseñada para este proyecto está conectada inalámbricamente al

broker de la red, con el objetivo que el usuario pueda observar la velocidad de la planta en tiempo real y las características del actuador, ahorrando dinero en la compra de pantallas táctiles que funcionarían como interfaces.

Después de realizar las pruebas se puede concluir que el comportamiento del sistema tanto de manera virtual como real es similar, teniendo en cuenta que la simulación hecha con herramientas de software matemático siempre va ser ideal y sirve como referencia para una correcta implementación de un sistema de control. La planta puede ser testada por medio de una aplicación la cual indica la velocidad del motor en tiempo real esto puede ser usado en la industria para control y mantenimiento preventivo teniendo un potencial enorme para un posterior desarrollo en la industria.

RECOMENDACIONES

La WNCS tiene gran capacidad de enlace con varios dispositivos, por lo tanto se recomienda incrementar varias plantas de trabajo con diferentes actuadores para implementar otros algoritmos de control más complejos, con el fin de estudiar y desarrollar más aplicaciones en la industria 4.0.

En la comunicación se recomienda realizar un estudio a fondo de los factores como la velocidad, latencia y seguridad de que puede existir en el intercambio de datos entre la planta y el bróker, así también entre el bróker y el dispositivo móvil con la aplicación de realidad aumentada para incrementar la eficiencia de este tipo de control.

En la parte de realidad aumentada se recomienda investigar a profundidad los beneficios que tiene Unity en conjunto con MQTT, para el desarrollo de una interfaz más completa que permita controlar el Set Point desde la APK instalada en el móvil y el reconocimiento automático de la IP del bróker, ya que esto ahorrara tiempo en la instalación del APK en el dispositivo evitando el ingreso de la IP manualmente en el script.

La Realidad Virtual al ser diseñada en Simulink por su gran capacidad de procesamiento matemático, se recomienda implementar varios modelos de distintas plantas a través de funciones de transferencia o de bloques con los modelos matemáticos característicos de la planta, así también el desarrollo de modelos 3D para predecir el comportamiento físico del sistema.

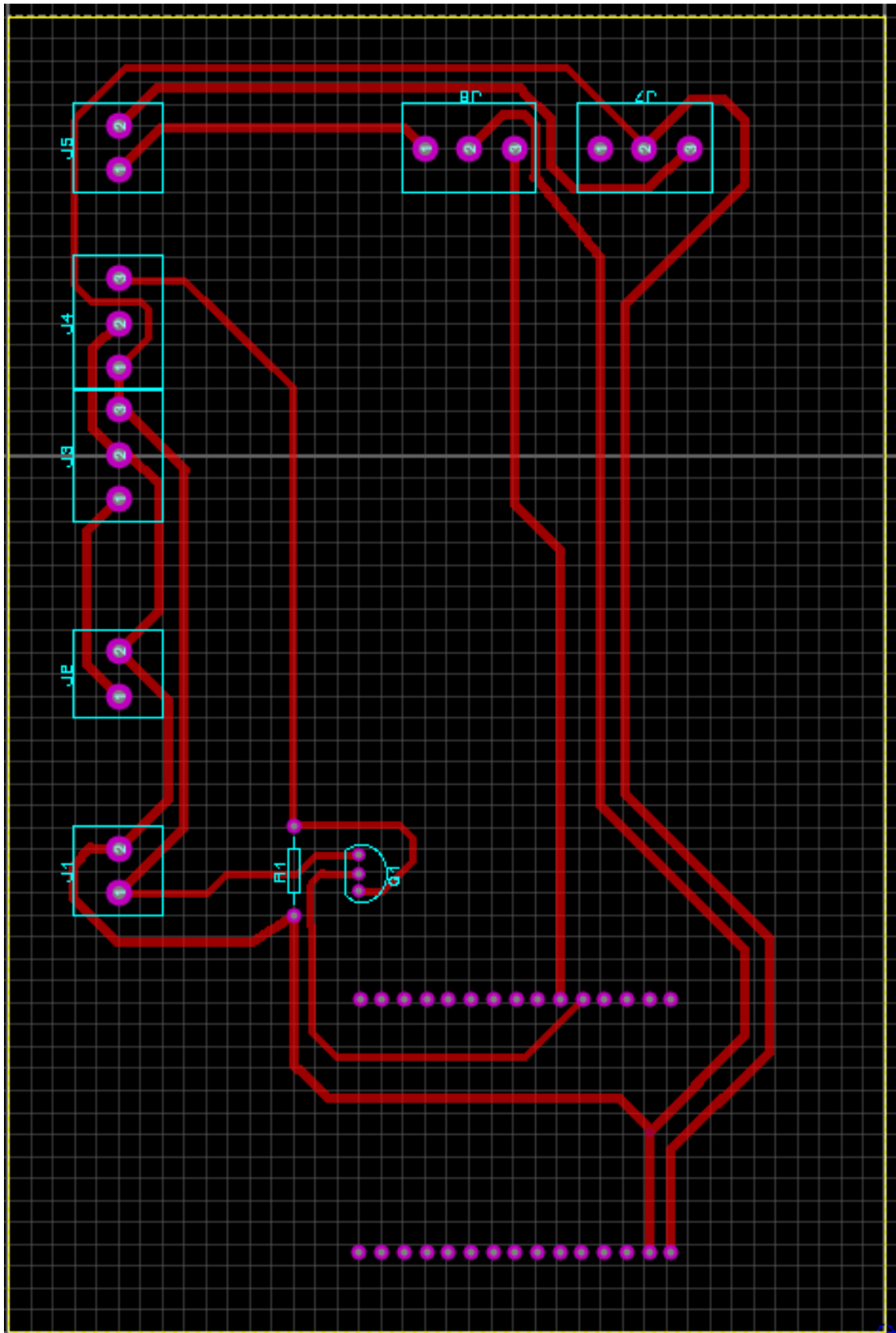
Considerando que la WNCS contiene los requerimientos principales para realizar el control de diferentes plantas o motores DC, se espera que este sistema sea implementado en la industria y sea tema de estudio e investigación para los docentes y estudiantes de la Universidad Politécnica Salesiana con el fin de seguir desarrollando la Automatización y control con el internet de las cosas en el país.

REFERENCIAS

- A, Carlos Pillajo, and Roberto Hincapié. 2018. *Wireless Network Control Systems De La Teoría a La Práctica*. ed. editorial Universitaria Abya-Yala. Quito-Ecuador.
- Chanthakit, Somphop. 2018. "MQTT Based Air Quality Monitoring System Using NodeMCU and Node-RED." : 3–7.
- Eridani, Dania, and Eko Didik Widiyanto. 2018. "Performance of Sensors Monitoring System Using Raspberry Pi through MQTT Protocol." *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*: 587–90.
- Escartín, Emilio R, Instituto Superior, and Politécnico José. 1998. "LA REALIDAD VIRTUAL , UNA TECNOLOGÍA EDUCATIVA A NUESTRO ALCANCE."
- Kim, Sung Lae et al. 2014. "Using Unity 3D to Facilitate Mobile Augmented Reality Game Development." *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*: 21–26.
- Manuel Báez, Álvaro Borrego, Jorge Cordero, Francisco Hernández Luis Cruz, Miguel González, Daniel Sanz David Palomero, José Rodríguez de Llera, and Pilar Torralbo Mariam Saucedo. 2014. "Introducción a Android." *E.M.E. Editorial* ©: 1–20. www.tecnologiaUCM.es.
- Nguyen, Vinh T., and Tommy Dang. 2017. "Setting up Virtual Reality and Augmented Reality Learning Environment in Unity." *Adjunct Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2017*: 315–20.
- Paganelli, Matteo, Paolo Sottovia, Francesco Guerra, and Yannis Velegrakis. 2019. "TuneR." *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*: 2945–48.
- Ramos-Escamilla, Maria. 2011. "Ingeniería Eléctrica." *Ingeniería e Investigación; num. 51 (2002); 13-15 Ingeniería e Investigación; num. 51 (2002); 13-15* 2248-8723 0120-5609.
<http://revistas.unal.edu.co/index.php/ingainv/article/view/21432>.
- Rattanapoka, Choopan, Somphop Chanthakit, Apatsaraporn Chimchai, and Amorntip Sookkeaw. 2019. "An MQTT-Based IoT Cloud Platform with Flow Design by Node-RED." *RI2C 2019 - 2019 Research, Invention, and Innovation Congress*.
- Ritchie, Peter. 2016. "Practical Microsoft Visual Studio 2015." *Practical Microsoft Visual Studio 2015*: 1–25.

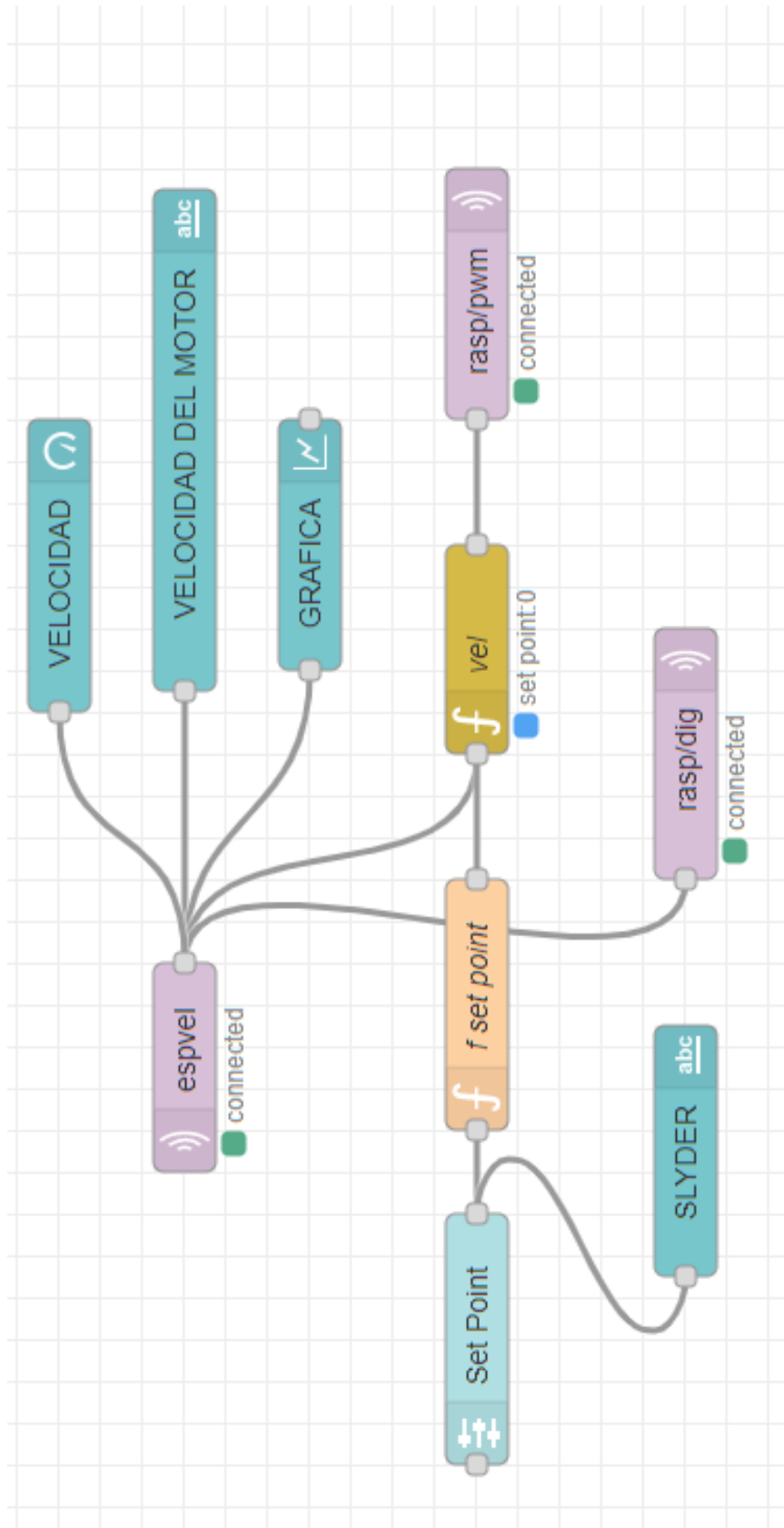
- Sandino, Luis A., Manuel Bejar, and Anibal Ollero. 2010. "Entorno Avanzado de Simulación y Experimentación En Tiempo Real." *Actas de las XXXI Jornadas de Automática* (June 2015): 1–8.
- Sun, Jiwei, Hailong You, Zhiyun Li, and Xinzhang Jia. 2009. "Research on Modeling Control Module of DC-DC Converter for Simulink." : 129–32.
- Syafiq, Mohamad, and Azwan Mohd. 2019. "EasyChair Preprint Control System Development for Hybrid Generator Project Control System Development for Hybrid Generator Project."
- Tabaa, Mohamed, Brahim Chouri, Safa Saadaoui, and Karim Alami. 2018. "Industrial Communication Based on Modbus and Node-RED." *Procedia Computer Science* 130: 583–88. <https://doi.org/10.1016/j.procs.2018.04.107>.
- Álamo, Teodoro. 2007. "Diseño Del Controlador PID." Universidad de Sevilla, Departamento de Ingeniería de Sistemas y Automática: 1–37.

ANEXO 2
PCB TOP COOPER



ANEXO 3

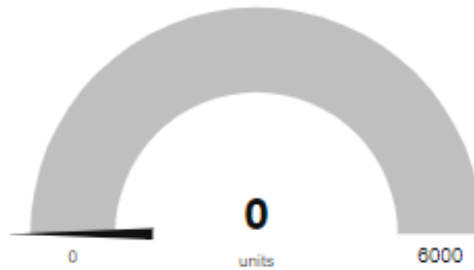
PROGRAMACION EN NODE-RED



ANEXO 4
DASHBOARD NODE-RED

GRAFICA

VELOCIDAD



Set Point

SLYDER

VELOCIDAD DEL MOTOR

- Con slider “Set Point” se controla la velocidad el motor manualmente.
- Con gráfica y un velocímetro para la ver el comportamiento de la velocidad del motor.

ANEXO 5

CODIGO NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <String.h>
volatile int contador=0, vel=0;
#define led 16
#define bt 5
#define led_pwm 15

// Update these with values suitable for
// your network.

const char* ssid = "CNTFIBRA-
ElectroSHARP";
const char* password = "SHARP1726793241";
const char* mqtt_server = "192.168.1.4"; // example
192.168.0.19
const char* topic_analog = "rasp/pwm";
const char* topic_dig = "rasp/dig";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0, lastvel=0;
char msg[50];
int value = 0, btn_las_state=0,
estado_bt=0, encendido=0;
int pot=0, pot_last=0, pot1=0;
int numer=0;

void setup() {
  pinMode(led, OUTPUT); // Initialize
the BUILTIN_LED pin as an output
  pinMode(bt, INPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  attachInterrupt(13,interrupcion0,RISIN
G);
  analogWrite(led_pwm,400);}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi
network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() !=
WL_CONNECTED) {
    delay(500);
    Serial.print("."); }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
```

```

Serial.println(WiFi.localIP());}

void callback(char* topic, byte*
payload, unsigned int length) {
    String string;

    for (int i = 0; i < length; i++) {
        string+=((char)payload[i]); }

    int pos = string.toInt();

    if ( strcmp(topic, topic_analog) == 0 ) {
        analogWrite(led_pwm,pos);
        // Serial.println(pos); }
        if ( strcmp(topic, topic_dig) == 0 ) {
            // Switch on the LED if an 1 was
            received as first character
            if ((char)payload[0] == '0') {
                // Serial.println("LOW");
                digitalWrite(led, LOW); // Turn the
                LED on (Note that LOW is the voltage
                level
                // but actually the LED is on; this is
                because
                // it is acive low on the ESP-01) }

            if ((char)payload[0] == '1') {
                Serial.println("HIGH");
                digitalWrite(led, HIGH); // Turn the
                LED off by making the voltage HIGH
            } }
            delay(1); }

```

```

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT
connection...");
        // Attempt to connect

        if (client.connect("ESP8266Client"))
        {
            Serial.println("connected");
            // Once connected, publish an
            announcement...
            client.publish("anesp", "hello
world");
            // ... and resubscribe
            client.subscribe(topic_analog);
            client.subscribe(topic_dig);

        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5
seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        } }
    void loop() {

        if (!client.connected()) {
            reconnect();
        }
        client.loop();
    }

```

```
long now = millis();
if (now - lastMsg > 150) {
  lastMsg = now;

  sprintf (msg, 75, "%ld", vel);
  Serial.print("Publish message: ");
  Serial.println(vel);
  client.publish("espvel", msg); }
```

```
if (now-lastvel>999){
  lastvel = now;
  vel=contador*3;
  contador=0; }}
```

```
ICACHE_RAM_ATTR      void
interrupcion0()
{
  contador++;
}
```

ANEXO 6

PROGRAMACION EN SCRIPT DE VISUAL STUDIO

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
using System.Net;
using System;
using uPLibrary.Networking.M2Mqtt;
using uPLibrary.Networking.M2Mqtt.Messages;
using uPLibrary.Networking.M2Mqtt.Utility;
using uPLibrary.Networking.M2Mqtt.Exceptions;

public class timer : MonoBehaviour
{
    private const int V = 1883;
    private const string IpString = "192.168.1.4";

    // Declaramos variables privadas para el cliente MQTT
    private MqttClient client;
    private string dat;
    /// <summary>
    /// ///
    /// </summary>
    public Text Textvel;
    public String tiempo = "";
    void Start()
    {
        // Creamos la instancia del cliente
#pragma warning disable CS0618 // El tipo o el miembro están obsoletos
        client = new MqttClient(IPAddress.Parse("192.168.1.4"), 1883);
#pragma warning restore CS0618 // El tipo o el miembro están obsoletos
        // se registra el mensaje que se recibe
        client.MqttMsgPublishReceived += client_MqttMsgPublishReceived;
        string clientId = Guid.NewGuid().ToString();
        client.Connect(clientId);
        // Se subscribe al topico hello y se obtiene los datos
        client.Subscribe(new string[] { "rasp/dig" }, new byte[] {
MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE });
    }
    void client_MqttMsgPublishReceived(object sender, MqttMsgPublishEventArgs e)
    {
        Debug.Log("Received: " + System.Text.Encoding.UTF8.GetString(e.Message));
        dat = System.Text.Encoding.UTF8.GetString(e.Message);
        //se convierte en dato a variable int
        tiempo = dat;
    }

    public void Update()
    {
        Textvel.text = tiempo;
    }
}
```