

**PARQUE DE DIVERSIONES 3D****INTEGRANTES:**

GIRALDO BARRERA ORLANDO STEVEN

GONZALEZ CAPACHO JUAN JOSE

PABON JAIMES MARLY

PRADA NIÑO LISBETH

**DOCENTE:**

ING. ANDRES ALFONSO MURGAS VILORIA

**GRUPO: E196**

**UNIDADES TECNOLOGICAS DE SANTANDER**

**REALIDAD AUMENTADA**

**BUCARAMANGA**

**MAYO DE 2024**

**TABLA DE CONTENIDO**

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>5</b>
<b>2.</b>	<b>JUSTIFICACIÓN .....</b>	<b>6</b>
<b>3.</b>	<b>MARCO REFERENCIAL.....</b>	<b>7</b>
<b>3.1</b>	<b>MARCO CONCEPTUAL.....</b>	<b>7</b>
<b>3.2</b>	<b>MARCO TEORICO.....</b>	<b>9</b>
<b>4.</b>	<b>DESARROLLO.....</b>	<b>10</b>
<b>4.1</b>	<b>PLANIFICACIÓN Y DISEÑO CONCEPTUAL: .....</b>	<b>10</b>
<b>4.2</b>	<b>MODELADO 3D: .....</b>	<b>10</b>
<b>4.3</b>	<b>IMPORTACIÓN DE MODELOS A UNITY: .....</b>	<b>33</b>
<b>4.4</b>	<b>IMPLEMENTACIÓN DE ATRACCIONES.....</b>	<b>33</b>
<b>4.5</b>	<b>PROGRAMACIÓN .....</b>	<b>36</b>
<b>4.6</b>	<b>ANIMACIONES .....</b>	<b>38</b>
<b>4.7</b>	<b>IMPLEMENTACIÓN FINAL .....</b>	<b>43</b>
<b>5.</b>	<b>CONCLUSIÓN .....</b>	<b>44</b>
<b>6.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>45</b>

**TABLA DE ILUSTRACIONES**

<b>Ilustración 1.</b> Carrito chocón 1 .....	11
<b>Ilustración 2.</b> Carrito chocón 2 .....	11
<b>Ilustración 3.</b> Carrito chocón 2 .....	11
<b>Ilustración 4.</b> Personaje 1.....	12
<b>Ilustración 5.</b> Personaje 2.....	12
<b>Ilustración 6.</b> Personaje 3.....	12
<b>Ilustración 7.</b> Pista de carritos chocones.....	13
<b>Ilustración 8.</b> Conos.....	13
<b>Ilustración 9.</b> Cajas obstáculos .....	14
<b>Ilustración 10.</b> Semáforo. ....	14
<b>Ilustración 11.</b> Letrero.....	14
<b>Ilustración 12.</b> Flores.....	15
<b>Ilustración 13.</b> Arboles.....	15
<b>Ilustración 14.</b> Carpa del circo.....	16
<b>Ilustración 15.</b> Puma.....	16
<b>Ilustración 16.</b> Bote de basura. ....	17
<b>Ilustración 17.</b> Banca.....	17
<b>Ilustración 18.</b> Mesa 1 .....	17
<b>Ilustración 19.</b> Arboles.....	18
<b>Ilustración 20.</b> Carrito de comida 1 .....	18
<b>Ilustración 21.</b> Carrito de Comida 2 .....	18
<b>Ilustración 22.</b> Mesa 2 .....	19
<b>Ilustración 23</b> Globo. ....	19
<b>Ilustración 24.</b> Barco Pirata .....	20
<b>Ilustración 25.</b> Dragon Guardian.....	20
<b>Ilustración 26.</b> Pirata 2.....	21
<b>Ilustración 27.</b> Pirata 1.....	21
<b>Ilustración 28.</b> Pirata 3.....	21
<b>Ilustración 29.</b> Banca.....	22
<b>Ilustración 30.</b> Bandera. ....	22
<b>Ilustración 31.</b> Lampara.....	22
<b>Ilustración 32.</b> Cañón. ....	23
<b>Ilustración 33.</b> Carrito de comida. ....	23
<b>Ilustración 34</b> Arboles.....	24
<b>Ilustración 35.</b> Rocas.....	24
<b>Ilustración 36.</b> Flores.....	25
<b>Ilustración 37</b> Perro. ....	27
<b>Ilustración 38.</b> Señalización 2 .....	27
<b>Ilustración 39.</b> Señalización 1 .....	27
<b>Ilustración 40.</b> Arboles. ....	28
<b>Ilustración 41.</b> Casa del Perro.....	28
<b>Ilustración 42.</b> Globo aerostático.....	29
<b>Ilustración 43.</b> Flores .....	29
<b>Ilustración 44.</b> Carrito de comida 4 .....	30
<b>Ilustración 45.</b> Entrada. ....	30
<b>Ilustración 46.</b> Terreno.....	31

<b>Ilustración 47.</b> Personaje Secundario 1 .....	31
<b>Ilustración 48.</b> Personaje Secundario 3. ....	32
<b>Ilustración 49.</b> Personaje Secundario 2. ....	32
<b>Ilustración 50.</b> Importación de Modelos a Unity. ....	33
<b>Ilustración 51.</b> Atracción de Barco Pirata. ....	34
<b>Ilustración 52.</b> Atracción de Carritos Chocones. ....	34
<b>Ilustración 53.</b> Atracción de Circo. ....	35
<b>Ilustración 54.</b> Rueda de la Fortuna. ....	35
<b>Ilustración 55.</b> Control de interacciones.....	36
<b>Ilustración 56.</b> Control del jugador. ....	37
<b>Ilustración 57.</b> Animación: Personaje secundario 1.....	38
<b>Ilustración 58.</b> Animación: Dragón Guardián. ....	39
<b>Ilustración 59.</b> Animación: Pirata 2.....	39
<b>Ilustración 60.</b> Animación: Pirata 3.....	40
<b>Ilustración 61.</b> Animación: Personaje secundario 3.....	40
<b>Ilustración 62.</b> Timeline Circo .....	41
<b>Ilustración 63.</b> Timeline Carritos chocones .....	41
<b>Ilustración 64.</b> Timeline Inicial .....	42
<b>Ilustración 65.</b> Timeline Barco Pirata. ....	42
<b>Ilustración 66.</b> Timeline La Rueda de la Fortuna. ....	43

## 1. INTRODUCCIÓN

El proyecto tiene como objetivo la creación de un parque de diversiones 3D utilizando el motor de desarrollo Unity. La iniciativa busca brindar a los usuarios una experiencia inmersiva y atractiva que les permita disfrutar de un recorrido interactivo por un mundo 3D, compuesto por cuatro atracciones: la rueda de la fortuna, el barco pirata, los carritos chocones y un circo.

En el presente documento se plasman las siete (7) etapas que conforman el desarrollo del proyecto de manera que se expone en primer lugar su planificación y diseño conceptual, en segundo lugar el respectivo modelado 3D para cada una de las atracciones y demás assets decorativos, en tercer lugar su importación al motor de desarrollo Unity para que posteriormente en el cuarto lugar se realice su respectiva ubicación y distribución en el parque de diversiones, se continua en quinto lugar con la programación del movimiento e interacciones que tendrá el usuario con las 4 atracciones del parque de diversiones, en sexto lugar se continua creando cada una de las animaciones que tendrán los objetos y las cámaras, para finalizar en su séptima etapa se realiza su exportación a la plataforma de Windows mediante la creación de un paquete que contendrá el ejecutable final.

Cabe resaltar que la implementación de este tipo de creaciones 3D en cuanto a costos se adapta muy bien dependiendo de los recursos disponibles, por lo que se pueden lograr proyectos muy bien desarrollados con poca inversión económica.

Para el desarrollo de este proyecto de creación de parque de diversiones 3D se utilizaron herramientas como el motor de desarrollo Unity, el software de modelado 3D Blender, la aplicación web de animación Mixamo, y repositorios en línea para los terrenos y texturas utilizados para la ambientación y animación del parque de diversiones 3D. Adicionalmente se utilizó la metodología de desarrollo Scrum realizando reuniones con el fin de visualizar el progreso del proyecto.

## 2. JUSTIFICACIÓN

Herramientas de desarrollo como Unity en la actualidad ofrecen la posibilidad de crear espacios 3D con experiencias atractivas e interactivas para el entretenimiento, por este motivo, surge la posibilidad de desarrollar proyectos innovadores que aprovechen el potencial de la tecnología para brindar experiencias de entretenimiento únicas.

Este proyecto propone utilizar el motor de desarrollo Unity para la creación de un parque de diversiones 3D, el cual contará con cuatro atracciones como lo son: la rueda de la fortuna, el barco pirata, los carritos chocones y un circo. Cada atracción estará diseñada con gráficos 3D y funcionalidades interactivas que permitirán a los usuarios explorarlas y experimentarlas de forma entretenida. La viabilidad de este proyecto es alta ya que los costos de desarrollo no son elevados a comparación con algunas otras obras de entretenimiento, lo que a la larga brinda una mayor rentabilidad si se decide comercializarlo.

La creación de un parque de diversiones 3D en Unity representa una oportunidad innovadora para satisfacer las necesidades de entretenimiento de una audiencia global, ya que el proyecto ofrece una experiencia atractiva y accesible, con un alto potencial de impacto y viabilidad comercial.

### 3. MARCO REFERENCIAL

#### 3.1 MARCO CONCEPTUAL

Para el desarrollo de este proyecto, se utilizaron las siguientes herramientas y recursos:

**3.1.1 UNITY:** Según (Unity Technologies, 2024), es un motor de juegos y plataforma de desarrollo multiplataforma que permite a los creadores diseñar, crear y publicar experiencias interactivas en 2D y 3D. Es utilizado por una amplia gama de desarrolladores, desde estudiantes y aficionados hasta estudios profesionales, para crear una gran variedad de aplicaciones, incluyendo:

- **Videojuegos:** Unity es uno de los motores de juegos más populares del mundo, y se ha utilizado para crear juegos de todos los géneros, desde juegos móviles casuales hasta juegos de consola AAA.
- **Simulaciones:** Unity se puede utilizar para crear simulaciones realistas y atractivas para una variedad de industrias, como la formación, la ingeniería y la atención médica.
- **Experiencias educativas:** Unity se puede utilizar para crear experiencias educativas interactivas que ayudan a los estudiantes a aprender de manera más efectiva.
- **Aplicaciones de realidad virtual y aumentada:** Unity es uno de los motores líderes para el desarrollo de aplicaciones de realidad virtual y aumentada.

**3.1.2 BLENDER 3D:** De acuerdo con (BLENDER, 2024), es un software de creación 3D gratuito y de código abierto que permite a los usuarios modelar, animar, renderizar y componer gráficos 3D. Es una herramienta versátil que se puede utilizar para una amplia gama de proyectos, incluyendo.

- **Creación de modelos 3D:** Blender ofrece una amplia gama de herramientas para modelar objetos 3D, desde simples formas geométricas hasta personajes y entornos complejos.

- **Animación:** Blender te permite animar tus modelos 3D para crear animaciones realistas y atractivas.
- **Renderizado:** Blender ofrece un potente motor de renderizado que te permite crear imágenes y vídeos de alta calidad de tus modelos 3D.
- **Composición:** Blender te permite componer tus imágenes y vídeos 3D con efectos especiales, como iluminación, sombras y texturas.

**3.1.3 METODOLOGIA SCRUM:** Es una metodología ágil para la gestión de proyectos, particularmente indicada para proyectos de desarrollo de software, que se caracteriza por su enfoque iterativo e incremental, basado en ciclos cortos de trabajo denominados sprints. (Trigos Gallego, 2012).

**3.1.4 PROGRAMACIÓN ORIENTADA A OBJETOS:** Conocida como POO por sus siglas, la programación orientada a objetos es un paradigma de programación el cual muchos de los lenguajes que actualmente están en auge adoptaron debido a que se basa en unificar los atributos y métodos que se ejecutan en los datos de un programa, al realizar dicha unificación este conjunto de información se define como objeto, en el cual los atributos son las características del objeto y los métodos son las acciones que este puede realizar. (Martínez Canelo, 2020).

**3.1.4 C#:** Es un lenguaje de programación orientado a objetos, multiparadigma y de propósito general desarrollado por Microsoft como parte de la plataforma .NET que se caracteriza por su sintaxis similar a C++ y Java, pero con un enfoque más moderno y seguro. (Microsoft, 2024)

### 3.2 MARCO TEORICO

**3.2.1 ANIMACION 3D:** Es la creación de imágenes en movimiento mediante el uso de modelos 3D, además estos modelos pueden ser objetos, personajes, escenarios o cualquier otra cosa que se pueda en este contexto la animación 3D se utiliza en una amplia gama de aplicaciones, incluyendo películas, videojuegos, televisión, realidad virtual y realidad aumentada. A diferencia de la animación tradicional, que se basa en dibujos o fotografías bidimensionales que se reproducen en rápida sucesión para crear la sensación de movimiento, la animación 3D trabaja con modelos digitales tridimensionales a los que se les da movimiento y vida a través de técnicas computacionales.

**3.2.2 MODELADO 3D:** Es el proceso de crear modelos 3D digitales los cuales pueden ser objetos, personajes, escenarios o cualquier otra cosa que se pueda imaginar y se utiliza en una amplia gama de aplicaciones, incluyendo animación, videojuegos, diseño de productos y arquitectura que diferencia del modelado tradicional, que se basa en materiales físicos como la arcilla o la madera, el modelado 3D utiliza herramientas computacionales para dar forma a objetos virtuales.

**3.2.3 TEXTURIZADO:** Son imágenes que se aplican a los modelos 3D para aportar detalles visuales y realismo, estas imágenes pueden contener patrones, manchas, desgaste y otros elementos que dan vida a las superficies, también es importante recalcar que el motor de desarrollo Unity admite diversos formatos de textura, incluyendo PNG, JPEG y TIFF.

#### 4. DESARROLLO

Para el desarrollo del presente proyecto se llevaron a cabo 8 etapas, como se describe a continuación:

##### 4.1 PLANIFICACIÓN Y DISEÑO CONCEPTUAL:

Se inició con la definición del concepto general del parque de diversiones, incluyendo las atracciones que se ofrecerían, la distribución del espacio y la temática del entorno. Esto a través de una lluvia día de ideas entre el equipo de trabajo sobre aquellos elementos necesarios que conformarían el parque de diversiones; De esta lluvia de idea dio como resultado un listado de cada uno de los modelos 3D necesarios para el proyecto, entre los cuales se pueden encontrar:

- Rueda de la fortuna
- Carritos chocones
- Barco pirata
- Circo
- Piratas
- Arboles
- Letreros
- Señalización
- Carritos de comida
- Sendero
- Entrada
- Skybox del cielo
- Pista de carritos chocones
- Globos
- Gente
- Bancas
- Lámparas
- Cerca
- Casita de perro
- Perro
- Mesas

##### 4.2 MODELADO 3D:

Utilizando la herramienta de modelado 3D Blender se crearon los modelos 3D de las atracciones, elementos decorativos y el entorno del parque de diversiones.

A continuación, se mostrará cada uno de los modelos 3D:

#### 4.2.1 Atracción: Carritos chocones

- **Carritos chocones:** Para estos se diseñaron tres (3) colores diferentes:

**Ilustración 1.** Carrito chocón 1.



Fuente: Autores del proyecto

**Ilustración 2.** Carrito chocón 2.



Fuente: Autores del proyecto

**Ilustración 3.** Carrito chocón 2.



Fuente: Autores del proyecto

- **Personajes de carritos chocones:** Para cada uno de los carritos chocones se diseñó un personaje diferente como se muestra en las siguientes ilustraciones:

**Ilustración 4.** Personaje 1.



Fuente: Autores del proyecto

**Ilustración 5.** Personaje 2.



Fuente: Autores del proyecto

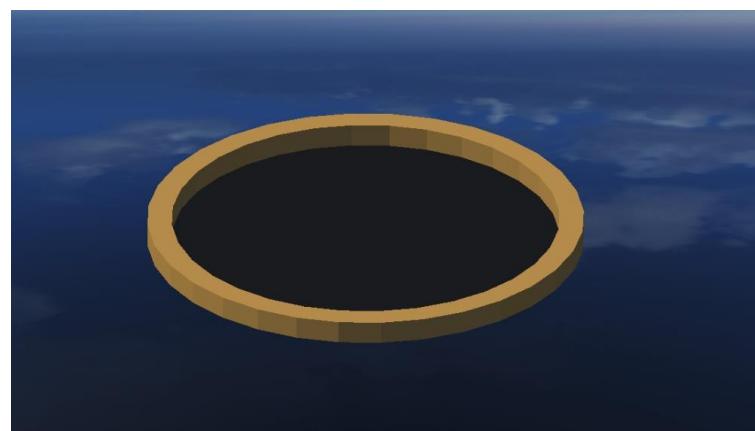
**Ilustración 6.** Personaje 3.



Fuente: Autores del proyecto.

- **Pista de carrito chocones:** Los carritos chocones y sus respectivos personajes están ubicados en la siguiente pista.

**Ilustración 7.** Pista de carritos chocones.



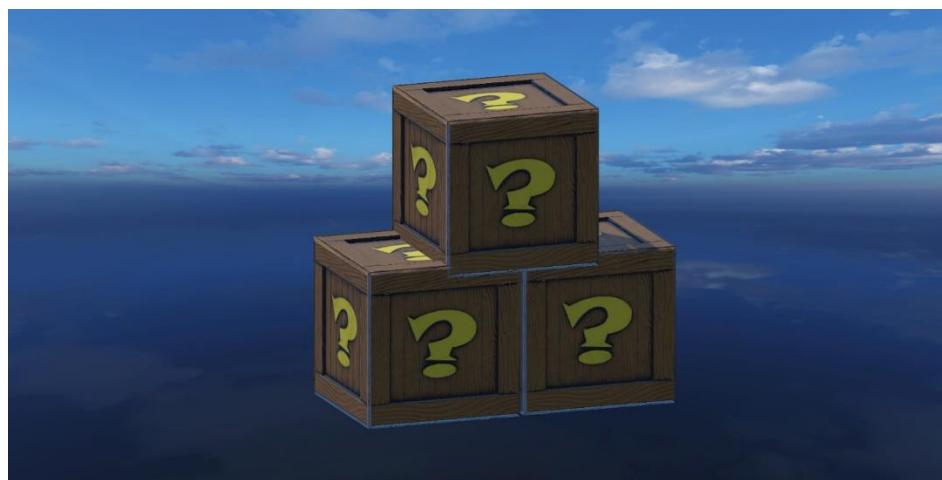
Fuente: Autores del proyecto

- **Decoraciones de la atracción y señalizaciones:** Se utilizaron elementos como conos, cajas obstáculo, semáforo, letreros, arboles y flores para la ambientación de la atracción carritos chocones.

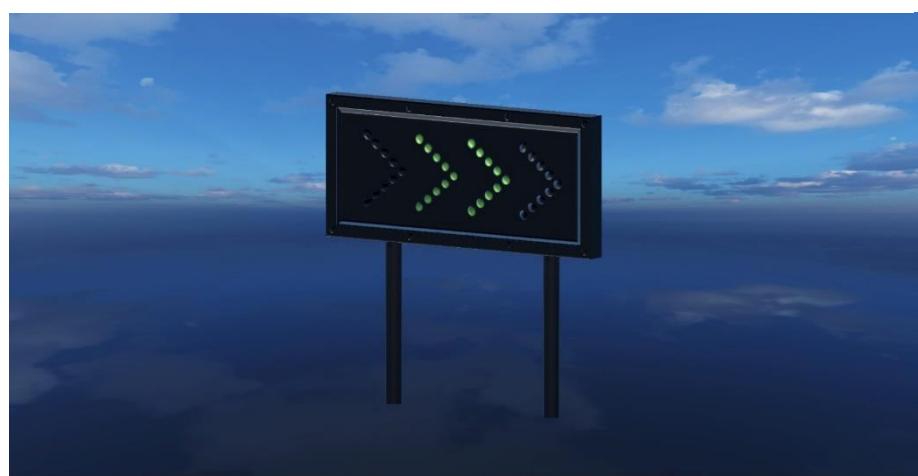
**Ilustración 8.** Conos.



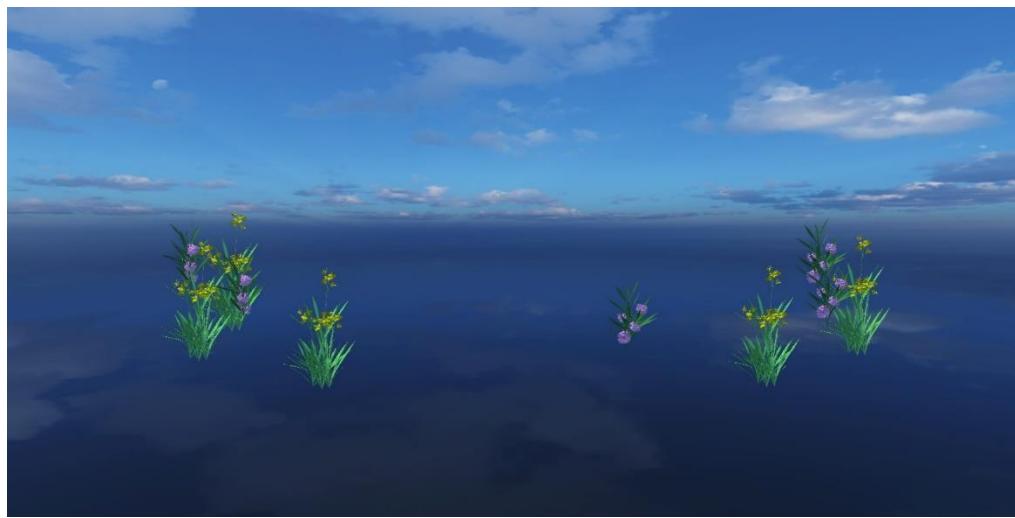
Fuente: Autores del proyecto.

**Ilustración 9.** Cajas obstáculos.

Fuente: Autores del proyecto.

**Ilustración 10.** Semáforo.**Ilustración 11.** Letrero.

Fuente: Autores del proyecto

**Ilustración 12.** Flores.

Fuente: Autores del proyecto

**Ilustración 13.** Arboles.

Fuente: Autores del proyecto

#### 4.2.2 Atracción: El circo

- **Carpa:** Para la atracción del circo se diseñó una tienda que se representa de la siguiente manera.

**Ilustración 14.** Carpa del circo.



Fuente: Autores del proyecto

- **Puma:** Teniendo en cuenta la atracción se optó por diseñar un puma para recibir al usuario a la entrada de la tienda del circo.

**Ilustración 15.** Puma.



Fuente: Autores del proyecto

- **Decoraciones de la atracción:** Se utilizaron elementos como carritos de comida, mesas, arboles, botes de basura, bancas y globos para la ambientación de la atracción.

**Ilustración 16.** Bote de basura.



Fuente: Autores del proyecto

**Ilustración 17.** Banca.



Fuente: Autores del proyecto

**Ilustración 18.** Mesa 1.



Fuente: Autores el proyecto.

**Ilustración 19.** Arboles.

Fuente: Autores el proyecto.

**Ilustración 20.** Carrito de comida 1.

Fuente: Autores el proyecto.

**Ilustración 21.** Carrito de Comida 2.

Fuente: Autores el proyecto.

**Ilustración 22.** Mesa 2.

Fuente: Autores el proyecto

**Ilustración 23** Globo.

Fuente: Autores el proyecto

**4.2.3 Atracción: El barco pirata.**

- **Barco Pirata:** Como objeto principal tenemos el barco pirata.

**Ilustración 24.** Barco Pirata.

Fuente: Autores el proyecto

- **Dragón guardián:** Encima del barco pirata se posiciona el siguiente dragón.

**Ilustración 25.** Dragon Guardian.

Fuente: Autores el proyecto

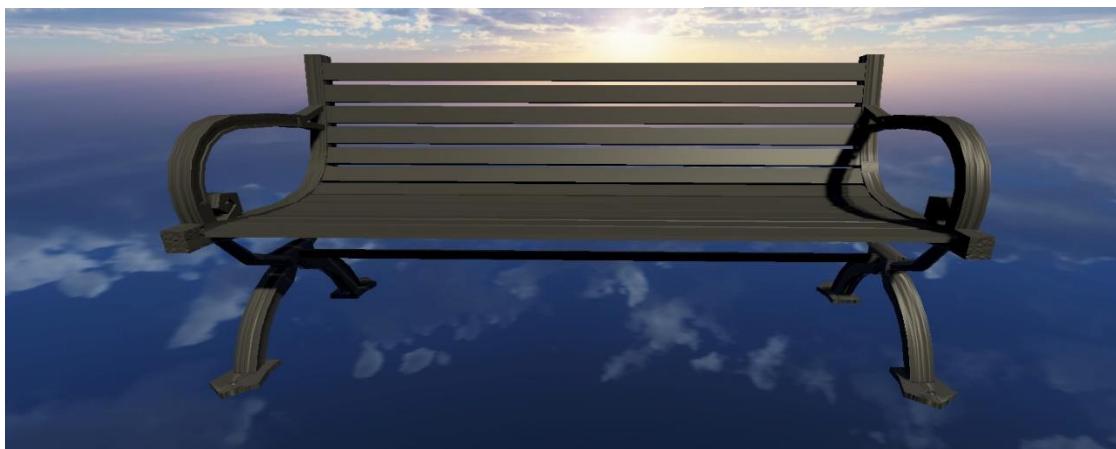
- **Piratas:** Para el recibimiento de la atracción se encuentran los dos personajes a continuación.

**Ilustración 27.** Pirata 1.**Ilustración 26.** Pirata 2.

Fuente: Autores el proyecto

**Ilustración 28.** Pirata 3.

- **Decoraciones de la atracción:** Se utilizaron elementos como carritos de comida, mesas, arboles, botes de basura, bancas, flores, banderas, lámparas y rocas para la ambientación de la atracción.

**Ilustración 29.** Banca.

Fuente: Autores el proyecto

**Ilustración 31.** Lámpara.**Ilustración 30.** Bandera.

Fuente: Autores el proyecto

**Ilustración 32.** Cañón.

Fuente: Autores el proyecto

**Ilustración 33.** Carrito de comida.

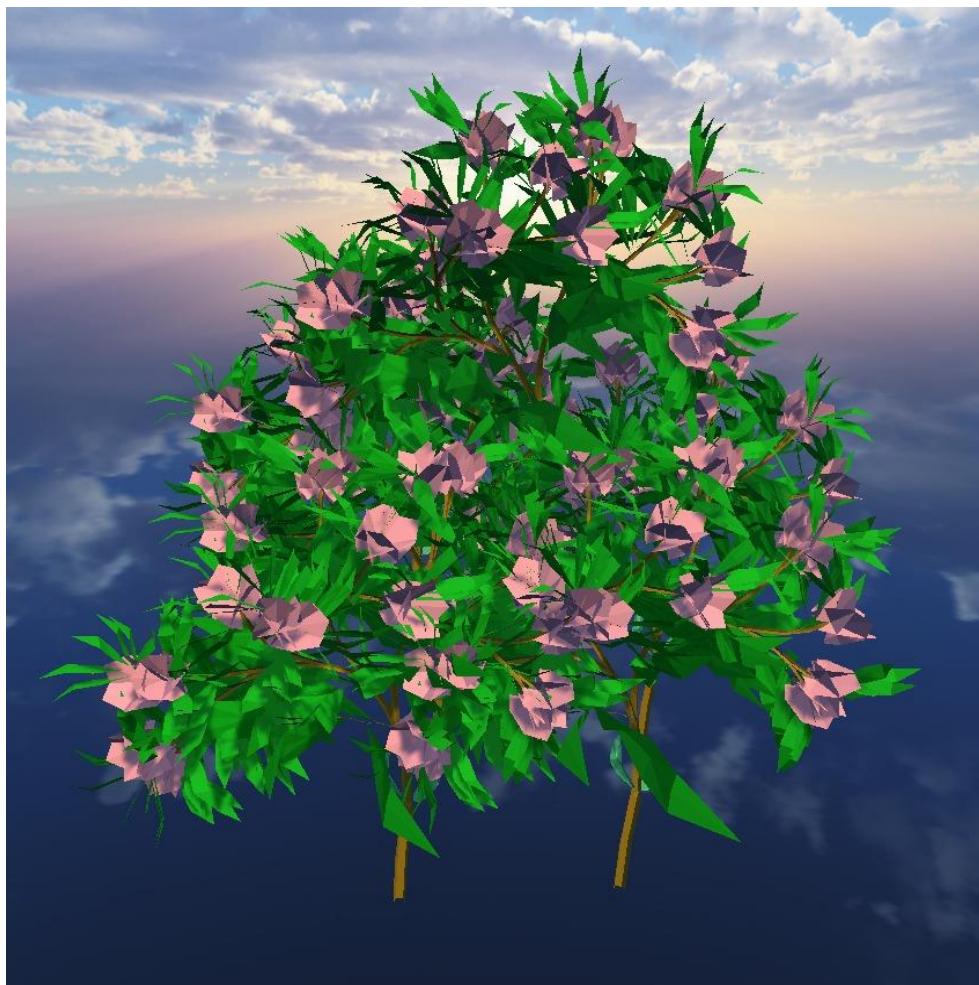
Fuente: Autores el proyecto

**Ilustración 35.** Rocas.

Fuente: Autores el proyecto

**Ilustración 34** Arboles.

Fuente: Autores el proyecto

**Ilustración 36.** Flores.

Fuente: Autores el proyecto

**4.2.4 Atracción: Rueda de la fortuna.**

- **Rueda de la fortuna:** Atracción enfocada en las vistas que tendrá el usuario.



Fuente: Autores el proyecto

**Ilustración 39.** Señalización 1.**Ilustración 38.** Señalización 2.

Fuente: Autores el proyecto

**Ilustración 37 Perro.**

**Ilustración 41.** Casa del Perro.

Fuente: Autores el proyecto

**Ilustración 40.** Arboles.

Fuente: Autores el proyecto

**Ilustración 42.** Globo aerostático.

Fuente: Autores del proyecto.

**Ilustración 43.** Flores

Fuentes: Autores del proyecto.

**Ilustración 44.** Carrito de comida 4.

Fuentes: Autores del proyecto.

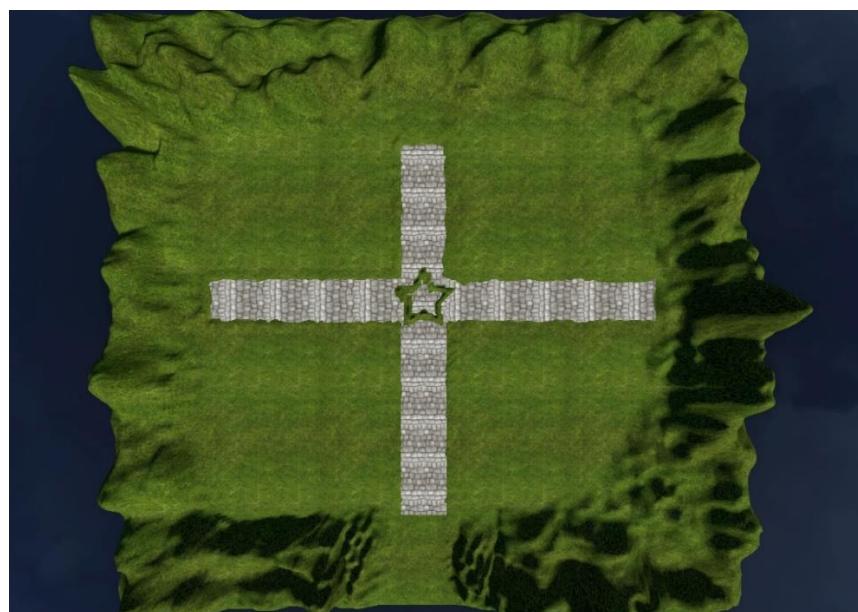
#### 4.2.5 Entrada:

**Ilustración 45.** Entrada.

Fuentes: Autores del proyecto

**4.2.6: Terreno:** Se caracteriza por la zona montañosa que delimita el parque y los senderos que dividen cada una de las atracciones.

**Ilustración 46.** Terreno.



Fuente: Autores del proyecto.

**4.2.7 Personajes secundarios:**

**Ilustración 47.** Personaje Secundario 1



Fuente: Autores del proyecto.

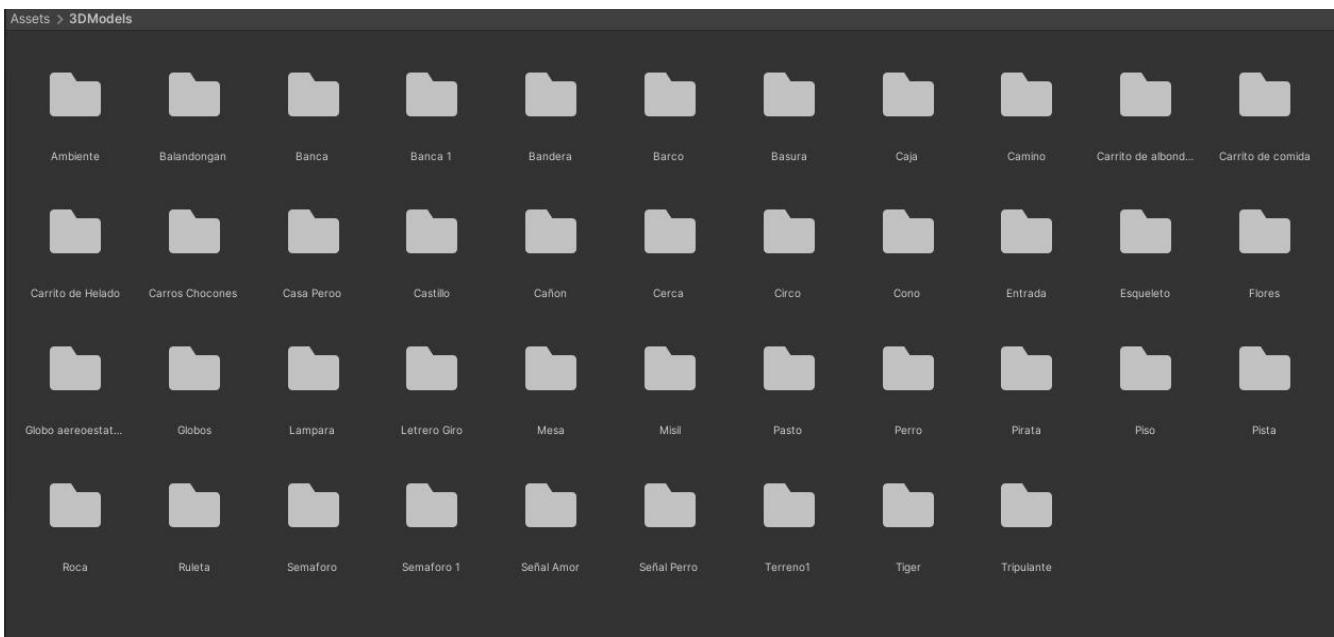
**Ilustración 49.** Personaje Secundario 2.**Ilustración 48.** Personaje Secundario 3.

Fuente: Autores del proyecto.

#### 4.3 IMPORTACIÓN DE MODELOS A UNITY:

Los modelos 3D creados en Blender se importaron al motor de desarrollo Unity. En la siguiente ilustración se mostrarán las importaciones realizadas:

**Ilustración 50.** Importación de Modelos a Unity.



Fuente: Autores del proyecto.

#### 4.4 IMPLEMENTACIÓN DE ATRACCIONES

Se ubicaron los modelos 3D de las atracciones en sus respectivas posiciones dentro del parque. De manera que el circo, el barco pirata, los carritos chocones y la rueda de la fortuna quedaron como se muestra en las siguientes ilustraciones:

**Ilustración 52.**Atracción de Carritos Chocones.

Fuente: Autores del proyecto.

**Ilustración 51.**Atracción de Barco Pirata.

Fuente: Autores del proyecto.

**Ilustración 53.** Atracción de Circo.

Fuente: Autores del proyecto.

**Ilustración 54.** Rueda de la Fortuna.

Fuente: Autores del proyecto.

## 4.5 PROGRAMACIÓN

Se utilizó el lenguaje de programación C# junto con el entorno de desarrollo Visual Studio para crear toda la lógica detrás de las acciones del jugador de tal manera que en las siguientes ilustraciones se mostrará cada uno de los scripts responsables del movimiento e interacciones de los objetos:

- **Control de interacciones:** Mediante el sistema de colisiones de Unity se detecta cuando el jugador está cerca de una atracción, para que en este momento se permita la interacción por medio de la captura del input de la tecla espaciadora, la cual al ser presionada activará la timeline pasada por referencia, la cual contendrá todas las animaciones de cada una de las atracciones.

**Ilustración 55.** Control de interacciones.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InteractionController : MonoBehaviour
{
    [SerializeField] GameObject _interactUI;
    [SerializeField] GameObject _timeline;
    [SerializeField] bool _canInteract;

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Player"))
        {
            _interactUI.SetActive(true);
            _canInteract = true;
        }
    }

    private void OnTriggerExit(Collider other)
    {
        if (other.gameObject.CompareTag("Player"))
        {
            _interactUI.SetActive(false);
            _canInteract = false;
        }
    }

    private void Update()
    {
        if (_canInteract && Input.GetKeyDown(KeyCode.Space))
        {
            _canInteract = false;
            _interactUI.SetActive(false);
            _timeline?.SetActive(false);
            _timeline?.SetActive(true);
        }
    }
}
```

Fuente: Autores del proyecto.

- **Control del jugador:** Mediante el uso del componente de Unity Character Controller y algunas variables de control como la velocidad y sensibilidad, se capturan los ejes horizontales y verticales, junto con el movimiento horizontal del mouse y las teclas Q y E, para aplicar el movimiento del jugador y la rotación de la cámara.

**Ilustración 56.** Control del jugador.

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
    [Header("Movement")]
    [SerializeField] public float _speed = 5.0f;
    [SerializeField] public float _sensitivity = 2.0f;
    [SerializeField] private CharacterController _characterController;
    [SerializeField] private float _rotationX = 0;
    [SerializeField] bool _canMove = true;

    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }

    private void Update()
    {
        if (!_canMove) return;

        float moveForward = Input.GetAxis("Vertical") * _speed * Time.deltaTime;
        float moveSideways = Input.GetAxis("Horizontal") * _speed * Time.deltaTime;

        Vector3 movement = transform.forward * moveForward + transform.right * moveSideways;
        _characterController.Move(movement);

        // Rotación de la cámara
        var xAxis = Input.GetAxis("Mouse X");
        var yAxis = -Input.GetAxis("Mouse Y");
        float mouseX = xAxis * _sensitivity;
        float mouseY = yAxis * _sensitivity;

        _rotationX += mouseY;
        _rotationX = Mathf.Clamp(_rotationX, -90, 90);

        transform.Rotate(0, mouseX, 0);
        Camera.main.transform.localRotation = Quaternion.Euler(_rotationX, 0, 0);

        if (Input.GetKey(KeyCode.Q))
        {
            transform.Rotate(0, -_sensitivity, 0);
        }
        if (Input.GetKey(KeyCode.E))
        {
            transform.Rotate(0, _sensitivity, 0);
        }
    }
}

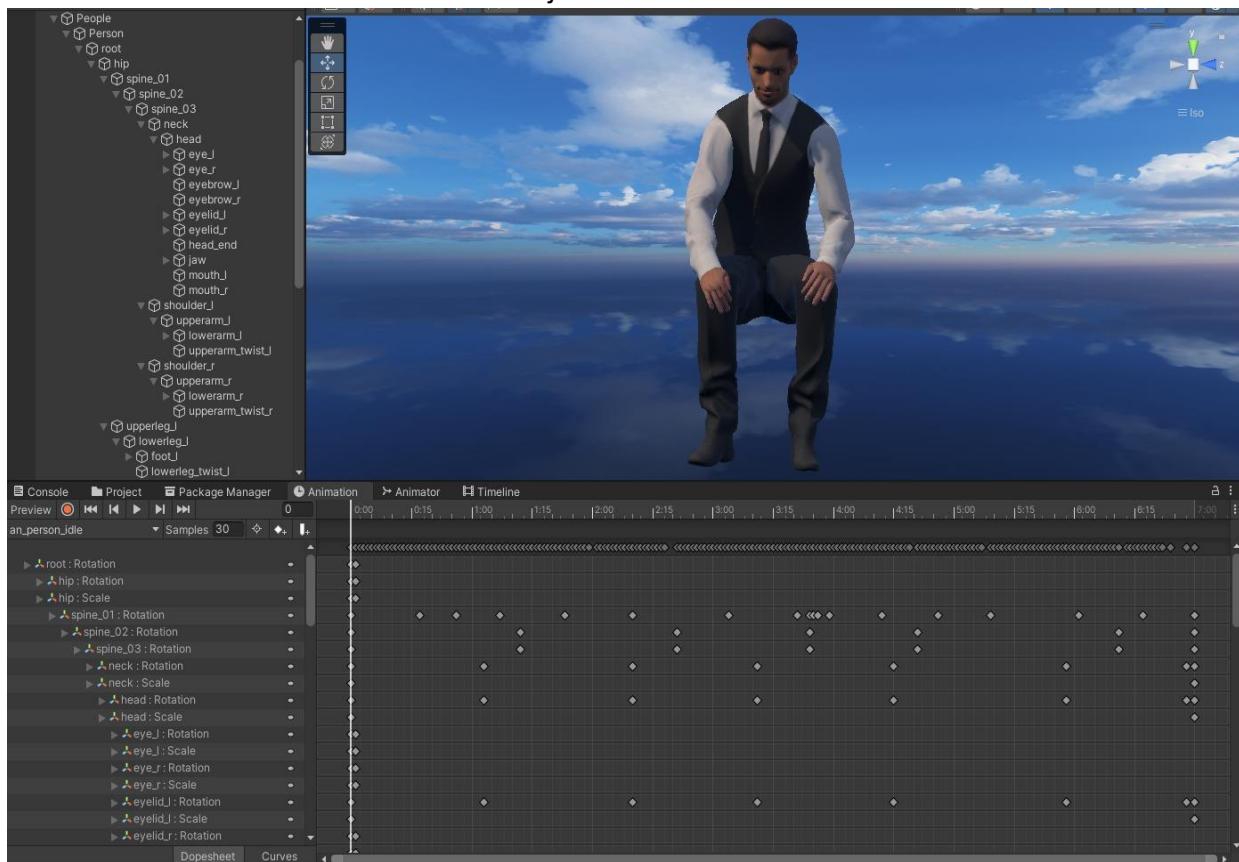
```

Fuente: Autores del proyecto.

## 4.6 ANIMACIONES

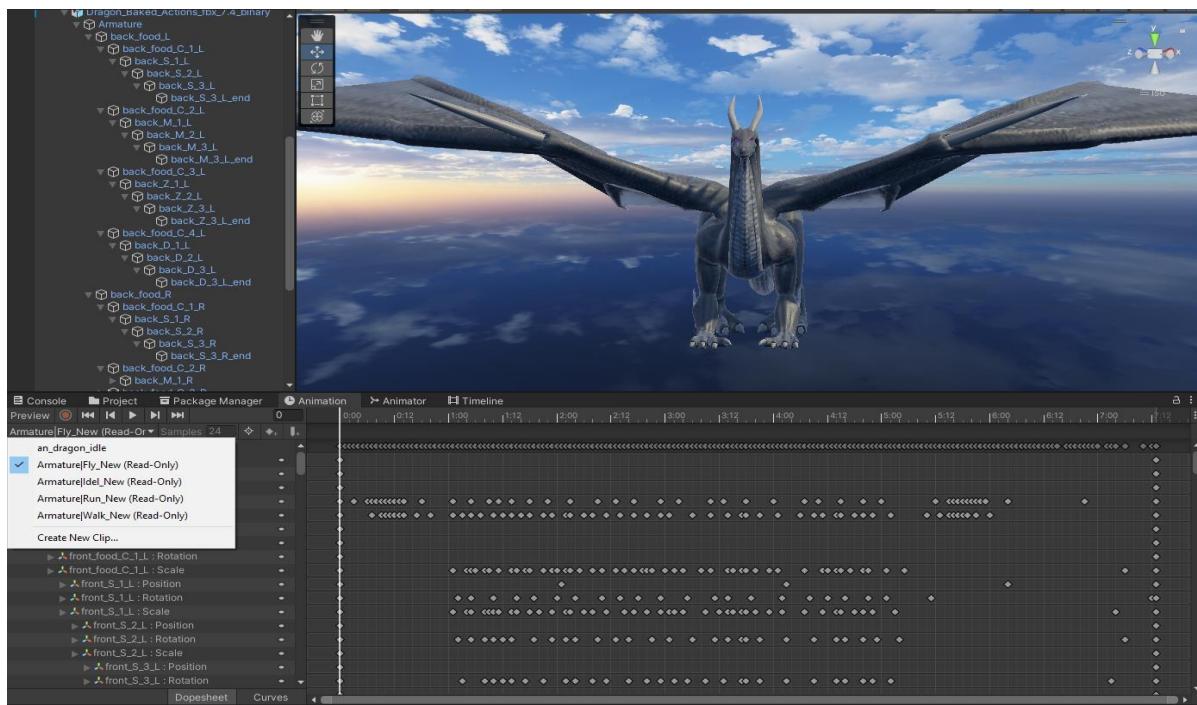
A continuación, se mostrarán algunas de las animaciones más relevantes que se desarrollaron para el proyecto, donde en su mayoría son animaciones realizadas dentro del motor Unity haciendo uso de las propiedades que ofrece el componente “transform”, y algunas realizadas desde la aplicación web Mixamo, la cual genera un rig automático para el modelo 3D, brindando una amplia biblioteca de animaciones.

**Ilustración 57.** Animación: Personaje secundario 1.



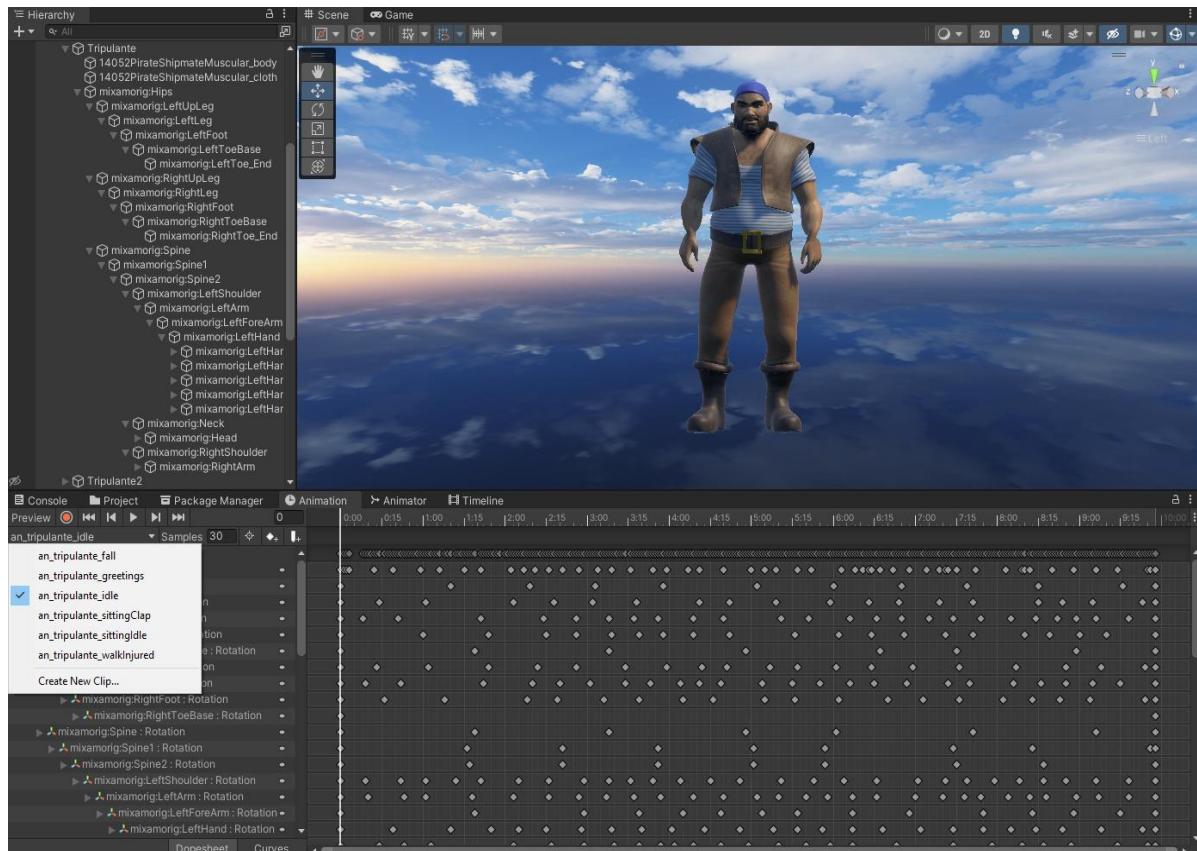
Fuente: Autores del proyecto.

**Ilustración 58.** Animación: Dragón Guardián.



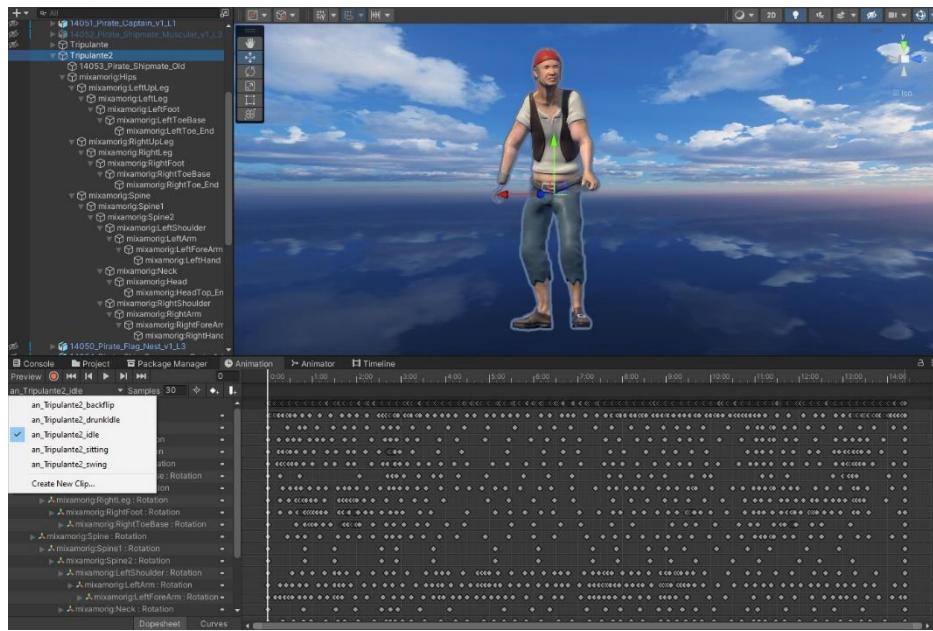
Fuente: Autores del proyecto.

**Ilustración 59.** Animación: Pirata 2.



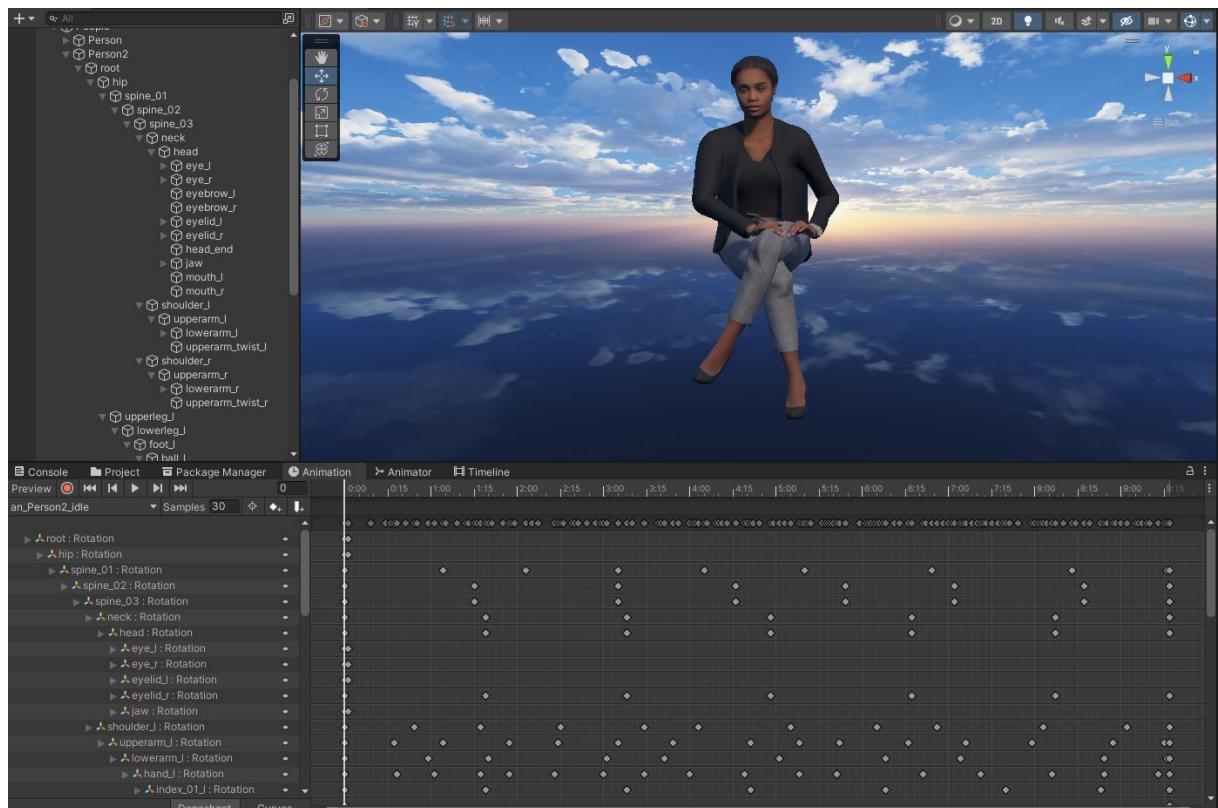
Fuente: Autores del proyecto.

**Ilustración 60.** Animación: Pirata 3.



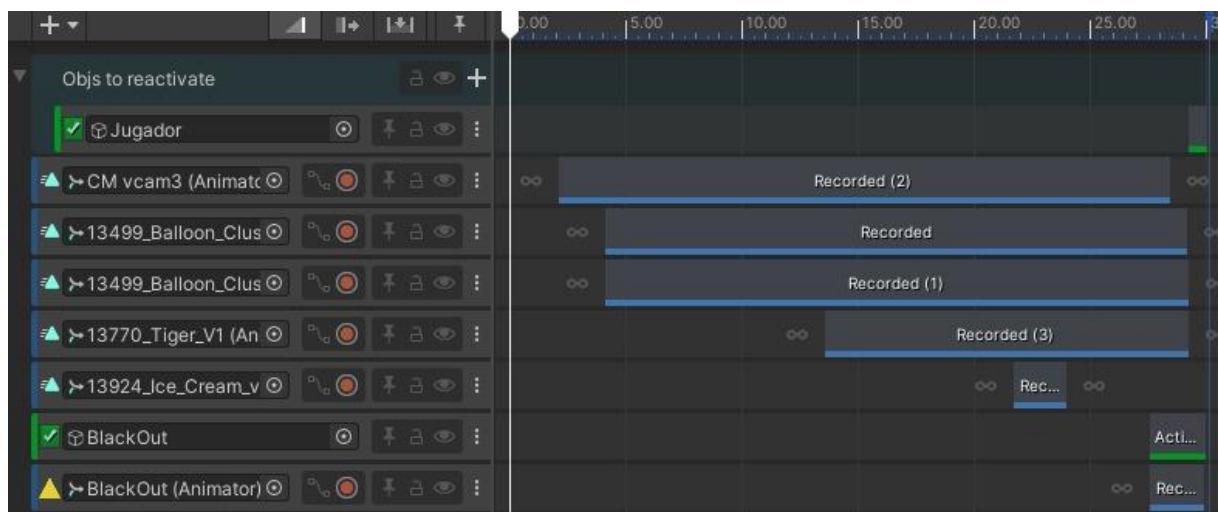
Fuente: Autores del proyecto.

**Ilustración 61.** Animación: Personaje secundario 3.



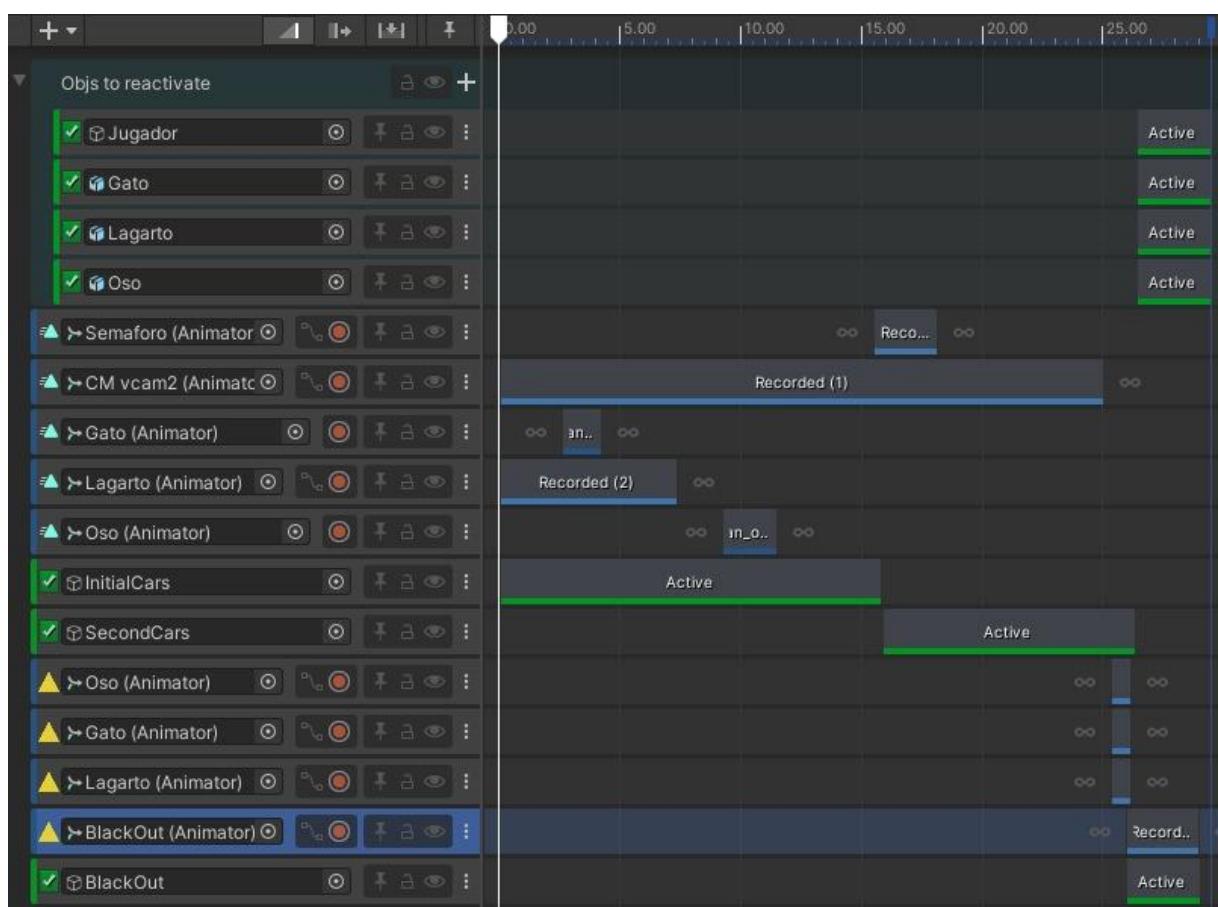
Fuente: Autores del proyecto.

**Ilustración 62.** Timeline Circo



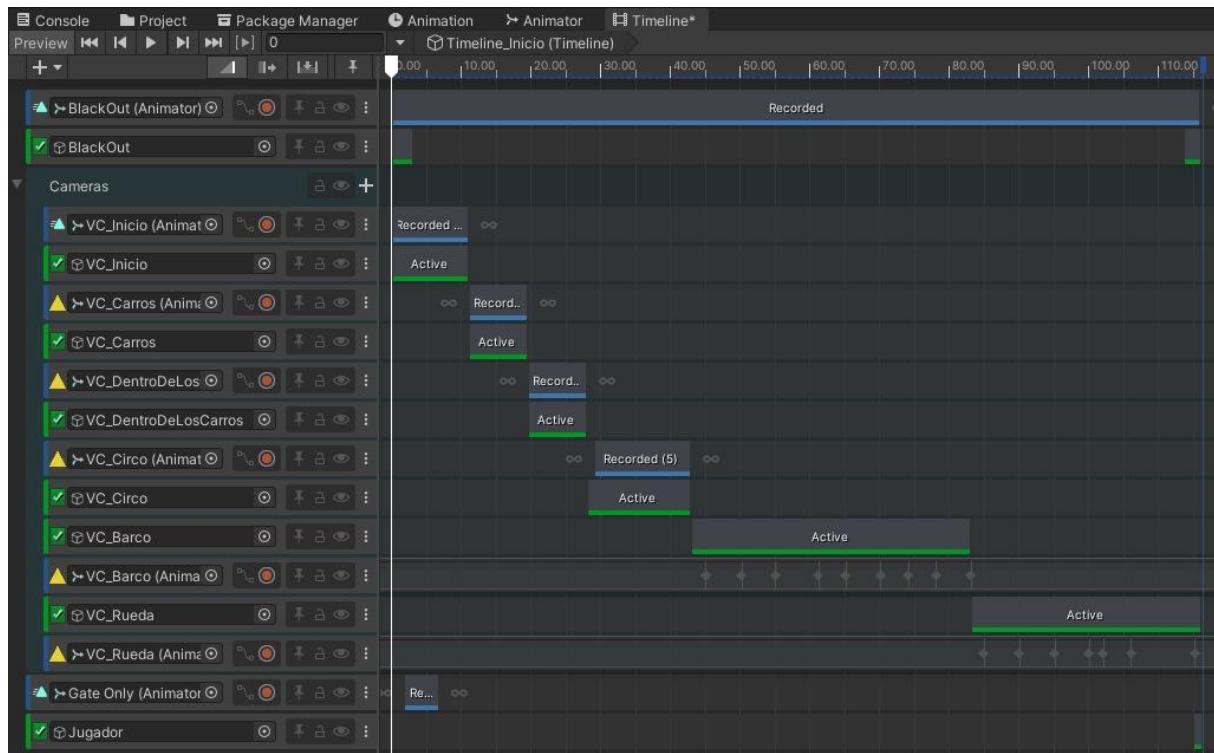
Fuente: Autores del proyecto.

**Ilustración 63.** Timeline Carritos chocones



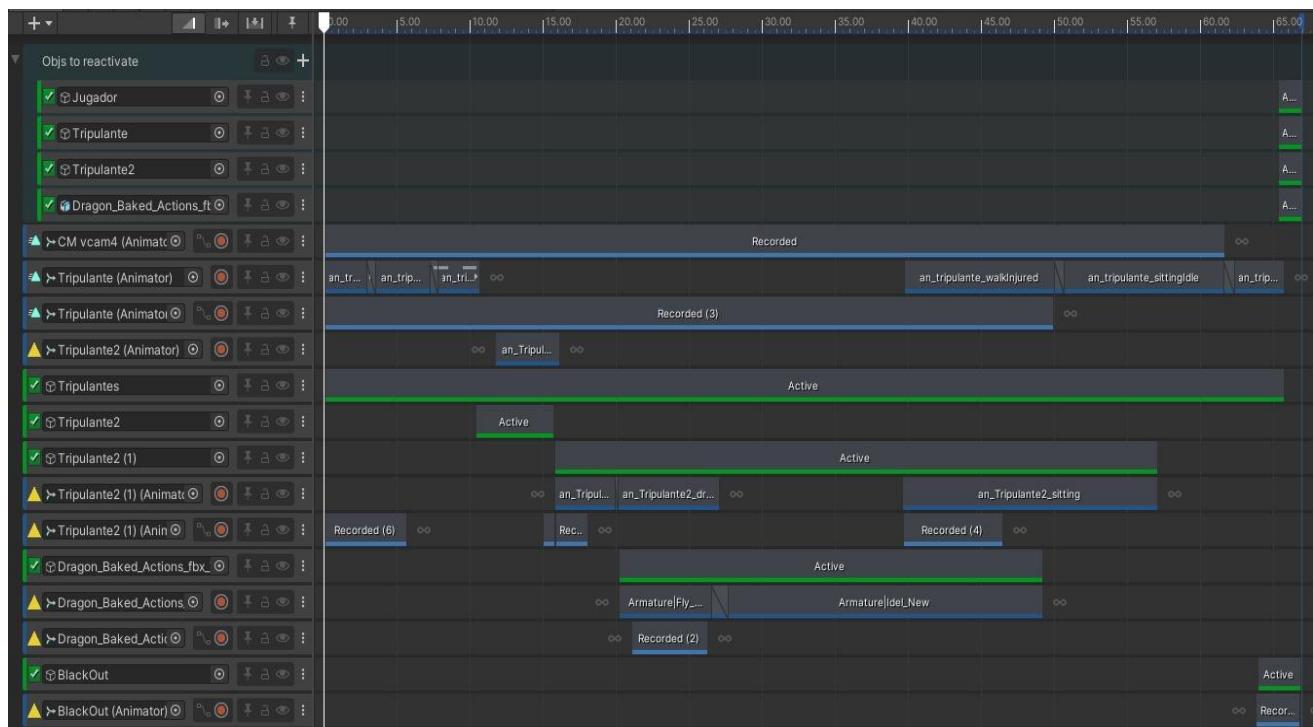
Fuente: Autores del proyecto.

**Ilustración 64.** Timeline Inicial



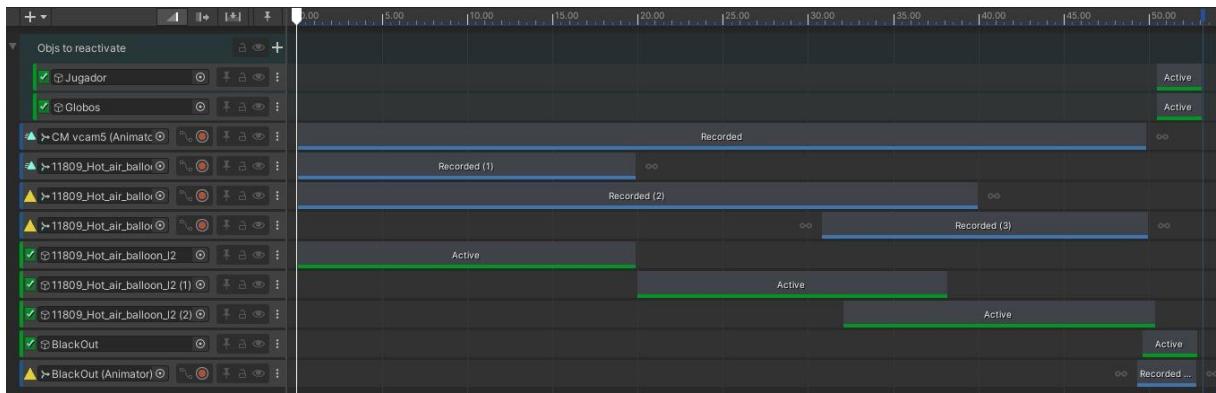
Fuente: Autores del proyecto.

**Ilustración 65.** Timeline Barco Pirata.



Fuente: Autores del proyecto.

**Ilustración 66.** Timeline La Rueda de la Fortuna.



Fuente: Autores del proyecto.

#### 4.7 IMPLEMENTACIÓN FINAL

Se exporta el proyecto desde Unity para la plataforma de Windows generando una compilación de todos los scripts del proyecto, traduciéndolos de C# a un lenguaje intermedio compatible con la plataforma de Windows, para luego, empaquetar los recursos del juego, como modelos, texturas, sonidos y otros assets, en un formato que el motor de Unity entiende y utiliza en la plataforma Windows. Durante este proceso, se aplican las optimizaciones establecidas en las configuraciones de cada asset, con el fin de reducir el tamaño del archivo final y mejorar el rendimiento. Por último se genera un ejecutable, junto con una carpeta de datos que contiene todos los recursos necesarios para la ejecución, donde se encuentran algunos archivos adicionales como bibliotecas y configuraciones específicas de la plataforma para asegurar que el juego funcione correctamente en el entorno. Este proceso se realiza de manera automática desde la pestaña "Build Settings" de Unity.

## 5. CONCLUSIÓN

El desarrollo del proyecto se llevó a cabo en siete etapas, desde la planificación y el diseño conceptual hasta la implementación final. Se prestó especial atención al modelado 3D de las atracciones, la construcción del terreno, la implementación de las funcionalidades interactivas y la ambientación del parque, con el fin de crear una experiencia amena y confortable para el usuario final.

A pesar de las limitaciones técnicas y de contenido, el proyecto de parque de diversiones 3D representa una propuesta innovadora, por lo que se recomienda continuar su desarrollo con la adición de nuevas atracciones, experiencias interactivas y la implementación de la realidad virtual para mejorar aún más la experiencia de los usuarios.

## 6. BIBLIOGRAFÍA

BLENDER. (2024). Obtenido de <https://www.blender.org>

Martínez Canelo, M. (2020). ¿Qué es la Programación Orientada a Objetos? Profile.  
<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

Microsoft. (2024). Dotnet. Obtenido de <https://dotnet.microsoft.com/es-es/languages/csharp>

Trigos Gallego, M. (12 de 06 de 2012). Openaccess. Obtenido de Openaccess:  
<https://openaccess.uoc.edu/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Unity Technologies. (2024). UNITY. Obtenido de <https://unity.com/support-services>