



MSc Dissertation in
Computer Engineering – Mobile Computing (MEI-CM)

***DEVELOPMENT A VIRTUAL REALITY MODEL
SIMULATION IN ORDER TO CONTROL A DRONE
USING A WEARABLE DEVICE IN A 3D ENVIRONMENT***
*(Desarrollo de un modelo de simulación de realidad virtual usando
un dispositivo portable en un entorno virtual)*

Christian Andrés Bustamante Crespo

Leiria, September 2017



***DEVELOPMENT A VIRTUAL REALITY MODEL
SIMULATION IN ORDER TO CONTROL A DRONE
USING A WEARABLE DEVICE IN A 3D ENVIRONMENT***
*(Desarrollo de un modelo de simulación de realidad virtual usando
un dispositivo portable en un entorno virtual)*

Christian Andrés Bustamante Crespo

Dissertation developed under the supervision of Doctor António Manuel de Jesus Pereira, professor at the School of Technology and Management of the Polytechnic Institute of Leiria and MSc. Graciela Guerrero Idrovo, Professor at Department of Computer Science from Universidad de Las Fuerzas Armadas ESPE.

Leiria, September 2017

Agradecimientos

Agradezco a Dios por darme capacidad, perseverancia y constancia para cumplir con las metas propuestas, por llevarme por encaminarme al éxito y sobretodo darme salud para poder cumplir con mis propias expectativas.

Quiero expresar mis agradecimientos a la institución Ecuatoriana que a través de una beca me permitió realizar mis estudios de maestría en el Instituto Politécnico de Leiria.

Agradezco a la Escuela Superior de Tecnología y Gestión (ESTG) de Leiria, por los medios y condiciones que ha puesto a mi disposición para la realización de este trabajo.

Agradezco a la Universidad de las Fuerzas Armadas ESPE, por los medios y condiciones que ha puesto a mi disposición para la realización de este trabajo.

Quiero expresar un profundo agradecimiento al Profesor Doctor Antonio Pereira por la orientación, el incentivo y todo el apoyo prestados a lo largo del proceso de desarrollo de este trabajo y de las publicaciones que nacieron a raíz de la investigación y desarrollo de este trabajo.

Agradezco también a la Profesora. Graciela Guerrero MSc por el apoyo durante el desarrollo de este trabajo y las publicaciones que se generaron a través del mismo.

Nota Previa

Del trabajo efectuado resultaron las siguientes publicaciones:

- C. Bustamante, G. Guerrero, N. Rodrigues y A. Pereira, A Virtual Reality UAV Simulation with Body Area Networks to promote the elders life quality, con el fin de generar un entorno virtual simulado para promover la calidad de vida de las personas mayores, publicada en la 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW), UTAD, Vila Real, Portugal, 2016;
- C. Bustamante, G. Guerrero, N. Rodrigues y A. Pereira , UAV flight simulator to improve elders' quality of life, como propuesta inicial del presente trabajo de investigación, publicada en la 5th World Conference on Information Systems and Technologies (WorldCist'17), Porto Santo Island, Madeira, Portugal, 2017;
- C. Bustamante, G. Guerrero, N. Rodrigues y A. Pereira, UAV simulator for grown-up people quality of life enhancement, como propuesta de mejora al presente trabajo de investigación, publicada en la 12nd. Iberian Conference in Information Systems and Technologies (CISTI2017), Lisboa, Portugal, 2017.

Resumen

Este trabajo propone diseñar, desarrollar y evaluar un simulador de realidad virtual (VR) para maniobrar un Vehículo Aéreo No Tripulado (UAV) únicamente con el movimiento de dispositivos portables, estos dispositivos utilizan sensores de movimiento (IMU) que son los que determinan los movimientos del (UAV), El entorno visual es reproducido de manera virtual utilizando gafas VR y un celular.

La información que genera los sensores es enviada a través de tecnología Bluetooth Low Energy (BLE) de ambos dispositivos (RPI) hacia el celular, donde se ejecuta el simulador. Esto con la finalidad de generar un sistema tecnológico orientado a mejorar la calidad de vida de las personas mayores fomentando el movimiento y entretenimiento en un entorno inmersivo y seguro.

El diseño propuesto en este trabajo, utiliza una arquitectura cliente-servidor con tecnología BLE (Bluetooth Low Energy), compuesta de 2 dispositivos Raspberry Pi (RPI) que actúan como controles remotos (Clientes BLE), y un Smartphone que permite la visualización del entorno 3D en el que corre el simulador de manera virtual (Servidor BLE).

Este trabajo empieza detallando el desarrollo y evaluación de un simulador de realidad virtual sobre plataforma Android para visualizar y manejar un vehículo aéreo no tripulado (UAV) en un ambiente virtual.

El entorno 3D propuesto en este trabajo, corresponde a un paisaje montañoso construido a través de un sistema panorámico y mallas de terreno 3D utilizando el motor de videojuegos Unity3D, además se desarrolla una librería nativa en JAVA para consumir los servicios BLE que son necesarios para la comunicación entre el celular y los RPI, esta librería es utilizada desde el simulador y permite recibir los mensajes que determinan las acciones que debe realizar el Drone en el ambiente virtual.

Posteriormente se realiza el desarrollo de una aplicación en Python que permite hacer uso del sensor IMU de la placa Sense HAT que esta integrada al RPI y los servicios BLE propios del RPI para que pueda funcionar como control remoto y maniobrar de manera inalámbrica el DRONE virtual del simulador. El Raspberry Pi junto a la placa Sense HAT demostraron ser dispositivos portables idóneos para ser usados como controles

remotos del Drone virtual, de esta manera se complementa el simulador juntando el celular con las gafas VR y los 2 RPI, uno para cada mano.

Una vez concluido el desarrollo del sistema bajo esta arquitectura, se documenta una estadística de pruebas relacionadas al uso del sistema en personas mayores, donde se aprecia el gusto, la diversión y la evolución en el uso del simulador bajo los diferentes modos de maniobra y visualización que el simulador ofrece. Según las pruebas realizadas, el simulador posee un entorno virtual muy realista, con paisajes panorámicos y mallas 3D, el sistema está desarrollado para ser ejecutado en 2 modos de visualización y 2 modos de dificultad diferentes los cuales se describen a continuación.

- Utilizando una cámara en Tercera persona: en donde el usuario visualiza al Drone y al entorno mientras lo maniobra (Posición de la cámara diferente a la posición del Drone);
- Utilizando una cámara en primera persona: en donde el usuario visualiza el entorno desde la perspectiva del Drone, (Posición de la cámara igual a la posición del Drone);
- Modo Fácil: en donde el usuario necesita realizar 4 movimientos entre ambos controles (RPI) para maniobrar completamente al Drone;
- Modo Difícil: en donde el usuario necesita realizar 6 movimientos entre ambos controles (RPI) para maniobrar completamente al Drone.

Los resultados muestran una acogida y gran interés por parte de los adultos, para ellos utilizar un simulador con tecnología VR resultó en extremo atractivo, además de la facilidad de manejo, ya que únicamente requieren mover las manos mientras controlan al Drone y finalmente se desarrolla y evalúa una aplicación de Android Nativa que funcione de la misma manera que el simulador y que permita maniobrar un DRONE real (Phantom 4) utilizando los mismos dispositivos portables (RPI) como controles de mando. De esta manera, se demostró que es posible controlar un Drone real (Phantom 4) utilizando el mismo mecanismo de maniobra que el que fue implementado en el simulador, esto se demostró al desarrollar una aplicación nativa utilizando el SDK del Phantom 4 y el mismo sistema de control a través de BLE que utiliza el simulador.

Palabras-clave: Simulation, 3D, UAV, VR, Bluetooth, BLE, Inertial Sensors, Elders, Well-being, Life quality. Virtual Environment, Wearable Devices, Real Time.

Abstract

The world population is aging, The quality life in elder people deteriorates with age, this is because of the sedentary lifestyle in which they live, consequently these countries must adapt much more rapidly to this aging. Therapy and Physical activity helps to improve health and well-being in the elders, bringing longevity and improves their life quality. The Technology has been used as a tool in various fields and there are some applications that have been developed to encourage sports through entertainment. Nintendo Wii and Kinect games had a positive impact on the overall well-being of the elderly, compared to a other group that played traditional board games.

The life quality of the elderly is impaired by the lack of physical activity, entertainment and lack of memory use. This work aims to solve this problem with the implementation of a system that promotes physical activity through body movements, generate entertainment through the control of a drone in a virtual environment and promote memory through remembering movement patterns that is due perform to be able to control the virtual Drone correctly.

This work makes use of wireless devices, wearable, light and with IMU sensors integrated in order to generate a body area network system that scans each arm movement and sends that information through Bluetooth Low Energy (BLE) technology to a virtual environment and control an UAV (Unnamed Aerial Vehicle) remotely.

The proposed system includes a 3D simulator that uses VR glasses for an immersive visualization in a virtual world, 2 Raspberry Pi devices with Sense HAT board integrated as hand controls, these plates are equipped with an IMU sensor that is responsible for the arms movement monitoring. This proposal allows an interconnection of portable devices through wireless technology and combine them with immersive virtual environments to generate a system that emulates the control of a Drone in a simulated environment.

In order to adapt the system and focus on older people, it is proposed to implement 2 visualization modes and 2 difficulty modes, the visualization modes determine the perspective in which the user looks at the environment while the user controls the Drone, The First perspective, is defined by a third person camera system, where the user is layers of watching the Drone while controlling it and the second mode corresponds to a first

person camera system, where the user's eyes are in the same drone position, this mode allows appreciate the environment in a better way, but does not allow to visualize the UAV. In the performed tests, each user had his / her own preferences, some of them liked to use the simulator with first persona camera configuration and others with third person camera configuration. this way the system is designed to adapt to the visualization mode of each individual.

With respect to the difficulty modes, the simulator offers the possibility to use the maneuver the drone through 2 mechanisms, in easy mode, where the user only needs to make 4 movements between both hands and the difficult mode where the user requires 6 movements between both hands in order to have a total drone control, this allows the user to develop cognitive skills (Eye-hand coordination) and improve physical and mental health, when using this system.

Through the tests of latency, it is determined that the transmission of data through Bluetooth Low Energy is extremely fast which means that the movements of hands that the user performs become real motions of the Virtual Drone in real time. A real time simulator is a system that offers reaction times similar to reality, this determines the efficiency of the simulator when it is used as training before using a real Drone.

Performed tests with older adults reveal an evolution of skills when using the system progressively in each of the different modes of use. The interest, the entertainment and the physical activity that the system promotes in the elderly, is enough to improve their quality of life by contributing in the reduction of the sedentary, promoting the physical activity and mental state.

To achieve the implementation of this system we propose a parallel development of a native Android application that makes use of the same system of body area networks of sensors to maneuver a real dron (Phantom 4) using only the movement of hands, with the object of validate the functions and movements of the virtual drone based on the movements of the real dron under the same control mechanism.

Key Words: Simulation, 3D, UAV, VR, Bluetooth, BLE, Inertial Sensors, Elders, Well-being, Life quality. Virtual Environment, Wearable Devices, Real Time.

Lista de figuras

Figura 1 - Sistema Virtual para rehabilitación de miembros superiores [12].....	10
Figura 2 - Pruebas de usabilidad SSN [17].....	11
Figura 3 - Video juego de movimiento de cuerpo entero [19].....	12
Figura 4 - Diagrama de la arquitectura general.....	15
Figura 5 - Diagrama de funcionalidad	16
Figura 6 - Raspberrry Pi 3 B	17
Figura 7 - Placa Sense HAT integrada al Raspberry Pi	18
Figura 8 - Comparación arquitectónica de las tecnologías Bluetooth [24]	20
Figura 9 - Distribución de canales en BLE [22].....	21
Figura 10 - Topología de comunicación en entorno conectado [26].....	25
Figura 11 - Diagrama de mensajería cliente-servidor [26]	25
Figura 12 - Diagrama de la estructura GATT [26]	26
Figura 13 - Diagrama de la estructura GATT usada en el simulador	27
Figura 14 - Escena generada a base de una imagen panorámica.....	30
Figura 15 - Configuración del Skybox utilizando la herramienta “Panorama to Cubemap” [28]	31
Figura 16 - Diseño del escenario virtual para el simulador	32
Figura 17 - Modelo 3D del dron e	33
Figura 18 - Circuito de objetivos aéreos.....	34
Figura 19 - Componente de Google VR	35
Figura 20 - Aplicación de realidad virtual y gafas VR [29]	35
Figura 21 - Sistema de rotación rotación VR [30].....	36
Figura 22 - Cámara en tercera persona con soporte VR	37
Figura 23 - Cámara en primera persona con soporte VR	37
Figura 24 - Modificación de cámara en tiempo de ejecución	38
Figura 25 - Escena de configuración de cámaras	38
Figura 26 - Eje de coordenadas propuesto en “AR Drone Quadrotor” [31].....	39
Figura 27 - Eje de coordenadas del dron e propuesto en este trabajo.....	40
Figura 28 - Función que actualiza la rotación Yaw del dron e.....	40
Figura 29 - Componentes se usan en el motor de física de Unity3D.....	41
Figura 30 - Diagrama de maniob re del dron e a través de rotaciones del RPI en Modo fácil	42
Figura 31 - Diagrama de maniob re del dron e a través de rotaciones del RPI en Modo difícil	43
Figura 32 - Diagrama de la arquitectura de software del simulador.....	44
Figura 33 - Estructura de la capa de bajo nivel.....	44
Figura 34 - Código de inicialización de las funciones BLE.....	45
Figura 35 - Entidad de la característica	47

Figura 36 - Evento que obtiene y administra una característica	47
Figura 37 - Función que obtiene las características desde el plugin JAR.....	48
Figura 38 - Configuración del plugin JAR (BLE_Plugin.jar) en Unity3D.....	48
Figura 39 - Estructura de la capa de alto nivel	49
Figura 40 - Componente que configura los servicios BLE.....	52
Figura 41 - Configuración del componente que controla el UAV y las cámaras.....	52
Figura 42 - Diagrama de flujo de la lógica del Simulador	53
Figura 43 - Arquitectura de software de la aplicación RPI	55
Figura 44 - Método para generar UUID de 128 bits	57
Figura 45 - Método de configuración y escritura de una característica.....	57
Figura 46 - Sistema de rotación del sensor IMU de la placa Sense HAT [37]	58
Figura 47 - Rango de valores angulares del sensor IMU	59
Figura 48 - Sistema de colores de la matriz LED para determinar el estado de conexión con el simulador ...	59
Figura 49 - Disminución de señal WIFI según a la distancia [25].....	61
Figura 50 - Diagrama de flujo de la comunicación entre los RPI y el simulador (Celular)	62
Figura 51 - Estadística de FM usando la configuración TPC en modo fácil.....	66
Figura 52 - Estadística de TU usando la configuración TPC en modo fácil	66
Figura 53 - Estadística de FM usando la configuración TPC en modo difícil.....	68
Figura 54 - Estadística de TU usando la configuración TPC en modo difícil	69
Figura 55 - Estadística de FM usando la configuración FPC en modo fácil.....	71
Figura 56 - Estadística de TU usando la configuración FPC en modo fácil	71
Figura 57 - Estadística de FM usando la configuración FPC en modo fácil.....	74
Figura 58 - Estadística de TU usando la configuración FPC en modo fácil	74
Figura 59 - Arquitectura General propuesta para de control del UAV real con los RPI.....	75
Figura 60 - Arquitectura General de software para el control del drone real.....	76
Figura 61 - Sistema de coordenadas del Phantom 4 [40].....	77
Figura 62 - Envío de datos al control remoto del Phantom 4.....	78
Figura 63 - Simulador DJI para el Phantom 4	79
Figura 64 - Marcas diferentes de gafas VR probadas con el Simulador	91
Figura 65 - Usuarios utilizando el Simulador Virtual	93
Figura 66 - Usuario que presentó incomodidad al usar el Simulador sin lentes	94
Figura 67 - Usuarios jóvenes utilizando el Simulador.....	95
Figura 68 – Control de un drone real (Phantom 4) utilizando los Raspberry Pi en un entorno simulado.....	96
Figura 69 - Interfaz grafica de la aplicación nativa que controla un drone real (Phantom 4)	97
Figura 70 - Control de un drone real (Phantom 4) utilizando los Raspberry Pi en un entorno real.....	98

Lista de tablas

Tabla 1 - Bluetooth Clásico vs BLE	19
Tabla 2 - Pruebas de Latencia en la transmisión de datos	62
Tabla 3 - Resultados de las pruebas usando la configuración TPC en modo fácil, primera vez	64
Tabla 4 - Resultados de las pruebas usando la configuración TPC en modo fácil, tercera vez	65
Tabla 5 - Resultados de las pruebas usando la configuración TPC en modo difícil, primer vez	67
Tabla 6 - Resultados de las pruebas usando la configuración TPC en modo difícil, tercera vez	68
Tabla 7 - Resultados de las pruebas usando la configuración FPC en modo fácil, primera vez	69
Tabla 8 - Resultados de las pruebas usando la configuración FPC en modo fácil, tercera vez.....	70
Tabla 9 - Resultados de las pruebas usando la configuración FPC en modo difícil, primera vez	72
Tabla 10 - Resultados de las pruebas usando la configuración FPC en modo difícil, tercera vez.....	73
Tabla 11 - Resultados de maniobras en base a factores de error.....	80

Lista de acrónimos

AD	Advertising Data
AES	Advanced Encryption Standard
API	Application Programming Interface
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
BR	Basic Rate
CPU	Central Processor Unit
EDR	Enhanced Data Rate
FHSS	Frequency Hopping Spread Spectrum
FPS	Frames Per Second
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GI	Global Illumination
GPU	Graphic Processor Unit
HMD	Head Mounted Display
IDE	Integrate Development Environment
IMU	Inertial Measurement Unit
IOT	Internet of Things
KBPS	Kilobit Per Second
L2CAP	Logical Link Control Adaptation Protocol
LE	Low Energy
LED	Light-Emitting Diode
MAC	Media Access Control
MBPS	Megabit Per Second
PC	Personal Computer
PHY	Physical Layer
RAM	Random Access Memory
RPI	Raspberry Pi

SDK	Software Development Kit
SIG	Special Interest Group
SMP	Security Manager Protocol
TIC	Information and Communication Technologies
UAS	Unnamed Aerial System
UAV	Unnamed Aerial Vehicle
USB	Universal Serial Bus
UUID	Universal Unique Identifier
VR	Virtual Reality

Tabla de Contenidos

AGRADECIMIENTOS	III
NOTA PREVIA	V
RESUMEN	VII
ABSTRACT	IX
LISTA DE FIGURAS	XI
LISTA DE TABLAS	XIV
LISTA DE ACRÓNIMOS	XVI
TABLA DE CONTENIDOS	XVIII
1. INTRODUCCIÓN	1
1.1. IDENTIFICACIÓN DEL PROBLEMA	3
1.2. OBJETIVOS	4
1.3. ESTRUCTURA DE LA TESIS	6
2. TRABAJO RELACIONADO	9
3. ARQUITECTURA GENERAL	15
3.1. TECNOLOGÍAS UTILIZADAS	16
3.1.1. TELÉFONO CELULAR GAMA ALTA	16
3.1.2. RASPBERRY PI 3B	17
3.1.3. PLACA SENSE HAT	18
3.1.4. BLUETOOTH LOW ENERGY	18
4. SIMULADOR VIRTUAL	29
4.1. UNITY3D	29
4.2. ENTORNO VIRTUAL DEL SIMULADOR	29
4.2.1. ESCENARIO 3D	30
4.2.2. CIRCUITO DE OBJETIVOS AÉREOS	33
4.3. CONFIGURACIÓN DE CÁMARAS	34
4.3.1. CÁMARA EN MODO VR	34
4.3.2. CÁMARAS EN MODOS FPC & TPC	36
4.4. MECÁNICA Y CONTROL DEL DRONE	39
4.5. ARQUITECTURA DE SOFTWARE	43
4.5.1. LIBRERÍA NATIVA DE SERVICIOS BLE	44

4.5.2.	LÓGICA DEL SIMULADOR	49
5.	CONTROLES REMOTOS RPI	55
5.1.	MODULO BLE	56
5.2.	MODULO IMU	57
5.3.	COMUNICACIÓN CON EL SIMULADOR	60
6.	PRUEBAS REALIZADAS	63
6.1.	PRUEBAS EN MODO TPC Y FÁCIL	64
6.2.	PRUEBAS EN MODO TPC Y DIFÍCIL	66
6.3.	PRUEBAS EN MODO FPC Y FÁCIL	69
6.4.	PRUEBAS EN MODO FPC Y DIFÍCIL	71
7.	PRUEBAS CON DRON REAL	75
7.1.	ARQUITECTURA GENERAL	75
7.2.	ARQUITECTURA DE SOFTWARE Y DESARROLLO	76
7.3.	PRUEBAS Y RESULTADOS	78
8.	CONCLUSIONES	81
8.1.	CONTRIBUCIONES PRINCIPALES	83
8.2.	TRABAJO FUTURO	84
9.	REFERENCIAS	87
10.	ANEXOS	91
A.	EJEMPLO DE 2 SET DE GATAS VR QUE SE USARON DURANTE EL DESARROLLO DE ESTE TRABAJO.	91
B.	IMÁGENES DE USUARIOS UTILIZANDO EL SIMULADOR VIRTUAL.	91
C.	USUARIO QUE TUBO PROBLEMAS AL USAR EL SIMULADOR CON LENTES	93
D.	USUARIOS JÓVENES QUE TAMBIÉN DISFRUTARON UTILIZANDO EL SIMULADOR	94
E.	CONTROL DE UN PHANTOM 4 UTILIZANDO LOS RASPBERRY PI DESDE UN ENTORNO SIMULADO.	95
F.	INTERFAZ DE USUARIO DE LA APLICACIÓN NATIVA PARA MANIOBRAR UN DRONE REAL.	96
G.	CONTROL DE UN PHANTOM 4 UTILIZANDO LOS RASPBERRY PI EN UN ENTORNO REAL.	97

1. Introducción

Las nuevas tecnologías han sido utilizadas en diversas áreas para solventar y dar solución a diferentes problemas, el combinar tecnologías es uno de los procesos de investigación que a dado lugar a grandes proyectos, propuestas e investigaciones, en esta medida, este trabajo pretende ser un sistema que mescle varias tecnologías para generar un sistema que emule el maniobre de un UAV en un entorno virtual y sea fácilmente maniobrado a través del movimientos de los brazos, estos movimientos son detectados a través de sensores que se encuentran en dispositivos portables, ligeros y que pueden ser manipulados cómodamente con cada mano. La información de cada sensor es trasmitida a través de tecnología inalámbrica hacia el simulador (Celular). Cada una de estas tecnologías son explicadas brevemente a continuación con el fin de dar una introducción al trabajo propuesto en esta investigación.

Las herramientas de desarrollo integrado (IDEs), para la creación de aplicaciones móviles se han mejorado drásticamente, ahora es posible encontrar herramientas gratuitas que integran componentes y tecnologías para facilitar y acelerar el desarrollo de aplicaciones graficas 2D y 3D de manera multiplataforma. Unity3D [1] es un motor gráfico diseñado para el desarrollo de videojuegos en varias plataformas y que hoy en día es usado por varias industrias y desarrolladores independientes ya que su potencial es amplio y posee la facilidad de crear y adaptar cualquier tipo de librería nativa para solventar problemas de compatibilidad con sistemas operativos específicos como iOS o Android.

Las tecnologías de comunicación inalámbrica han llegado a satisfacer cualquier tipo de comunicación y especialmente la tecnología de Bluetooth se la encuentra en casi cualquier dispositivo inteligente, como relojes, teléfonos, impresoras, auriculares y hasta lámparas, esta tecnología que inicialmente fue desarrollada para trasmisión de datos en rangos cortos ahora posee estándares y protocolos de comunicación de bajo consumo energético, lo que lo hace una tecnología inalámbrica idónea para ser usado entre dispositivos portables que requieren bajo consumo de energía como sensores [2]. Esta nueva especificación se la conoce como Bluetooth de baja energía (Bluetooth Low Energy o BLE) y permite generar enlaces de comunicación sencillos en aplicaciones de muy baja potencia.

El vehículo aéreo no tripulado (UAV) o el sistema aéreo no tripulado (UAS) es un vehículo aéreo capaz de volar sin un operador humano a bordo [3]. Aunque los UAV se

usan principalmente en aplicaciones militares, actualmente se los usa también en trabajos o proyectos científico, seguridad pública y tareas comerciales. Un UAV puede ser controlado remotamente, semi - automático, automático, o una combinación de los mismos, capaz de realizar tantas tareas como se puede imaginar [4]. Estos vehículos han tenido bastante auge tanto para ser usados en eventos deportivos, militares o como entretenimiento. Los drones modernos con posibilidad de usar sus respectivos SDK tienden a tener precios poco accesibles para la mayoría de personas. Este tipo de vehículo generalmente es maniobrado utilizando un control remoto o a través de una aplicación móvil, es de fácil control y las versiones más modernas poseen GPS integrado lo que les permite recorrer rutas previamente programadas [5].

En cuanto al uso de UAVs enfocados en la mejora de calidad de personas mayores, no se ha encontrado trabajo relacionado, pero si trabajos con tecnología acorde a la usada en este trabajo, un ejemplo es AR .Drone [6] el cual propone un prototipo que combina Sensores y movimientos de usuarios para ser utilizados en el juego "IMAV [7]", donde se genera un mapa con una ruta a base de texturas, las cuales son escaneadas y convertidas en coordenadas de ubicación que utiliza el Drone para realizar un seguimiento a través de reconocimiento de patrones, los cuales representan la trayectoria y la posición, todo en un área de vuelo determinada. Los autores se basaron en dos estructuras principales: un filtro de Kalman [8] y un controlador no lineal para guiar el trabajo en cada misión. La investigación concluye que el uso de las dos estructuras se puede utilizar para guiar el drone en diferentes caminos a pesar de encontrar cualquier perturbación. Una de las limitaciones que se obtuvo fue el aspecto en entornos cerrados, por lo que se recomendó aumentar el tamaño de la marca y mantener el entorno con buena iluminación.

La evolución de ordenadores también ha dado origen a computadoras mucho más pequeñas, como es el caso de las Raspberry Pi (RPI), un nano computador, tan pequeño como la palma de la mano, liviana y con alto nivel de procesamiento, su sistema operativo basado en Linux posee una interfaz grafica y permite usar varios tipos de lenguajes de programación. Es por esto que este dispositivo a sido utilizado para proyectos que involucren las IoT. Otra ventaja es que a través de su ranura GPIO se puede adaptar o integrar diferentes tipos de placas o tarjetas para maximizar sus posibilidades en el desarrollo de aplicaciones específicas, como es el caso de [9], en donde se integran varios componentes a una RPI para generar un sistema educativo lúdico para niños especiales, o

la placa Sense HAT, una tarjeta que posee varios tipos de sensores además de una matriz de leds que funcionan como un monitor de leds y que es la que se usa en este trabajo.

La propuesta de este trabajo, integra varias de las tecnologías mencionadas anteriormente para generar un sistema que sirva como apoyo en la mejora de la calidad de vida de las personas mayores, ya que por el sedentarismo tienden a perder condición física y mental de manera progresiva. La propuesta que presenta esta tesis es un sistema de realidad virtual que pueda ser ejecutada en cualquier celular inteligente utilizando cualquier tipo de gafas VR que se encuentren hoy en día en el mercado. Este simulador presenta un escenario realista y un DRONE virtual que es maniobrado a través de los movimientos de 2 Raspberry Pi. La comunicación e interacción entre el celular y las computadoras portables se lo realiza de manera inalámbrica utilizando el estándar BLE, donde el celular actúa como servidor y las computadoras como clientes. El simulador está diseñado para ser visualizado en 2 perspectivas diferentes, en primera y tercera persona, adicionalmente el sistema ofrece 2 tipos de formas para maniobrar el DRONE, una simple en la que el DRONE es controlado con 4 movimientos de las manos (2 por mano) y una compleja en la que requiere un total de 6 movimientos de las manos (3 por mano).

Adicionalmente al Simulador, esta tesis desarrolla una aplicación de Android Nativa para controlar un DRONE real, (Phantom 4) utilizando las mismas nano computadoras que el simulador, esto con el objetivo de poder realizar una abstracción más realista del funcionamiento, animación y movimiento del DRONE en el ambiente virtual.

1.1. Identificación del Problema

Mejorar la calidad de vida de las personas mayores utilizando un sistema que involucre la integración de varias tecnologías puede ser una solución o una base para futuras investigaciones que pretendan ayudar a este grupo específico de personas. A continuación se describen los problemas que se pretende solventar en este trabajo, los mismos que se han clasificado acorde a cada factor que tiene relevancia en el desarrollo y búsqueda de solución de este trabajo.

Con respecto al usuario final

La salud y la calidad de vida de los adultos mayores se va deteriorando con la edad, esto es debido a varios factores como el sedentarismo y la falta de actividad física en la que

se ven envueltos a su edad. El poder promover de alguna manera la movilidad, el deporte y el entretenimiento de los adultos mayores es una forma de mejorar su salud y su calidad de vida.

Con respecto al uso de la tecnología por el usuario final

Para la mayoría de personas mayores, la tecnología actual a veces les resulta difícil de entender por lo que aplicaciones que usualmente utilizan las personas jóvenes como medios de entretenimiento u deporte pueden resultar poco viables para enfocarse en un grupo de personas mayores.

Con respecto al uso de Simuladores de UAVs

Hoy en día, los UAVs se lo puede encontrar en varios tamaños y marcas, sin embargo, sus precios suelen ser poco accesibles para la mayoría de usuarios y fracasar en la maniobra, ya sea por motivos ambientales o error humano puede resultar en una pérdida económica o daños colaterales. Este problema se ve solucionado a través de simuladores, ya que a través de entornos virtuales es posible usarlos como entrenamiento o practica antes del uso con dispositivos reales, en este caso de Drones. Un simulador para maniobrar Drones puede resultar difícil se lo ejecuta usando dispositivos de mando reales, por esta razón se a propuesto el realizar un simulador que gestione las acciones de mando a través simples movimientos de brazos.

1.2. Objetivos

El objetivo general propuesto en este trabajo, es generar una solución tecnológica que solvete los problemas anteriormente mencionados. En este contexto se propone generar una aplicación que fomente el movimiento de las personas mayores a través de un sistema de entretenimiento portable, inmersivo y de fácil utilización. En esta medida el objetivo recae en el diseño, desarrollo e implementación de un Simulador de Realidad Virtual para el manejo de un vehículo aéreo no tripulado utilizando sensores externos en computadoras portables y tecnología BLE como medio de comunicación inalámbrica. El desarrollo se lo realiza sobre el motor Unity3D, el cual permite diseñar y generar todo el ambiente Grafico, Física y Matemática implicada en el funcionamiento de la simulación, incluyendo la integración de la tecnología BLE a través de un Plugin JAR para la

comunicación con las nano computadoras y Google VR para generar una visualización inmersiva utilizando cualquier tipo gafas VR.

Al ser un simulador controlado por el movimiento de las manos, se disminuye al mínimo la dificultad de manejo de la aplicación usando esta tecnología y Al ser una aplicación de realidad virtual el usuario visualizara el entorno a través de unas gafas lo que permite que sus manos estén libres únicamente para mover las nano computadoras y así maniobrar el dron virtual.

El enfoque del simulador en un dron virtual da la ventaja de ser maniobrado de forma fácil para los adultos mayores y al ser un dispositivo que viaja por vía aérea, el problema de esquivar obstáculos terrestres se reduce a cero, además que en el simulador el ambiente es optimo al no verse afectado por torrentes de aire, lluvia o cualquier otro tipo de efectos naturales adversos.

El objetivo analizado y planteado anteriormente solventa los problemas identificados dando lugar a objetivos específicos que se mencionan a continuación.

Diseñar un entorno virtual realista, amigable y de confort visual para una mejor acogida por parte de los adultos mayores

Un ambiente realista y simple genera un desgaste visual menor y por esta razón se pretende crear un escenario atractivo, tranquilo y natural que no obstaculicen la vista del usuario con objetos extras.

Diseñar una arquitectura tecnología que permita integrar y controlar un simulador virtual a través de sensores de red de área corporal de manera inalámbrica, con el objetivo de facilitar el maniobre del Drone a los adultos mayores

Este objetivo consiste en diseñar una arquitectura tecnológica que permita integrar las tecnologías de realidad virtual, Bluetooth (BLE) y manejo de sensores IMU (Inertial Measurement Unit) a través de dispositivos portables (Raspberry Pi) con el objetivo de facilitar el uso y fomentar el movimiento de brazos durante el uso del simulador, una arquitectura de este tipo permite usar dispositivos portables sin cables, lo que facilita el movimiento tanto de los brazos como del cuerpo, promover la movilidad de los adultos mayores es una forma eficaz de mejorar su salud al disminuir el sedentarismo.

Adaptar el simulador para que posea 2 modos de visualización y 2 modos de dificultad de maniobra con el objeto de disminuir la monotonía y promover el entretenimiento y uso del sistema

Se pretende mejorar la calidad de vida de los adultos mayores al ofrecer un sistema de entretenimiento a través de varios modos de uso en un mismo simulador, los adultos mayores pueden tener problemas visuales y debido a esto se ha visto necesario el ofrecer 2 modos de visualización al momento de maniobrar un Drone, de esta manera cada usuario puede disfrutar del uso del simulador con la configuración que prefiera, adicionalmente a esto se pretende generar 2 formas de control del Drone, una simple, en donde el adulto no requiere realizar demasiados movimientos para lograr controlar al Drone en su totalidad y un modo mas complejo en donde el adulto requiere de mayor destreza para lograrlo, esto fomenta la auto competitividad y contribuye al estimular las funciones cognitivas (Coordinación Ojo-mano) del usuario además de fomentar la memoria espacial.

El movimiento esta relacionado con el deporte y el entretenimiento con la diversión, un sistema que permita al adulto estar activo y entretenido es un sistema que mejora su calidad de vida y el buen vivir.

Desarrollar y testear una aplicación nativa en Android que utilice los mismos dispositivos portables que el simulador virtual para maniobrar un dron real

Como objetivo específico final se pretende desarrollar una aplicación nativa en Android que haga uso de dispositivos portables para maniobrar un dron real, esto con el objetivo de comparar la funcionalidad entre ambas aplicaciones y realizar una abstracción más realista del movimiento y animación que debe tener el dron virtual para ser lo más realista posible, adicionalmente se valorara el uso de un control inalámbrico bajo la tecnología BLE con sistemas de hardware reales y posibilitar el estudio en futuras investigaciones.

1.3. Estructura de la tesis

Una vez analizados los objetivos definidos anteriormente, el presente trabajo comienza por presentar en el Capítulo 2, investigaciones relacionadas con este trabajo, donde se aprecia las tecnologías similares a las propuestas en este trabajo orientadas a mejorar la calidad de vida de las personas.

En el Capítulo 3 se define la arquitectura general que comprende todo el sistema de Simulación Virtual, donde se detalle el funcionamiento y las tecnologías involucradas en el mismo.

En el Capítulo 4 se explica el diseño gráfico 3D y la integración con las librerías de Google VR para generar un ambiente virtual, lúdico e hiper-realista, además se explica la arquitectura de software que comprende el desarrollo y análisis de 2 módulos, uno perteneciente a un plugin nativo desarrollado para proveer de servicios BLE al simulador y el otro que corresponde a la lógica de funcionamiento y maniobra del dron virtual.

El Simulador está configurado para maniobrar al dron a través de los Raspberry Pi los cuales serán analizados en el Capítulo 5, donde se explicará su arquitectura de software y configuración para que funcionen como mandos remotos, además se explicara el modo de uso que se les dio a los sensores IMU de la placa Sense HAT y se detalla la comunicación de los mismos con el celular (Simulador), la interacción entre el usuario con el sistema y las pruebas de latencia entre la comunicación Cliente – Servidor.

Una vez concluido el sistema virtual se procederá a explicar en el Capítulo 7 el desarrollo de la Aplicación Nativa que permite maniobrar un Dron real con los mismos mandos que se maniobra el Dron virtual del simulador, se detalla la arquitectura, comunicación y pruebas de funcionamiento utilizando el Phantom 4 como dispositivo UAV de estudio.

En el Capítulo 8 se describen un conjunto de pruebas realizadas en un grupos personas mayores, se detalla un cuadro comparativo y resultados obtenidos.

Para terminar en el Capítulo 9 se describe las conclusiones, recomendaciones y trabajo futuro relevantes con este trabajo.

2. Trabajo Relacionado

La evolución de las tecnologías de la información y de comunicación tanto a nivel de software como hardware han sido la fuente de grandes investigaciones y han dado lugar al desarrollo de sistemas automáticos, inteligentes y sobretodo útiles en diversas áreas de las ciencias.

La computación gráfica ha llevado la industria de videojuegos a los más exigentes estándares de diseño y producción, la capacidad gráfica de los escenarios son cada vez más realistas sin mencionar que los sistemas de hardware tanto de computadoras de escritorio como de teléfonos celulares también se han incrementado, esto permite cada vez más poder generar entornos más realistas usando dispositivos de hardware más pequeños, la tendencia de la computación gráfica intenta cada vez más de generar entornos virtuales y formas de visualización más inmersivos, como es el caso de las aplicaciones de realidad virtual, este tipo de aplicaciones están diseñadas para ser usadas utilizando equipos HMD (Head mounted displays) o gafas VR que permiten incrementar el rango de visión a base de una visualización en 2 cámaras, lo que da lugar a una visualización panorámica, que aumenta la sensación de realismo y mejora así la experiencia del usuario. Esta tecnología a sido estudiada y direccionada en varios aspectos, por ejemplo en [10], se analizo la viabilidad de manejar TICs a través de entornos de realidad virtual en personas con discapacidad motora, una de las ventajas de la realidad virtual es que la mayoría de las acciones por parte del usuario solo requiere del movimiento de la cabeza, por ende este trabajo puede ser una base para proyectos de investigación de apoyo a este tipo de personas. En el área de la medicina, [11] se aprovechó esta tecnología para reducir el nivel de fobia a las alturas a través de terapias en ambientes virtuales, este estudio demostró la reducción de fobia medida en base a parámetros que determinan el nivel de miedo que posee una persona. Se han desarrollado aplicaciones que fomentan la rehabilitación de pacientes promoviendo el movimiento de brazos o piernas a través de sistemas virtual, es el caso de [12], un sistema lúdico que utiliza el Kinect de Microsoft para detectar los movimientos de los miembros superior de un paciente y así fomentar su rehabilitación a través de interacciones reales con el sistema. Ver Figura 1.



Figura 1 - Sistema Virtual para rehabilitación de miembros superiores [12]

El poder mejorar la calidad de vida de los adultos mayores, en colaboración con la tecnología, es uno de los puntos de interés de la atención médica. Se ha evidenciado que actividades físicas o mentales como juegos de mesa, o aplicaciones tecnológicas interactivas están asociados a reducir el riesgo de fatiga mental y emocional [13].

Se han desarrollado varios estudios para examinar la importancia de la calidad de vida y la salud en la vejez. La importancia de la actividad física o la movilidad es una forma de mejorar las condiciones orgánicas y la degeneración física. Los estudios muestran el efecto beneficioso de un estilo de vida activo, especialmente si la autonomía física y funcional puede permanecer en la vejez. La reducción de la capacidad fisiológica evidente en el envejecimiento puede afectar la calidad de vida. Se ha demostrado que la actividad física produce un aumento del rango de movimiento, de la fuerza muscular y de los niveles funcionales de autonomía [14]. El ejercicio regular ayuda a minimizar los efectos fisiológicos de un estilo de vida sedentario, aumentando la esperanza de vida activa al limitar el desarrollo y progresión de condiciones crónicas e incapacitantes. También hay beneficios psicológicos y cognitivos derivados de la participación regular en el ejercicio de los adultos mayores [15].

Las investigaciones realizadas para solventar el problema de sedentarismo en las personas mayores abarcan varias tecnologías y atacan el problema de varios modos, por ejemplo, el proyecto propuesto en [16], define un entorno musical interactivo para una mejor expresividad a través de movimientos corporales, el sistema anima al usuario a ejercer movimientos a través de sonidos, resultados y análisis del proyecto reveló amplia aceptación y funcionalidad, Los ancianos se ven muy feliz de usar el sistema, con sus

movimientos de cabeza, brazos y espalda, pueden producir varios tipos de sonidos musicales.

Otro proyecto interesante se lo describe en SSN [17], ellos usan un enfoque social para mejorar la calidad de vida de las personas mayores a través de tecnologías de la información y comunicación (TIC). Los autores propusieron una amigable Interfaces para ver contenido web, como videos, juegos, actividades físicas guiadas con el objetivo de fomentar la actividad de forma remota, hoy en día muchas personas buscan rutinas de ejercicio en YouTube como reemplazo de un guía de gimnasio. Su propuesta tiene varias ventajas y se centra principalmente en la eliminación de la sensación de estar solo y sedentario. En la Figura 2, se muestra una imagen de su usabilidad.



Figura 2 - Pruebas de usabilidad SSN [17]

En “Games for a Better Life” [18] en colaboración con SASCO (Singapur Amalgamated Services Co-operative Organization), exploró los beneficios físicos y psicológicos en los adultos mayores utilizando el Nintendo Wii, a través de juegos deportivos y juegos de memoria, mientras que otro grupo realizaba los mismos juegos de manera tradicional. El análisis de los resultados mostró que las personas mayores que jugaron a través de videojuegos mejoraron significativamente su bienestar general en comparación con el grupo de personas que realizaron los juegos de manera tradicional,

concluyendo que la herramienta utilizada contribuye positivamente a este grupo de personas.

Otro trabajo similar fue realizado por Gerling [19], donde se muestra que con la cámara Kinect, se realizó una aplicación dirigida a las personas mayores para que puedan ejercitarse y divertirse al mismo tiempo. En el juego se plantearon una serie de ejercicios estáticos y dinámicos para que el usuario los imitase, ver Figura 3. Al final del juego se presentaron los porcentajes de efectividad: 77% de aprendizaje, índice de felicidad 62%, cansancio después de hacer el juego 47% y dificultad de uso 55%. Los autores concluyen que los videojuegos que utilizan el movimiento controlado, proporcionan una mayor motivación para los ancianos, atrayéndolos a la realización de la actividad física para tener una vida más activa y duradera.



Figura 3 - Video juego de movimiento de cuerpo entero [19]

No solo aplicaciones de realidad virtual sino también aplicaciones mas simples, y lúdicas han sido usadas enfocadas a este grupo de personas como es el caso de [20] donde se discuten las características que pueden influir en la experiencia de un anciano usando dispositivos de videojuegos, algunos de ellos pueden ser: demografía, habilidades sensoriales y motoras, usabilidad, adaptabilidad, costo e incluso la motivación de cada persona. Para ello, los autores experimentaron con diferentes dispositivos y plataformas como Nintendo DS, Nintendo Wii, PlayStation con el fin de realizar diferentes actividades, probando la capacidad del cerebro, los movimientos físicos, la experiencia tecnológica y la motivación obtenida de un grupo de personas mayores. Finalmente se hizo evidente la aceptación de estas herramientas tecnológicas, ya que el resultado fue una experiencia

agradable en la ejecución de las actividades y mejoraron la calidad de vida tanto física como mental.

3. Arquitectura General

A continuación se describe la arquitectura general propuesta del simulador de realidad virtual, esta arquitectura esta compuesta por un dispositivo celular (Servidor), el cual contiene la aplicación del simulador que fue desarrollado utilizando librerías de Google VR para dar soporte visual al ser utilizadas conjuntamente con gafas VR , dos Raspberry Pi (Clientes), que son utilizados como controles para maniobrar al dron virtual y poseen una placa Sense HAT integrada, la misma que contiene los sensores de movimiento (IMU). El Servidor y los Clientes se comunican a través de tecnología Bluetooth Low Energy (BLE) que es gestionada de manera nativa a través de un Plugin nativo desarrollado en Java para proveer de servicios BLE al simulador y permitir la interconexión de dispositivos, tal y como se muestra en la Figura 4.

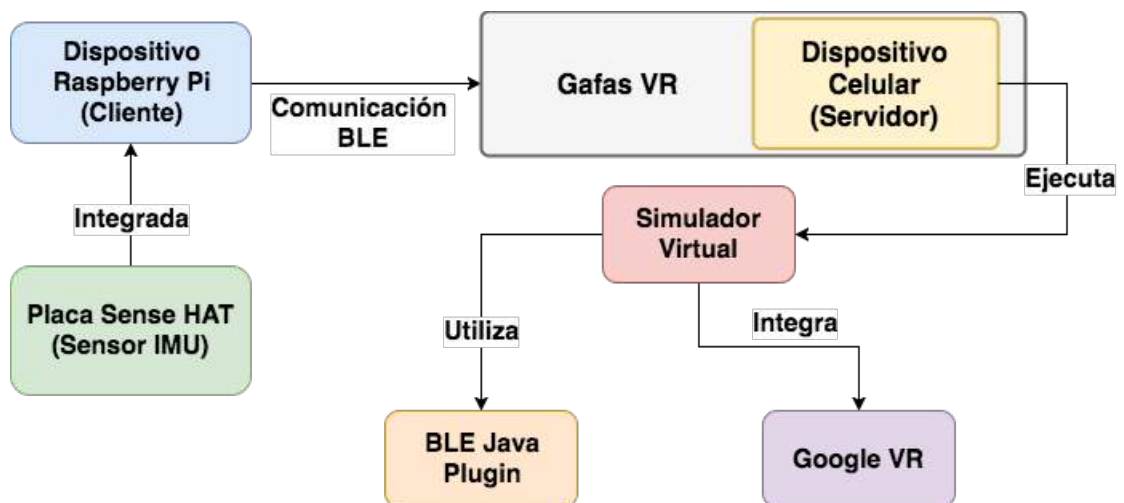


Figura 4 - Diagrama de la arquitectura general

Cada cliente realiza un escaneo en busca del servicio levantado por el servidor, esto se logra identificando un UUID único que es establecido tanto en las aplicaciones clientes como en el servidor, una vez encontrado, la conexión es automática y cada cliente procede a enviar a través de un JSON la información que obtiene de los sensores IMU al servidor, esta información se genera al mover cada uno de estos dispositivos utilizando el movimiento de manos y brazos como se representa en la Figura 5.

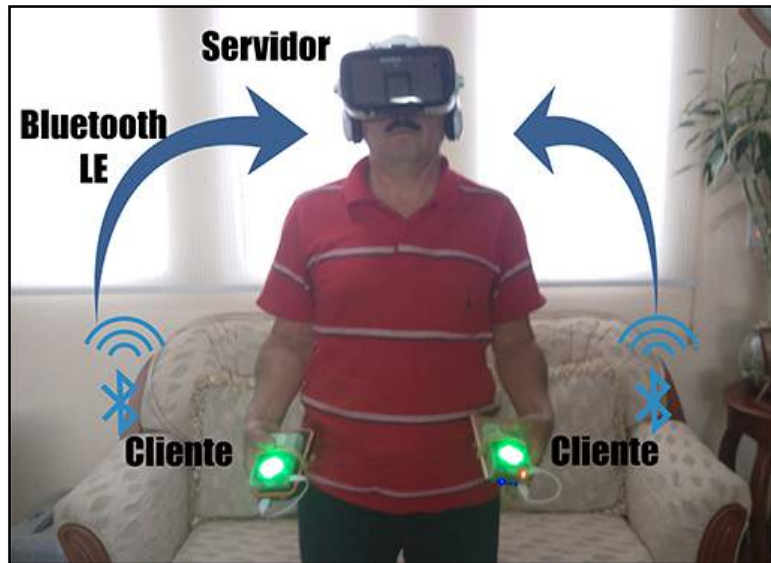


Figura 5 - Diagrama de funcionalidad

3.1. Tecnologías Utilizadas

A continuación, se detallan cada una de las tecnologías y dispositivos utilizados en el diseño de esta arquitectura.

3.1.1. Teléfono Celular Gama Alta

El simulador de realidad virtual está diseñado para ser ejecutado en cualquier teléfono Android de gama alta, el mismo que debe cumplir con ciertas características para su correcto funcionamiento, las cuales son las siguientes:

- Debe poseer como mínimo 2GB de RAM para una ejecución fluida;
- Debe poseer un procesador gráfico decente con soporte de ya que una aplicación de realidad virtual debe procesar la escena 2 veces simultáneamente y esto podría dar lugar a un LAG (Retardo en la imagen debido a un bajo procesamiento grafico);
- Debe tener un sistema operativo Android 5.0, en adelante, esto es porque las funciones de BLE a nivel de servidor (Advertising Callbacks) fueron añadidas a partir de esta versión;
- Debe poseer sensores de giroscopio, acelerómetro, barómetro y compas, esto es debido a que el SDK de Google VR utiliza estos sensores para realizar los movimientos de la cámara según el movimiento de la cabeza de la persona;

- Debe poseer un tamaño recomendado de 5.5 pulgadas, esto garantiza una pantalla más amplia y da lugar a una inmersión más completa.

3.1.2. Raspberry Pi 3B

Raspberry Pi 3 modelo B (Figura 6) es la última versión de la familia de este tipo de computadoras portables, se escogió este dispositivo para ser usado como mandos del simulador por las características que posee y porque es directamente compatible con la placa Sense HAT que ofrece los sensores necesarios y una matriz de leds para visualizar los estados de conexión con el Servidor. Esta computadora ha sido utilizada para un sinnúmero de proyectos tecnológicos [21] gracias a las características que posee.



Figura 6 - Raspberrry Pi 3 B

Las características más relevantes de este dispositivo en este trabajo son las siguientes:

- 1.2GHz 64-bit quad-core ARMv8 CPU;
- Bluetooth 4.1;
- Bluetooth Low Energy (BLE);
- 1 GB RAM;
- 4 USB ports.

Este dispositivo es extremadamente ligero y funciona con solo 5 voltios de energía lo que significa que puede ser conectado a un cargador externo de celular para su funcionamiento. De esta manera el RPI es un dispositivo perfecto para ser usado como de mando en el simulador.

3.1.3. Placa Sense HAT

La placa Sense HAT es una placa diseñada para ser integrada al Raspberry pi, ver Figura 7 y fue diseñada inicialmente para ser usada en las misiones de Astro pi, donde fue lanzada a la estación espacial internacional en el 2015 y ahora es accesible para ser comprada por cualquier usuario, generalmente es usada para proyectos de investigación a través del Raspberry Pi.



Figura 7 - Placa Sense HAT integrada al Raspberry Pi

Las características más relevantes de esta placa son las siguientes:

- Posee una matriz de leds de 8x8, que es útil para desplegar contenido visual acorde a las necesidades de cada proyecto, en el caso de este trabajo se lo usa para identificar el estado de conexión con el servidor a través de la visualización de circunferencias en 3 colores diferentes;
- Posee un sensor IMU (Unidad de medida inercial) compuesta por Giroscopio, Acelerómetro, Magnetómetro y Barómetro, este dispositivo utiliza todos estos sensores para realizar el cálculo de 3 rotaciones particulares, (Yaw, Pitch y Roll) las cuales se utilizan para determinar los movimientos del DRONE.

3.1.4. Bluetooth Low Energy

Existen un montón de protocolos Wireless para uso en tecnologías de información y comunicación TICs, pero lo que hace que BLE sea tan interesante es que es el más sencillo

para implementar la comunicación entre pequeños dispositivos y en cualquier plataforma móvil actual (iOS, Android, Windows phones, Raspberry Pi, etc.) además de su muy bajo coste de energía.

BLE es la nueva especificación que se introdujo desde la versión Bluetooth 4.0 [22] por Bluetooth Special Interest Group (SIG). Se ha diseñado como una tecnología complementaria a Bluetooth clásico para garantizar un consumo de energía bajo, y menor tiempo de conexión. A pesar del uso de la misma banda de frecuencia 2.4GHZ y las similitudes compartidas, BLE es considerado un nuevo estándar ya que esta orientado a aplicaciones diferentes. BLE está diseñado para la transmisión de pequeñas cantidades de datos (tiempos de transmisión muy pequeños) y por lo tanto de ultra-bajo consumo de energía. No está pensado para mantener una conexión entre dispositivos por un largo tiempo transmitiendo grandes cantidades de datos a alta velocidad. Esto permite que los dispositivos estén activos solo cuando se les pide la transmisión de datos. En la Tabla 1 se muestra las diferencias mas relevantes entre ambas tecnologías.

Especificaciones Técnicas	Bluetooth Clásico	Bluetooth BLE
Radio Frecuencia	2.4 GHZ	2.4 GHZ
Distancia / Rango	10-100 metros	50 metros
Velocidad	1-3 Mbps	1 Mbps
Nodos conectados	7	7+
Security	56 a 128 bit	128 bit AES
Latencia	100+ ms	< 3ms
Trasmisión de voz	Yes	No
Topología de Red	Punto a punto, scatternet	Punto a punto, estrella
Consumo de energía	<30 mHA	< 20 mHA
Principales usos	Teléfonos móviles, auriculares, audio estéreo, automóviles, PCs, etc.	Teléfonos móviles, juegos, PCs, deporte y fitness, médicos, automotores, industriales, electrodomésticos, etc.

Tabla 1 - Bluetooth Clásico vs BLE

Como se puede apreciar en la tabla ambas tecnologías difieren en algunas características. Para la implementación de esta tecnología en base a la propuesta de este trabajo, los aspectos más relevantes son la latencia, la cual BLE supera por mucho a la tecnología clásica, la topología, al ser estrella permite generar una arquitectura cliente – servidor con un amplio número de conexiones, como se observa en la tabla el número de

conexiones límite en la versión clásica es de 7 dispositivos hacia un dispositivo maestro, mientras que BLE no tiene esa limitante. Por último, el consumo de energía es mínimo y al utilizar computadoras de bajo consumo eléctrico, es más viable una implementación BLE. Es por esta razón que varios proyectos han adoptado la tecnología BLE como medio de comunicación entre dispositivos que requieren de un consumo muy bajo de energía, como es el caso de [23], un sistema inteligente para detección de automóviles, el cual integra placas Arduino, Raspberry Pi, teléfonos celulares, y sensores en una arquitectura con una comunicación a base de BLE desde la versión Bluetooth 4.0.

Otro aspecto que se debe considerar al hablar de estas tecnologías es que la configuración clásica y la nueva (BLE), no son compatibles directamente, esto se debe a su arquitectura y protocolos de comunicación, en la Figura 8 se muestra un gráfico con las diferencias arquitectónicas entre ambas tecnologías.

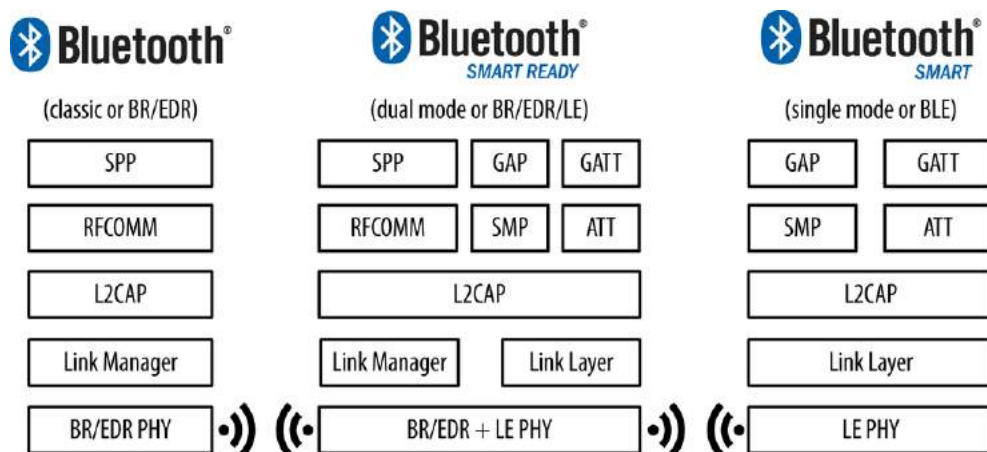


Figura 8 - Comparación arquitectónica de las tecnologías Bluetooth [24]

A continuación se define cada una de las tecnologías Bluetooth.

- **Bluetooth clásico (BR/EDR):** implementa el estándar inalámbrico clásico Bluetooth;
- **Monomodo (Bluetooth Smart):** implementa el estándar BLE y puede comunicarse con otros módulos monomodo y con los de modo dual pero no con los que sólo soportan BR/EDR;
- **Modo dual (BR/EDR/LE, Bluetooth Smart Ready):** implementa ambos estándares, BR/EDR y BLE y es capaz de conectarse con cualquier dispositivo Bluetooth.

Como se aprecia en el gráfico, existen algunos dispositivos que portan ambas configuraciones, esto con la finalidad de tener mayor compatibilidad y funcionalidad. La propuesta de este trabajo pone énfasis en la arquitectura BLE y sus capas las cuales se explican a continuación.

LE PHY (Physical layer)

La capa física contiene los circuitos de comunicaciones analógicos utilizados para modular y desmodular señales analógicas y transformarlas en símbolos digitales. El BLE puede comunicar en más de 40 canales a través de la frecuencia de 2.4000 GHz a 2.4835 GHz [25]. 37 de estos canales se utilizan para datos de conexión y los últimos tres canales (37, 38 y 39) se utilizan como canales de publicidad para establecer conexiones y enviar datos de difusión. BLE utiliza una técnica llamada espectro de dispersión de salto de frecuencia, en la cual la radio salta entre canales en cada evento de conexión tal como se muestra en la Figura 9. El valor del salto se comunica cuando se establece la conexión, por lo que es diferente para cada nueva conexión establecida. Esta técnica minimiza el efecto de cualquier interferencia de radio.

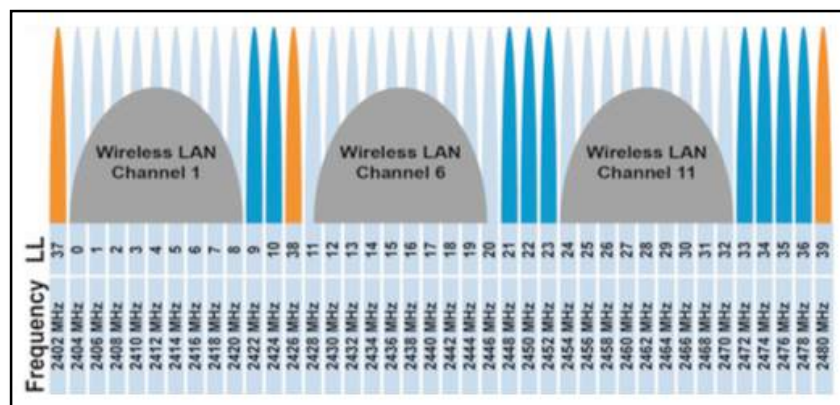


Figura 9 - Distribución de canales en BLE [22]

Link Layer

La capa de enlace es la parte que interactúa directamente con la capa física y suele implementarse como una combinación de hardware y software personalizados. La capa de enlace define los siguientes roles para sus dispositivos, basados en grupos lógicos:

- **Anunciante:** dispositivo que envía paquetes publicitarios;
- **Maestro:** Dispositivo que inicia una conexión con el anunciante.

La capa de enlace también se encarga de la dirección del dispositivo Bluetooth - un número de 48 bits que identifica de forma exclusiva un dispositivo.

L2CAP (Logical Link Control and Adaptation Protocol)

La capa L2CAP (control Lógico de enlace y protocolo de adaptación) está a cargo de dos tareas:

- Toma varios protocolos de las capas superiores y los encapsula en el formato estándar de paquetes BLE (y viceversa);
- Se encarga de la fragmentación y recombinación: toma grandes paquetes de las capas superiores y los descompone en trozos que encajan en el tamaño máximo de carga útil de 27 bytes de los paquetes BLE en el lado de transmisión, y viceversa, recibe múltiples paquetes que han sido fragmentados y recombinados En un solo paquete grande que luego se envía a la parte superior.

La capa L2CAP está a cargo de dos protocolos principales: el ATT (Attribute Protocol) y el SMP (Security Manager Protocol).

ATT (Attribute Protocol)

El (ATT) es un simple protocolo de cliente / servidor basado en los atributos presentados por un dispositivo. El ATT constituye la base del intercambio de datos en aplicaciones BLE. Un cliente solicita datos de un servidor y el servidor envía datos a sus clientes. Cada servidor contiene datos organizados en forma de atributos, a cada uno de los cuales se le asigna un identificador de atributo de 16 bits, un identificador universalmente único (UUID), un conjunto de permisos y un valor. El identificador de atributo es simplemente un identificador utilizado para acceder a un valor de atributo, mientras que el UUID se utiliza para especificar el tipo y la naturaleza de los datos en el valor. El cliente envía las solicitudes de escritura o lectura apropiadas, y el servidor responde según ellas.

Cuando un cliente desea leer o escribir valores de atributo desde o hacia un servidor, envía una solicitud de lectura o escritura al servidor con el identificador. El servidor responde entonces con el valor de atributo o una respuesta de recibo. En el caso de una operación de lectura, el cliente tiene que analizar el valor y comprender el tipo de datos basado en el UUID del atributo. Por otra parte, durante una operación de escritura, se

espera que el cliente proporcione datos que correspondan con el tipo de atributo y el servidor es libre de rechazar la operación si ese no es el caso.

El sistema que se propone en este trabajo debe mantener una comunicación BLE entre el dispositivo central (Celular) y 2 dispositivos periféricos (Raspberry Pi) mientras el simulador este activo.

SMP (Secure Manager Protocol)

Este Protocolo está compuesto por una serie de algoritmos de seguridad para generar e intercambiar claves de seguridad entre conexiones BLE. Este protocolo proporciona apoyo para los tres siguientes procedimientos:

- **Emparejamiento:** Proceso en el cual se genera una clave de cifrado común y temporal para poder pasar a una conexión cifrada segura;
- **Vinculación:** Secuencia de emparejamiento seguida por la generación e intercambio de claves de seguridad permanentes, que se almacenan en la memoria no volátil y que crean un vínculo entre dos dispositivos, lo que permite establecer conexiones futuras sin tener que llevar a cabo el proceso de emparejamiento de nuevo;
- **Cifrado de restablecimiento:** Este procedimiento define cómo usar las claves de seguridad almacenadas en cada dispositivo después de una conexión exitosa y usarlas en posteriores conexiones para restablecer una conexión cifrada segura sin tener que pasar los 2 procedimientos anteriores.

GAP (Generic Access Profile)

GAP es un acrónimo para el perfil de acceso genérico [26], y controla las conexiones y la publicidad en Bluetooth. GAP es lo que hace que un dispositivo sea visible para el mundo exterior y determina cómo dos dispositivos pueden o no, interactuar entre sí.

Roles

El GAP define varios roles para los dispositivos, los cuales se clasifican en centrales y periféricos.

- **Periféricos:** son dispositivos pequeños, de baja potencia, de bajos recursos, que pueden conectarse a dispositivos centrales mucho más potentes. Un

ejemplo de periférico es un glucómetro, un medidor de pulsaciones o un Raspberry Pi;

- **Centrales:** Un dispositivo central generalmente es un teléfono móvil o una Tablet que tienen una capacidad de proceso mucho mayor.

El GAP define 2 topologías de comunicación, La topología Broadcast en la que un dispositivo periférico envía información a varios dispositivos centrales sin establecer una conexión, y la topología en modo Conectado, que es cuando un dispositivo central establece una conexión con uno o más dispositivos periféricos, en donde ambos utilizan los servicios y características del GATT para comunicarse en ambas direcciones.

GATT (Generic Attribute Profile)

GATT define la manera en que dos dispositivos BLE pueden comunicarse usando los Servicios y Características[26]. La comunicación se realiza mediante un protocolo conocido como ATT, que se usa para almacenar los servicios, características y datos relacionados en una tabla usando identificadores de 16-bit para cada entrada en la tabla.

Establecer una conexión también es la única forma de permitir la comunicación bidireccional, en la que el dispositivo central puede enviar datos al periférico y viceversa.

Topología en entorno conectado

La Figura 10 muestra como los dispositivos BLE funcionan en un entorno conectado. Un periférico sólo puede conectarse a un dispositivo central (como un teléfono móvil) a la vez, pero el dispositivo central puede conectarse a varios periféricos.

Sin embargo, una vez que se establece una conexión entre un periférico y un dispositivo central, la comunicación puede tener lugar en ambas direcciones.

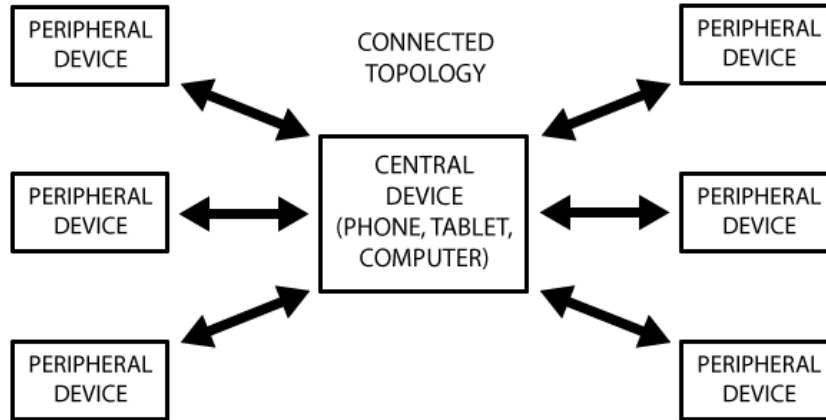


Figura 10 - Topología de comunicación en entorno conectado [26]

Transacciones GATT

El periférico se conoce como el servidor GATT, que contiene los datos de búsqueda ATT y las definiciones de servicio y características, y el cliente GATT (el teléfono / tableta), que envía solicitudes a este servidor [26].

Todas las transacciones son iniciadas por el dispositivo maestro, el GATT Client (central), que recibe la respuesta del dispositivo esclavo, el GATT Server (periférico).

Al establecer una conexión, el periférico sugerirá un 'intervalo de conexión' al dispositivo central, y el dispositivo central intentará volver a conectar cada intervalo de conexión para ver si hay nuevos datos disponibles.

La Figura 11 ilustra el proceso de intercambio de datos entre un periférico (el Servidor GATT) y un dispositivo central (el Cliente GATT), con el dispositivo central iniciando cada transacción.

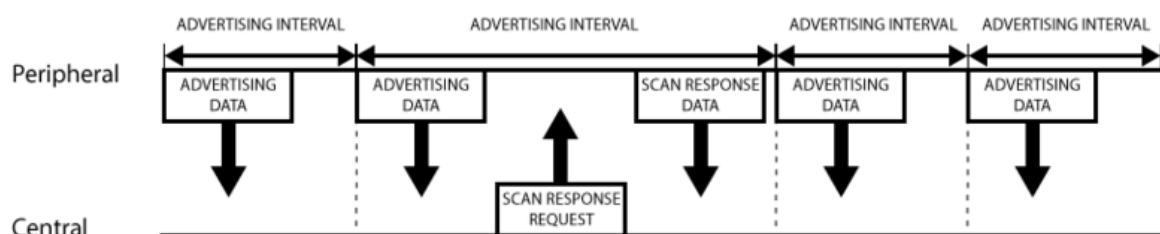


Figura 11 - Diagrama de mensajería cliente-servidor [26]

Servicios y Características

Las transacciones GATT en BLE se basan en objetos anidados de alto nivel denominados Perfiles, Servicios y Características, que se pueden ver en la Figura 12, donde se ilustra la estructura básica del GATT [26].

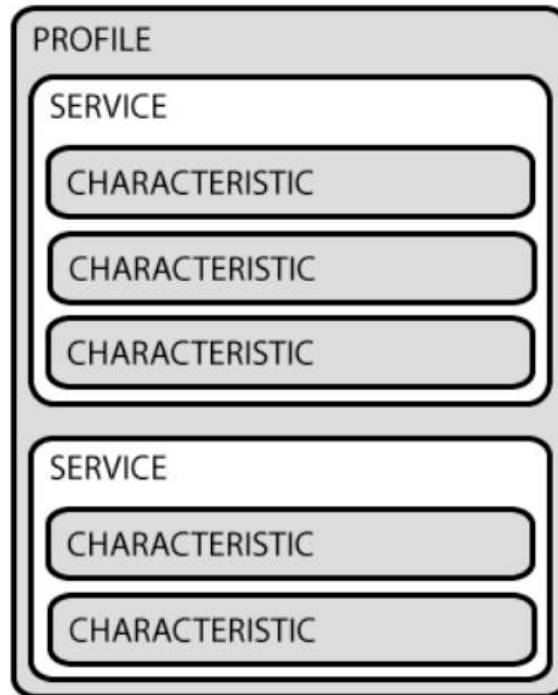


Figura 12 - Diagrama de la estructura GATT [26]

Perfiles

Un perfil es una colección predefinida de Servicios que ha sido especificada por el Bluetooth SIG o por el fabricante del periférico. Por ejemplo, el perfil de frecuencia cardiaca combina el servicio de frecuencia cardiaca y el servicio de información de dispositivos[26].

Servicios

Los servicios se utilizan para dividir datos en entidades lógicas y contienen trozos específicos de datos llamados características. Un servicio puede tener una o más características y cada servicio se distingue de otros servicios por medio de un ID numérico único denominado UUID, que puede ser de 16 bits (para servicios BLE adoptados oficialmente) o de 128 bits (para servicios personalizados)[26].

Características

Las características encapsulan un único tipo de dato. De forma similar a los Servicios, cada Característica se distingue a través de un UUID predefinido de 16 o 128 bits.

Las características son el elemento principal que vamos a usar para interactuar con nuestro periférico BLE. Cada característica posee propiedades, como el tipo de accesos a ella, la cual puede ser de lectura o escritura, adicionalmente poseen descriptores, los cuales son atributos que permiten definir a un valor característico. Y por último el valor, el cual contiene la información específica que se pretende transmitir.

En base a esta estructura, se a diseñado el perfil de comunicación con el servicio y las características necesarias para la transmisión de datos de los RPI al Simulador, el mismo que se lo puede apreciar en la Figura 13.

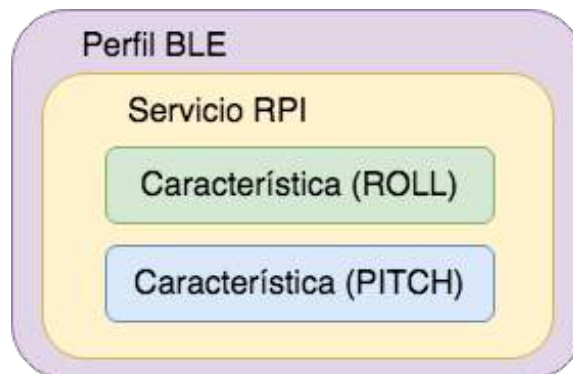


Figura 13 - Diagrama de la estructura GATT usada en el simulador

4. Simulador Virtual

El simulador virtual es una aplicación desarrollada en la herramienta Unity3D y compilada para ser ejecutada sobre la Plataforma Android, esta aplicación tiene su propia arquitectura de software e integra varios componentes que permiten el funcionamiento de la manera esperada. este Capítulo será explicado en 4 partes, primero se hablará en términos generales acerca de la herramienta que se utilizó en el desarrollo de esta aplicación. Segundo, se hablará sobre la construcción del entorno virtual, el cual comprende todo el escenario 3D, y el circuito aéreo diseñado para proporcionar objetivos de vuelo para el Drone. Tercero, se realizará la integración con Google VR y la configuración para una visualización en 2 perspectivas. Cuarto, se explicará la mecánica y control del Dron, incluyendo los componentes que permiten la emulación de Física real a través del Motor de Unity3D y por último se explicará la arquitectura de software de la aplicación la cual integra un Plugin JAR desarrollado para dar soporte BLE al simulador y permitir gestionar la transmisión de datos desde los dispositivos portables (RPI).

4.1. Unity3D

Unity3D es un motor de juego desarrollado por Unity3D Technologies. Las principales características de la misma son su amplio soporte de plataformas y su facilidad de uso. El editor es visual, puede arrastrar y soltar objetos en la escena del juego y adjuntar componentes, como Terrenos, Luces, sonidos, script, etc. Los scripts pueden estar en lenguaje C# o JavaScript. Unity3D soporta una amplia gama de contenido “Add-ons” de entrada, ya sea de forma nativa o mediante Plugins de terceros, y soporte para varias tecnologías de VR, entre los que se encuentran el Google VR. Además, proporciona una tienda de contenido 3D gratuito y de pago para acelerar el proceso de creación de escenarios 2D y 3D, en esta medida se ha utilizado el store de Unity3D para obtener un modelo de un dron estilo militar y las herramientas del propio editor para la creación de un Terreno lo más realista posible.

4.2. Entorno Virtual del Simulador

El entorno tridimensional del simulador comprende la creación del escenario 3D, el modelo 3D del Drone y el circuito de objetivos aéreos, los mismos que son explicados a continuación.

4.2.1. Escenario 3D

Existen 2 maneras de generar una perspectiva tridimensional en la realidad virtual, la primera es a través de Imágenes Panorámicas [27], lo que se hace es generar un cubo de 6 imágenes de una imagen panorámica y posicionar la cámara principal en el centro, de este modo con la tecnología VR el cubo se visualiza como una esfera donde los bordes se suavizan dando el efecto visual característico de inmersión, este tipo de visualización es la más óptima en cuanto a rendimiento, realismo y facilidad de desarrollo se refiere, ya que al ser imágenes panorámicas del mundo real proporcionan una calidad visual casi igual a la realidad pero con el inconveniente de que la cámara principal siempre permanece estática, esto quiere decir que no es posible movilizarse a través del entorno de manera libre, para solventar este problema se debe generar cambios de posición de la cámara en otro cubo de 6 nuevas imágenes como lo hacen los recorridos 360 de Google, la otra ventaja es el rendimiento ya que el GPU del dispositivo solamente debe procesar 6 imágenes. La primera versión del simulador se la realizó bajo este mecanismo, generando un entorno muy realista a través de la adición de un efecto de sombra para el dron y un sistema de partículas para visualizar hojas tal y como se aprecia en la Figura 14.



Figura 14 - Escena generada a base de una imagen panorámica

Este efecto visual se produce a través de un Shader (Archivo de código que utiliza APIs gráficas OpenGL, DirectX o Vulkan para el procesamiento gráfico usando la GPU del dispositivo). Este Shader utiliza 6 imágenes generadas a partir de una imagen panorámica y utilizado a través de un material dentro de un componente conocido como Skybox, el mismo que al ser añadido a la cámara principal, esta genera un ambiente de fondo de la imagen panorámica. En la figura 15 se puede apreciar el material configurado con el Shader y las 6 imágenes que fueron usadas para este efecto, además se visualiza el proceso de conversión de la imagen panorámica en 6 diferentes imágenes a través de una herramienta gratuita de la tienda de Unity3D.

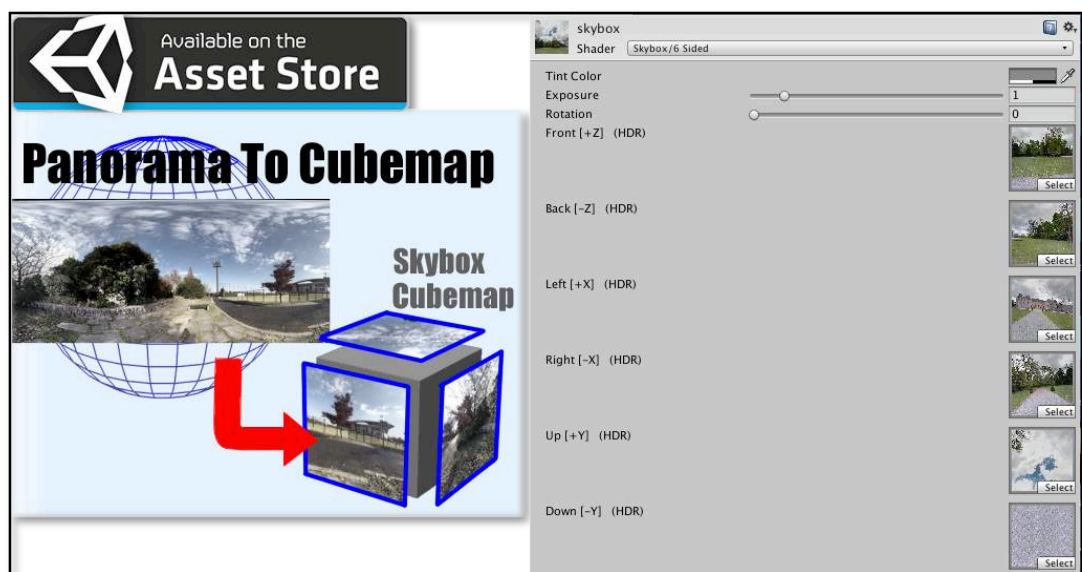


Figura 15 - Configuración del Skybox utilizando la herramienta “Panorama to Cubemap” [28]

Este diseño 3D fue fácil de implementar, pero como se mencionó no ofrece la posibilidad de que el observador avance libremente por el escenario, esto se debe a que no existe un mundo virtual como tal, solamente es un fondo panorámico que permite únicamente visualizar en 360 grados. Uno de los requerimientos funcionales del simulador es la posibilidad de maniobrar el Drone siendo espectador (cámara en tercera persona) y viendo lo que el Dron ve (cámara en primera persona).

Para poder configurar el sistema de cámaras en el simulador de manera que se pueda maniobrar tanto en tercera como en primera persona, es necesario generar un ambiente más complejo combinando un fondo panorámico con un terreno real a base de polígonos, para esto se utilizó el objeto Terrain de Unity3D y se lo editó con su propia

herramienta de edición dando lugar a una malla de polígonos a la cual se le agregó modelos 3D de rocas para aumentar el realismo, este es el segundo método de generación de ambientes virtuales, tiene la ventaja que se puede mover dentro del entorno pero tiene el inconveniente de que el GPU requiere procesar miles de polígonos y sus respectivas texturas, es por esto que para reproducir ambientes virtuales a base de polígonos se recomienda hacerlo en computadoras de Escritorio ya que poseen tarjetas gráficas lo suficientemente potentes como para procesar ambientes tan complejos y visualizarlos a través de los conocidos HMD (Head Mounted Display) como el OculusRift o el HTC Vive, por esta medida, para construir un Escenario3D que será visualizado a través de tecnología VR por un Celular, es necesario limitar el número de polígonos y evitar el uso de sombras, además de usar otras técnicas de optimización. El entorno 3D simula la sombra del dron para que no sea directamente procesada a través de la física de luces de Unity3D conocido como GI (Global Illumination), esto se debe a que este sistema de generación de sombras es dinámica y computacionalmente pesada para dispositivos móviles, en vez de esto se usó una configuración a base de cámaras y un componente que permite que una cámara específica muestre solamente una proyección de sombra sobre el terreno en las coordenadas X y Z en las que se encuentre el dron.

Dicho esto, el escenario 3D para el simulador se lo realizó como un paisaje montañoso y un fondo panorámico para visualizar las nubes a través de un Skybox tal como se aprecia en la Figura 16.



Figura 16 - Diseño del escenario virtual para el simulador

El Objeto de estudio es un dron el cual se lo obtuvo de la tienda de Unity3D, un dron genérico de 4 rotores, estilo militar tal y como se aprecia en la Figura 17.

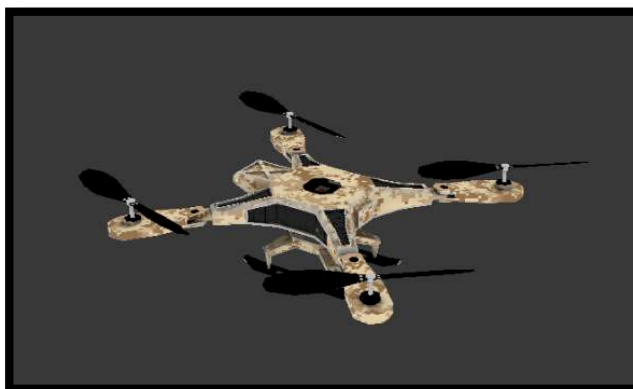


Figura 17 - Modelo 3D del dron

4.2.2. Circuito de objetivos aéreos

En esta sección se explicará la parte del diseño que corresponde a un circuito guiado de objetivos aéreos que aparecen secuencialmente durante todo el manejo del simulador, aunque el usuario puede controlar el Dron como desee ya que es un sistema realizado con el fin de fomentar el movimiento a través del entretenimiento, también existen objetivos aéreos que fueron puestos como modo de juego por 2 razones.

- **Primera:** Para aumentar la dinámica del uso del simulador. Este objetivo es exclusivo para ludificar y mejorar la experiencia de usuario;
- **Segundo:** Para poder evaluar el avance del usuario en base a la experiencia de maniobra ganada con relación al tiempo.

Este sistema de circuitos (Figura 18) se ha realizado a base de circunferencias con efecto GLOW color azul para que pueda ser apreciado fácilmente y distinguido entre el terreno que es predominantemente de color verde. Este circuito fue implementado en modo estático para poder evaluar el avance de una persona mayor con el continuo uso del sistema, de esta manera un circuito al no ser aleatorio, permite mejorar la memoria de la persona al tener que recordar y reconocer una ruta específica. Hay que tomar en cuenta que el grupo de personas al que se enfoca este trabajo no poseen la precisión y velocidad de reacción como una persona joven, es por esto que la modificación del circuito en un modo aleatorio sería enfocado más para el uso a modo de juego y no de simulador en personas

más jóvenes, este punto de vista se lo explica en el Capítulo 9 como recomendación para trabajos futuros.



Figura 18 - Circuito de objetivos aéreos

4.3. Configuración de Cámaras

El simulador virtual debe poder ser utilizado bajo 2 perspectivas de visualización, a través de una cámara en primera persona, en cuyo caso el observador mira y maniobra al dron como si lo hiciera en la vida real, por otro lado, la cámara en tercera persona permite que los ojos del observador estén en el dron, esto quiere decir que la experiencia es semejante a viajar en un avión. Estas configuraciones deben estar en modo VR usando las librerías de Google VR, de esta manera esta sección se dividirá en 2 partes, la primera hablará de la integración de Google VR en la escena y la segunda de la configuración de las cámaras en primera y segunda persona.

4.3.1. Cámara en modo VR

Google VR es un conjunto de librerías que permiten configurar un escenario tradicional en un sistema de realidad virtual de tal manera que sean visualizados a través de gafas VR como las Google Cardboard [29], este plugin puede ser descargado de la página oficial de Google. Una vez importado, esta herramienta proporciona un componente Figura 19 que se encarga de duplicar la cámara principal de la escena para convertirla en una doble cámara de visualización inmersiva, lo que caracteriza la realidad virtual.

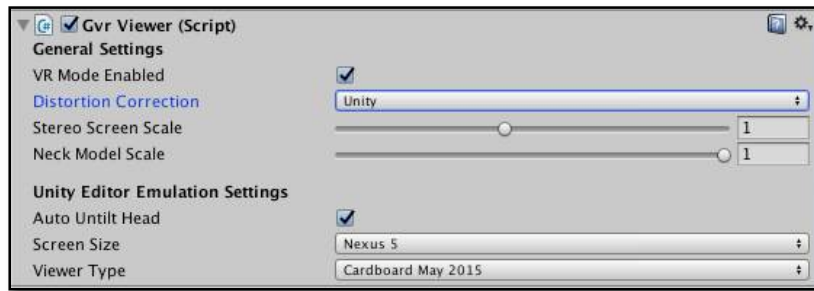


Figura 19 - Componente de Google VR

Una vez agregado el componente en la escena, la Cámara principal se auto configura de tal manera que la visualización esta ya lista para ser utilizada mediante unas gafas de VR tal y como se muestra en la Figura 20. Cuando se hace uso de estas librerías, al iniciar la aplicación Google pide al usuario instalar un paquete de soporte de servicios para Google VR, esto con el fin de optimizar la experiencia de usuario al ejecutar una aplicación que hace uso de sus librerías. En la figura se puede apreciar el modelo de las primeras gafas hechas a base de cartón, de ahí el nombre Google Cardboard, pero hoy en día hay varias marcas y modelos en diferentes materiales que mejoran la experiencia de usuario al proveer opciones de integración de audio, sujetador de cabeza e incluso soporte para ser usado con lentes, este último es muy necesario en este sistema ya que muchas personas adultas requieren el uso de lentes para poder ver correctamente. Por esta razón a propuesto utilizar un dispositivo que provee de todas las características mencionadas para evitar problemas de usabilidad e incomodidad.



Figura 20 - Aplicación de realidad virtual y gafas VR [29]

El sistema de rotación de cámaras que utilizan las aplicaciones de realidad virtual se basan en 3 ángulos de orientación que se obtienen a partir de los sensores de giroscopio del teléfono celular, estos movimientos permiten internamente rotar el conjunto de cámaras acorde al movimiento de rotación de la cabeza del usuario tal y como se muestra en la Figura 21.

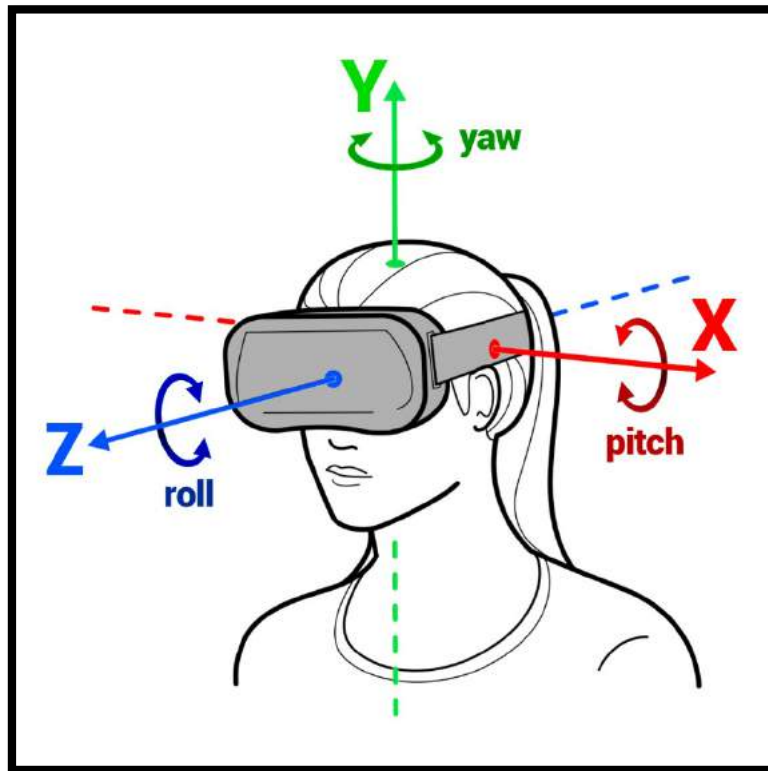


Figura 21 - Sistema de rotación rotación VR [30]

La librería de Google VR obtienen internamente todos los valores generados por los sensores IMU del teléfono, los movimientos de rotación de la cabeza se gestionan sobre cada uno de los ejes y sobrescriben los vectores de rotación de las cámaras en Unity3D para emular una cabeza real en un entorno virtual.

4.3.2. Cámaras en modos FPC & TPC

Una vez configurada la escena para que soporte VR es momento de configurar la cámara principal para que se posicione en diferentes puntos estratégicos y poder así proporcionar una simulación en 2 perspectivas diferentes. La primera es la configuración de cámara en tercera persona, en la cual el usuario mira al dron desde un punto lejano tal y como se observa en la Figura 22 y la cámara en primera persona es aquella en la que el usuario mira lo que el dron ve, tal como se muestra en la Figura 23.



Figura 22 - Cámara en tercera persona con soporte VR

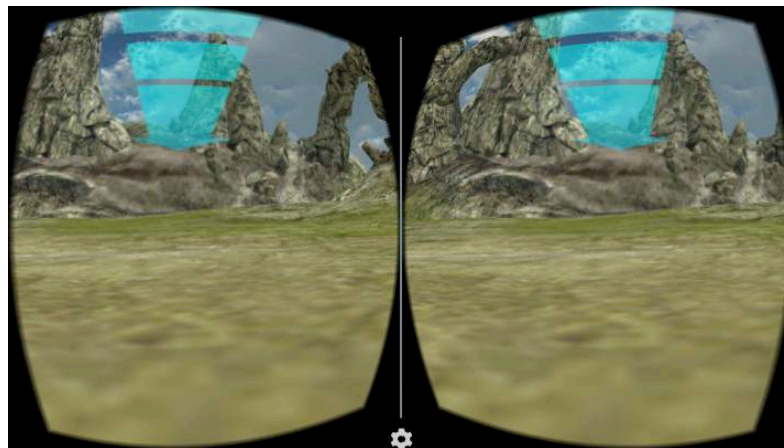


Figura 23 - Cámara en primera persona con soporte VR

La aplicación utiliza un componente “GlobalValues” desarrollado para que funcione con el patrón singleton y es el encargado de guardar la configuración a partir de una escena de opciones para después configurar las cámaras en el simulador. Por defecto, la cámara está configurada para funcionar en una perspectiva de tercera persona, y en caso de ser cambiada a primera persona se reubica a la cámara en la posición en la que se encuentra el GameObject “FPCSpawn” que fue ubicado en el punto exacto del dron donde se desea ubicar la cámara y adicionalmente se le emparenta con el mismo objetivo para que el dron se mueva juntamente con la cámara. Por último, se deshabilita los componentes que renderizan al dron y que dan movilidad a los rotores, esto se hace ya que en primera persona cualquier visualización del dron o de los rotores girando genera un obstáculo visual que deteriora la experiencia de usuario bajo esta configuración. En la Figura 24 se puede apreciar la jerarquía en la que se diseñó la parte gráfica del dron, el componente

“GlobalValues” con la opción de cámara en primera persona “FPC” y el fragmento de código encargado de pasar de una configuración en tercera “TPC” a primera persona “FPC”.

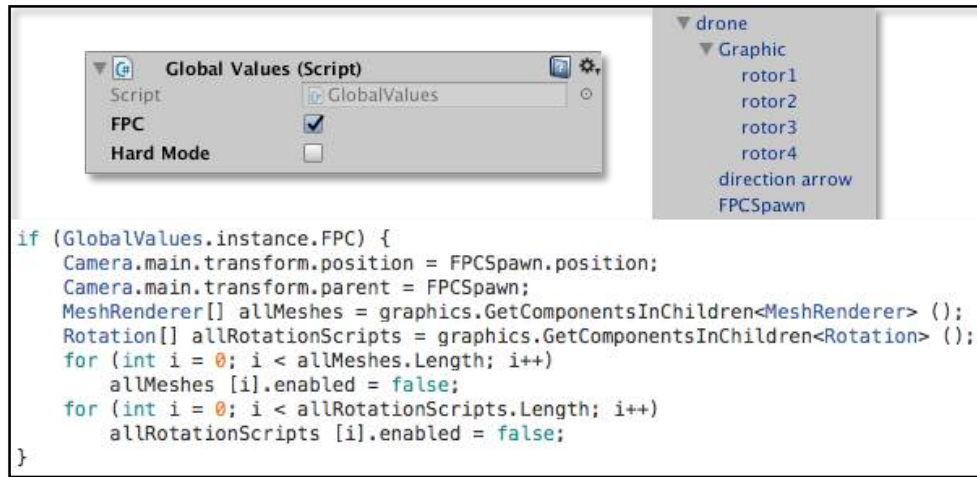


Figura 24 - Modificación de cámara en tiempo de ejecución

Este cambio de configuración en cámaras se ejecuta en tiempo de ejecución dependiendo de la opción escogida por el usuario, para este efecto se ha desarrollado una ventana previa a la aplicación del simulador que permite escoger el modo de visualización y la dificultad de maniobre para el Dron tal y como se muestra en la Figura 25.

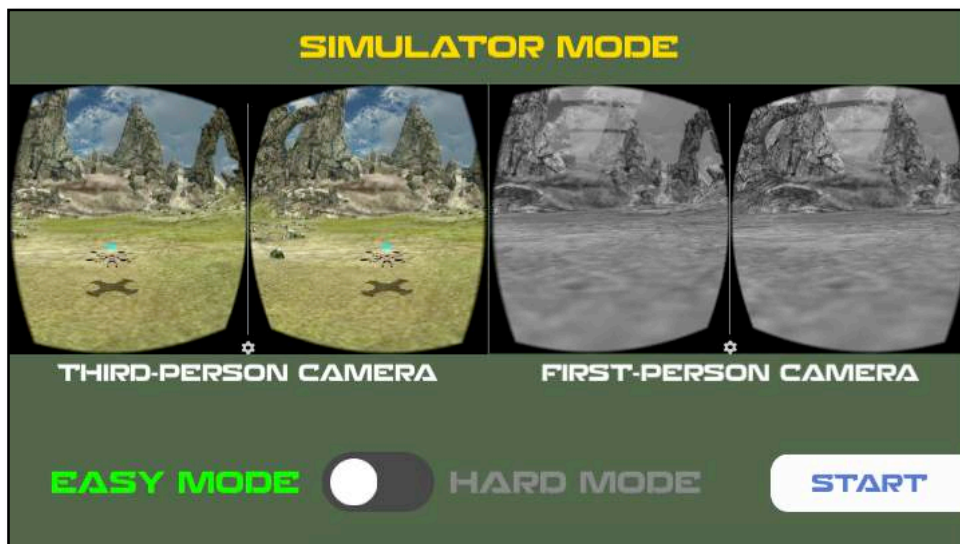


Figura 25 - Escena de configuración de cámaras

4.4. Mecánica y control del Drone

El dron es un tipo de UAV que puede realizar 8 tipos de movimientos basados en 4 funciones específicas, la función de acelerar o desacelerar que permiten el movimiento vertical del dron, las rotaciones Pitch y Roll que generan el movimiento horizontal y la rotación Yaw que permite girar al dron sobre su propio eje. Las rotaciones Pitch y Roll provocan que la nave gire frontalmente o lateralmente y con ello modificar la dirección de la fuerza de propulsión que generan los rotores para generar el movimiento de traslación.

El dron virtual fue diseñado para cubrir todos los movimientos, con excepción de 2 movimientos de traslación, hacia la izquierda y derecha, significa que en la mecánica se ignoran por completo las rotaciones Pitch y Roll, esta decisión fue tomada ya que con el resto de movimientos es capaz de funcionar sin ningún tipo de limitaciones además que las traslaciones laterales generan mareo en un entorno virtual.

Algunos artículos usan un modelo matemático bajo un sistema de coordenadas relativo al propio avión (local) [31]. Donde definen tres ejes perpendiculares de tal manera que el origen es el centro de masa, el eje X se dirige a través del frente del avión, el eje Y a través de la derecha del avión y el eje Z pasa a través de la parte inferior de la aeronave como se muestra en la Figura 26, por otro lado la mecánica del dron virtual fue construido bajo un sistema de coordenadas relativas al mundo (Global) como se muestra en la Figura 27, en la que el eje Y representará el movimiento vertical y el eje Z el horizontal, adicionalmente se muestra la rotación Yaw que es sobre el eje Y y permite una rotación horizontal sobre su propio eje, esta decisión se la tomo en base a la forma en que trabaja Unity3D y el sistema de coordenadas que usa al procesar la física en el Drone virtual.

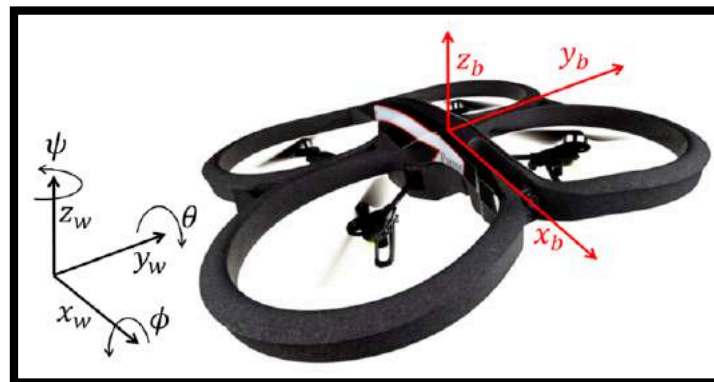


Figura 26 - Eje de coordenadas propuesto en “AR Drone Quadrotor” [31]

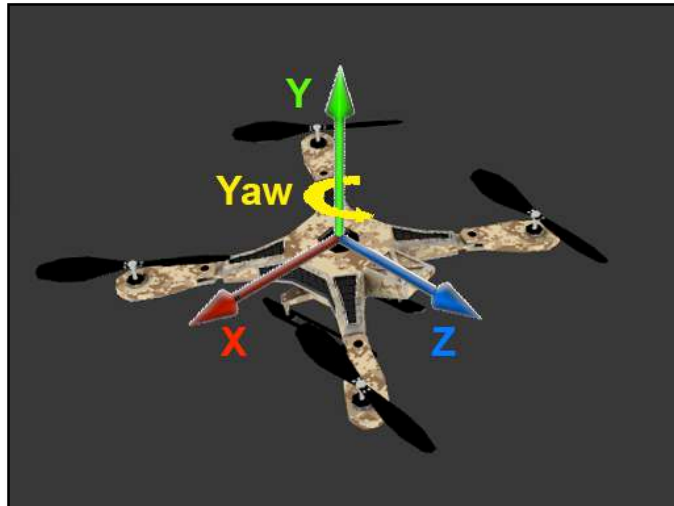


Figura 27 - Eje de coordenadas del drone propuesto en este trabajo

Con el objetivo de asemejar el movimiento al de un Drone real, la física utilizada en la rotación de los rotores se lo basó en [32] donde se configuró el modelo 3D del Drone de tal manera que cada rotor sea un polígono completamente independiente al resto de su malla, de esta forma se hizo posible girar cada rotor individualmente y se programó una función que anime a todo el dron a través de rotaciones bajo circunstancias específicas. Figura 28, si el dron está avanzando horizontalmente y se ejecuta una rotación Yaw, inmediatamente se provoca una ligera rotación hacia el eje al que este girando. Esto se logra realizando interpolaciones lineales en cada valor del vector rotación para suavizar el movimiento y asignar este nuevo vector interpolado al vector rotación principal del dron.

```

public void updateAnimation(DroneMotion value)
{
    switch (value)
    {
        case DroneMotion.Yaw_Left:
            rotationAngle.z = Mathf.Lerp (rotationAngle.z, maxRotation, Time.deltaTime * 5);
            break;
        case DroneMotion.Yaw_Right:
            rotationAngle.z = Mathf.Lerp (rotationAngle.z, -maxRotation, Time.deltaTime * 5);
            break;
        case DroneMotion.None:
            rotationAngle.z = Mathf.Lerp (rotationAngle.z, 0, Time.deltaTime * 5);
            break;
    }
    thisTransform.localEulerAngles = rotationAngle;
}

```

Figura 28 - Función que actualiza la rotación Yaw del drone

Para que el dron pueda reaccionar a las fuerzas Físicas de Unity3D se requiere 3 componentes esenciales. Ver Figura 29.

- **Transform:** Este componente es el encargado de almacenar los vectores posición, rotación y escala de un GameObject. A través de una instancia a este objeto se modifican los vectores posición y rotación para maniobrar el vehículo;
- **BoxCollider:** Este componente permite que el objeto responda a las colisiones internas con otros objetos, como el suelo, rocas, etc. Es indispensable para evitar que el objeto atravesara obstáculos;
- **Rigidbody:** Este componente permite que el objeto que lo porta reaccione a la física del motor Unity3D, esto significa que el objeto es afectado por la gravedad y cualquier fuerza que se le aplique directa o indirectamente.

Como se mencionó anteriormente se puede observar que el componente Rigidbody está configurado para impedir las rotaciones en los ejes X y Z ya que estos ejes son exclusivos para las rotaciones frontales y laterales que el Dron virtual no necesita hacerlas.

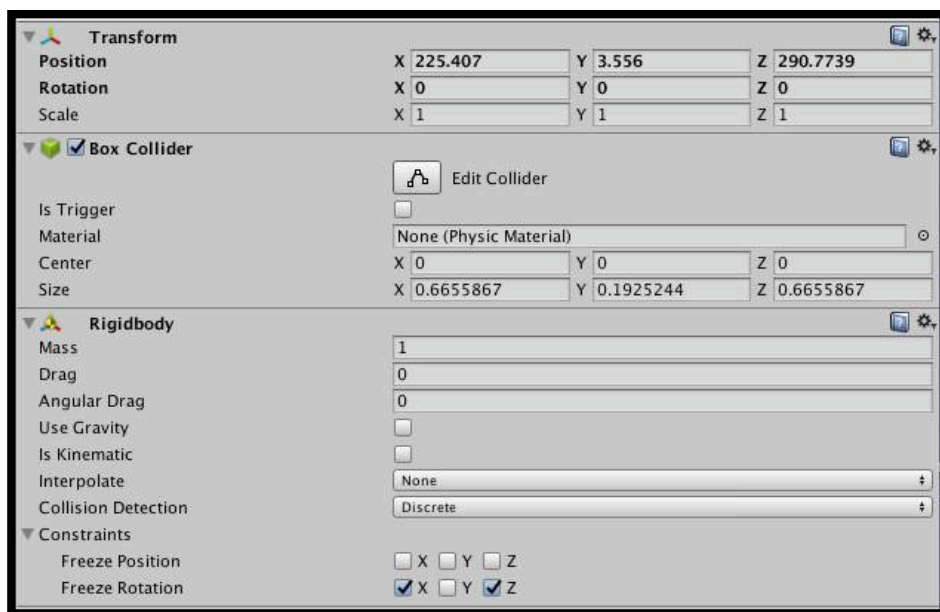


Figura 29 - Componentes se usan en el motor de física de Unity3D

La física interna de Unity3D a través de los componentes estudiados permiten una movilidad realista en un entorno simulado, ahora se explicará los eventos encargados de

que esto ocurra a través de 2 mecanismos de maniobre propuestos con la finalidad de dar al usuario 2 alternativas de control, las cuales son las siguientes.

- **Modo Fácil:** Este sistema de control permite maniobrar el dron a través de 4 movimientos, con rotación YAW a la izquierda con la mano izquierda y a la derecha con la mano derecha y movimientos verticales, ascender y descender con cualquier mano. En este sistema el dron se mueve horizontalmente hacia delante indefinidamente. Ver Figura 30;
- **Modo Difícil:** Este sistema de control es más complejo y permite maniobrar el dron a través de 6 movimientos, rotaciones sobre su eje “YAW” a la izquierda y derecha con la mano izquierda, movimientos horizontales con la mano derecha y movimientos verticales con ambas manos al mismo tiempo. Ver Figura 31.



Figura 30 - Diagrama de maniobre del dron a través de rotaciones del RPI en Modo fácil



Figura 31 - Diagrama de maniobras del dron a través de rotaciones del RPI en Modo difícil

4.5. Arquitectura de Software

Una vez analizada la parte gráfica del sistema será más fácil entender su arquitectura y funcionamiento. La arquitectura utilizada para el desarrollo del sistema de simulación está compuesta por dos capas: una capa de alto nivel y una capa de bajo nivel, representadas en la Figura 32. la capa de alto nivel está conformada por la parte funcional del simulador, y un módulo de gestión de servicios BLE, todo desarrollado en C# y la capa de bajo nivel es un archivo .JAR (librería de clases) que se desarrolló utilizando el lenguaje de programación Java y es administrado por el componente “BLE_Behaviour” de la capa de alto nivel.

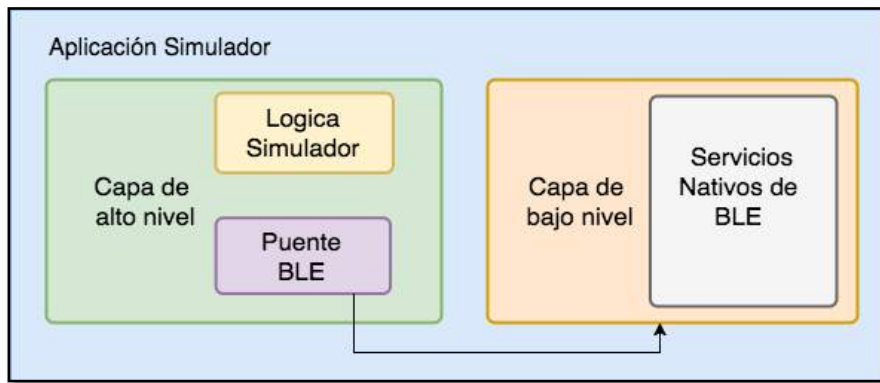


Figura 32 - Diagrama de la arquitectura de software del simulador

A continuación, se describen los componentes implicados en esta arquitectura, se empieza hablando de la librería nativa de servicios BLE que corresponden a la capa de bajo nivel del simulador y posteriormente se hablará de la capa de alto nivel que hace uso de Plugin desarrollado en la capa de bajo nivel.

4.5.1. Librería Nativa de Servicios BLE

Para poder proveer de servicios BLE al simulador virtual desarrollado en Unity3D, es necesario desarrollar primero un plugin JAR [33] que pueda ser usado desde Unity3D para la comunicación con los dispositivos Periféricos. Esto se debe a que Unity3D utiliza como lenguaje de programación C# a través de las librerías MONO, pero tanto el Framework MONO como el Framework de Unity3D no poseen librerías que permitan el uso de la tecnología BLE a través de C#.

Esta librería de clases (Plugin nativo de la capa de bajo nivel) es desarrollada en el IDE Android Studio y su estructura consta de 3 Clases Principales y una Clase (Entidad) que contiene la información de la Característica tal y como se muestra en la Figura 33.

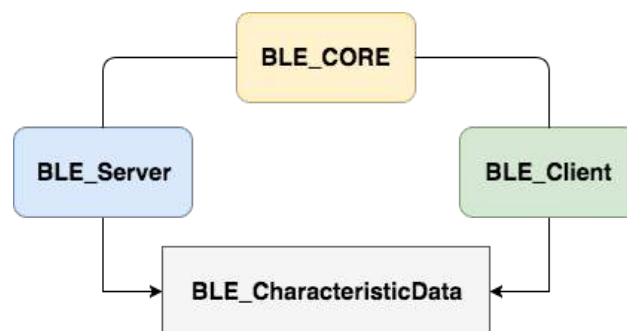


Figura 33 - Estructura de la capa de bajo nivel

La Clase BLE_CORE utiliza el patrón singleton ya que requiere ser accedida desde las clases BLE_Server y BLE_Client, esto se debe a que en la clase BLE_CORE se realizan las instancias de las clases que proveen los servicios de BLE pertenecientes al Framework de Android y deben ser accedidas a través de una instancia única. La Clase BLE_Client no es relevante para el funcionamiento del simulador, ya que las funciones de clientes las realizan los dispositivos periféricos, esto se lo ha desarrollado con el objeto de realizar las primeras pruebas de comunicación entre 2 teléfonos celulares.

La clase BLE_Core inicializa variables de comunicación que son enviadas desde el simulador como se muestra en la Figura 34.

```
public boolean initialize(String gameObjectName,
                        int advertiseMode,
                        int advertisePowerLevel,
                        int deviceMode,
                        int maxClientConnections)
{
    if(mBluetoothAdapter==null)
    {
        if(mBluetoothManager == null) {
            mBluetoothManager =
                (BluetoothManager) UnityPlayer
                    .currentActivity
                    .getSystemService(Context.BLUETOOTH_SERVICE);
        }
        mBluetoothAdapter = mBluetoothManager.getAdapter();
        this.gameObjectName = gameObjectName;
        this.advertiseMode = advertiseMode;
        this.advertisePowerLevel = advertisePowerLevel;
        this.deviceMode = deviceMode;
        this.maxClientConnections = maxClientConnections;
    }
}
```

Figura 34 - Código de inicialización de las funciones BLE

Los parámetros que se utilizan en la función describen la forma en la que el servidor establecerá la comunicación con los periféricos y se describen a continuación:

gameObjectName: es el nombre del GameObject activo en Unity3D donde se encuentran una lista de métodos que serán ejecutados desde el Plugin JAR como si fuesen Eventos, esto es posible gracias a una función que provee la librería de Unity3D (Unity.dll) que es utilizado como librería dentro del proyecto JAVA y permite enviar información en formato de texto desde JAVA hacia C# a través del siguiente método:

UnityPlayer.UnitySendMessage(String gameObjectName, String methodName, String parameter);

El primer parametro **gameObjectName** representa el nombre del objeto que tiene como función administrar y gestionar los mensajes que son enviados desde el Plugin JAR por el método antes mencionado.

El segundo parámetro **methodName** representa el nombre de la función que debe ser ejecutada dentro cualquier clase que posea el **gameObjectName**.

El tercer parámetro **parameter** representa el parámetro de la función **methodName** el cual es enviado en formato de texto, es por esta razón que todos los mensajes se envían en formato JSON, para poder enviar varios parámetros en una sola cadena de texto.

Como ejemplo de uso de este método mostramos el caso de conexión de un dispositivo donde alertamos al simulador a través de una función que recibe como parámetro la dirección MAC del dispositivo que se conecta.

Metodo en Unity3D dentro del GameObject BLE_Behaviour C#

```
void _OnClientConnected(string deviceName)
{
    onClientConnected (deviceName);
}
```

Funcion ejecutada desde el Plugin JAR

```
UnityPlayer.UnitySendMessage("BLE_Behaviour", "_OnClientConnected", device.getAddress());
```

De esta manera se notifica en Unity3D cada uno de los eventos generados por los servicios BLE desde el plugin JAR.

Una vez que un dispositivo periférico ha realizado una conexión exitosa empieza a enviar la información, la misma que es recibida a través de un objeto propio del SDK de Android "BluetoothGattCharacteristic" por medio de un evento que es ejecutado cada vez que el valor de una característica cambia.

El plugin se ha configurado para almacenar estas características en una Lista de objetos de la entidad "BLE_CharacteristicData" Figura 35. La cual posee como datos miembros el UUID que identifica a dicha característica y un JSON que contiene la información de la característica.

```

public class BLE_CharacteristicData {
    public String uuid;
    public String jsonData;
    public BLE_CharacteristicData(String uuid, String jsonData) {
        this.uuid = uuid;
        this.jsonData = jsonData;
    }
}

```

Figura 35 - Entidad de la característica

La forma en que Android recibe las notificaciones de una característica por parte de los dispositivos periféricos se las lleva a cabo de un respectivo evento que se auto ejecuta cuando una característica a sido sobrescrita por parte del periférico, en cuyo caso se procede a generar una nueva instancia de la entidad BLE_CharacteristicData con la información de esa Característica e inmediatamente agregada al Buffer de Características como se muestra en la Figura 36.

```

@Override
public void onCharacteristicWriteRequest(BluetoothDevice device,
                                        int requestId,
                                        BluetoothGattCharacteristic characteristic,
                                        boolean preparedWrite,
                                        boolean responseNeeded,
                                        int offset,
                                        byte[] value) {
    super.onCharacteristicWriteRequest(device,
        requestId,
        characteristic,
        preparedWrite,
        responseNeeded,
        offset,
        value);
    if(!isServerReady) return;
    characteristicWrites.
        add(new BLE_CharacteristicData(characteristic
            .getUuid()
            .toString(), new String(value)));
}

```

Figura 36 - Evento que obtiene y administra una característica

En Unity3D existe una función que se auto ejecuta 30 veces por segundo, dentro de la cual se hace el llamado al método que se muestra en la Figura 37 y que es el encargado de obtener en formato JSON la información de cada característica que esté disponible en el Buffer para posteriormente ser procesada para dar la movilidad respectiva al Drone.

```

public String getCharacteristicWrite()
{
    if(bleServer == null) return null;
    if(!bleServer.characteristicWrites.isEmpty())
    {
        JSONObject jsonObject = null;
        try {
            jsonObject = new JSONObject();
            jsonObject.put("uuid",bleServer.characteristicWrites.get(0).uuid);
            jsonObject.put("jsonData",bleServer.characteristicWrites.get(0).jsonData);
        } catch (JSONException e) {
            showToastShort(e.getMessage());
        }
        bleServer.characteristicWrites.remove(0);
        return jsonObject.toString();
    }
    return null;
}

```

Figura 37 - Función que obtiene las características desde el plugin JAR

De esta manera queda explicado el desarrollo del Plugin que conforma la capa de bajo nivel de la arquitectura, la cual representa un archivo .JAR con las funciones necesarias para usar y consumir servicios BLE de forma nativa, esta librería es gestionada a través del componente ServerManager que se lo desarrollo en C#. Este Plugin está configurada para proveer de servicios BLE al simulador. Ver Figura 38.

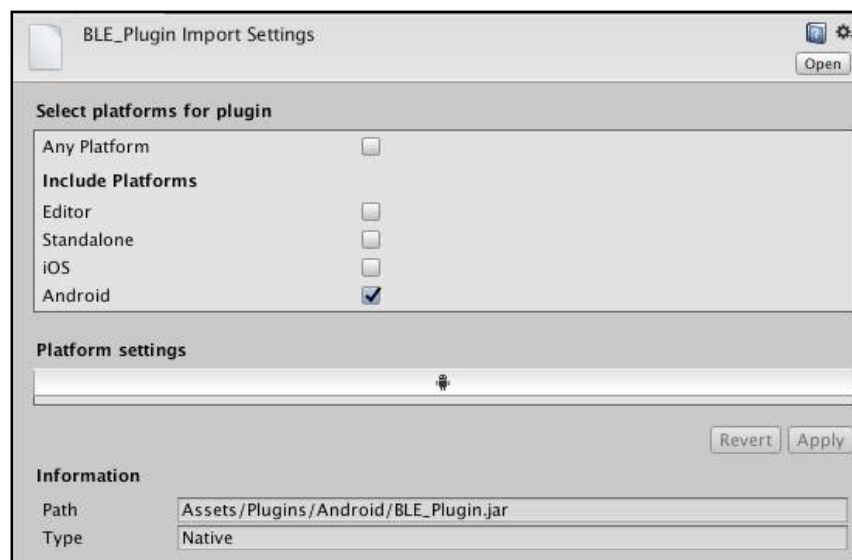


Figura 38 - Configuración del plugin JAR (BLE_Plugin.jar) en Unity3D

4.5.2. Lógica del Simulador

Para la emulación del Drone en el ambiente virtual es necesario procesar la información que llega de cada uno de los RPI a través del Plugin nativo explicado anteriormente. Para este efecto, la capa de alto nivel contiene varios componentes que han sido desarrollados para realizar determinadas tareas, tanto de mensajería como de emulación, desde la ventana de menú hasta el sistema de simulación 3D. Ver Figura 39.

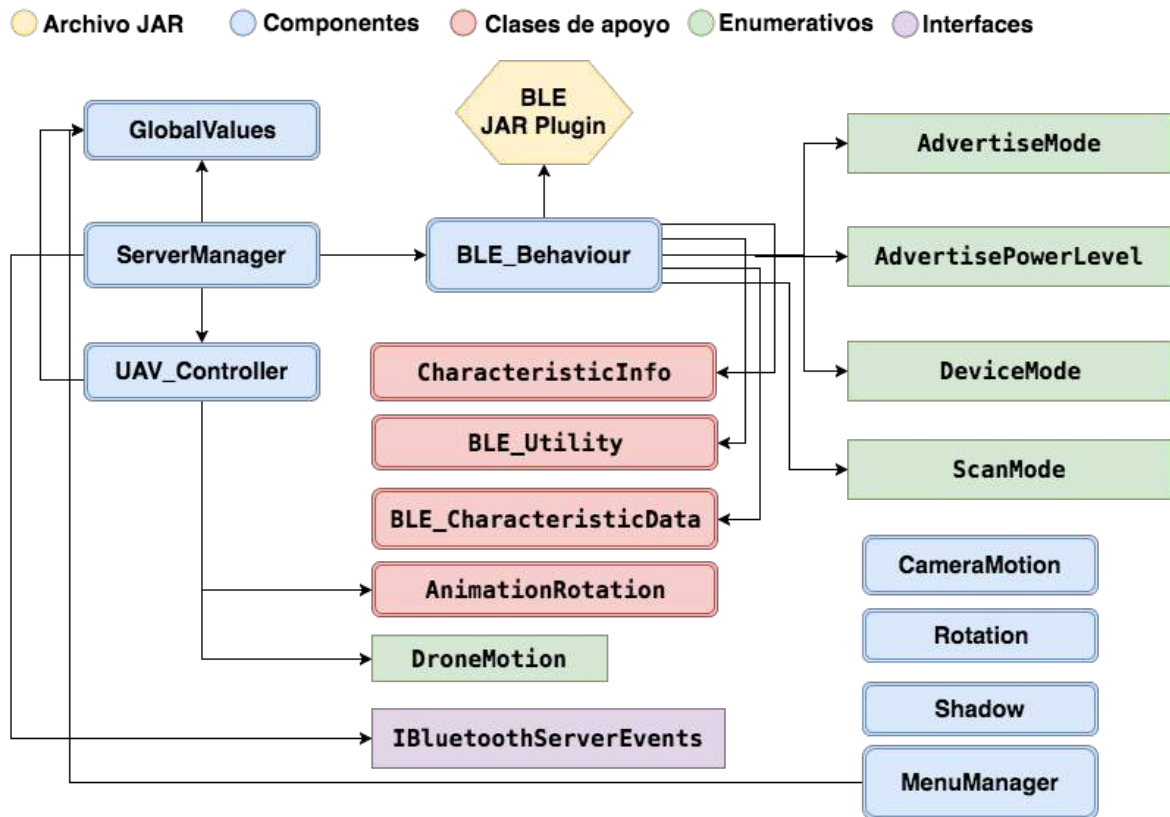


Figura 39 - Estructura de la capa de alto nivel

Componentes de la capa de alto Nivel

- **BLE JAR Plugin:** Archivo JAR, exclusivo para plataformas Android, librería de clases que provee de soporte de conexión y mensajería a través de la tecnología BLE desarrollado con el fin de ser administrado desde Unity3D;
- **BLE_Behaviour:** Este componente inicializa y consume los servicios del plugin JAR (BLE_Plugin.jar) y genera los eventos y delegados necesarios

para la conexión BLE. Los modos de publicidad fueron configurados según los parámetros definidos por Android [34] y son los siguientes:

- **Scan Mode:** Este modo está configurado en latencia baja y potencia alta. Este modo no es utilizado por el Simulador, fue desarrollado para pruebas piloto entre teléfonos celulares;
- **Advertise Mode:** Este modo está configurado en latencia baja y potencia alta para permitir una mayor velocidad de conexión con los RPIs [34];
- **Advertise Power Level:** Este modo está configurado para ofrecer un mayor rango de visibilidad del paquete publicitario [34];
- **Device Mode:** Esta opción está seteada en servidor ya que el plugin debe inicializarse para que funcione como servidor, la opción de cliente se la desarrollo solo con motivos experimentales en las pruebas piloto entre 2 celulares;
- **CharacteristicInfo (Characteristics):** Es una lista con las características que se pretende usar en el simulador, las cuales son las de pitch y roll de cada RPI;
- **BLE_CharacteristicData:** Esta clase de apoyo es una entidad que transforma el json proveniente del plugin JAR y crea un objeto que permite almacenar el uuid y la información obtenida para posteriormente procesarla a través del componente ServerManager;
- **BLE_Utility:** Esta clase de utilidad contiene funciones que permiten generar UUIDs para enviarlas al Plugin nativo, también es la encargada de transformar la información JSON que llega desde el plugin para su posterior procesamiento.

Adicionalmente este componente permite asignar el modo en que se levantarán los servicios BLE tal como se muestra en la Figura 40;

- **ServerManager:** Este componente se encarga de recibir la información que llega de los Raspberry Pi, procesarla y transmitirla al componente encargado de maniobrar el Drone a través de eventos creados utilizando la siguiente interfaz de funciones:

- **IBluetoothServerEvents:** Esta es una interfaz implementada en este componente y permite generar eventos que son llamados directamente desde el código nativo del Plugin JAR y funcionan como disparadores para los eventos y delegados dentro del componente BLE_Behaviour.

La interfaz es implementada y cada método de la misma es llamado desde el Plugin nativo enviando los parámetros específicos de cada una;

- **UAV_Controller:** Este componente gestiona los movimientos del dron a través de la información que gestiona el componente ServerManager, también responde a los eventos Trigger que se producen al cruzar a través de los objetivos del circuito aéreo. Este componente almacena instancias de los objetos FPCSpawn y Graphics con los cuales cambia el modo de cámara a primera persona y modifica el texto 3D que muestra el estado de conexión de los RPI en el simulador. Para gestionar la lógica del Drone virtual, hace uso de las siguientes clases propias del UAV:
 - **AnimationRotation:** Esta clase permite dar una ligera rotación al Drone en base a sus movimientos para generar mayor realismo de movimiento, esta rotación es administrada por este componente ya que depende de la heurística y el modo de maniobra para que se modifiquen los ángulos de rotación 3D del Drone virtual;
 - **DroneMotion:** este modo determina la dirección de movimiento que debe efectuar en Drone virtual en base a los resultados del flujo de ejecución durante todo el proceso de simulación.

La configuración de este componente desde el editor de Unity3D se lo puede apreciar en la Figura 41;

- **GlobalValues:** Este componente se encarga de almacenar temporalmente las variables que determinan el modo de dificultad y de maniobra del simulador, además de controlar un componente que permite la reproducción del sonido del Drone;
- **CameraMotion:** Este componente permite rotar y trasladar la cámara (Solo en modo TPC) para evitar que el UAV salga del rango de visión del usuario;
- **Rotation:** Este componente se encarga de animar los rotores del UAV;
- **Shadow:** Este componente se encarga de rotar y trasladar la sombra simulada acorde a la posición del UAV.

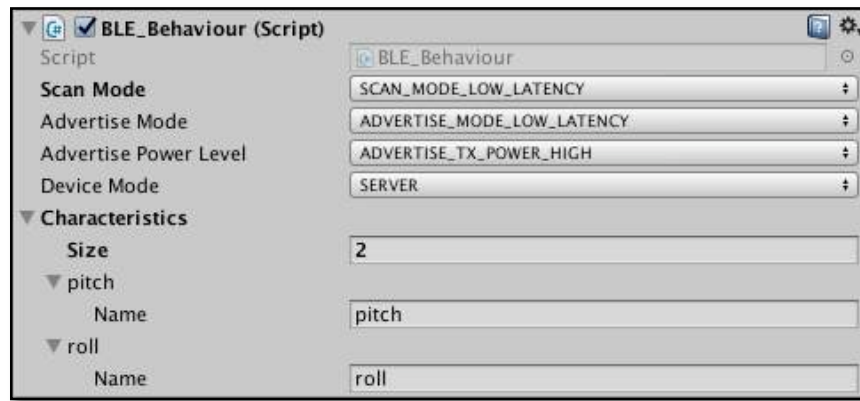


Figura 40 - Componente que configura los servicios BLE

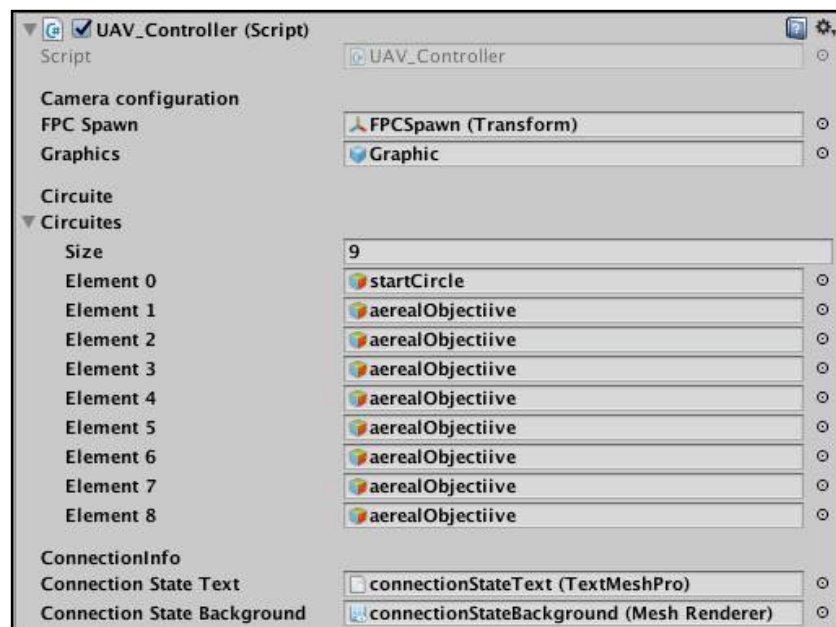


Figura 41 - Configuración del componente que controla el UAV y las cámaras

La lógica del simulador depende del modo de maniobre que se seleccione en la ventana de menú de la aplicación, la información que llega de los RPI siempre es la misma y es el simulador el que maneja y gestiona las acciones que debe realizar el Drone en base al ID del dispositivo (Si es mano izquierda o derecha), tipo de movimiento (Rotación o aceleración) y dificultad de maniobre (Fácil o difícil). Este proceso de heurística se puede ver reflejado a través del diagrama de flujo en la Figura 42.

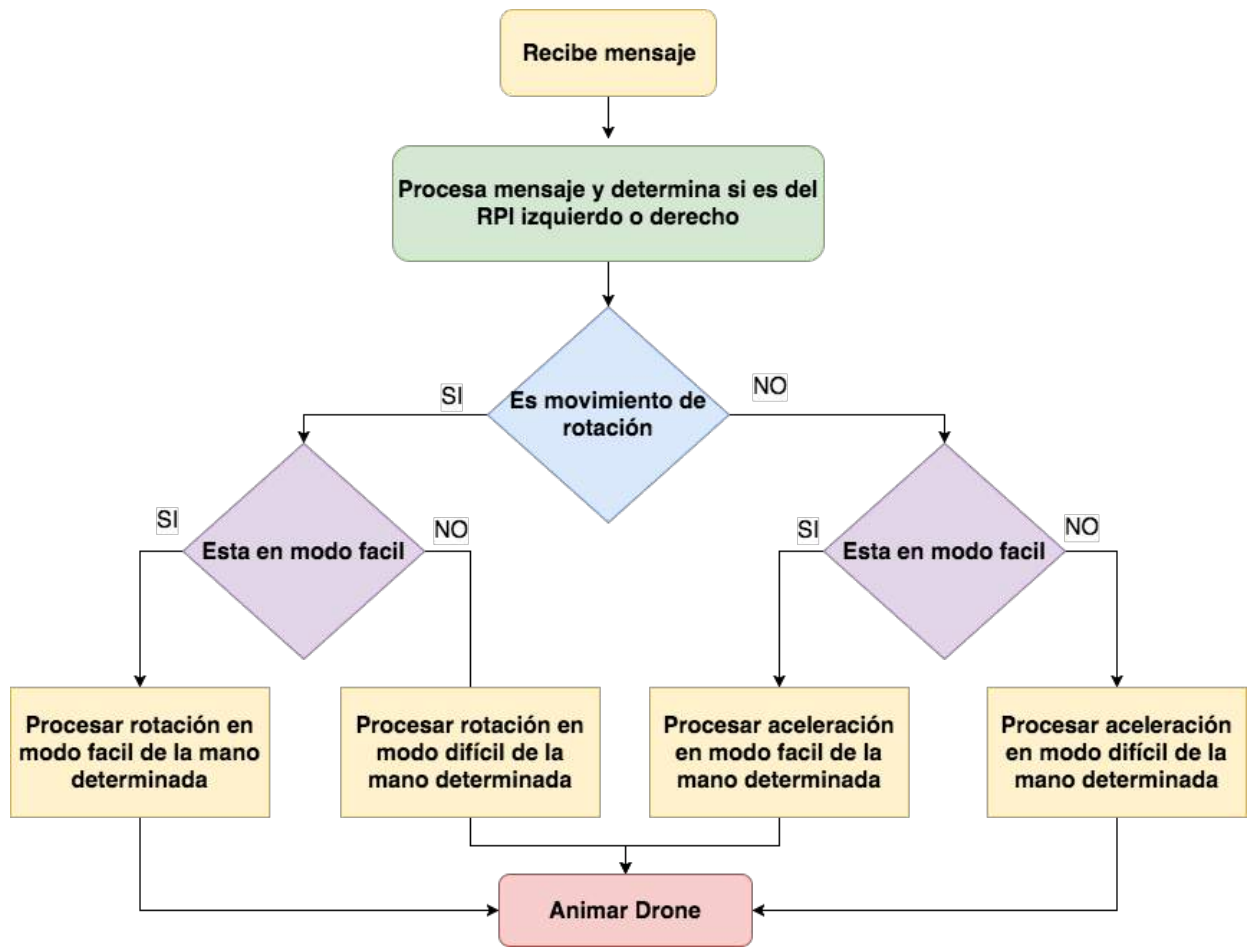


Figura 42 - Diagrama de flujo de la lógica del Simulador

En el Anexo A se muestra 2 marcas de gafas VR que fueron usadas durante el desarrollo del simulador para probar la visualización inmersiva que se obtiene al utilizar las librerías de Google VR.

5. Controles Remotos RPI

Se utilizaron dispositivos Raspberry Pi por el alto poder de procesamiento que poseen además de que ya cuentan con tecnología Wireless integrada y una ranura GPIO con la que se puede adaptar tarjetas adicionales como es el caso de la placa Sense HAT.

Estos dispositivos están configurados para que realicen las conexiones BLE de manera automática, justo después de ser prendidos, están configurados para usar la matriz de LED de la placa Sense HAT con el objeto de visualizar los estados de conexión con el Simulador, se desarrolló un pequeño script que permite apagar estos dispositivos al conectar cualquier periférico de entrada con puerto USB, esto se lo realizó ya que al igual que cualquier dispositivo computacional que corre sobre un sistema operativo no es recomendable apagarlo bruscamente desconectando la fuente de poder, adicionalmente se instaló las librerías necesarias para poder utilizar los servicios BLE desde Python.

Se utilizó el lenguaje de programación Python ya que tanto las librerías de BLE como las librerías de la placa Sense HAT están hechas sobre este lenguaje.

La arquitectura de software Figura 43, propuesta para ser desarrollada en Python consiste en dos módulos, el primero está enfocado en el uso de una librería que permite implementar los servicios de BLE y el segundo está enfocado en el manejo del sensor IMU de la placa Sense HAT a través de su propia librería, utilizando ambos módulos se envía la información en formato JSON a través de una característica específica que posteriormente será gestionada por el simulador para dar movimiento al dron.

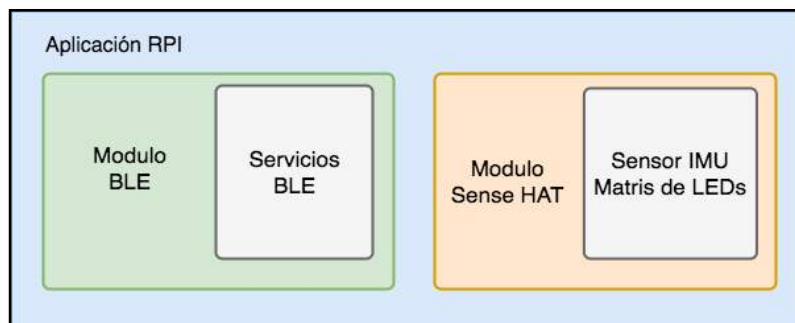


Figura 43 - Arquitectura de software de la aplicación RPI

5.1. Modulo BLE

Este módulo fue desarrollado utilizando la librería BluefruitLE de Adafruit [35], la cual hace uso de las librerías BlueZ en su versión 5.33, cualquier versión superior a esa provoca errores de compilación.

BlueZ proporciona soporte para las capas y protocolos básicos de Bluetooth. Es flexible, eficiente y utiliza una implementación modular. Entre sus características se encuentran:

- Implementación modular completa;
- Seguridad multiprocesamiento simétrica;
- Procesamiento de datos multi-hilo;
- Compatibilidad con varios dispositivos Bluetooth;
- Real abstracción de hardware;
- Interfaz de socket estándar para todas las capas;
- Soporte de seguridad a nivel de dispositivos y servicios.

La librería de Adafruit soporta el uso de Tecnología BLE pero en modo experimental, esto puede ocasionar problemas de conexión que ha surgido en el simulador. Para solventar este problema, la conexión se ha incluido en un bucle iterativo que permite una reconexión automática en caso de fallo.

Basado en la estructura GATT, el servicio que ofrece cada dispositivo RPI son 2 características, una por cada tipo de movimiento, y cada característica se envía en modo JSON según el siguiente formato: {"DeviceID",name,"Value", value}, donde "name" puede ser "R" o "L" los cuales representan los RPI de la mano Derecha e Izquierda respectivamente y value está en el rango de enteros [-1:1], con este formato, el simulador sabe que dispositivo envió la información y que acción debe asignarle al dron en base al valor obtenido, cada característica posee un UUID único el cual es asignado tanto en el simulador como en la aplicación de los RPI, de esta forma el simulador puede identificar el tipo de movimiento validando el UUID de la característica obtenida.

Para la comunicación BLE se a personalizado un servicio con un UUID de 128 bits generado por código utilizando el nombre del paquete de la aplicación Android como parámetro de la función, tal como se muestra en la Figura 44.

```

public static string getGuidFromString(string value)
{
    MD5 md5 = MD5.Create();
    byte[] myStringBytes = Encoding.ASCII.GetBytes (value);
    byte[] hash = md5.ComputeHash(myStringBytes);

    return new Guid(hash).ToString();
}

```

Figura 44 - Método para generar UUID de 128 bits

Este módulo usa una función Figura 45, que construye el JSON en base a los parámetros antes mencionados y los envía realizando una escritura de valor sobre una respectiva característica.

```

def writeCharacteristic(characteristic, value):
    jsonObj = {
        'deviceID': deviceID,
        'value': int(value)
    }
    characteristic.write_value(json.dumps(JsonObj))

```

Figura 45 - Método de configuración y escritura de una característica

Cuando se realiza una escritura de un valor de cualquier característica, el simulador recibe un evento de sobre-escritura de la característica afectada, con la información del valor en ella.

5.2. Modulo IMU

La placa Sense Hat utiliza un sensor de movimiento llamado IMU, que permite medir el tipo de movimiento que experimenta el dispositivo.

IMU significa Unidad de Medición Inercial [36] y conforma tres sensores en uno los cuales se describen a continuación:

- **Un giroscopio:** mide el momento y la rotación;
- **Un acelerómetro:** mide las fuerzas de aceleración y se puede utilizar para encontrar la dirección de la gravedad;
- **Un magnetómetro:** mide el propio campo magnético de la Tierra.

Este módulo fue desarrollado utilizando la librería oficial de Sense HAT para el manejo del sensor IMU, para generar valores acordes a 3 tipos de movimiento, ROLL, YAW y PITCH, internamente la placa usa todos sus sensores para generar valores precisos que reflejen los movimientos y rotaciones de la placa Sense HAT tal y como se muestra en la Figura 46.

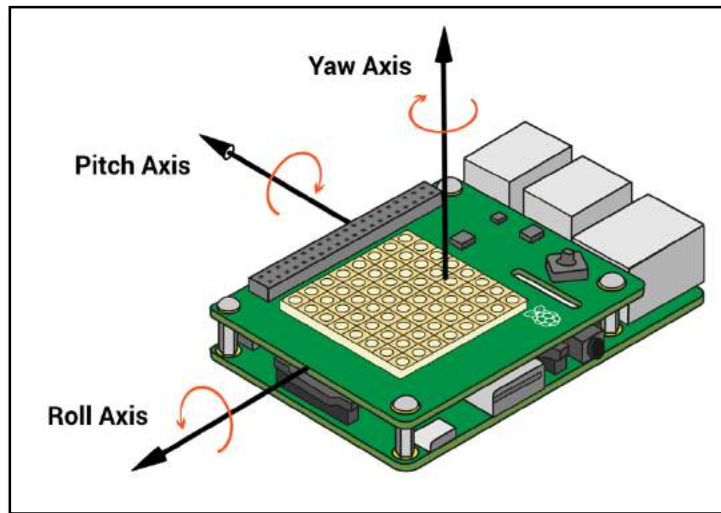


Figura 46 - Sistema de rotación del sensor IMU de la placa Sense HAT [37]

De los 3 tipos de rotaciones que se pueden usar, solamente las rotaciones ROLL y PITCH pueden ser usados desde cualquier ángulo sin afectar el valor generado por el sensor IMU, esto se debe a que, para controlar el dron virtual, el usuario es capaz de girar 360 grados los cuales afectan exclusivamente a la rotación YAW de este sensor, por esta razón solamente se utilizó las rotaciones ROLL y PITCH para maniobrar al dron tal y como se explicó en el Capítulo 4.

En la Figura 47 muestra el rango de ángulos que se uso para el envío de los valores a través de la característica, en donde se utilizan banderas que permiten filtrar el envío de valores redundantes, esto con el objeto de evitar cuellos de botella y mejorar el rendimiento en la comunicación inalámbrica.


```
pitch = sense.get_orientation()['pitch'] # pitch rotation
roll = sense.get_orientation()['roll'] # roll rotation
if (pitch > 270 and pitch < 315 and pitchMotion != 1 and rollMotion == 0):
    pitchMotion = 1
    writeCharacteristic(char_X, pitchMotion)
elif (pitch > 45 and pitch < 90 and pitchMotion != -1 and rollMotion == 0):
    pitchMotion = -1
    writeCharacteristic(char_X, pitchMotion)
elif (pitch < 45 or pitch > 315) and pitchMotion != 0 and rollMotion == 0:
    pitchMotion = 0
    writeCharacteristic(char_X, pitchMotion)
if (roll > 270 and roll < 315 and rollMotion != -1 and pitchMotion == 0):
    rollMotion = -1
    writeCharacteristic(char_Y, rollMotion)
elif (roll > 45 and roll < 90 and rollMotion != 1 and pitchMotion == 0):
    rollMotion = 1
    writeCharacteristic(char_Y, rollMotion)
elif (roll < 45 or roll > 315) and rollMotion != 0 and pitchMotion == 0:
    rollMotion = 0
    writeCharacteristic(char_Y, rollMotion)
```

Figura 47 - Rango de valores angulares del sensor IMU

Este módulo también gestiona el sistema de colores que muestra la matriz de LEDs de la placa Sense HAT para informar al usuario del estado de conexión con el Simulador, el estado de conexión se ha clasificado en base a 3 colores tal y como se muestra en la Figura 48.

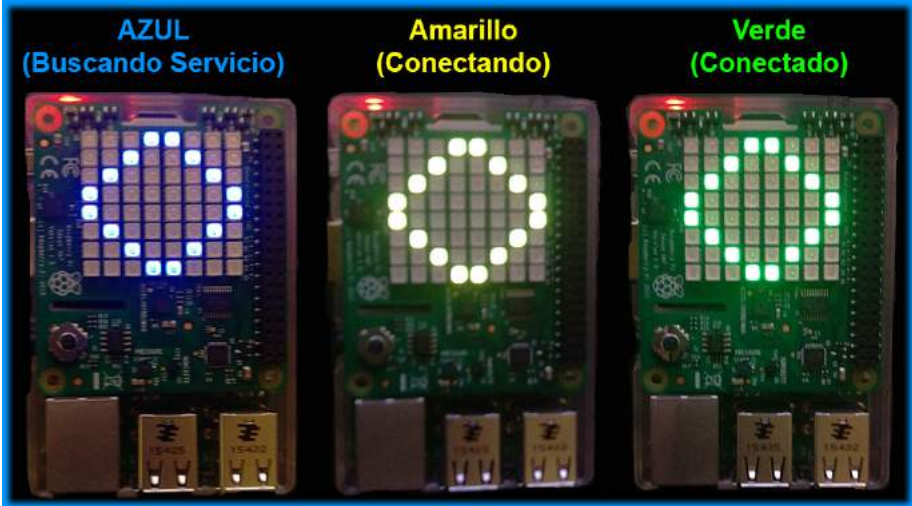


Figura 48 - Sistema de colores de la matriz LED para determinar el estado de conexión con el simulador

5.3. Comunicación con el Simulador

La comunicación con el simulador se la realiza a través de la tecnología Wireless (BLE), la misma que usa un servicio proporcionado por el Servidor en base a una estructura GATT como se explicó en el Capítulo 2. Las características de este servicio son los tipos de movimiento (PITCH y ROLL) que realiza cada RPI y sus valores corresponden al ángulo de inclinación de cada RPI como se explicó en el Capítulo 5, de esta manera la aplicación Simulador inicializa y levanta el servicio a través de una publicidad, cada RPI escanea los servicios proporcionados por la publicidad para inicializar la conexión. Una vez establecida la comunicación, cada RPI sobrescribe los valores de cada característica generando una respuesta inmediata por parte de los eventos en el Simulador. Estos eventos se ejecutan automáticamente recibiendo la información en formato JSON, procesándola y usándola para maniobrar el dron virtual.

La comunicación entre el simulador y los RPI se ejecutan con muy baja latencia, esto es por la fuerza de señal que posee en base a la distancia que existe entre ambos dispositivos la cual es suficientemente corta como para que la señal de radio frecuencia no presente problemas. La distancia máxima entre ambos dispositivos es de 85 cm en una persona de 1.70 metros de altura y se mide en relación a la distancia entre los ojos (Donde va las gafas con el celular) y las manos (donde van los RPI). La distancia máxima recomendada para uso de BLE es de 50 metros. En [25] se realizan pruebas de rendimiento en base a la fuerza de la señal con respecto a la distancia e interferencia, las pruebas se realizan desde una distancia de 0.5 m a 3.5 metros, y refleja una disminución de la fuerza de señal directamente proporcional a la distancia entre los dispositivos emisor y receptor, ver Figura 49.

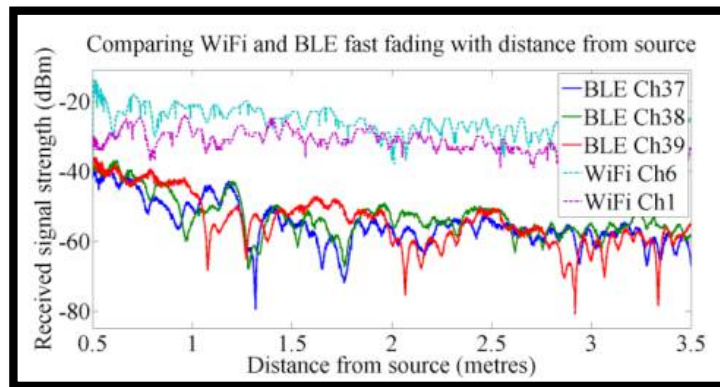


Figura 49 - Disminución de señal WIFI según a la distancia [25]

Como se aprecia en el estudio anterior, la señal de BLE disminuye en base a varios factores, entre los más importantes la distancia entre el dispositivo emisor y el receptor. Si consideremos el efecto negativo de este estudio para estimar la pérdida de señal de los RPI con el simulador en el peor escenario, sería suponiendo que el sistema es usado por “Boban Marjanovic”, el jugador activo más alto de la NBA hasta la fecha con una altura de 2.22 metros, esto significa que la distancia máxima entre su cabeza y sus manos sería de 1.11 metros que aun así mantendría una señal de comunicación excelente dando como resultado un tiempo de latencia mínimo.

La latencia es uno de los factores en las que aplicaciones de ejecución en tiempo real como simuladores o videojuegos sufren pérdidas de FPS (Frames per second) lo que lleva a una mala experiencia de usuario. En [38] se generan pruebas de rendimiento comparando BLE con otras tecnologías y a pesar que BLE obtuvo menos rendimiento, el tiempo de latencia es mínimo, en especial para una transmisión de datos muy pequeña.

Para efectos de estudio y análisis de comunicación, se generaron pruebas de latencia entre la comunicación ya que las funciones BLE del simulador son administradas nativamente y esto implica un posible aumento de latencia en tiempo de ejecución.

El escenario de pruebas se lo realizó a nivel local en cada dispositivo, sincronizando las horas de ambos dispositivos, se almacena la hora en que un RPI escribía una característica y el tiempo en que el simulador reciba la característica ya procesada, esta diferencia está determinada en milisegundos y documentada en la Tabla 2.

Característica recibida	Latencia en milisegundos
Arriba (10 veces)	20 ms
Parar (10 veces)	18 ms
Izquierda (10 veces)	21 ms
Derecha (10 veces)	23 ms
Parar (10 veces)	21 ms
Abajo (10 veces)	12 ms

Tabla 2 - Pruebas de Latencia en la transmisión de datos

Los resultados de estas pruebas reflejan ser positivos, la latencia generada en la transmisión de datos es tan pequeña que no provoca ningún efecto LAG en el simulador.

Para finalizar esta sección se muestra en la Figura 50, el diagrama de flujo de la comunicación entre la aplicación Simulador y los RPI.

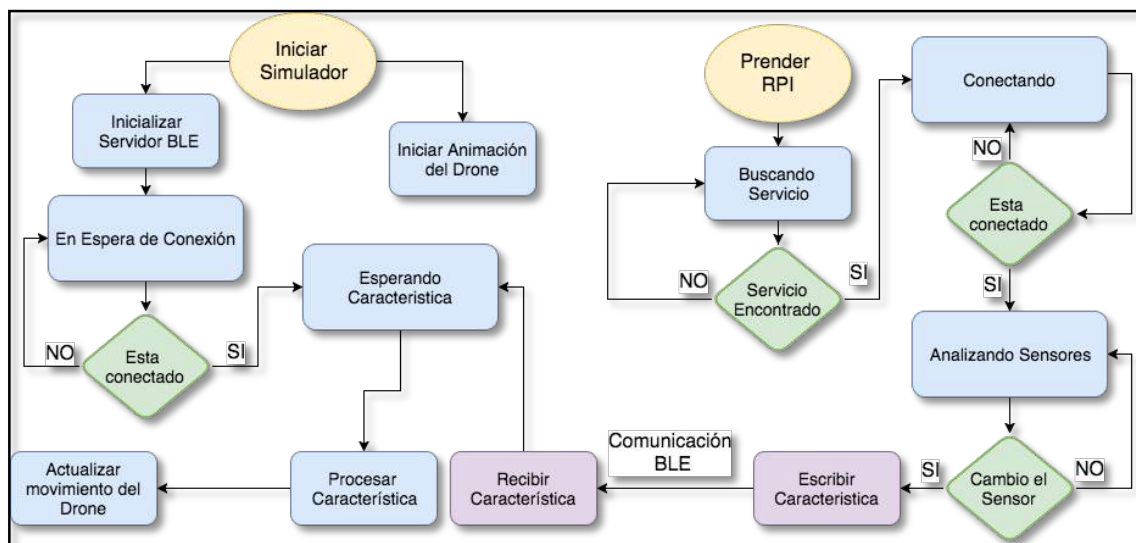


Figura 50 - Diagrama de flujo de la comunicación entre los RPI y el simulador (Celular)

6. Pruebas Realizadas

Las pruebas realizadas se estructuraron en base a las funciones del simulador. La primera función corresponde al modo de visualización durante el maniobro del dron, que puede ser una perspectiva de visión en primera y tercera persona y la segunda función corresponde al modo de dificultad de mando del dron que puede ser fácil o difícil, de esta manera se puede realizar una combinación de ambas funciones para generar un total de 4 modos. Cada uno de estos modos fue testeado en un grupo de personas mayores de entre 55 y 65 años de edad.

Para todas las pruebas se definieron parámetros con rangos de valor entre [0:5], los mismos que se explican a continuación:

- **EV (Experiencia visual):** Este parámetro representa la facilidad que tiene el usuario para visualizar el entorno y el dron en el ambiente virtual, de esta manera cada usuario califica su experiencia visual. Cabe decir que la experiencia visual depende mucho del tipo de resolución que posea el celular y del modo de la cámara en la que se ejecute el simulador. Este parámetro no es relevante para una comparación en función del uso, por lo cual se lo analizó una sola vez;
- **FM (Facilidad de Maniobro):** Este parámetro representa la facilidad o dificultad que experimento el usuario en base al modo de control del dron usando los RPI. Este parámetro es muy relevante por lo que fue evaluado y comparado para monitorear las mejoras en la coordinación Ojo-Mano en función de la memoria y del uso;
- **EU (Experiencia de Usuario):** Este parámetro evalúa la satisfacción personal y el gusto por la aplicación en el modo en que se usó. Este parámetro no es relevante para una comparación en función del uso, por lo cual se lo analizó una sola vez;
- **TU (Tiempo de Uso):** Este parámetro representa el tiempo de uso en segundos que el usuario se tardó en completar todo el circuito de objetivos aéreos. Para determinar que se ha completado este circuito el simulador emite un sonido de éxito al cruzar por toda la ruta. Este parámetro fue determinado por el moderador del sistema de pruebas que en este caso es mi codirectora Graciela Guerrero o mi persona. Este parámetro es igualmente

relevante y proporcional al parámetro FM ya que un usuario con mayor destreza debería terminar el circuito en un menor tiempo.

Las pruebas que se realizaron y que se detallan a continuación fueron probadas en cada usuario 3 veces de las cuales los parámetros más relevantes se los compara en función del primer y tercer uso del sistema mientras que los parámetros menos relevantes se los analiza una sola vez.

6.1. Pruebas en modo TPC y Fácil

Estas pruebas se las hicieron utilizando el simulador con la configuración de la cámara en tercera persona y en modo fácil dando como resultado los valores mostrados en la Tabla 3.

Usuário	EV	FM	EU	TU
1	3	2	3	111
2	5	3	4	101
3	-	-	-	INF
4	2	2	2	160
5	4	2	3	95
6	2	3	2	140
7	3	2	2	136
8	5	3	4	99
9	2	3	2	135
10	2	4	3	100
11	3	3	3	99
12	3	1	2	145
13	2	4	3	120
14	4	3	3	121
15	3	2	2	144
16	-	-	-	INF
17	5	3	4	97
18	3	2	2	136
19	4	3	3	90
Promedio	3	3	3	119

Tabla 3 - Resultados de las pruebas usando la configuración TPC en modo fácil, primera vez

Análisis

Los resultados de esta prueba muestran un promedio de 3 en cada uno de los parámetros establecidos, las experiencias visuales tuvieron inconvenientes especialmente en usuarios con lentes muy grandes, los cuales tuvieron que probar el sistema sin lentes y

esto les dificulto el uso del mismo, es el caso de las 2 pruebas sombreadas en color rojo, ellos no usaron el simulador por una mala experiencia visual, aun así, el mayor porcentaje de los usuarios de pruebas disfrutaron usar el sistema. Se puede concluir a través del parámetro EV que el nivel de visibilidad es medio y el parámetro EU indico un gusto por el modo de uso del sistema.

Para propósitos evaluativos se les pidió al mismo grupo de personas usar el simulador en el mismo modo el mayor número de veces posibles y la curiosidad que tuvieron por usar el sistema facilito este proceso, de este modo se estuvo moderando el uso del sistema por un periodo de 2 horas en donde cada uno de los usuarios tuvieron la oportunidad de usar el simulador 3 veces. La tercera vez de uso del simulador se monitoreo el parámetro FM y TU dando como resultado los valores de la Tabla 4. Se omitió las pruebas en los individuos que no decidieron utilizar el sistema por motivos antes mencionados.

Usuário	EV	FM	EU	TU
1	3	3	3	107
2	5	5	4	90
3	-	-	-	-
4	2	3	2	120
5	4	4	3	96
6	2	3	2	121
7	3	3	2	122
8	5	4	4	87
9	2	3	2	114
10	2	4	3	97
11	3	4	3	85
12	3	3	2	105
13	2	4	3	88
14	4	3	3	101
15	3	3	2	121
16	-	-	-	-
17	5	3	4	89
18	3	4	2	123
19	4	3	3	99
Promedio	3	3	3	103

Tabla 4 - Resultados de las pruebas usando la configuración TPC en modo fácil, tercera vez

Análisis

Los resultados en el tercer intento muestran avances tanto en FM como en TU, los usuarios parecen adaptarse a la forma de maniobre del sistema, se puede llegar a la conclusión de que el uso continuo del sistema mejora la habilidad de maniobre, lo que

significa que se estimula la memoria y las habilidades cognitivas (Coordinación Ojo-Mano).

En base al análisis de ambas tablas se puede concluir una mejora en la habilidad de maniobrar el dron, Figura 51, y en el tiempo requerido para concluir el circuito de objetivos aéreos, Figura 52.

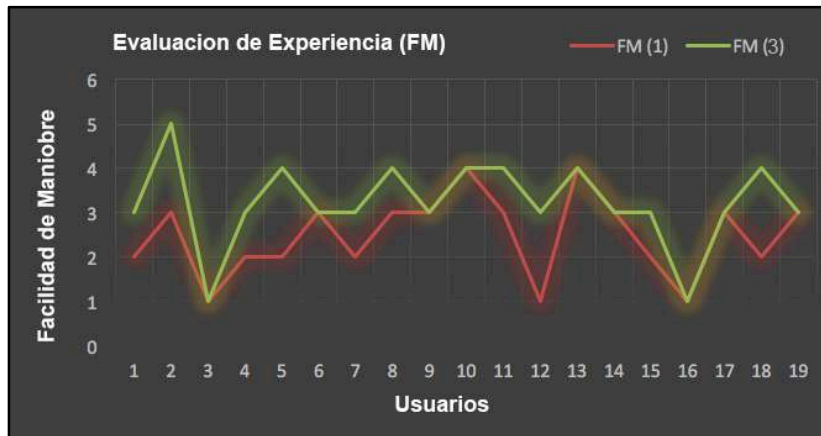


Figura 51 - Estadística de FM usando la configuración TPC en modo fácil

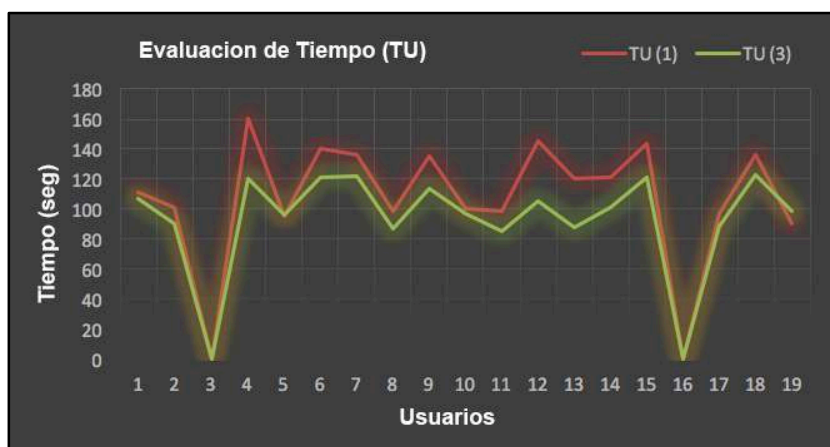


Figura 52 - Estadística de TU usando la configuración TPC en modo fácil

6.2. Pruebas en modo TPC y Difícil

Estas pruebas se las hicieron utilizando el simulador con la configuración de la cámara en tercera persona y en modo difícil dando como resultado los valores mostrados en la Tabla 5.

Usuário	EV	FM	EU	TU
1	2	1	3	215
2	4	2	3	200
3	2	1	2	185
4	3	1	2	205
5	4	2	3	189
6	3	3	2	174
7	3	2	2	187
8	2	3	4	160
9	3	2	3	186
10	4	2	3	191
11	3	1	3	208
12	2	1	2	211
13	2	2	3	199
14	4	3	3	169
15	3	2	2	207
16	3	2	3	202
17	4	2	4	193
Promedio	3	2	3	193

Tabla 5 - Resultados de las pruebas usando la configuración TPC en modo difícil, primer vez

Análisis

En este modo la experiencia visual fue igual que en el modo fácil, ya que la dificultad no cambia el entorno, sino que dificulta el maniobrar, de la misma manera el parámetro EU se mantiene casi igual.

El cambio de tipo de maniobra a modo difícil aumento la dificultad para maniobrar el dron y el tiempo en terminar el circuito. Esta prueba se hizo omitiendo a los 2 sujetos que no podían usar el simulador, uno de los problemas en este caso era que algunos usuarios no recordaban bien los patrones de movimiento para mover el dron lo que provocó algo de confusión en este modo, aun así, se les pidió usar el sistema 3 veces para ver el avance del mismo, el cual se refleja en la Tabla 6.

Usuário	EV	FM	EU	TU
1	2	3	3	160
2	4	2	3	180
3	2	3	2	156
4	3	2	2	187
5	4	3	3	158
6	3	4	2	155
7	3	3	2	151
8	2	3	4	138
9	3	4	3	159
10	4	3	3	159
11	3	2	3	175
12	2	2	2	164
13	2	3	3	161
14	4	3	3	155
15	3	4	2	160
16	3	3	3	162
17	4	3	4	129
Promedio	3	3	3	159

Tabla 6 - Resultados de las pruebas usando la configuración TPC en modo difícil, tercera vez

Análisis

La tercera vez que se utilizó el sistema se aprecia una mejora en comparación a la primera vez, tal como ocurrió en el modo fácil, los usuarios se adaptaron a la nueva forma de maniobra y optimizaron la velocidad y tiempo. Incluso pocos de ellos dijeron que es más fácil este tipo de maniobra que el modo fácil.

En base al análisis de ambas tablas se puede concluir una mejora en la habilidad de maniobrar el dron, Figura 53, y en el tiempo requerido para concluir el circuito de objetivos aéreos, Figura 54.



Figura 53 - Estadística de FM usando la configuración TPC en modo difícil



Figura 54 - Estadística de TU usando la configuración TPC en modo difícil

6.3. Pruebas en modo FPC y Fácil

Estas pruebas se las hicieron utilizando el simulador con la configuración de la cámara en primera persona y en modo fácil dando como resultado los valores mostrados en la Tabla 7.

Usuário	EV	FM	EU	TU
1	3	3	2	82
2	5	4	2	77
3	4	4	3	79
4	3	3	3	81
5	4	3	2	72
6	5	4	3	97
7	3	2	3	98
8	5	4	2	87
9	3	3	2	80
10	5	4	4	81
11	5	3	3	71
12	3	3	3	91
13	2	3	3	87
14	4	3	3	75
15	3	2	3	68
16	5	4	2	73
17	5	4	3	92
Promedio	4	3	3	81

Tabla 7 - Resultados de las pruebas usando la configuración FPC en modo fácil, primera vez

Análisis

Estas pruebas mejoran notablemente la EV pero disminuyo un poco en EU, los usuarios gustaron más del uso en este modo que en tercera persona, ya que algunos mencionaron que era como viajar en un avión.

Con esta configuración de cámaras y en modo fácil se aprecia en gran diferencia la facilidad de uso del sistema, claro que hay que considerar que el mismo grupo de personas han usado ya el sistema un total de 6 veces, 3 por cada una de las 2 configuraciones anteriores, esto implica una adaptación al sistema incluso con otra configuración, la ventaja de esta configuración es que el usuario mira lo que el Drone ve y por ello el nivel de visibilidad es muy considerable.

Tal y como lo hicimos en los casos anterior, cada usuario uso el sistema un total de 3 veces, por lo que la Tabla 8. refleja los datos obtenidos en el tercer uso del sistema.

Usuário	EV	FM	EU	TU
1	3	4	2	81
2	5	5	2	72
3	4	5	3	73
4	3	5	3	79
5	4	4	2	75
6	5	5	3	88
7	3	4	3	87
8	5	5	2	81
9	3	4	2	73
10	5	5	4	79
11	5	5	3	75
12	3	4	3	88
13	2	5	3	82
14	4	4	3	72
15	3	4	3	75
16	5	5	2	67
17	5	5	3	80
Promedio	4	5	3	78

Tabla 8 - Resultados de las pruebas usando la configuración FPC en modo fácil, tercera vez

Análisis

La tercera vez de uso, los usuarios mostraron una destreza bastante alta, este modo fue el favorito para muchos, en comparación con las otras pruebas, en esta prueba el promedio del parámetro FM llega a 5, el tiempo no vario demasiado con relación al otro y es porque cruzar todo el circuito lleva alrededor de 1 minuto.

Realizando las comparaciones en base al análisis de ambas tablas sobre los parámetros FM y TU, se puede apreciar la mejora en la habilidad de maniobra, Figura 55, y en el tiempo requerido para concluir el circuito de objetivos aéreos, Figura 56.



Figura 55 - Estadística de FM usando la configuración FPC en modo fácil

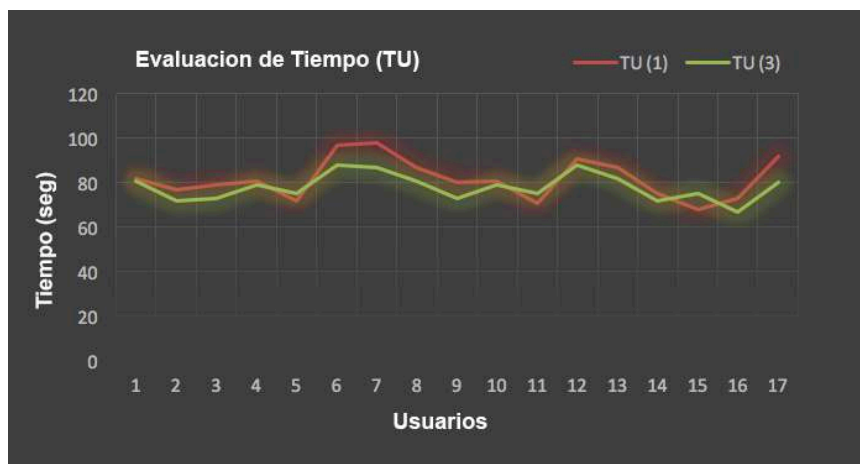


Figura 56 - Estadística de TU usando la configuración FPC en modo fácil

6.4. Pruebas en modo FPC y Difícil

Estas pruebas se las hicieron utilizando el simulador con la configuración de la cámara en primera persona y en modo difícil dando como resultado los valores mostrados en la Tabla 9.

Usuário	EV	FM	EU	TU
1	3	2	2	110
2	3	3	2	101
3	4	3	2	98
4	3	2	3	98
5	4	4	2	92
6	3	3	3	110
7	3	2	2	115
8	3	3	2	104
9	3	2	2	111
10	3	3	2	99
11	4	4	3	91
12	3	4	2	95
13	2	2	3	120
14	3	3	2	102
15	3	3	3	107
16	3	3	3	100
17	3	4	2	87
Promedio	3	3	2	102

Tabla 9 - Resultados de las pruebas usando la configuración FPC en modo difícil, primera vez

Análisis

La EV de estas pruebas fueron buenas, pero un poco menor que en modo fácil, esto se dio al mareo que generó el maniobrero en modo difícil.

De igual manera el parámetro EU bajo, muchos usuarios presentaron mareo incluso más que en el modo fácil, ya que no solamente giraban la cámara con el movimiento de la cabeza sino también con las manos y el hecho de avanzar y retroceder les resultó algo un poco incómodo.

Por otro lado, el parámetro FM fue un poco bajo ya que al igual que las pruebas con la configuración en modo difícil, no les gustó mucho la forma de maniobrero, aunque ya estaban más habilidosos. La Tabla 10, muestra los datos reflejados en el tercer uso del sistema.

Usuário	EV	FM	EU	TU
1	3	1	2	141
2	3	2	2	118
3	4	3	2	120
4	3	2	3	134
5	4	2	2	104
6	3	3	3	110
7	3	2	2	112
8	3	3	2	107
9	3	2	2	112
10	3	3	2	115
11	4	2	3	119
12	3	2	2	120
13	2	2	3	128
14	3	3	2	102
15	3	3	3	107
16	3	3	3	100
17	3	1	2	132
Promedio	3	2	2	116

Tabla 10 - Resultados de las pruebas usando la configuración FPC en modo difícil, tercera vez

Análisis

Este caso fue un poco diferente al resto, los usuarios presentaron mareo en el uso y envés de mejorar, en algunos casos la eficiencia de uso disminuyó. Parece que un maniobro de este tipo en esta configuración de cámara resulta molesto en la orientación y a pesar de que la experiencia visual es mucho mejor que las pruebas de cámara en tercera persona, la EU en general no estuvo muy agradable.

Realizando las comparaciones en base al análisis de ambas tablas sobre los parámetros FM y TU, se puede apreciar la diferencia en la habilidad de maniobro, Figura 57, y en el tiempo requerido para concluir el circuito de objetivos aéreos, Figura 58. en este caso la línea verde que representa mejora en las pruebas anteriores, ahora representa pérdida.



Figura 57 - Estadística de FM usando la configuración FPC en modo fácil

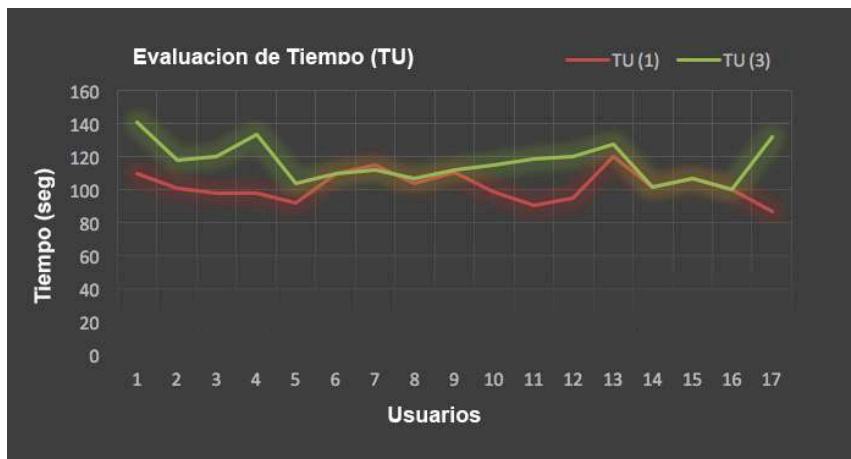


Figura 58 - Estadística de TU usando la configuración FPC en modo fácil

En el Anexo B se puede apreciar algunos de los usuarios que hicieron uso del simulador durante las pruebas realizadas.

En el anexo C se aprecia el usuario que tubo incomodidad al usar las gafas VR con sus lentes puestos, por lo que tubo que usarlo sin lentes.

En el anexo D se aprecia a 2 de estudiantes (Personas jóvenes) que disfrutaron e hicieron uso del Simulador.

7. Pruebas con Dron Real

Uno de los objetivos de esta tesis es realizar pruebas con un dron real, por esta razón, se incluyó el desarrollo de una aplicación para Android que utilice la misma estructura y servicios del plugin JAR para maniobrar un dron real de la misma manera que se lo maniobra en el simulador.

7.1. Arquitectura General

En base a lo anteriormente mencionado, se ha utilizado un Phantom 4 [39] como objeto de estudio y su respectivo SDK para usarlo en el desarrollo de esta aplicación. Como se observa en la Figura 59, la arquitectura está compuesta por el teléfono celular que corre la aplicación que gestiona la comunicación con los RPI a través de la tecnología BLE y envía los eventos de maniobra hacia el control remoto del Phantom 4 usando una conexión USB y este a su vez, envía las ordenes a través de tecnología Wireless a una frecuencia de 2.4Ghz hacia el dron. Cabe recalcar que la conexión USB entre el teléfono celular y el control remoto del dron es indispensable para la comunicación y el envío de ordenes que requiere el control para maniobrar al dron. A pesar que este dron tiene soporte para una conexión a través de Bluetooth entre el teléfono y el control remoto, se tubo que utilizar la conexión USB ya que el Teléfono ya mantiene una conexión de Bluetooth con los RPI por lo que se tubo que utilizar un método diferente.

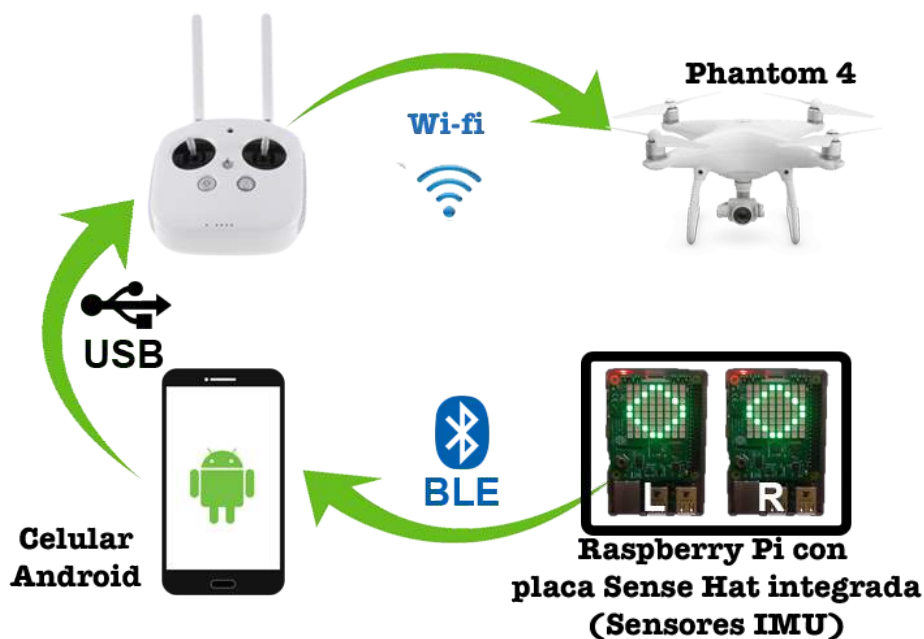


Figura 59 - Arquitectura General propuesta para de control del UAV real con los RPI

El SDK de este dron permite realizar aplicaciones para Android e iOS, ofreciendo acceso a las funciones del dron, de esta manera se generó una aplicación en Android para maniobrar el Phantom 4 utilizando el sistema de maniobras en modo difícil que posee el simulador, esto se decidió ya el modo fácil obliga al dron a avanzar indefinidamente y esto en un dispositivo real puede generar accidentes, lo que ya ha ocurrido durante el proceso de pruebas de esta aplicación.

7.2. Arquitectura de Software y Desarrollo

La arquitectura de software de esta aplicación está definida por 2 módulos, el módulo que maneja las funciones BLE que recibe los mensajes de los RPI y el módulo que gestiona las funciones del dron a través de su SDK, tal como se observa en la Figura 60.

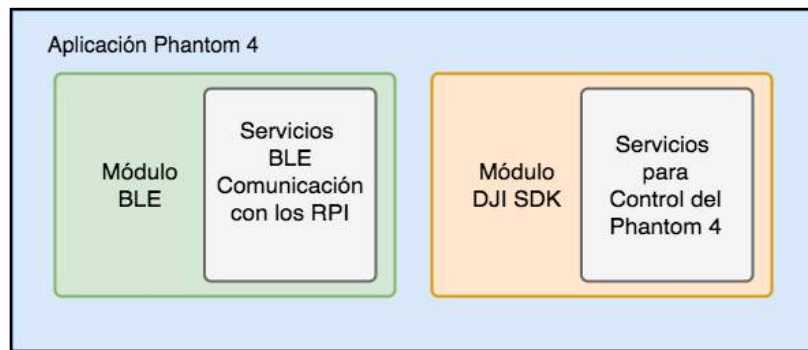


Figura 60 - Arquitectura General de software para el control del dron real

El Phantom 4 posee 4 tipos de movimientos, Throttle, para ascender o descender, PITCH para avanzar o retroceder, ROLL para trasladarse de lado a lado y YAW rotar sobre su propio eje [40]. El sistema de coordenadas y los movimientos de traslación y rotación están definidas en la Figura 61.

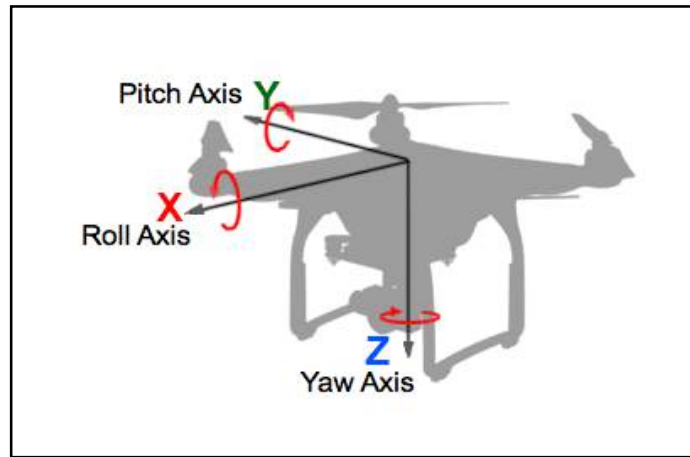


Figura 61 - Sistema de coordenadas del Phantom 4 [40]

En base a su sistema de coordenadas, se debe modificar las instrucciones para que se adapte y funcione sin alterar el código de los RPI quedando de las siguientes maneras.

- Los movimientos PITCH de ambos RPI manipulan los movimientos THROTTLE del phantom 4;
- Los movimientos ROLL del RPI de la mano izquierda controlan los movimientos YAW del phantom 4;
- Los movimientos ROLL del RPI de la mano derecha controlan los movimientos PITCH del phantom 4.

Con esta configuración, el sistema de maniobra del phantom 4 es idéntico al sistema de maniobra del dron virtual en modo difícil.

De esta manera se recibe la información de las características de igual manera que se lo hace en el plugin JAR, pero en vez de enviarlos a través de un evento a la clase BLE_Behaviour del simulador se las envía a través de una instrucción de mando al control remoto del phantom 4 el cual maniobra el dron con los nuevos 4 valores configurados como se muestra en la Figura 62.

```

@Override
public void run() {
    if (DJIApplicationController.isFlightControllerAvailable()) {
        FlightController flightController = DJIApplicationController
            .getAircraftInstance()
            .getFlightController();
        if(flightController != null)
        {
            flightController
                .sendVirtualStickFlightControlData(new FlightControlData(
                    pitch,
                    roll,
                    yaw,
                    throttle),
                    new CommonCallbacks.CompletionCallback() {
                        @Override
                        public void onResult(DJIErrors djiError) {
                        }
                    });
        }
    }
}
}

```

Figura 62 - Envió de datos al control remoto del Phantom 4

En el caso que se produzca una desconexión por parte de cualquier RPI se envía las 4 variables con valor de 0, de esta forma el dron deja de hacer cualquier movimiento y entra en modo de “Standby” hasta que el RPI se reconecte al teléfono. Esto evita cualquier accidente o daño colateral por falla de mando, aun así, es recomendable que haya un segundo usuario con el control remoto del dron en mano en caso de alguna emergencia. Esta aplicación no esta diseñada para ser usada por adultos mayores ni por jóvenes, esta echa con propósitos de investigación dentro de esta tesis con el único y exclusivo propósito de comparar la funcionalidad y manejo del dron real y el virtual a través de los mandos RPI.

7.3. Pruebas y Resultados

Durante y después del desarrollo de esta aplicación se la probó a través del Simulador de pruebas que provee la empresa DJI [41] de la cual procede el Phantom 4. Ver Figura 63.

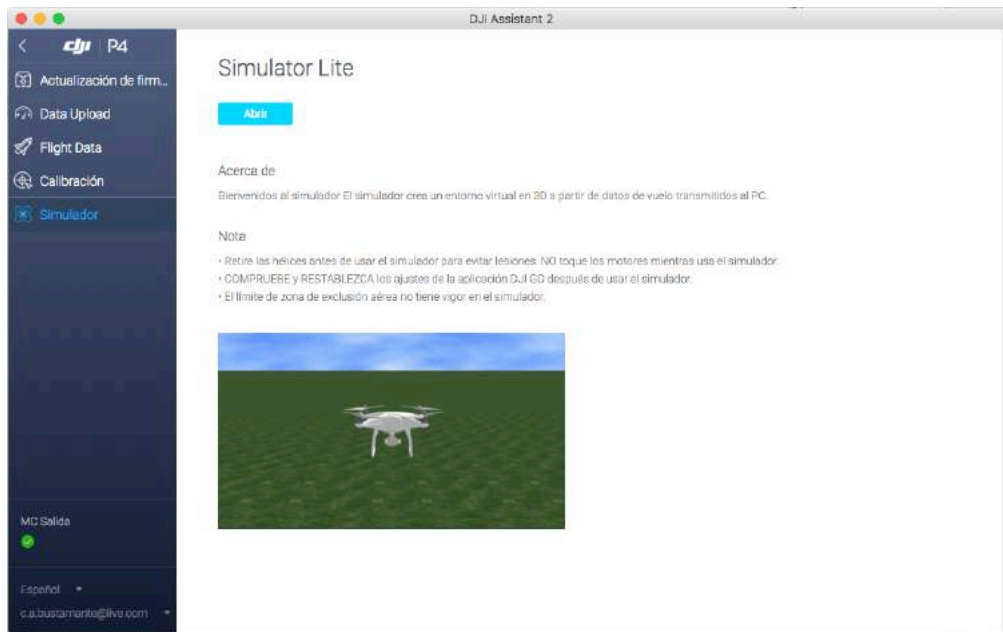


Figura 63 - Simulador DJI para el Pahantom 4

Este simulador permite realizar pruebas y depurar la aplicación realizada usando el SDK antes de realizar pruebas reales.

Los controladores de vuelo de las aeronaves DJI soportan un modo de simulación que permite un desarrollo más rápido y seguro de las aplicaciones. El controlador de vuelo acepta comandos de control, y los utiliza para simular la información del sensor y del estado generando los mismos movimientos que tendría en pruebas reales;

Esta aplicación (DJI Assitant) es útil para realizar pruebas de aplicación exhaustivas cada vez que se actualice el maniobre del dron a través de una aplicación mobil.

Una vez efectuado las pruebas simuladas, se realizaron pruebas reales, las mismas que dieron resultados positivos, efectivamente el sistema de controles RPI puede ser usado para manipular un Phantom 4 del mismo modo que se lo hace en el simulador con el modo de control difícil.

La experiencia y maniobre es diferente, ya que existen factores naturales como el viento u obstáculos que dificultan el control del dron, además de que como se mencionó en el Capítulo 5, las librerías de BLE en Phytton están en modo experimental y por ende las conexiones BLE no son 100% estables, en base a esto, se a podido documentar un cuadro de errores que se han producido en base a los factores de fallo antes mencionados con

resultados no esperado en algunos casos. La Tabla 11, muestra 3 escenarios que se han escogido para las pruebas reales de esta aplicación, en cada escenario se a evaluado el nivel de fuerza del viento en un rango de [0:5], los obstáculos encontrados en cada día variaron dependiendo del lugar de pruebas y en la mayoría de los casos representan vehículos estacionados o arboles cercanos, La intermitencia de conexión de los RPI es idéntica que en el simulador y se documentaron el número de desconexiones que presentaron los dispositivos en cada escenario, finalmente se documentó también accidentes que se produjeron y que no llegaron a daños mayores pero son una muestra de lo peligroso que es maniobrar un dron utilizando dispositivos o métodos de maniobre diferentes a los recomendados para hacerlo.

	Viento	Obstáculos	Desconexión RPI	Accidentes
Día despejado	1	2	1	0
Día nublado	3	1	2	2
Día lluvioso	5	1	1	1

Tabla 11 - Resultados de maniobre en base a factores de error

En el Anexo E, se muestra las pruebas del Drone (Phantom 4) realizadas a través del simulador virtual (DJI Assistant), en las que se prueba la comunicación BLE para probar los movimientos que debería realizar en un entorno real.

En el Anexo F, se muestra una captura de la interfaz de usuario de la aplicación ejecutada en un ambiente real .

En el Anexo G, se muestras algunas de las capturas en las que se maniobra el Drone (Phantom 4) utilizando los Raspberry Pi en un ambiente real.

8. Conclusiones

Gracias a la investigación realizada, se a podido estudiar, desarrollar e implementar el sistema de simulación propuesto en este trabajo, las pruebas realizadas generaron resultados positivos y demostraron la viabilidad de uso de este sistema con el enfoque que se lo a definido a través de las publicaciones científicas realizadas.

En primera instancia se demostró que el motor de videojuegos Unity3D fue una herramienta poderosa que permite integrar fácilmente varios componentes gráficos para generar entornos realistas, fluidos y con soporte a una configuración en modo VR, además de facilitar el uso de Física interna que fue útil para la emulación de la gravedad del Drone virtual.

En segunda instancia se demostró que las librerías Google VR se las puede integrar fácilmente y convertir el sistema de visualización tradicional que posee Unity3D en un sistema de visualización VR el mismo que requiere del uso de cualquier tipo de gafas VR para generar la inmersión completa en el ambiente virtual simulado.

En tercera instancia se demostró que es factible desarrollar e integrar librerías nativas que permiten agregar funcionalidades extras que la herramienta Unity3D no posee, en este caso se agrego funciones de comunicación inalámbrica a través de la tecnología Bluetooth Low Energy (BLE), para poder transmitir los datos de los dispositivos RPI al celular. Este Plugin fue desarrollado en JAVA y funciona exclusivamente sobre plataformas Android, ya que esta desarrollado utilizando el SDK de Android.

En cuarta instancia se demostró que los Raspberry Pi con la placa Sense HAT integrada en ella, representan un dispositivo portable poderoso, ligero y que requiere únicamente una batería portable para ser utilizada, El Raspberry Pi ya posee un sensor de Bluetooth con la especificación de BLE, por lo que no fue necesario integrar ninguna placa externa para proveer de servicios BLE a este dispositivo, la placa Sense HAT posee el sensor IMU que es indispensable para el maniobre del UAV y adicionalmente incorpora una matriz de LEDs de 8x8, la cual es útil para determinar el estado de conexión entre el RPI y el Servidor.

En quinta instancia se demostró que el sistema de maniobre propuesto para el simulador es también posible realizarlo en un UAV real, para lo cual se desarrollo una aplicación en Android que permita maniobrar un Drone real (Phantom 4) utilizando los

RPI, las pruebas realizadas demostraron que el maniobrar con los RPI generan movimientos muy similares entre el Drone virtual y el Drone real.

Por último, a través de las pruebas realizadas se demostró la acogida por parte de las personas mayores y el interés en el uso del sistema, a través de los parámetros que se crearon se determinó que el simulador es un medio de entretenimiento que evita el sedentarismo y promueve el movimiento y la diversión. El simulador fue desarrollado para ser usado utilizando 2 perspectivas de visualización, una con un sistema de cámaras en primera persona y otra con un sistema de cámaras en tercera persona, esto permite que el usuario decida el modo de visualización que le sea de mayor gusto, de esta forma se evita una pérdida de interés o la generación de monotonía al usar el sistema. Adicionalmente al sistema de visualización en 2 cámaras, el simulador ofrece 2 métodos de maniobrar, uno más complejo que otro, esto permite que los usuarios no pierdan interés y se promueva la auto competitividad al usar el sistema. El maniobrar de un Drone usando solamente el movimiento de las manos promueven el desarrollo de la capacidad cognitiva (Coordinación Ojo-Mano) y el uso de la memoria, lo que es un factor a favor al usar el sistema.

La propuesta de esta tesis fue desarrollada, implementada, publicada y probada, los objetivos planteados han sido concluidos y los resultados fueron tal y como se esperaron, las tecnologías utilizadas tanto en software como en hardware permitieron una integración y dieron fruto a un sistema que pudo ser estudiado y monitoreado. El uso de las librerías de Google VR dentro de la herramienta Unity3D permitió generar entornos tridimensionales acorde a la propuesta en este trabajo, el paisaje montañoso diseñado como escenario para correr el simulador ofreció una experiencia visual muy realista. La librería de clases desarrollada en JAVA se la pudo integrar y utilizar desde C# para proveer de servicios BLE al simulador. Los RPI junto con su placa Sense HAT demostraron ser un dispositivo ligero y portable para ser usado como mandos remotos a través del uso de su sensor IMU. La tecnología BLE demostró para lo que está hecha, ya que al ser la tecnología Wireless de bajo consumo de energía fue excelente para ser implementada y utilizada entre dispositivos móviles, a pesar de que la conexión se la desarrolló en diferentes lenguajes y para diferentes plataformas, no hubo problemas de conexión y comunicación. El objeto de estudio (UAV), fue excelente para ser manipulado a través de mandos remotos gracias a sus múltiples opciones de control que posee. Las pruebas realizadas demostraron que el simulador puede ser una forma de entretenimiento tecnológico y rehabilitación que

fomenta las habilidades y la función mental, se observó los resultados utilizando el simulador en todas las combinaciones a nivel de cámara y de dificultad. De esta manera cada usuario tubo sus preferencias entre una u otra configuración y esto se lo pudo saber en base a la documentación parametrizada en que se realizaron las pruebas. Las pruebas con el dron real demostraron que es posible maniobrarlo bajo el mismo mecanismo y con los mismos dispositivos que el Dron virtual, sin embargo, no se lo recomienda a modo de diversión sino solo a modo de investigación.

Palabras-clave: Simulation, 3D, UAV, VR, Bluetooth, BLE, Inertial Sensors, Elders, Well-being, Life quality. Virtual Environment, Wearable Devices, Real Time..

8.1. Contribuciones Principales

Como resultado de este trabajo y en función de los objetivos específicos, se generaron las siguientes contribuciones principales.

Diseñar un entorno virtual realista, amigable y de confort visual para una mejor acogida por parte de los adultos mayores

Este sistema contribuyo al proponer, diseñar e implementar un entorno tridimensional hiper-realista, atractivo, simple y de confort visual, en donde el adulto disfruto del paisaje visual mientras maniobraba el Dron. Entre las características mas importantes se destaca el uso de mallas 3D reales, imágenes panorámicas y sonidos reales, todo esto permitió que el simulador ofrezca una sensación de inmersión muy parecido a la realidad. El escenario fue optimizado para ser utilizado en dispositivos móviles (Celulares) y adaptado para ser visualizado usando gafas VR de cualquier tipo.

Diseñar una arquitectura tecnología que permita integrar y controlar un simulador virtual a través de sensores de red de área corporal de manera inalámbrica, con el objetivo de facilitar el maniobre del Dron a los adultos mayores

Este trabajo contribuyo al proponer, diseñar e implementar una arquitectura que este conformada por la integración de varias tecnologías de vanguardia para generar una solución al problema de sedentarismo y baja calidad de vida que sufren los adultos mayores, entre las características mas importantes de esta contribución esta el uso de

realidad virtual para generar un ambiente inmersivo el cual fomenta el entretenimiento de los adultos mayores, una interconexión entre dispositivos sensoriales y un ambiente virtual a través de una red de área corporal, con el objeto de generar un modo de control simple pero con el potencial de promover el movimiento y mejorar la salud física y mental de los adultos mayores.

Desarrollar y testear una aplicación nativa en Android que utilice las mismas nano computadoras que el simulador virtual para maniobrar un DRONE real

Como contribución final, se desarrolló una aplicación en Android que permite utilizar los RPI para maniobrar un dron real, en este caso un Phantom 4 de la empresa DJI, las pruebas realizadas, determinaron que es posible maniobrar un dron real a base de movimientos corporales (Manos). Esta aplicación no se recomienda para ser usada como entretenimiento sino solo para propósitos de investigación tecnológica.

8.2. Trabajo Futuro

Las pruebas demostraron la posibilidad de enfocar el sistema desarrollado en este trabajo a personas más jóvenes, o bajo otros mecanismos de control, como la tecnología BLE utiliza una topología de estrella, es posible desarrollar un sistema con más de 2 dispositivos sensoriales, de este modo se plantea como trabajo futuro el desarrollar un sistema que pueda ser usado para rehabilitación, deporte o entretenimiento a base de una red de área corporal con sensores que no solamente determinen el movimiento de manos sino también de piernas y pies.

A razón de este estudio se vio la posibilidad de generar una especie de videojuego en vez de un simulador a través de un circuito en modo aleatorio, lo que requiere de una destreza de maniobre más rápida y una memoria espacial más precisa, este punto de vista se lo debe tomar en cuenta para el desarrollado en trabajos futuros orientado a personas más jóvenes y modificar el sistema del dron para que sea maniobrado irrealmente a mayor velocidad y con una mecánica distinta. Tal como es el caso de videojuegos bajo las consolas de Nintendo Wii o de Microsoft Kinect donde los usuarios deben utilizar movimientos para la interacción con el sistema, un trabajo futuro con realidad virtual y sensores en todo el cuerpo sería una forma de mejorar la experiencia virtual y visual que las consolas mencionadas.

Con el fin de ampliar la cantidad de dispositivos en los que pueda ser ejecutado este simulador, es posible generar un Plugin nativo bajo el lenguaje de programación Objective-C y con esto agregar soporte de Bluetooth Low Energy para dispositivos iOS.

9. Referencias

- [1] "Unity - Game Engine". Unity3d.com. N.p., 2016. Web. 2 Sept. 2016. Available: <https://unity3d.com/es>
- [2] E. Mackensen, M. Lai and T. M. Wendt, "Bluetooth Low Energy (BLE) based wireless sensors," 2012 IEEE Sensors, Taipei, 2012, <http://ieeexplore.ieee.org/document/6411303>
- [3] Zarza, Leonardo Arcadio Tarza, Sistema de aeronaves no tripuladas, Escuela Superior de Guerra Conjunta, 2013
- [4] Bento, M. D. F. (2008). Unmanned Aerial Vehicles : An Overview. insideGNSS, 54–61. Retrieved from <http://www.insidegnss.com/auto/janfeb08-wp.pdf>
- [5] Cavoukian, A. (2012). Privacy and Drones : Unmanned Aerial Vehicles. Privacy by Design, Canada, 1(1), 1–27. Available: <http://www.publicsafety.gc.ca/lbrr/archives/cnmcs-plcng/cn29822-eng.pdf>
- [6] Dijkshoorn, N., & Visser, A. (2011). Integrating Sensor and Motion Models to Localize an Autonomous AR. Drone. International Journal of Micro Air Vehicles, 3(4), 183. <http://doi.org/10.1260/1756-8293.3.4.183>
- [7] "(IMAV) International Micro Air Vehicles: Indoor Competition". Dgon-imav.org. N.p., 2016. Web. 2 Sept. 2016.
- [8] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), 35. <http://doi.org/10.1115/1.3662552>
- [9] Guillermo Climent González, Desarrollo de plataforma de aplicaciones de ayuda para niños con necesidades especiales sobre raspberry pi, uam, 2014
- [10] Cristian Fernández Del Pozo, Realidad virtual para personas que tienen discapacidad motora, UAM, 2016
- [11] Baños Rivera, R.M.; Botella Arbona, C.; Perpiñá, c.; Quero Castellano, S. Tratamiento mediante realidad virtual para la fobia a volar: un estudio de caso Clínica y Salud, vol. 12, Colegio Oficial de Psicólogos de Madrid Madrid, España, <http://www.redalyc.org/articulo.oa?id=180618319006>
- [12] Francisco Moreno, Jordan Ojeda, Esmitt Ramírez J., Christiam Mena, Omaira Rodríguez, Un Framework para la Rehabilitación Física en Miembros Superiores con Realidad Virtual
- [13] Rezende, L. F. M. de, Rey-López, J. P., Matsudo, V. K. R., & Luiz, O. do C. (2014). Sedentary behavior and health outcomes among older adults: a systematic review. BMC Public Health, 14(1), 1–9. <http://doi.org/10.1186/1471-2458-14-333>
- [14] Pernambuco, C. S. (2012). Quality of life, elderly and physical activity. Health, 04(02), 88–93. <http://doi.org/10.4236/health.2012.42014>
- [15] Chodzko-Zajko, W. J., Proctor, D. N., Fiatarone Singh, M. A., Minson, C. T., Nigg, C. R., Salem, G. J., & Skinner, J. S. (2009). Exercise and physical activity for older adults. Medicine and Science in Sports and Exercise, 41(7), 1510–1530. <http://doi.org/10.1249/MSS.0b013e3181a0c95c>

- [16] Reis, L. S., Reis, G., Barroso, J., & Pereira, A. (2012). AMIGA - An interactive musical environment for Gerontechnology. *Procedia Computer Science*, 14(Dsai), 208–217. <http://doi.org/10.1016/j.procs.2012.10.024>
- [17] Marcelino, I., Laza, R., & Pereira, A. (n.d.). Research Article SSN : Senior Social Network for improving quality of life, 1–14. <http://www.hindawi.com/journals/ijdsn/aip/2150734/>
- [18] Jung, Y., Li, W., Gladys, C., & Lee, K. M. (2009). Games for a Better Life : Effects of Playing Wii Games on the Well-Being of Seniors in a Long-Term Care Facility. *Proceedings of the Sixth Australasian Conference on Interactive Entertainment*, 0–5. <http://doi.org/10.1145/1746050.1746055>
- [19] Gerling, K., Livingston, I., Nacke, L., & Mandryk, R. (2012). Full-body motion-based game interaction for older adults. *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12*, 1873–1882. <http://doi.org/10.1145/2207676.2208324>
- [20] Wijnand IJsselsteijn, Henk Herman Nap, Yvonne de Kort, Karolien Poels, *Digital Game Design for Elderly Users*, Eindhoven University of Technology, The Netherlands, 2007, <http://doi.org/10.1145/1328202.1328206>
- [21] *Raspberry Pi Projects Book*, The Magpi Magazine.
- [22] Yassir Akhayad, *Bluetooth 4.0 Low Energy: Análisis de las prestaciones y aplicaciones para la automoción*, 2016
- [23] Daniel Gómez Fernández, *Sistemas de sensores inteligentes para detección de automóviles*, UAM, 2014
- [24] Luis Garijo Gutierrez, Francisco Falcone Lanas, *Diseño y estudio de un sistema de comunicación inalámbrico basado en tecnología bluetooth low energy con desarrollo de protocolo propio de enrutamiento*, Universidad Politecnica de Navarra, España, 2016
- [25] R. Faragher, R. Harle, *An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications*, University of Cambridge, UK
- [26] Kevin Townsend, *Introduction to Bluetooth Low Energy*, Adafruit, 2014, available at <https://cdn-learn.adafruit.com/downloads/pdf/introduction-to-bluetooth-low-energy.pdf>
- [27] Dr.Hussein Abdul-Ridha Mohammed1 , Rasha Asaad Kamil 2, *Applying the Concepts of Panorama in Virtual Reality Using the Map Principle on a Library Website*, 2014
- [28] Yutaka Yoshisaka, *Panorama To Cubemap*, Unity Asset Store, Version: 1.0.0 (Dec 16, 2013), <https://www.assetstore.unity3d.com/en/#!/content/13616>
- [29] Samartha S M, Ritvik S Kotwal and Vidyashri V Mutnal, *Exploring the world of virtual reality gaming with Google Cardboard and Unity*, *International Journal of Modern Trends in Engineering and Research*, 2016

- [30] ASHWIN JAISHANKER, Virtual reality 101 – The different types of VR headsets, YourStory , 2016, Available at <https://yourstory.com/2016/06/virtual-reality-headset/>
- [31] Vago Santana, L., Brandao, A. S. A. S., Sarcinelli-Filho, M., Carelli, R., Santana, L. V., Brandao, A. S. A. S., ... Carelli, R. (2014). A Trajectory Tracking and 3D Positioning Controller for the AR . Drone Quadrotor. 2014. <http://doi.org/10.1109/ICUAS.2014.6842321>
- [32] Elfeky, M., Elshafei, M., Saif, A. A., & Al-malki, M. F. (2016). Modeling and Simulation of Quadrotor UAV with Tilting Rotors, 14(4), 1–9. <http://doi.org/10.1007/s12555-015-0064-5>
- [33] Unity Technologies, JAR plug-ins, Unity Documentation, <https://docs.unity3d.com/Manual/AndroidJARPlugins.html>
- [34] Android, Android developer reference, AdvertiseSettings, <https://developer.android.com/reference/android/bluetooth/le/AdvertiseSettings.html>
- [35] Tony DiCola, Adafruit Industries, Bluefruit LE Python Library, 2017, available: <https://cdn-learn.adafruit.com/downloads/pdf/bluefruit-le-python-library.pdf>
- [36] Guadalupe Crespo Quirós, Sistema de enlace robusto para la teleoperación de un UAV (vehículo aéreo no tripulado) en la plataforma robótica ARGOS, UAM, 2015
- [37] Raspberry Pi, Raspberry Pi Learning Resources, Getting Started with the Sense HAT, <https://www.raspberrypi.org/learning/getting-started-with-the-sense-hat/worksheet/>
- [38] Konstantin Mikhaylov 1, Nikolaos Plevritakis 2 and Jouni Tervonen, Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciiTI, J. Sens. Actuator Netw. 2013
- [39] DJI, Phantom 4, <https://www.dji.com/phantom-4>
- [40] DJI, Mobile SDK, Flight Control, https://developer.dji.com/mobile-sdk/documentation/introduction/flightController_concepts.html
- [41] DJI, Testing, Profiling & Debugging, Aircraft Simulator, <https://developer.dji.com/mobile-sdk/documentation/application-development-workflow/workflow-testing.html>

10. Anexos

A. Ejemplo de 2 set de gafas VR que se usaron durante el desarrollo de este trabajo.

En la figura 64 se muestran 2 marcas diferentes de gafas VR en las que se a probando el simulador, a pesar de que ambas ofrecen la misma calidad visual, la marca de BOBOVR z4 ofreció mayor confort, además que trae integrado 2 auriculares que permiten encerrar el sonido y mejorar la experticia sonora durante el uso del mismo.



Figura 64 - Marcas diferentes de gafas VR probadas con el Simulador

B. Imágenes de usuarios utilizando el Simulador Virtual.

Las imágenes de la Figura 65, muestran algunos de los usuarios que utilizaron el simulador virtual en todas sus configuraciones. El lugar de prueba fueron las instalaciones de la Universidad de las Fuerzas Armadas, donde los docentes de mayor edad se ofrecieron voluntariamente a probar y disfrutar del producto de este trabajo.





Figura 65 - Usuarios utilizando el Simulador Virtual

C. Usuario que tubo problemas al usar el Simulador con lentes

Como se explico en el Capitulo 66, entre los usuarios que utilizaron el Simulador hubo 2 que tubo problemas al usarlo con lentes puesto, en la Figura 65, se aprecia uno de los usuarios que utilizo el sistema sin lentes, aun así dijo que era algo interesante pero no se sintió muy cómodo al usarlo sin sus lentes.



Figura 66 - Usuario que presentó incomodidad al usar el Simulador sin lentes

D. Usuarios jóvenes que también disfrutaron utilizando el Simulador

A pesar que el este trabajo fue orientado para ser usado en personas mayores, fue probado por personas mas jóvenes, los cuales disfrutaron mucho del mecanismo de maniobre que posee este simulador, es por ello que se habla en trabajos Futuros de una propuesta mas compleja de implementación para personas mas jóvenes. En la Figura 67, se muestra 2 de los usuarios que mas interés tuvieron en utilizar el simulador.



Figura 67 - Usuarios jovenes utilizando el Simulador

E. Control de un Phantom 4 utilizando los Raspberry Pi desde un entorno simulado.

Tal como se explica en el capítulo 7, la empresa desarrolladora del Phantom 4 (DJI) a puesto a disposición, un simulador orientado a los desarrolladores con le objetivo de permitir probar el desarrollo de aplicaciones nativas en un ambiente simulado antes de probarlas en un ambiente real. En la Figura 68, se muestra una captura en la que se aprecia como se realizaron las pruebas utilizando el Simulador (DJI Assistenat 2) para las pruebas simuladas. Incluso en un ambiente simulado, el Phantom 4 debe estar prendido y conectado al computador a través de un cable USB, el control remoto del UAV debe estar conectado de igual manera en el celular y los dispositivos portables (Raspberry Pi) conectados a través de BLE al celular. Al estar conectado el UAV al computador, el simulador recibe las acciones que debería procesar el Dron real pero las abstrae y las reproduce a través del simulador DJI.



Figura 68 – Control de un drone real (Phantom 4) utilizando los Raspberry Pi en un entorno simulado

F. Interfaz de usuario de la aplicación nativa para maniobrar un Drone real.

En la Figura 69, se aprecia la interfaz de usuario de la aplicación nativa en funcionamiento, esta aplicación contiene 3 botones los cuales permiten que el Dron despegue (TAKE OFF), se dirija al punto de inicio (GO HOME) y que aterrice (LANDING), a pesar de que el Drone es controlado por los Raspberry Pi, es necesario dar una orden de despegue antes de proceder con el maniobre, de igual manera el Drone requiere una orden especifica de aterrizaje para que aterrice de forma segura. Tal como se explico en el capitulo 7, por razones de seguridad fue necesario realizar las pruebas con la supervisión de una segunda persona que este pendiente con el control remoto del Phantom 4 en caso de alguna emergencia.

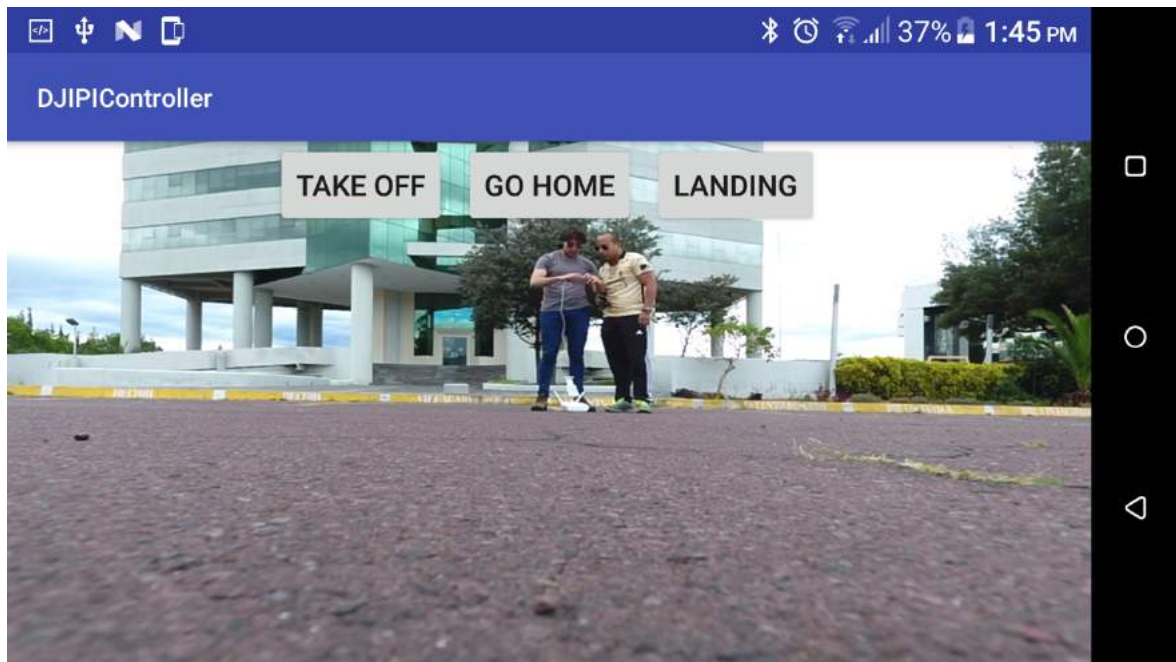


Figura 69 - Interfaz grafica de la aplicacion nativa que controla un dron real (Phantom 4)

G. Control de un Phantom 4 utilizando los Raspberry Pi en un entorno real.

El maniobrar del Phantom 4 a través de los Raspberry Pi, fue una experiencia totalmente increíble y emocionante, estas pruebas fueron realizadas por mi persona y siempre en compañía de otra, a diferencia de maniobrar el dron a través de simulador desarrollado en este trabajo, maniobrar un Dron real con el mismo mecanismo es un poco diferente, el viento o la inestabilidad que posee el Dron hacen que la experiencia sea un poco mas compleja. En la figura 70 se muestran capturas en las que se a probado la aplicación nativa y con ello se a podido maniobrar un dron real a través de dispositivos portables.



Figura 70 - Control de un drone real (Phantom 4) utilizando los Raspberry Pi en un entorno real