

**UNIVERSIDAD DE EL SALVADOR**  
**FACULTAD DE INGENIERÍA ARQUITECTURA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS INFORMÁTICOS**



**TÉCNICAS DE SIMULACIÓN**

**CICLO I - 2024**

**EVALUACIÓN: PROYECTO – ETAPA II**

**TEMA DE PROYECTO:**

**DISEÑO E IMPLEMENTACIÓN DE UN MODELO DE SIMULACIÓN A TRAVÉS  
DE UN ENTORNO DE REALIDAD VIRTUAL**

**TUTOR: MSc. Luis Kelman Belloso**

<b>ALUMNOS</b>		<b>CARNET</b>
Carlos Alexander De León Gutiérrez		DG12003
Cristian Antonio Escalante Hernández		EH18010
Kevin Stanley Melgar Rivera		MR21083
Andrea Esmeralda Raymundo Salazar		RS21037
Antonio Natanael Cabezas Sánchez		CS21062

**SEDE: Central – UES**

**Ciudad Universitaria, 18 de Junio de 2024**

## INDICE

INTRODUCCIÓN.....	3
OBJETIVOS.....	3
CAPITULO I – GENERALIDADES.....	4
CAPITULO II – ANALISIS.....	12
CAPITULO III – PLANTEAMIENTO DEL PROBLEMA.....	14
CAPITULO IV – REQUERIMIENTOS.....	15
CAPITULO V – ESTRUCTURA DE UNA VR APP.....	16
CAPITULO VI – MODELO DE SIMULACIÓN.....	20
CAPITULO VII – IMPLEMENTACIÓN EN UNITY.....	23
CONCLUSIONES.....	29
ANEXOS.....	29

# INTRODUCCIÓN

La realidad virtual cada día es mas importante en las diferentes actividades humanas, ya que permite una experiencia enriquecida a los usuarios, esto puede ser aprovechado en el enfoque de nuevas metodologías de de enseñanza y educación.

El siguiente informe esta estructurado de la siguiente manera:

Capitulo 1: Se brinda contexto general sobre simulación y realidad virtual

Capitulo 2: Realizamos un análisis del problema, y su contexto tomando en cuenta el Laboratorio VIROO de la Universidad de El Salvador

Capitulo 3: Se realiza el planteamiento del problema, y las variables que intervienen

Capitulo 4: Describimos los requerimientos generales del proyecto.

Capitulo 5: Describimos el modelo de simulación elegido, y sus características.

Capitulo 6: Mostramos su implementación en Unity.

# OBJETIVOS

## Objetivo General

- Implementar un modelo de simulación, mediante el paradigma de realidad virtual, utilizando Unity 3D

## Objetivo Específicos

- Determinar las características del modelo de simulación
- Comprender las funcionalidades básicas de aplicación de VR
- Investigar sobre los entornos de realidad virtual, conceptos principales, y componentes.
- Desarrollar el modelo de simulación elegido, utilizando Unity
- Realizar pruebas de la aplicación VR

# CAPITULO I – GENERALIDADES

## ANTECEDENTES

La historia de la Realidad Virtual empezó a principios del siglo XX, allá por el año 1935. En ese año, encontramos la primera referencia específica a unas gafas de Realidad Virtual, cuyo autor fue el escritor de ciencia ficción Stanley G. Weinbaum en su cuento titulado “Las gafas del Pígalión”.

El cuento gira alrededor del invento de un profesor, que permite a quien lo usa, transportarse a otros mundos y lugares a través de unas gafas.

Las primeras gafas de Realidad Virtual fueron creadas en 1960 por Morton Heiling, y por ello se le conoce como el padre de la Realidad Virtual.

Este cinematógrafo e inventor, había desarrollado el Sensorama en 1957, una máquina bastante aparatosa que permitía al usuario disfrutar de experiencias multisensoriales. Su nuevo invento, diseñado para ser utilizado como un equipo portátil en la cabeza del usuario, se denominó aparato de televisión estereoscópica para uso individual, fue este el inicio de la historia de la realidad virtual.



**Sensorama. Fuente: Wikipedia.org**

A pesar de que ese casco de Realidad Virtual actualmente puede parecer muy rudimentario, el concepto era ya muy similar a los equipos actuales, y por ello se considera que fueron las primeras gafas de Realidad Virtual de la historia.

## **MARCO TEÓRICO**

### **SIMULACIÓN DE PROCESOS**

#### **SISTEMA. EXPERIMENTO. MODELO**

El modelado y la simulación son formas de adquirir conocimiento acerca del comportamiento de los sistemas. Por tal motivo, conviene definir qué se entiende por sistema. Puede considerarse que un sistema es "cualquier objeto cuyas propiedades se desean estudiar". Es decir, "cualquier fuente potencial de datos puede considerarse que es un sistema" (Urquía & Martín, 2013). Así, por ejemplo, una fábrica con máquinas, personal y galpón sería un sistema. Una manera de conocer el comportamiento de un sistema es experimentar con él.

Un experimento es "el proceso de extraer datos de un sistema sobre el cual se ha ejercido una acción externa" (Urquía & Martín, 2013). Por ejemplo, el jefe de producción de una fábrica puede ensayar diferentes procedimientos de distribución y ubicación de las máquinas para establecer qué combinación muestra un mejor equilibrio entre costo y tiempo de proceso. Aun cuando sea posible experimentar sobre el sistema real, en ocasiones no es recomendable por el alto coste económico. Considérese un empresario que debe decidir si ampliar o no las instalaciones de su fábrica, para lo cual necesita estimar si la ganancia potencial que supondrían las nuevas instalaciones justifica el costo de realizar la ampliación. Experimentar con el sistema real supondría realizar la ampliación con el fin de evaluar su rendimiento económico, lo cual no resulta razonable.

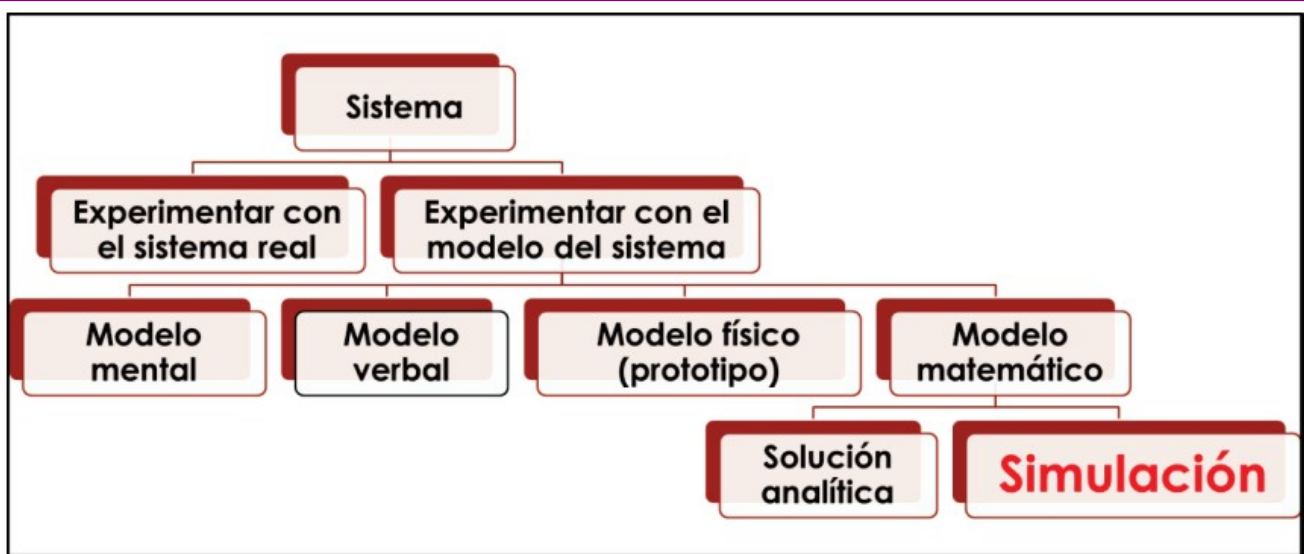
Una alternativa a la experimentación con el sistema real consiste en realizar un modelo del sistema y experimentar con el modelo. Un modelo "es una representación de un sistema desarrollada para un propósito específico" (Urquía & Martín, 2013). Regresando al ejemplo de la ampliación de la fábrica, en lugar de experimentar con el sistema real, puede realizarse un modelo de la operación de cada una de las configuraciones de la fábrica (la actual y la ampliada) y comparar el comportamiento de los modelos. En definitiva, experimentar con un modelo resulta menos costoso y más seguro que experimentar directamente con el sistema real. Además, con un modelo adecuado se pueden ensayar condiciones de operación extremas que son impracticables en el sistema real.

#### **LA SIMULACIÓN**

La simulación "es una imitación del funcionamiento de un sistema real por medio de un modelo que se comporta de forma análoga" (Vaán 2014). Es decir, la simulación es una forma de reproducir las condiciones de un sistema real a través de un modelo matemático o computarizado, con el objetivo de estudiar, evaluar, predecir, rediseñar, probar, mejorar y optimizar el desempeño de dicho sistema.

Se recomienda usar simulación en los siguientes casos:

- Cuando el sistema real no existe, es costoso, peligroso, consume mucho tiempo o es imposible de construir y experimentar con prototipos (por ejemplo. un nuevo computador o un reactor nuclear).
- Cuando se tenga necesidad de estudiar el pasado, presente o futuro del sistema en tiempo real, tiempo expandido o tiempo comprimido (sistemas de control a tiempo real, estudios en cámara lenta, crecimiento poblacional).
- Cuando el sistema es tan complejo que su evaluación analítica es difícil. ya sea porque el modelado matemático es imposible o porque el modelo matemático no tiene solución analítica o numérica (ecuaciones diferenciales no lineales, problemas estocásticos).



Simulación en Ingeniería – Fuente: Elaboración Propia

# SISTEMA DE REALIDAD VIRTUAL

## Definición

Un sistema de realidad virtual es un sistema el cuál crea un ambiente tridimensional por medio de una computadora. Es decir, la única manera interactuar con este entorno virtual es mediante una computadora, para lo cual se vuelve necesario el uso de periféricos especiales para la transmisión de información del mundo real hacia el mundo virtual.



**Visor de Realidad Virtual**

Estos periféricos de entrada y salida de información incluyen gafas, auriculares, guantes o inclusive trajes; a mayor número de periféricos mayor grado de inmersión

Dentro del listado de los periféricos utilizados, los que se vuelven más prescindibles a medida el software y hardware para la realidad virtual evoluciona son los utilizados como entrada el envío de información de las acciones del usuario.

## Componentes

Como todos los sistemas digitales, está sustentado por dos partes: hardware y software.

En la parte del hardware se encuentra la computadora, que procesa todo el código fuente, ejecuta el motor de renderización y demás elementos del sistema para transmitirlo a los periféricos antes mencionados, todo orquestado por la computadora el tiempo real con base a las acciones del usuario que está interactuando con el entorno virtual.

Por el lado del software, se encuentran las piezas gráficas, sonoras y el código fuente que define como se comportarán las piezas gráficas cuando el usuario interactúe con ellas.

Por ejemplo, un clavo en un entorno virtual que simula un ambiente de manufactura de productos contruidos con madera es la pieza gráfica, pero la interacción de este clavo con un martillo sostenido por el usuario cuando este lo golpeé sobre una superficie para dejarlo clavado se obtiene del comportamiento definido por el código fuente del entorno virtual, el cual definió el comportamiento del clavo a quedarse sostenido sobre la superficie en la que fue puesto previo a ser golpeado con el martillo por el usuario.



Con lo anterior claro, se puede resumir que un sistema de realidad virtual está construido por los siguientes elementos:

Componente	Descripción	Tipo
<b>Entorno virtual</b>	<p>Es el primer paso de todos. Consiste en el diseño y construcción de un entorno virtual mediante tecnología 3D para la creación de texturas, efectos visuales y de audio.</p> <p>Para este propósito se hace uso de herramientas para el diseño 3D y motores que rendericen los diseños creados.</p>	Software
<b>Visualización y audio</b>	Mediante periféricos de visión se introduce al entorno virtual modelado, mientras que la adición de periféricos de audio permite generar una inmersión total en dicho entorno.	Hardware
<b>Interacción del usuario</b>	<p>Aunque de la mano con el componente de hardware, debido a que son el medio por el cual se transmite información de lo que el usuario hace dentro del entorno, es clasificado dentro del software debido a la importancia de la programación del comportamiento del ambiente y sus elementos respecto a las acciones del usuario en este.</p> <p>Además, a medida que la construcción de sistemas evoluciona periféricos de entrada para la transmisión de información son cada vez menos necesarios, siendo reemplazados por sensores de movimiento o cámaras dentro del mismo periférico de visión.</p>	Software
<b>Procesamiento en tiempo real</b>	Con los medios de entrada (periféricos) y el comportamiento de los elementos del ambiente (programación) listos, se debe de orquestar la interacción entre ambos para que en tiempo real estos puedan interactuar de manera fluida.	Software

<b>Retroalimentación háptica</b>	<p>Es un aporte a un nivel mayor de inmersión. Además de la interacción visual y auditiva, permitir la respuesta de los elementos mediante sensaciones hápticas como vibraciones, sensaciones de textura permiten una mayor inmersión en la realidad virtual y abstracción del mundo real.</p> <p>El detalle es, que para lograr este tipo de inmersión es necesario que los periféricos utilizados sean de una mayor gama, debido a las características de respuesta especiales con las que cuentan.</p>	<b>Hardware</b>
----------------------------------	---	-----------------

## GRAFICAS EN 3D POR COMPUTADORA

Son gráficos que utilizan una representación tridimensional de datos geométricos (a menudo cartesianos) que se almacenan en el ordenador con el propósito de realizar cálculos y representar imágenes 2D, que se pueden almacenar para verlas más tarde o mostrarlas en tiempo real. Además, el término se puede referir al proceso de creación de dichos gráficos, o al campo de estudio de técnicas y tecnología relacionadas con los gráficos 3D.

Un gráfico 3D difiere de uno bidimensional principalmente por la forma en que ha sido generado. Este tipo de gráficos se originan mediante un proceso de cálculos matemáticos sobre entidades geométricas tridimensionales producidas en un ordenador, y cuyo propósito es conseguir una proyección visual en dos dimensiones para ser mostrada en una pantalla o impresa en papel.

En general, el arte de los gráficos tridimensionales es similar a la escultura o la fotografía, mientras que el arte de los gráficos 2D es análogo a la pintura. En los programas de gráficos por computadora esta distinción es normalmente difusa: algunas aplicaciones 2D utilizan técnicas 3D para alcanzar ciertos efectos como iluminación, mientras que algunas aplicaciones 3D primarias hacen uso de técnicas 3D.

### Creación de gráficos 3D

El proceso de creación de gráficos 3D por computadora puede ser dividido en estas tres fases básicas:

1. **Modelado**
2. **Composición de la escena**
3. **Rénder (creación de la imagen final)**

### Modelado

La etapa de modelado consta de ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversas técnicas de modelado; Constructive Solid Geometry,

modelado con NURBS y modelado poligonal son algunos ejemplos. Los procesos de modelado puede incluir la edición de la superficie del objeto o las propiedades del material (p.e., color, luminosidad, difusión, especularidad, características de reflexión, transparencia u opacidad, o el índice de refracción), agregar texturas, mapas de relieve (bump-maps) y otras características.

El proceso de modelado puede incluir algunas actividades relacionadas con la preparación del modelo 3D para su posterior animación. A los objetos se les puede asignar un esqueleto, una estructura central con la capacidad de afectar la forma y movimientos de ese objeto. Esto ayuda al proceso de animación, en el cual el movimiento del esqueleto automáticamente afectara las porciones correspondientes del modelo.

El modelado puede ser realizado por programas dedicados (p.e. Lightwave, Rhinoceros 3D, Moray, un componente de una aplicación (Shaper, Loftter en 3D Studio o por un lenguaje de descripción de escenas (como en POV-Ray. En algunos casos, no hay una distinción estricta entre estas fases; en dichos casos, el modelado es solo una parte del proceso de creación de escenas, por ejemplo, con Caligari trueSpace).

### **Composición de la escena**

Esta etapa involucra la distribución de objetos, luces, cámaras y otras entidades en una escena que será utilizada para producir una imagen estática o una animación. Si se utiliza para Animación, esta fase, en general, hace uso de una técnica llamada "Keyframing", que facilita la creación de movimientos complicados en la escena. Con la ayuda de la técnica de keyframing, en lugar de tener que corregir la posición de un objeto, su rotación o tamaño en cada cuadro de la animación, solo se necesita marcar algunos cuadros clave (keyframes). Los cuadros entre keyframes son generados automáticamente, lo que se conoce como 'Interpolación'.

La iluminación es un aspecto importante de la composición de la escena. Como en la realidad, la iluminación es un factor importante que contribuye al resultado estético y a la calidad visual del trabajo terminado. Por eso, puede ser un arte difícil de dominar. Los efectos de iluminación pueden contribuir en gran medida al humor y la respuesta emocional generada por la escena, algo que es bien conocido por fotógrafos y técnicos de iluminación teatral.

### **Tesselation y Mallas**

El proceso de transformar la representación de objetos, como el punto medio de coordenadas de una esfera y un punto en su circunferencia, en una representación poligonal de una esfera, se conoce como tessellation. Este paso es usado en el rénder basado en polígonos, donde los objetos son descompuestos de representaciones abstractas primitivas como esferas, conos, etc., en las denominadas mallas, que son redes de triángulos interconectados.

Las mallas de triángulos son populares ya que está probado que son fáciles de 'renderizar' usando Scanline rendering.

Las representaciones poligonales no son utilizadas en todas las técnicas de rénder, y en estos casos, el paso de tessellation no es incluido en la transición de representación abstracta y la escena 'renderizada'.

## **Renderizado**

Rénder se llama al proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado con tomar una foto o filmar la escena en la vida real, después que se terminó de armar. Generalmente se buscan imágenes de calidad fotorrealista, y para este fin se han desarrollado muchos métodos especiales. Las técnicas van desde el rénder de alambre (wireframe rendering), pasando por el rénder basado en polígonos, hasta las técnicas más modernas como: Scanline Rendering, Raytracing o Radiosity.

El software de rénder puede simular efectos cinematográficos como lens flare, Profundidad de campo, o Motion blur (desenfoque de movimiento). Estos artefactos son, en realidad, un producto de las imperfecciones mecánicas de la fotografía física, pero como el ojo humano está acostumbrado a su presencia, la simulación de dichos efectos aportan un elemento de realismo a la escena. Se han desarrollado técnicas con el propósito de simular otros efectos de origen natural, como la interacción de la luz con la atmósfera o el humo. Ejemplos de estas técnicas incluyen sistemas de partículas que pueden simular lluvia, humo o fuego, Muestreo volumétrico para simular niebla, polvo y otros efectos atmosféricos, y Cáusticas para simular el efecto de la luz al atravesar superficies refractantes.

El proceso de rénder requiere gran procesamiento, ya que requiere simular gran cantidad de procesos físicos complejos. La capacidad de procesamiento se ha incrementado rápidamente a través de los años, permitiendo un grado superior de realismo en los rénders. Estudios de cine que producen animaciones generadas por computadora hacen uso, en general, de lo que se conoce como Render Farm (granja de rénder) para generar imágenes de manera más rápida.

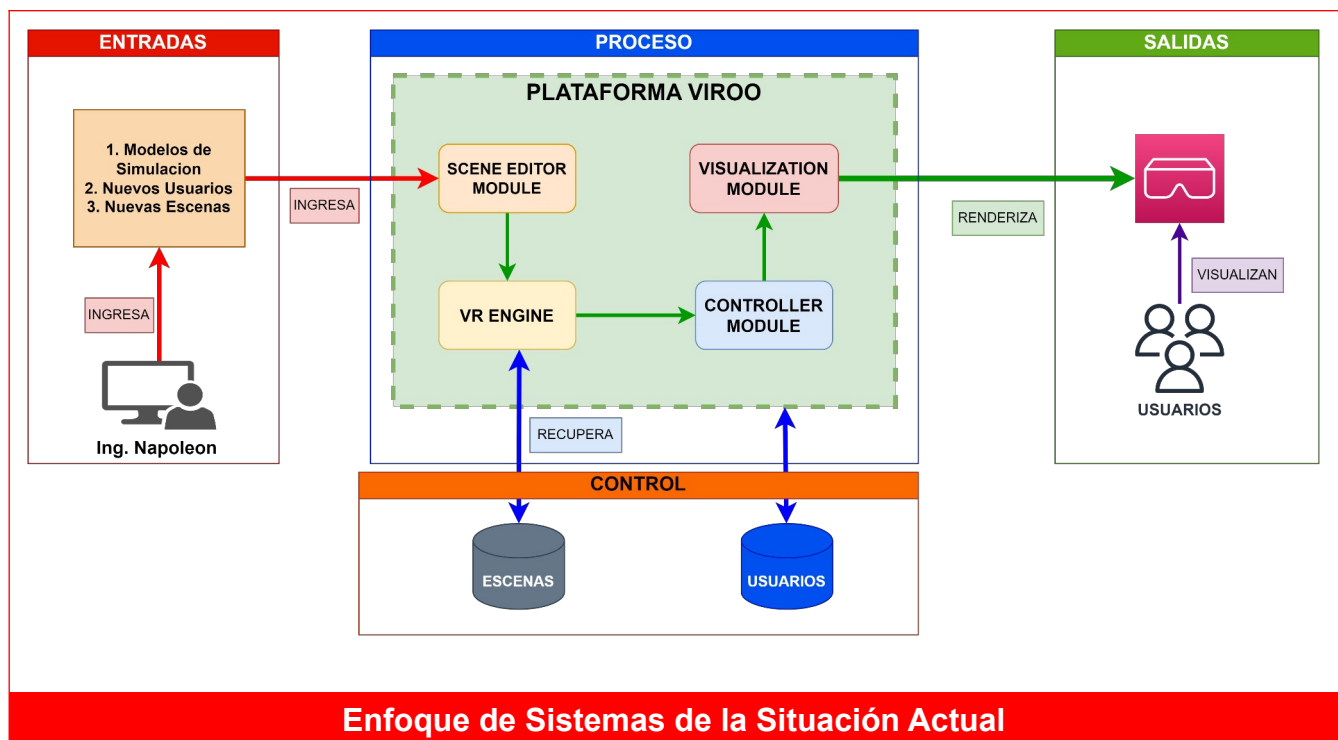
# CAPITULO II – ANALISIS

## ANÁLISIS DE LA SITUACIÓN ACTUAL

La Universidad de El Salvador da un paso más en la formación de sus profesionales y se adapta a los cambios tecnológicos que enfrenta la educación en el mundo, a partir del año 2022, el centro educativo cuenta con un Laboratorio de Realidad Virtual.

El diseño y equipamiento del espacio estuvo a cargo de la empresa española VIROO, que utiliza la tecnología denominada: Virtualware.

Utilizaremos el enfoque de sistemas para iniciar el análisis:



## Descripción de Elementos

### Entradas:

- 1. Modelos de Simulación:** Los estudiantes desarrollan nuevos modelos de simulación que son agregados a la base de datos de la plataforma.
- 2. Nuevos Usuarios:** Solo el administrador de la plataforma VIROO, puede agregar nuevos usuarios
- 3. Nuevas Escenas:** El administrador también puede agregar nuevas escenas.

## **Procesos**

- 1. SCENE EDITOR MODULE:** Permite editar y teleportar al usuario a diferentes escenas guardadas en la base de datos.
- 2. VR ENGINE:** Contiene el motor de simulación de leyes físicas, con el cual funcionan los modelos que se integraran.
- 3. CONTROLLER MODULE:** Permite controlar la navegación sobre la escena, o ir a una nueva.
- 4. VISUALIZATION MODULE:** Envía la información de renderizado a las gafas de realidad virtual del usuario.

## **Salidas:**

Escenas de realidad virtual interactivas con las cuales el usuario, puede interactuar y simular eventos mediante los componentes de hardware.

## **Control:**

- 1. Base de datos Escenas:** Contiene la información de las escenas guardadas en el sistema.
- 2. Base de Datos Usuarios:** Contiene la información de los usuarios y sus interacciones con el sistema.

**Medio Ambiente:** Todas demás facultades y departamentos de la Universidad de El Salvador

# CAPITULO III – PLANTEAMIENTO DEL PROBLEMA

## FORMULACIÓN DEL PROBLEMA

La educación tradicional basada en la lectura o la memorización de datos va dejando paso a nuevas estrategias educativas. Cada vez son más los estudios que señalan que el acceso a una gran cantidad de información no equivale a un aprendizaje eficaz. Por ello, muchos entornos educativos ya apuestan por experiencias cognitivas que mejoren la concentración y la retención de los alumnos.

Las Tecnologías de la Información y Comunicación (TIC) son una herramienta crucial para potenciar y enriquecer el conocimiento de los alumnos en todas las etapas educativas. El uso de la Realidad Virtual en las aulas ya forma parte de una nueva forma de aprender.

Estar realmente inmerso en el aprendizaje, con todos los sentidos enfocados, detona en los alumnos una máxima motivación en el conocimiento. En este sentido, la Realidad Virtual en las aulas se ha convertido en una herramienta educativa con gran potencial, y es capaz de generar nuevos entornos de conocimiento sin perder la practicidad y amenidad de la experiencia.

Es un mundo virtual compuesto por elementos visuales mediante gráficos que incluye sonidos ambientales precisos y las características del entorno generan una experiencia inmersiva. Además, existe una retroalimentación sensorial producida por la posición del usuario, que deberá interactuar con el entorno para sentir la realidad.

### **Enunciado del problema.**

**Desarrollar un modelo de simulación de eventos discretos de la Máquina de Turing, e implementarlo en un sistema de realidad virtual, mediante modelado de graficas 3D por computadora.**

La importancia histórica radica en que proporcionó el marco conceptual esencial para la construcción de computadoras modernas. La Máquina de Turing no solo impulsó el desarrollo tecnológico, sino que también transformó la comprensión misma de lo que es posible computar.

# CAPITULO IV – REQUERIMIENTOS

## DEFINICIÓN DE REQUERIMIENTOS

### Listado de Requerimientos Funcionales

ID Requerimiento	Nombre	Descripción
RF-001	Gestionar Vistas Interactivas	Permite interactuar con el modelo de simulación mediante un entorno de realidad virtual.
RF-002	Gestionar Escena	Carga o actualiza la escena dependiendo de la interacción del usuario.
RF-003	Cargar Modelo 3D	Carga y renderiza los modelos en 3D de la aplicación
RF-004		

### Listado de Requerimientos No Funcionales

ID Requerimiento	Nombre	Descripción
RNF-001	Plataforma	Implementar la simulación mediante un stack tecnología compatible con el marco de trabajo de realidad virtual VIROO.
RNF-002	Idioma	Desarrollar la documentación y texto necesario dentro de la simulación en idioma español, debido al lugar de implementación objetivo (laboratorio de VR UES).
RNF-003	Intuitivo	Generar una experiencia intuitiva para el usuario. Establecer un flujo simple de pasos a realizar para completar la simulación.



# CAPITULO V – ESTRUCTURA DE UNA VR APP

## CONTEXTO TEÓRICO

En el desarrollo de software son muy utilizados los patrones de diseño para plantear la arquitectura de la aplicación a desarrollar, para nuestra solución, proponemos el siguiente patrón:

### Patrón MVC

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador; es decir: por un lado define componentes para la representación de la información y, por otro lado, para la interacción del usuario.

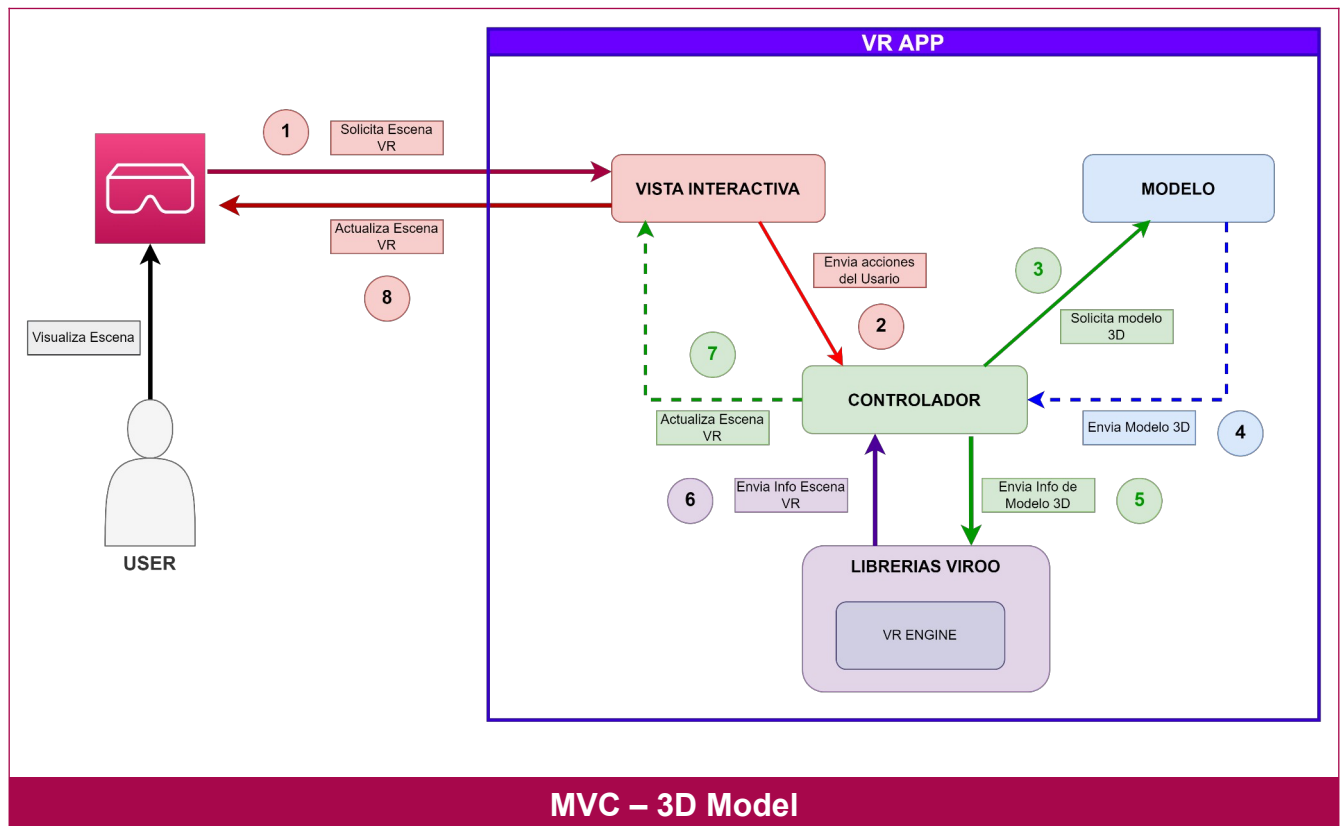
**El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).

**El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos).

**La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida

Sin embargo para nuestra propuesta realizaremos una modificación al patrón, esto debido a que estaremos trabajando con un modelo 3D, por tanto proponemos usar: **MVC-3D**

## ARQUITECTURA DE UNA VR APP



## DESCRIPCIÓN DE LOS COMPONENTES DE UNA VR APP

**Modelo:** Contendrá las graficas del modelo de simulación en 3D

**Controlador:** Gestiona las peticiones que viene de la vista interactiva, como actualizaciones de escena, cambio de escena, o actualización del modelo.

**Vista Interactiva:** Es la que interactúa con el usuario a través del componente hardware, actualizando los componentes o textura de su escena

**Librerías VIROO:** Se utilizaran los componentes y librerías de la plataforma VIROO, para desplegar nuestra aplicación.

DIAGRAMA GENERAL DE CASO DE USO

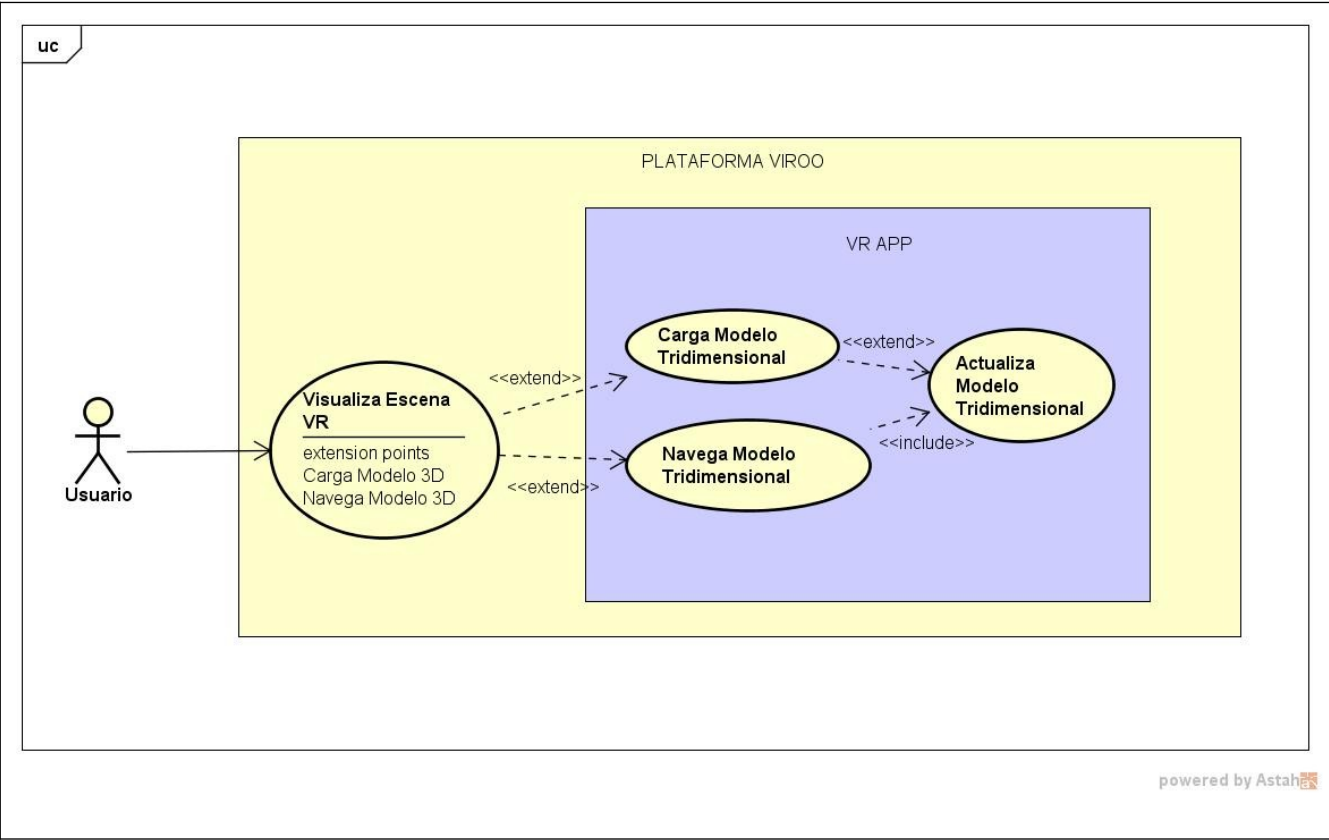
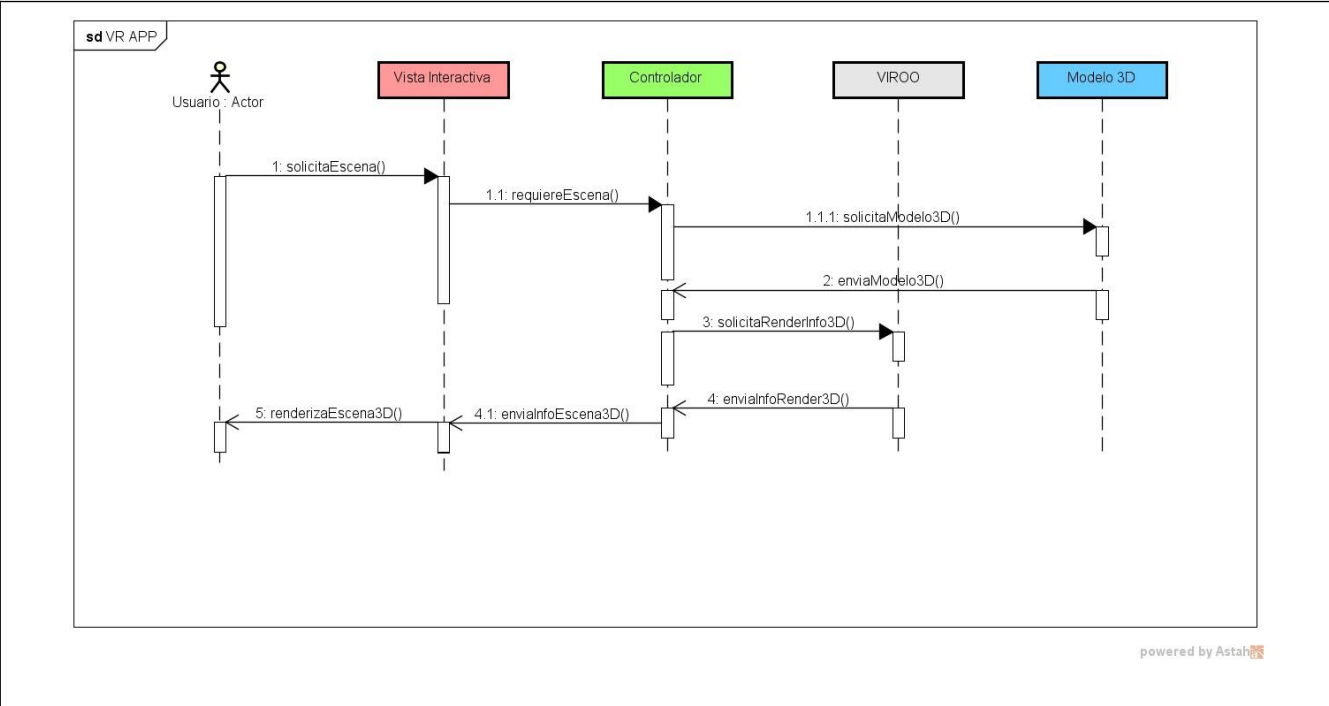



DIAGRAMA GENERAL DE SECUENCIA

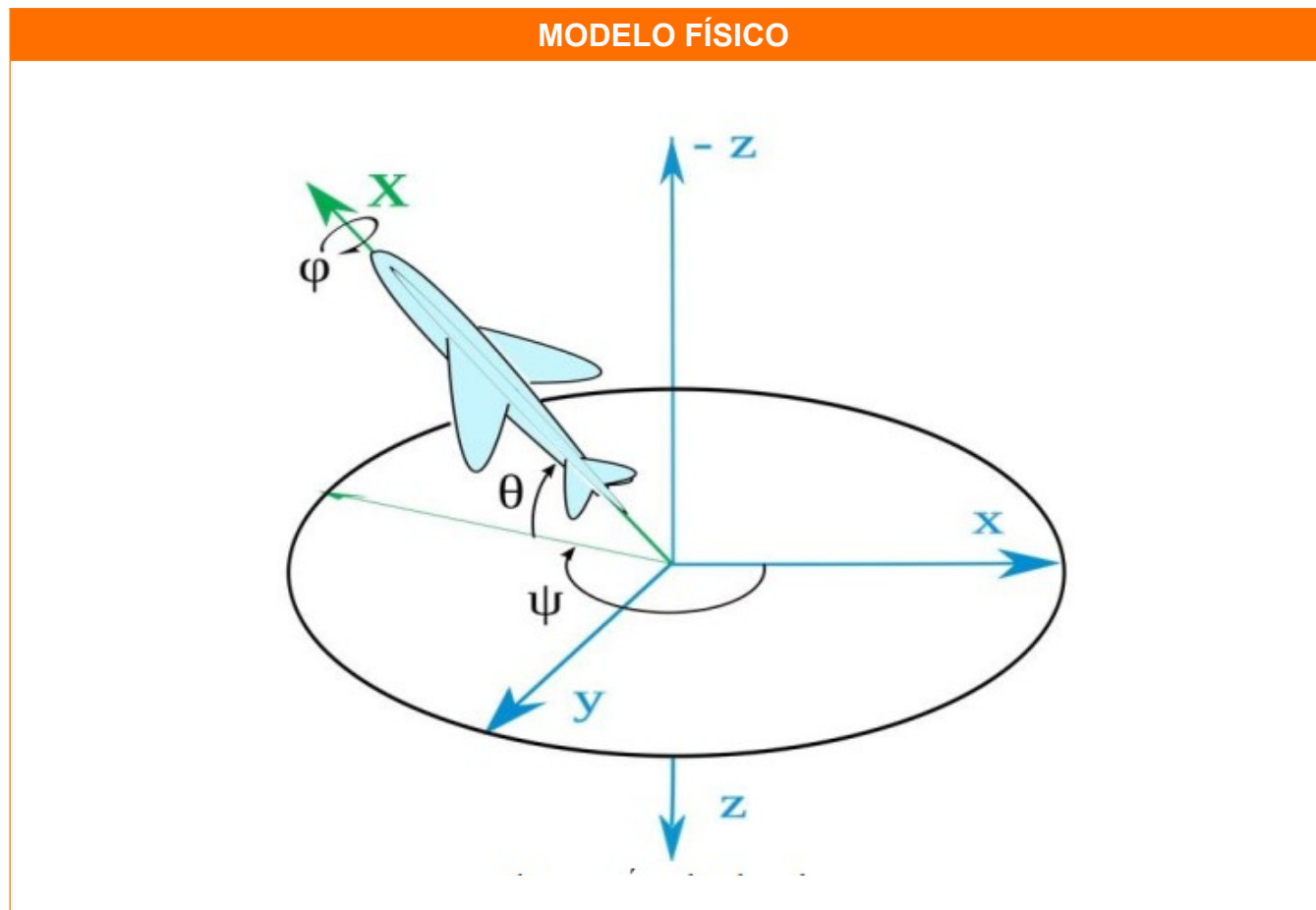


# CAPITULO VI – MODELO DE SIMULACIÓN

## ELECCIÓN DEL MODELO

ELECCION DEL MODELO	
<b>SIMULADOR DE VUELO</b>	
DESCRIPCION	IMPLEMENTACION
Un simulador de vuelo es una conjunción de herramientas de software y de hardware que permiten crear una réplica fiel, precisa y tan semejante a la realidad como sea posible del comportamiento de un determinado aeronave en vuelo.	

## CARACTERÍSTICAS DEL MODELO



## MODELO MATEMATICO - ECUACIONES DE MOVIMIENTO

Definición 7.3 (ecuaciones 3DOF de movimiento 3D)

$$m\dot{V} = T - D - mg \sin \gamma; \quad (7.1.3.1)$$

$$mV\dot{\chi} \cos \gamma = L \sin \mu; \quad (7.1.3.2)$$

$$mV\dot{\gamma} = L \cos \mu - mg \cos \gamma; \quad (7.1.3.3)$$

$$\dot{x}_e = V \cos \gamma \cos \chi + W_x; \quad (7.1.3.4)$$

$$\dot{y}_e = V \cos \gamma \sin \chi + W_y; \quad (7.1.3.5)$$

$$\dot{h}_e = V \sin \gamma; \quad (7.1.3.6)$$

$$\dot{m} = -T\eta. \quad (7.1.3.7)$$

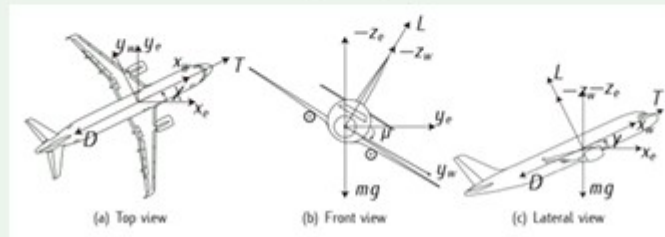
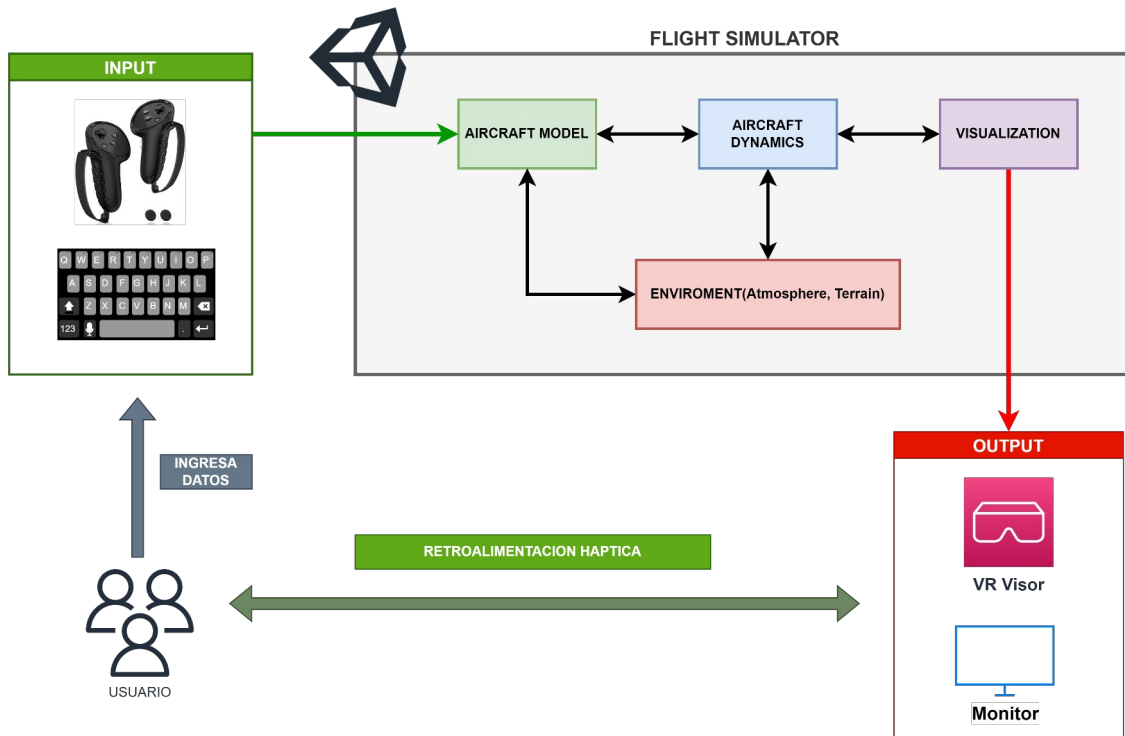


Figura 7.2: Fuerzas aéreas.

## DISEÑO DE SIMULACIÓN DEL MODELO

### DIAGRAMA COMPONENTES DE FLIGHT SIMULATOR



EN DONDE:

AIRCRAFT MODEL

MODELO FISICO

AIRCRAFT  
DYNAMICS

MODELO  
MATEMATICO



# CAPITULO VII – IMPLEMENTACIÓN EN UNITY

En este capítulo, se explora como se implemento en Unity, los modelos y componentes del modelo de simulación

## IMPLEMENTACIÓN DEL MODELO FÍSICO



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class moveTrail : MonoBehaviour {
6
7      public int moveSpeed = 500;
8
9      // Update is called once per frame
10     void Update () {
11         transform.Translate(-Vector3.right * Time.deltaTime * moveSpeed);
12         Destroy(gameObject, 5);
13     }
14
15     private void OnTriggerEnter(Collider other)
16     {
17         if (other.gameObject.tag == "sphere")
18         {
19             other.GetComponent<Renderer>().enabled = false;
20         }
21     }
22 }
23 }

```

## IMPLEMENTACIÓN DEL MODELO MATEMÁTICO

### MOVIMIENTO

$$\vec{F} = \frac{d(m\vec{V})}{dt}$$

### IMPLEMENTACIÓN EN UNITY





```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlaneMovement : MonoBehaviour
6  {
7      Rigidbody p_rigidBody;
8
9      public float speed = 0;
10     public float maxSpeed = 10;
11     public float acceleration = 10;
12     public float deceleration = 10;
13
14     float tiltAngle = 90.0f;
15     float tiltAroundX;
16     float tiltAroundZ;
17     float tiltAroundY;
18
19     float changeY;
20
21     // Use this for initialization
22     void Start()
23     {
24         p_rigidBody = GetComponent<Rigidbody>();
25     }
26
27     // Update is called once per frame
28     void Update()
29     {
30         Vector2 input = OVRInput.Get(OVRInput.Axis2D.SecondaryThumbstick);
31         //Debug.Log(input.x + ", " + input.y);
32         Vector2 input2 = OVRInput.Get(OVRInput.Axis2D.PrimaryThumbstick);
33         //Debug.Log(input2.x + ", " + input2.y);
34     }
```

## VELOCIDAD

$$V_x = V_{ix} + \left( \frac{V_e \cdot k_h}{m_a} \right) \cdot t$$

## IMPLEMENTACIÓN EN UNITY

```
1  /* ACCELERATE, DECELLERATE, YAW */
2      if (input.y > .7)
3      {
4          p_rigidBody.AddForce(transform.right * -200);
5      }
6      if (input.x < -.7)
7      {
8          transform.Rotate(transform.up, Time.deltaTime * -30);
9      }
10     if (input.y < -.7)
11     {
12         p_rigidBody.AddForce(transform.right * 30);
13     }
14     if (input.x > .7)
15     {
16         transform.Rotate(transform.up, Time.deltaTime * 30);
17     }
18
```

## ROTACIÓN

Ecuación Cinemática de Rotación:

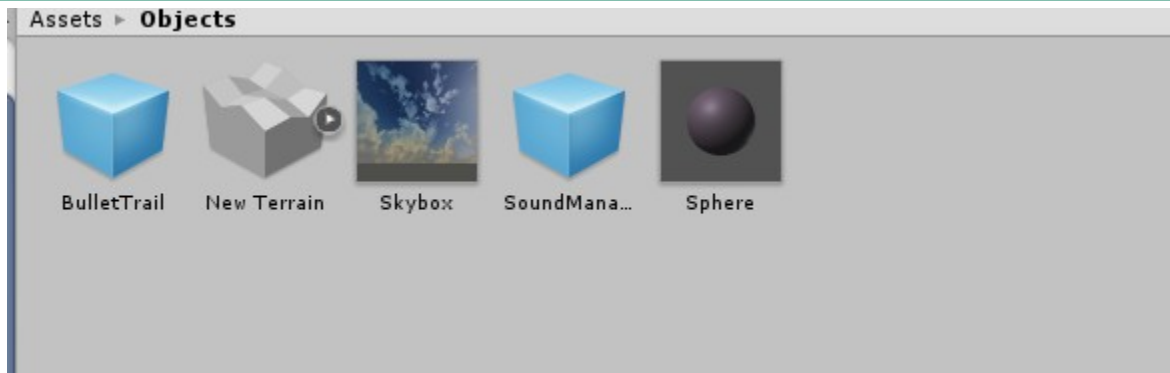
$$\dot{q} = \frac{1}{2} \mathbf{H}(\vec{\omega}) q$$

## IMPLEMENTACIÓN EN UNITY

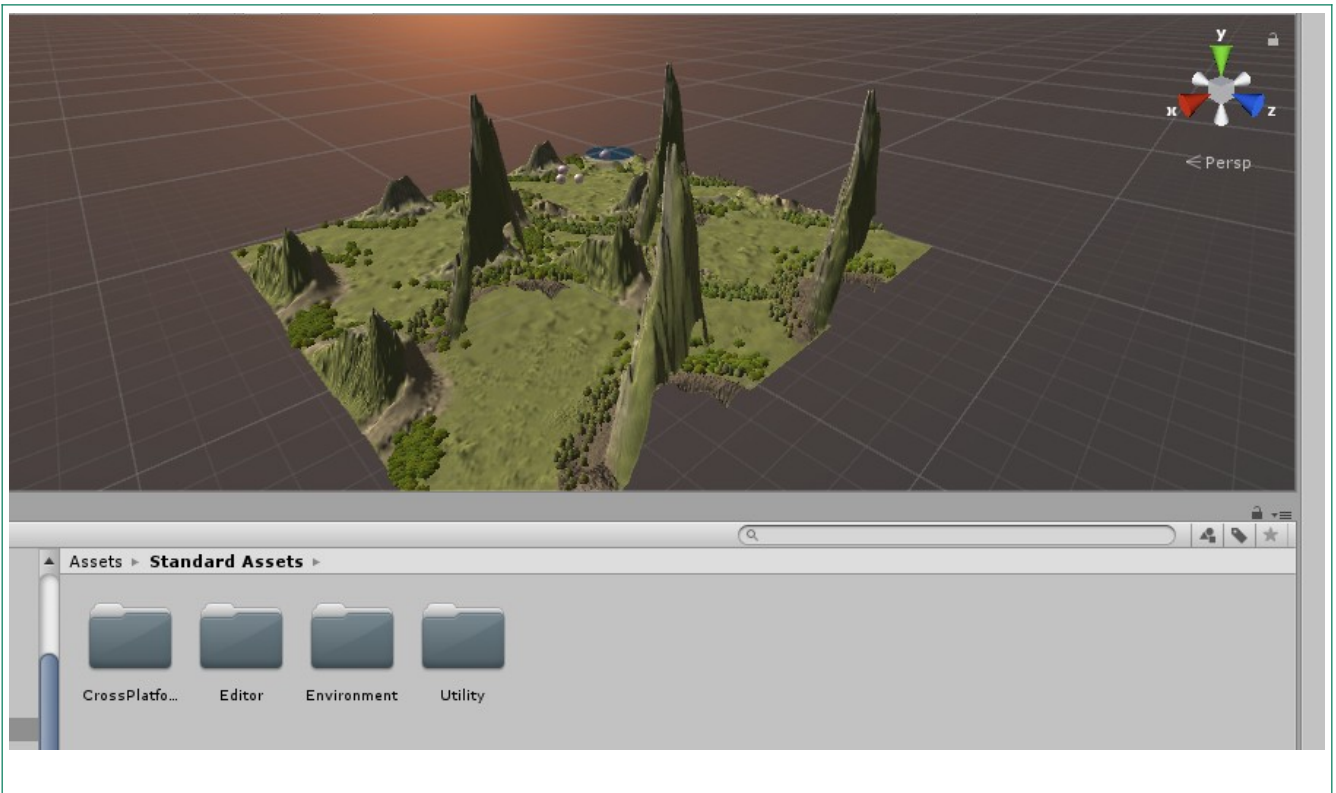
```
1  /* ROLL */
2  if (input2.x > 0 && input2.y < .5 && input2.y > -.5)
3  {
4      transform.RotateAround(transform.position, transform.right, Time.deltaTime * 30f);
5  }
6
7  if (input2.x < 0 && input2.y < .5 && input2.y > -.5)
8  {
9      transform.RotateAround(transform.position, -transform.right, Time.deltaTime * 30f);
10 }
```

## IMPLEMENTACIÓN DEL ENVIROMENT

### ENVIROMENT



### IMPLEMENTACIÓN EN UNITY



# CONCLUSIONES

Se pudo realizar una simulación a partir del modelo matemático. A pesar de ser muy simple debido a que no se tuvieron en cuenta perturbaciones exteriores, como el cambio de la densidad del aire a diferentes alturas, este modelo permite dar una idea global del funcionamiento del vehículo, y abre paso a trabajos futuros sobre la automatización del avión ala voladora.

Existen distintas formas de modelar una aeronave, ya sean grandes o pequeñas y varían dependiendo del diseño y configuración, una forma de comprobar la respuesta de un sistema no solo es modelando matemáticamente, sino también identificando la respuesta de un sistema físico y comparándolo con el sistema modelado. Al momento de comparar el modelado matemático con el modelo físico existen variaciones incluso entre el modelo para aeronaves grandes ya que incluye más parámetros, especialmente cuando hablamos de la altura o traslación vertical existe una variación muy grande entre lo que se modeló matemáticamente con la respuesta obtenida y el sistema identificado, esto porque estamos usando un modelo estático que al momento de desplazarse hacia arriba lo hace de forma absoluta sin considerar un pequeño movimiento hacia adelante. Por esta razón existirá una variación entre los modelos teóricos y los vistos físicamente, sin embargo, son una buena aproximación para lo que tendremos de forma tangible.

# ANEXOS

## Bibliografía

- Pangilinan, E., Lukas, S., & Mohan, V. (2019). *Creating Augmented and Virtual Realities: Theory and Practice for Next-Generation Spatial Computing*. O'Reilly Media.
- McDougall, S. D. J., Dawkins, S., & McDougall, J. (2022). *Understanding virtual reality: Challenging Perspectives for Media Literacy and Education*. Routledge.
- A, M. A. G., Vexo, F., & Thalmann, D. (2023). *Stepping into Virtual Reality*. Springer.
- Ross, S. M. (2012). *Simulation*. Academic Press.
- Law, A. M. (2023). *Simulation Modeling and Analysis, Sixth Edition*. McGraw-Hill Companies.