



**UNIVERSIDAD AMERICANA DE ACAPULCO**  
"EXCELENCIA PARA EL DESARROLLO"

---

**FACULTAD DE INGENIERÍA EN COMPUTACIÓN**

INCORPORADA A LA UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO  
CLAVE DE INCORPORACIÓN 8852-16

**"DESARROLLO DE ESCENARIOS VIRTUALES PARA  
SOFTWARE DE ROBÓTICA EDUCATIVA EN 3D"**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE  
**INGENIERA EN COMPUTACIÓN**

PRESENTA  
**GABRIELA DIEGO HERNÁNDEZ**

DIRECTOR DE TESIS  
**M.C. DAVID JAIME GONZÁLEZ MAXINEZ**



ACAPULCO, GUERRERO, FEBRERO 2017



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **AGRADECIMIENTOS**

A la Facultad de Ingeniería de la Universidad Americana de Acapulco que forma parte importante de este logro.

De igual manera agradezco a todos aquellos profesores que me motivaron a seguir adelante con este proyecto y en especial al director de mi tesis, al M.C. David Maxinez por la paciencia y apoyo que me brindo durante la realización de este trabajo.

## **DEDICATORIA**

Este trabajo está dedicado a:

Mis padres, por su paciencia, comprensión y apoyo en todo momento, ya que sin ellos, sin su ayuda y motivación, yo nunca hubiera llegado tan lejos.

A mi hija Danya Sofía, porque ella fue mi mayor motivación para salir adelante, ya que a pesar de todo, su sonrisa me daba el aliento necesario para no rendirme y luchar contra todo.

# ÍNDICE GENERAL

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I: PRESENTACIÓN</b> .....	3
1.1 Planteamiento del problema.....	3
1.2 Justificación.....	3
1.3 Hipótesis.....	4
1.4 Objetivo General .....	5
1.5 Objetivos Específicos .....	5
<b>CAPÍTULO II: ESCENARIOS VIRTUALES</b> .....	7
2.1 Introducción.....	7
2.2 Realidad virtual.....	8
2.3 Escenario virtual.....	9
2.4 Escenarios para robots.....	13
<b>CAPÍTULO III: CARACTERIZACIÓN DE ESCENARIOS</b> .....	16
3.1 Introducción.....	16
3.2 Tema .....	18
3.3 Objetivo .....	18
3.3.1 Tipo de robot a construir .....	19
3.3.2 Caracterización y utilería del escenario .....	21

3.4 Escenarios para robots.....	21
3.4.1 Robot Androide.....	21
3.4.2 Robot Zoomórfico .....	24
3.4.3 Robot Móvil.....	26
3.4.4 Robot Poliarticulado.....	28
3.4.5 Robot Híbrido.....	30

## **CAPÍTULO IV: DESARROLLO E**

<b>IMPLEMENTACIÓN .....</b>	<b>34</b>
4.1 Introducción.....	34
4.2 Plataformas de diseño.....	34
4.3 Construcción de escenarios con Unity 3D.....	40
4.4 Características de Unity 3D.....	41
4.5 Creación de nuevo proyecto.....	48
4.6 Construcción de un escenario .....	52
4.6.1 Terreno .....	53
4.6.2 Características del terreno.....	54
4.6.3 Textura del terreno.....	56
4.6.4 Relieve del terreno.....	61
4.6.4.1 Creación de Heightmap.....	61
4.6.5 Creación de océanos, lagos y ríos.....	72
4.6.6 Creación de vegetación del terreno .....	75
4.6.7 Creación del cielo del terreno .....	87

4.6.8 Añadiendo personaje al terreno .....	90
4.6.9 Ejecución del escenario .....	93
4.6.10 Compilación del proyecto.....	96
<b>CAPITULO V: RESULTADOS .....</b>	<b>101</b>
5.1 Descripción de la plataforma interactiva anterior.....	101
5.2 Descripción de la plataforma interactiva actual .....	106
<b>CAPÍTULO VI: CONCLUSIÓN Y TRABAJO A FUTURO .....</b>	<b>114</b>
<b>BIBLIOGRAFÍA .....</b>	<b>116</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>118</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>123</b>
<b>APÉNDICE A .....</b>	<b>125</b>
Convertor de Imagen por medio de XnConvert .....	125



# INTRODUCCIÓN

Este proyecto forma parte de un desarrollo global de software educativo, el fin de todo programa de esta índole es propiciar el aprendizaje de una manera virtual interactiva y fácil de comprender.

La robótica hoy en día es una rama de estudio que se está volviendo cada día más fuerte por los diversos avances tecnológicos. La enseñanza en esta área se implementa mediante el desarrollo de escenarios virtuales en 3D, siendo de bajo costo y fácil de entender, tanto el alumno como el profesor.

En general la mayoría de los programas educativos están diseñados en 2D “dos dimensiones”, es decir, los objetos incluidos en los escenarios se encuentran con desplazamiento de un punto A, a un punto B, las figuras, imágenes o simplemente el texto que se muestran están de forma plana y lo que se pretende con este proyecto es brindar un cambio a la visualización de la multimedia que se manejará en el mismo, esto es tercera dimensión 3D, logrando con ello que el alumno aprenda de manera divertida y la enseñanza se lleve a cabo más sencilla ya que su forma de interacción será más abierta.

**CAPÍTULO I**

**PRESENTACIÓN**

# **CAPÍTULO I: PRESENTACIÓN**

## **1.1 Planteamiento del problema**

Este proyecto se basa en el desarrollo de una plataforma virtual que busca enseñar la función de cada uno de los diversos robots utilizados en la actualidad en sus diferentes campos de aplicación de manera interactiva para el usuario, acortando la brecha económica entre la enseñanza de la robótica con elementos físicos de alto costo, en comparación con el software didáctico correspondiente de bajo costo y que sin embargo debe cumplir totalmente con su equivalente físico.

## **1.2 Justificación**

El desarrollo virtual enfocado al proceso de enseñanza aprendizaje tiene muchas ventajas con respecto a la enseñanza aprendizaje actual, requiere de aspectos físicos (utilería) que inevitablemente genera un costo y que muchas escuelas en nuestro país no son capaces de absorber. En cuestiones de robótica dicho costo aún es más elevado y por lo tanto, no todas las instituciones cuentan con el capital requerido para implementar la materia o algún complemento de enseñanza para dicha materia.

## 1.3 Hipótesis

Demostrar que los beneficios del software educativo diseñados en 3D presentan muchas ventajas en comparación con el uso físico de dispositivos o elementos que apoyan el proceso de enseñanza aprendizaje, algunos de estos beneficios que esperamos obtener son:

- Aprendizaje interactivo basado en la técnica de constructivismo “Aprender-Haciendo”.
- Bajo costo en el uso de materiales al ser un desarrollo virtual.
- Cambio de escenarios que generan innovación y creatividad.
- Fortalecer la innovación en el desarrollo y construcción de robots.
- Motivar el enfrentamiento de retos para la programación de robots.
- El aprendizaje puede realizarse fuera del aula.
- Reforzar los conocimientos por medio de la práctica constante.
- El aprendizaje puede ser de manera autónoma.
- Evaluar del aprendizaje en forma directa.

## **1.4 Objetivo General**

Mostrar el procedimiento de diseño y creación de escenarios virtuales en tercera dimensión “3D” para ejemplificar el uso de los diferentes tipos de robots y su campo de aplicación.

## **1.5 Objetivos Específicos**

1. Analizar los diferentes tipos de software utilizados para el diseño en 3D y utilizar el más apropiado para esta aplicación.
2. Mostrar el procedimiento de diseño de la utilería utilizada para cada escenario de acuerdo al tipo de robot.
3. Explicar la función de la plataforma a utilizar, conocer el área de trabajo de la pantalla principal y sus herramientas principales.
4. Describir como se arma un escenario por medio de la construcción de un terreno.
5. Aprender a diseñar el relieve de un terreno tales como valles y montañas por medio de heightmaps.
6. Brindar realismo al terreno por medio de texturas como pasto, hierbas, árboles y agua.
7. Mostrar cómo se realiza la ejecución del escenario y como se visualizará al terminar el procedimiento de armado.

## **CAPÍTULO II**

### **ESCENARIOS VIRTUALES**

## **CAPÍTULO II: ESCENARIOS VIRTUALES**

### **2.1 Introducción**

Desde el inicio de la humanidad, el hombre primitivo como medio de subsistencia ha tratado de comprender el comportamiento de fenómenos naturales para lograr adaptarse a estas manifestaciones, en un principio utilizaron los elementos de su entorno para mejorar su condición de confort, por ejemplo, crearon algunos utensilios o instrumentos para llevar el agua a la boca, inventaron armas rústicas que les permitían defenderse de otros hombres y/o animales, etcétera.

Su escenario natural, que es lo que forma parte de todo su ecosistema, permitió que el ser humano lograra aprender a transformar y con ello satisfacer necesidades vitales como alimentación, vestido y hogar...etc.

Conocer se volvió parte de una necesidad para el hombre pero no solo desde tiempos primitivos sino hasta ahora en la actualidad la necesidad persiste. Hoy en día, esta necesidad se aplica de diversas maneras para entender-conocer lo que pasa en la realidad, y busca de diversas formas obtener ese entendimiento, el cual en la actualidad puede hacerse mediante experimentación física o en un entorno virtual.

Por manera virtual se puede entender que es una simulación por computadora, dinámica y tridimensional en la mayoría de sus casos, con alto contenido gráfico destinada a crear una realidad, la cual se conforma de escenas u objetos de apariencia real tomando como base un entorno físico para generar en el usuario, la sensación de estar inmerso en él.

## **2.2 Realidad virtual**

Actualmente muchas empresas se han dedicado a desarrollar escenarios virtuales en un contexto denominado realidad virtual, siendo el primero y más grande nicho de mercado el área de entretenimiento y de los videojuegos pero, que posteriormente se ha extendido a otros campos como a la medicina, ver figura 2.1, la arqueología, la creación artística, el entrenamiento militar o simulaciones de vuelo (Wikimedia Foundation Inc., 2016).

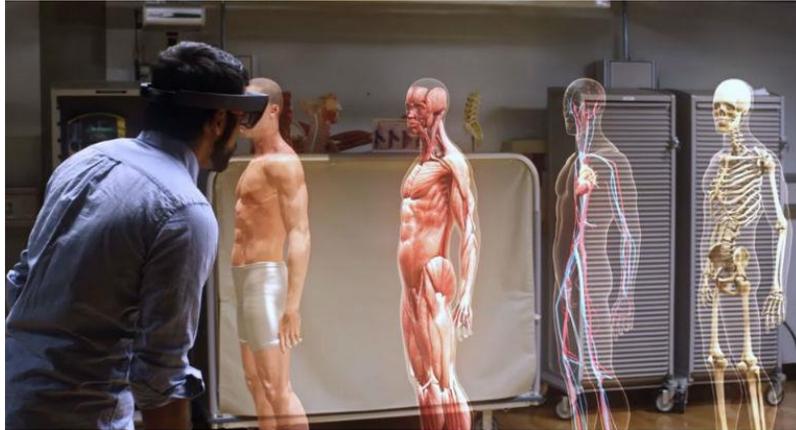


Figura 2.1. Realidad virtual aplicada en el área de medicina.

En cuanto a educación, la realidad virtual avanza a pasos agigantados, aunque queda mucho por hacer pero aun así, las ventajas son muchas para todos los alumnos y que mejor, es para todas las edades.

Por ejemplo una ventaja interesante, es que son pocas las empresas que estén creando contenido para la educación, ya que toda la atención y avances se están realizando en la industria del entretenimiento.

### **2.3 Escenario virtual**

Un escenario virtual es una simulación en computadora dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y variables complejas, durante la cual el usuario ingresa a situaciones que aparentan ser reales resultando

inmerso en ambientes altamente participativos siendo de origen artificial como se aprecia en la figura 2.2.



Figura 2.2. Ejemplo de escenario virtual que muestra la interacción de robots en una fábrica.

Como características en su diseño se pueden tener:

- Responde a la metáfora de "mundo" que contiene "objetos".
- Se expresa en lenguaje gráfico bidimensional o tridimensional.
- Su comportamiento es dinámico y opera en tiempo real.
- Su operación está basada en la incorporación del usuario en el "interior" del medio computarizado.
- Posee la capacidad de reaccionar ante el usuario, ofreciéndole, en su modalidad más avanzada, una experiencia inmersiva, interactiva y multisensorial (2009).

Actualmente, se puede encontrar en internet software educativo o herramientas que manejan contenido en 3D, esto con la finalidad de que la interacción con el usuario y el aprendizaje sea de lo más amigable.

Un ejemplo es el de 3D Tools para SMART, con esta herramienta, los objetos y modelos pueden manipularse en tres dimensiones y etiquetarse desde una gran variedad de ángulos y perspectivas, permitiendo así que los alumnos comprendan mejor los objetos físicos de cualquier asignatura pero más especialmente en ciencias, tecnología y matemáticas (Technologies S. , 2016).

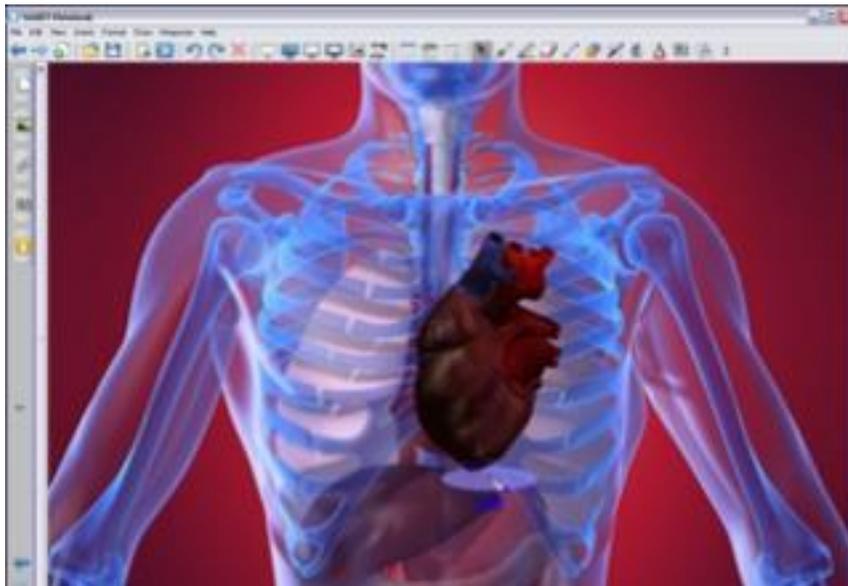


Figura 2.3. Uso del software de SMART en el área de medicina.

Como puede observarse en la figura 2.3, el uso de programas tridimensionales fomenta a un gran desarrollo para el aprendizaje ya que al mostrar imágenes u objetos lo más real posible, los alumnos aprenden todo esto de manera más exacta y se comprende mucho mejor lo que se pretende mostrar.

Actualmente se encuentran disponibles muchos programas educativos que trabajan en diversas plataformas y se pueden trabajar los objetos de manera tridimensional.

En la tabla 2.1 se muestran los desarrollos actuales:

Software o herramienta	Descripción
Cabri 3D	Software educativo dedicado al área de geometría en el que el armado de figuras se lleva a cabo en 3D logrando un entorno más amigable con el usuario (Cabri, 2009).
Winplot	Con Winplot pueden generar gráficos de función lineal, cuadrática, hiperbólica, exponencial, geométrica y trigonométrica, aplicadas a distintas áreas de conocimiento: demografía, biología, física, química, etc. Con este programa los alumnos podrán ejercitar

	con ecuaciones explícitas, paramétricas, implícitas y cilíndricas, generar curvas simples, tubos e incluso representar ecuaciones diferenciales tanto en dos como en tres ejes (2D y 3D (Winplot, 2011)).
Tinkercad	Software de diseño en 3D con herramientas sencillas para la elaboración de dibujos, formas, etc. Serás guiado a través del proceso de diseño 3D a través de "lecciones", que enseñan los conceptos básicos antes de pasar a las técnicas de modelado más complejas (Autodesk, 2016).

Tabla 2.1. Ejemplos de software en 3D aplicados en diferentes áreas de estudio.

## 2.4 Escenarios para robots

Para realizar un escenario es importante conocer cuál es el hábitat natural de aplicación de cada robot, a fin de hacer más clara la caracterización, considere la tabla 2.2. En la que se muestran la **morfología de los diversos tipos de robots y las características de cada uno de ellos** (Maxinez, 2012).

Tipo de Robot	Descripción	
	Androide	Robot con apariencia semejante a la de un humano.
	Zoomórfico	Robot con apariencia semejante a la de un animal.
	Móvil	Robot con capacidad de desplazamiento por medio de plataformas móviles.
	Poli articulado	Robot con capacidad de movimiento por medio de brazos para recolectar o mover objetos.
	Híbrido	Robot caracterizado por tener una combinación o más de la morfología de los robots antes mencionados.

Tabla 2.2. Descripción de cada tipo de robots existentes.

## **CAPITULO III**

### **CARACTERIZACIÓN DE ESCENARIOS**

# CAPÍTULO III: CARACTERIZACIÓN DE ESCENARIOS

## 3.1 Introducción

Una caracterización se va a entender como todo aquello que comprende un escenario virtual, tales como los árboles, cajas, montañas o simplemente los objetos en los que va a interactuar el niño y/o joven en el escenario dispuesto en su robot.

El armar un escenario va depender del tipo de robot que se vaya a armar y en consecuencia no todos los escenarios tendrán la misma forma o mejor dicho, no tendrán el mismo ambiente, por ejemplo, si el niño armará un robot **poliarticulado** el escenario virtual debe estar hecho con base al comportamiento de ese tipo de robot siendo este una fábrica o una construcción para que su objetivo en el escenario será mover objetos grandes o pesados de un lugar a otro, ver figura 3.1.



Figura 3.1. Robot poliarticulado trabajando para una ensambladora de coches.

En general la descripción de cada escenario deberá tener los siguientes puntos.

- Tema
- Objetivo
- Caracterización del escenario
- Utilería del escenario

## 3.2 Tema

Se puede entender que será el asunto a tratar; en este proyecto cada escenario tendrá un tema que va a depender del tipo de robot que se arme, su función y trabajo a desarrollar, por ejemplo:

**Tema:** La exploración del planeta Marte.

**Descripción.** El planeta Marte es considerado un planeta telúrico<sup>1</sup>, por lo tanto tiene en su superficie volcanes, valles y desiertos. Su suelo es rocoso y presenta marcas de impactos contra otros cuerpos celestes, se le conoce como “planeta rojo” por la coloración rojiza de su suelo, causada por la oxidación de los minerales de hierro que lo conforman y su atmosfera es muy fina y tenue, está compuesta principalmente por dióxido de carbono, nitrógeno y argón (GeoEnciclopedia, s.f.).

## 3.3 Objetivo

En el objetivo se establece la meta a alcanzar de manera general.

---

<sup>1</sup> Los planetas telúricos son aquellos que tienen como núcleo roca o piedra, que son además sólidos y se encuentran cerca del Sol estos son: Mercurio, Venus, Tierra y Marte.

**Ejemplo.** Se requiere en la exploración del planeta recoger piedras y polvo marciano de diversos lugares y superficies.

### 3.3.1 Tipo de robot a construir

En base al objetivo se sugiere la construcción de un robot con capacidad de desplazamiento y que pueda recoger piedras y polvo marciano.

**Ejemplo.** Se propone **robot híbrido**, el cual puede desplazarse de un punto a otro y que además tiene capacidad para coger piedras o polvo a través de brazos o palas mecánicas. En la figura 3.2 se muestra un robot híbrido que fue utilizado para la exploración del planeta Marte.

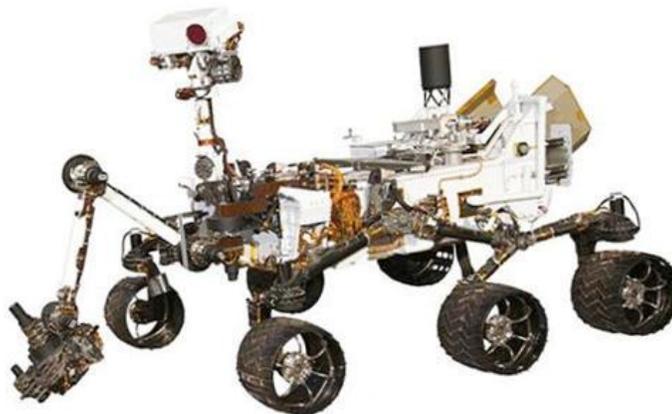


Figura 3.2. Robot híbrido utilizado para la exploración de Marte.

**Trabajo a desarrollar:** Consiste en construir un escenario o realidad virtual acorde al tema que se trata.

**Ejemplo.** Construir escenario en base a las características del planeta Marte. En la figura 3.3, se muestra el suelo del planeta Marte el cual puede ser tomado como referencia para construir el escenario.

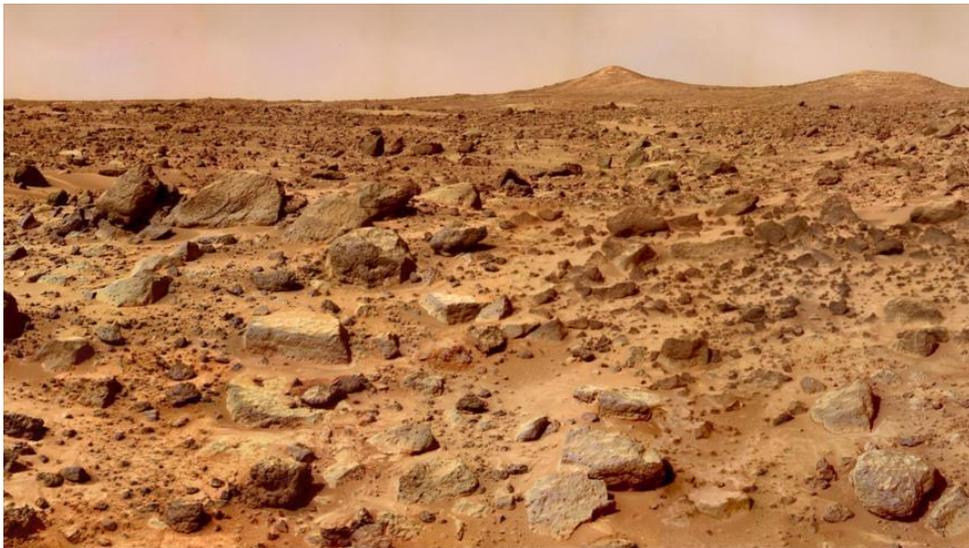


Figura 3.3. Escenario donde se visualiza el suelo del planeta Marte.

### **3.3.2 Caracterización y utilería del escenario**

**Consiste en reunir dentro de un esquema o boceto, todos los elementos utilizados para la ambientación** incluyendo la utilería necesaria para recrear el escenario natural del planeta Marte.

**Ejemplo: Elementos para la caracterización.**

- Montañas
- Valles
- Desiertos
- Rocas
- Tierra o polvo de tono rojo

## **3.4 Escenarios para robots**

Para realizar una caracterización apropiada a cada tipo de robots es necesario profundizar en el comportamiento o morfología de cada robot.

### **3.4.1 Robot Androide**

Como se explicó, es la denominación que se le da a un robot cuya apariencia es parecida a la de un ser humano y de igual manera imita

algunos aspectos de su conducta de manera autónoma, como se muestra en la figura 3.4.



Figura 3.4. Robot androide.

Por lo tanto, tiene capacidad de desplazamiento mediante piernas y se apoya con movimiento de brazos y cabeza facilitando la realización de tareas del ser humano.

Por su morfología su hábitat natural, es el ambiente de un ser humano, en sus diferentes facetas de actividad. Empresarios, investigadores, chefs, etc.

**Utilería del escenario virtual.** En este caso depende del entorno en donde se encuentre el ser humano y la función que realiza, por ejemplo:

1. **Empresas:** Se encuentran: Ejecutivos, presidente, recepcionistas, vigilancia, azafata de ferias y/o congresos, cajeros, empacadores.
  - **Utilería para el escenario:** Mesas, sillas, libros, computadoras, oficinas, escritorios, libreros, anaqueles, archiveros, cestos de basura, teléfonos, lámparas, plantas de adorno, impresoras, lapiceros.
  
2. **Ciencia:** Realizando investigaciones espaciales y oceánicas.
  - **Utilería para el escenario:** para escenarios espaciales son trajes espaciales, herramientas para armado y desarmado de objetos, botiquín médico, contenedores para muestras, videocámaras, estrellas, cielo espacial, planetas, naves espaciales; para investigaciones oceánicas son tanques de oxígeno, trajes de buceo, googles, botiquín médico, contenedores para muestras, objetos de medición, videocámaras, animales marinos, plantas marinas, piedras acuáticas, arrecifes, arena.
  
3. **Medicina:** Realizando procedimientos quirúrgicos de invasión mínima y transportando muestras biológicas o químicas.
  - **Utilería para el escenario:** Camas, camillas, medicinas, anaqueles, cestos de basura, escritorios, libros y libretas, computadoras, instrumento médico, aparatos médicos, lámparas.
  
4. **Búsqueda y rescate:** Realizando actividades riesgosas para el ser humano y con mayor precisión en caso de accidentes o desastres naturales.

- **Utilería para el escenario:** Edificaciones destruidas, arboles, montañas deslavadas, animales vivos y muertos, ríos, lagos o mares, vidrios rotos, coches, ropa, muebles.
5. **Hogar:** Realizando tareas domésticas del hogar como: cocinar, mover objetos de un lado a otro, realizar limpieza de una habitación, vigilancia.
- **Utilería para el escenario:** Habitaciones, camas, mesas, sillas, aparatos electrónicos, baños, mascotas, vasos, cacerolas, ollas, cuchillos, libreros, lámparas.

Así es como se debe realizar el desglosamiento de caracterización para cada tipo de escenario.

### 3.4.2 Robot Zoomórfico

Como se explicó, la morfología de este tipo de robot se caracteriza por ser similar a la de un animal, la forma de desplazamiento de éste se va a diferenciar dependiendo del tipo de clase animal al que se arme dicho robot, por ejemplo, si es un mamífero, podemos encontrar gatos, perros, caballos, ornitorrincos, canguros, delfines y ballenas, en este desglosamiento se encuentran animales acuáticos, los cuales su tipo de movilidad es por medio de aletas a diferencia de los terrestres que se

pueden desplazarse por medio de sus patas, como se muestra en la figura 3.5.



Figura 3.5. Robot zoomórfico.

Este robot, se puede desplazar en un hábitat salvaje o aunado con el ser humano en el hogar o desastres naturales.

**Utilería del escenario virtual.** En este caso depende del entorno en donde puedan interactuar este tipo de robots, por ejemplo:

1. **Desastres naturales:** Hace referencia a las enormes pérdidas materiales y vidas humanas ocasionadas por eventos o fenómenos naturales como los terremotos, inundaciones, tsunamis, deslizamientos de tierra, deforestación, contaminación ambiental y

otros por lo tanto puede realizar actividades riesgosas para el ser humano y con mayor precisión en caso de accidentes o desastres naturales.

- **Utilería para el escenario:** Como puede ser variado de acuerdo a la situación, podemos encontrar principalmente edificaciones derrumbadas, escombros, personas, arboles, vehículos, animales, fuego y para el caso de un tsunami, el mar.

**2. Hogar:** Se puede realizar actividades como compañía para el ser humano o vigilancia del hogar.

- **Utilería para el escenario:** Una casa, habitaciones, personas, camas, muebles, plantas, sillas, aparatos electrónicos.

### **3.4.3 Robot Móvil**

Estos robots se caracterizan por tener capacidad de desplazamiento por medio de plataformas móviles que pueden ser por medio de llantas o bandas. Estos pueden tener forma de coches y su movimiento será similar a uno tal y como se muestra en la figura 3.6.



Figura 3.6. Robot móvil utilizado por la NASA para exploración de Marte.

**Utilería del escenario virtual.** El entorno en el que puede realizar funciones propias del robot son muchas, por su morfología se puede utilizar tanto en la ciencia como para uso personal, algunos ejemplos son los siguientes:

1. **Ciencia:** Puede ser utilizado para exploración de otros planetas, recolectando muestras y proporcionando nueva información para la ciencia.
  - **Utilería para el escenario:** La exploración varía de acuerdo al planeta a investigar pero se puede encontrar rocas, montañas, polvo o tierra, valles, desiertos.  
Para realizar investigaciones de nuevas áreas en el planeta Tierra.
  - **Utilería para el escenario:** El entorno puede llevar árboles, plantas, animales, piedras, agua, valles, montañas, desiertos.

2. **Entretenimiento:** Se puede encontrar en exposiciones de robótica y en concursos de velocidad o seguidores de línea.
  - **Utilería para el escenario:** Para ambos ejemplos se encuentra personas, plataformas con diseños específicos para realizar una competencia de carrera, sillas, computadoras, mesas.
  
3. **Vigilancia:** Se puede montar un CCTV en un robot móvil que permita observar no solo un área específica sino que tenga capacidad de desplazamiento a libre demanda por el operador.
  - **Utilería del escenario:** cámaras de video, edificios, patios, calles, habitaciones, personas, arboles, plantas, mesas, sillas, aparatos electrónicos como televisores y/o computadoras.

#### 3.4.4 Robot Poliarticulado

Su capacidad de movimiento principalmente es por medio de brazos robóticos los cuales su función es recolectar o mover objetos. Su morfología es similar a la de un brazo humano, ver figura 3.7.



Figura 3.7. Robot poli articulado utilizado por empresas.

**Utilería del escenario virtual.** Por su morfología se puede encontrar en fábricas, hospitales y edificios, desempeñando tareas que requiera de una fuerza superior al humano o de mayor precisión. Un ejemplo es como se muestra en la figura anterior, que es un brazo robótico utilizado por empresas industriales para el armado de objetos.

- 1. Empresas industriales:** Por su capacidad de fuerza se puede utilizar para realizar trabajos pesados como es el armado de coches, transportando objetos pesados de un lugar a otro.
  - **Utilería para el escenario:** En una ensambladora de autos se requiere partes de autos a ensamblar, bandas para desplazamiento de objetos, maquinaria para armado de automóviles, personal, cajas, aparatos electrónicos como computadoras, cables, la fábrica.

2. **Medicina:** En algunas situaciones, se pueden requerir de brazos robóticos para realizar operaciones a personas o animales con un alto grado de precisión.
- **Utilería para el escenario:** Hospital, quirófano, personal médico, computadoras, camilla, aparatos médicos, instrumento médico y el paciente.

### 3.4.5 Robot Híbrido

Este robot se caracteriza por ser una combinación de una o más morfologías de los robots anteriormente descritos tal y como se muestra en la figura 3.8.



Figura 3.8. Robot híbrido utilizado para realizar diversas acciones.

Se puede desplazar por medio de ruedas, bandas, piernas, puede tener o no brazos y su función puede ser muy variada ya que al tener una o más morfologías de otros robots, su función puede variar de acuerdo a su armado.

**Utilería del escenario virtual.** Se puede encontrar en diferentes ambientes:

1. **Seguridad:** En algunas situaciones, los robots móviles son utilizados para realizar actividades riesgosas tal como desactivar bombas o abrir cajas que pueden contener un explosivo o un arma biológica.
  - **Utilería para el escenario:** Edificios, calles, personas, automóviles, arboles, plantas, montañas, valles, cajas, botes de basura, paquetes que puedan contener artefactos explosivos, bombas.
  
2. **Empresas industriales:** Se puede utilizar para realizar empacamientos, moviendo objetos de un lugar a otro y/o armar los mismos.
  - **Utilería del escenario:** Fábrica, cajas, materia prima, maquinaria de fábricas, herramientas, computadoras, sillas, personal.
  
3. **Construcciones:** Su función puede servir para mover bloques pesados y transportar material.

- **Utilería del escenario:** Edificación en construcción, obreros, material de construcción como tabiques, cemento, varillas, carretillas, palas, picos, tierra, agua, maquinaria pesada.

## **CAPÍTULO IV**

### **DESARROLLO E IMPLEMENTACIÓN**

# **CAPÍTULO IV: DESARROLLO E IMPLEMENTACIÓN**

## **4.1 Introducción**

Para el diseño y construcción de escenarios es necesario conocer perfectamente el hábitat donde se desenvolverá, por tanto cada escenario tendrá objetos diferentes ya que el medio en el que se encontrará cada robot tendrá una tarea en específico, por ejemplo como ya se mencionó anteriormente si tomamos un robot móvil que tiene como objetivo realizar una exploración en el planeta Marte, este tendrá que desplazarse a través de campo rocoso, los objetos de ese escenario probablemente serán montañas, tierra y piedras, siendo éstos los que van a estar interactuando principalmente con el robot.

Por otro lado, en este capítulo se describe detalladamente como se construye un escenario y cuáles son las herramientas de diseño más apropiadas para este fin.

## **4.2 Plataformas de diseño**

Actualmente existen muchas plataformas de diseño en 3D para la creación de juegos, películas, virtualización de diseños para arquitectura etc. Los más utilizados son los siguientes:

1. **3D Studio Max:** Con su arquitectura modular basada en plugin<sup>2</sup>, es uno de los programas de animación 3D más utilizado, especialmente para la creación de videojuegos, anuncios de televisión, en arquitectura o en películas (Autodesk Inc., 2016). Ejemplo: Figura 4.1.

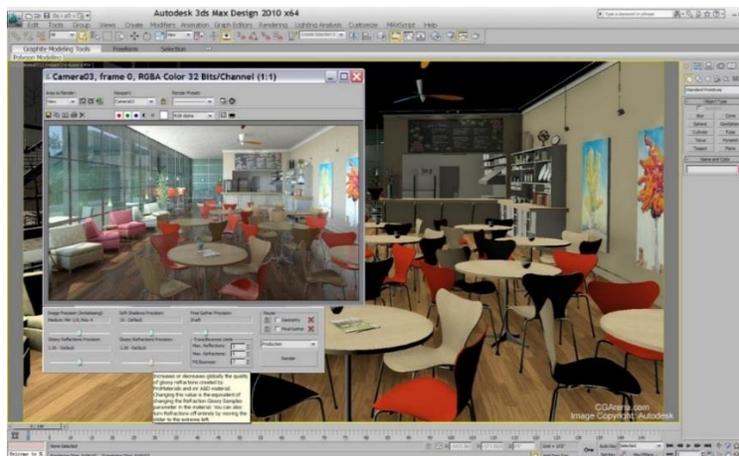


Figura 4.1. Uso de 3D Studio Max para la creación de escenarios.

2. **Blender:** Es un programa informático multiplataforma, dedicado especialmente al modelado, iluminación, renderizado<sup>3</sup>, animación y

---

<sup>2</sup> **Plugin:** Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es un módulo aparte que se incluye opcionalmente en una aplicación

<sup>3</sup> **Renderizado:** Es un término usado en jerga informática para referirse al proceso de generar una imagen o vídeo mediante el cálculo de iluminación (GI, Graphics Illumination) partiendo de un modelo en 3D.

creación de gráficos tridimensionales. (Blender, 2016) Ejemplo: Figura 4.2.



Figura 4.2. Uso de Blender para modelar y animar gráficos.

- Autodesk Maya:** Es un programa informático dedicado al desarrollo de gráficos 3D por ordenador, efectos especiales y animación (Autodesk Inc., 2016). Ejemplo: Figura 4.3.



Figura 4.3. Uso de Autodesk Maya para la elaboración de animaciones y efectos especiales.

4. **Unity 3D:** Motor de videojuego<sup>4</sup> que permite crear juegos para diversas plataformas y desarrollar contenidos interactivos en 2D y 3D (Technologies, 2016). Ver figura 4.4.

---

<sup>4</sup> Un **motor de videojuego** es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego. Del mismo modo existen motores de juegos que operan tanto en consolas de videojuegos como en sistemas operativos. La funcionalidad básica de un motor es proveer al videojuego de un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico.



Figura 4.4. Diseño de escenas para videojuegos en Unity 3D.

En la tabla 4.1, se muestran las ventajas y desventajas de cada plataforma.

Software	Ventajas	Desventajas
<b>3D Studio Max</b>	Renderización más realista.	Solo es compatible con la plataforma de Windows.
	Modelado y animación en 3D.	El software tiene un costo.
	Capacidad de desarrollo de videojuegos.	
<b>Blender</b>	Software libre.	Interfaz compleja de entender.
	Multiplataforma.	

	Utiliza muy poco espacio en el disco.	
	Se puede programar en Python.	
	Posee un motor de Juegos interno.	
<b>Autodesk Maya</b>	Es compatible con las plataformas Mac OS y Windows.	Motor de render con características complejas.
	Permite realizar animación, efectos visuales, renderización en 3D.	Gestión de las versiones.
	Permite personalización de interfaz y herramientas.	
<b>Unity 3D</b>	Fácil de usar.	Espacio ocupado por los proyectos.
	Dos lenguajes para los scripts (C# y JavaScript).	Rendimiento.
	Arrastrar y soltar para referenciar casi todo.	Gestión de las versiones.
	Asset Store.	Política de precios y costes.
	Multiplataforma.	Invasión de juegos.

Tabla 4.1. Ventajas y desventajas de software dedicado a animación en 3D.

### 4.3 Construcción de escenarios con Unity 3D

El software que se utilizará para este desarrollo es Unity 3D, por ser un motor de videojuego que permite crear juegos para diversas plataformas y desarrollar contenidos interactivos en 2D y 3D, su interfaz es sencilla de entender y permite descargar el software con un tipo de licencia gratuita.

En la figura 4.5, se muestra cuál es su campo de aplicación.



Figura 4.5. Software Unity y sus campos de aplicación.

## 4.4 Características de Unity 3D

Antes de comenzar se debe considerar lo siguiente. Las especificaciones necesarias para trabajar con Unity son:

- **Para desarrollo**

- OS: Windows 7 SP1+, 8, 10; Mac OS X 10.8+. Windows XP y Vista no son compatibles; y las versiones de servidor de Windows & OS X no se han probado.
- GPU: Capacidades de tarjeta de vídeo con DX9 (modelo de shader 2.0). Todo lo que se haya lanzado desde 2004 debería funcionar. El resto depende principalmente de la complejidad de sus proyectos.

- **Para ejecutar juegos de Unity**

Por lo general, el contenido desarrollado con Unity puede ejecutarse bastante bien en todas partes. Qué tan bien se ejecuta depende de la complejidad de su proyecto. Requisitos más detallados:

- OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+

- Tarjeta de vídeo: capacidades DX9 (shader modelo 2.0); por lo general, todo lo que se haya lanzado desde 2004 debería funcionar.
- CPU: compatible con el conjunto de instrucciones SSE2.
- Web Player (obsoleto): Requiere un navegador que soporte plugins, como IE, Safari y algunas versiones de Firefox

Es importante mencionar que tiene cuatro tipos de licencias como se muestra en la figura 4.6, son las siguientes:

- Personal
- Plus
- Pro
- Enterprise (A medida)



Figura 4.6. Tipos de licencia para Unity 3D.

En la tabla 4.2, se muestran las diferencias de cada tipo de licencia.

Personal	
Todas las prestaciones del motor	Todas las plataformas
Actualizaciones continuas	Sin regalías
Pantalla de inicio: Pantalla de inicio MWU	Capacidad de ingresos: \$100 mil
Unity Multiplayer: 20 usuarios simultáneos	Unity Ads
Acceso a beta	Gestión de puestos flexible
Informes de ejecución	Software educativo para certificación de Unity
Kits de Assets	Soporte Premium
Acceso al código fuente	
Plus	
Todas las prestaciones del motor	Todas las plataformas
Actualizaciones continuas	Sin regalías
Pantalla de inicio: Animación personalizada o ninguna	Capacidad de ingresos: \$200 mil
Unity Multiplayer: 50 usuarios simultáneos	Unity Ads
Acceso a beta	Gestión de puestos flexible

Informes de ejecución	Software educativo para certificación de Unity
Kits de Assets: 20% descuento	Soporte Premium
Acceso al código fuente	
<b>Pro</b>	
Todas las prestaciones del motor	Todas las plataformas
Actualizaciones continuas	Sin regalías
Pantalla de inicio: Animación personalizada o ninguna	Capacidad de ingresos: Ilimitado
Unity Multiplayer: Multiplayer personalizado	Unity Ads
Acceso a beta	Gestión de puestos flexible
Informes de ejecución	Software educativo para certificación de Unity: Acceso de 3 meses
Kits de Assets: 40% descuento	Soporte Premium
Acceso al código fuente	
<b>Enterprise*</b>	
Todas las prestaciones del motor	Todas las plataformas
Actualizaciones continuas	Sin regalías

Pantalla de inicio: Animación personalizada o ninguna	Capacidad de ingresos: Ilimitado
Unity Multiplayer: Multiplayer personalizado	Unity Ads
Acceso a beta	Gestión de puestos flexible
Informes de ejecución	Software educativo para certificación de Unity: Acceso de 3 meses
Kits de Assets: 40% descuento	Soporte Premium
Acceso al código fuente	

Tabla 3.2. Tipos de licencia para Unity 3D.

- \* Moldeable a las necesidades de la empresa (Technologies, 2016).

En este desarrollo se utilizara la licencia de tipo personal, ya que no se requerirán del todo tipo de herramientas que se encuentran en las otras licencias.

Para obtener la plataforma puede visitar su sitio web dando clic en el siguiente vínculo o bien, buscarlo en su navegador web.

<http://unity3d.com/es/unity>

Una vez dentro de la página, elegirá la licencia que más le convenga, ver figura 4.7, se le pedirá que cree una cuenta gratuita que es generada por Unity para poder descargar el software de la página, comprar en la tienda Asset Store, participar en la comunidad de Unity y para permitirte comprar o suscribirte a los diferentes productos que ofrece el mismo, ver figura 4.8.



Figura 4.7. Elección de licencia en Unity.



Figura 4.8. Descargando Unity Personal.

Una vez creada la cuenta y descargado el software, el programa nos mostrará la siguiente ventana, figura 4.9, se ingresará la misma cuenta que se creó al descargar el software.

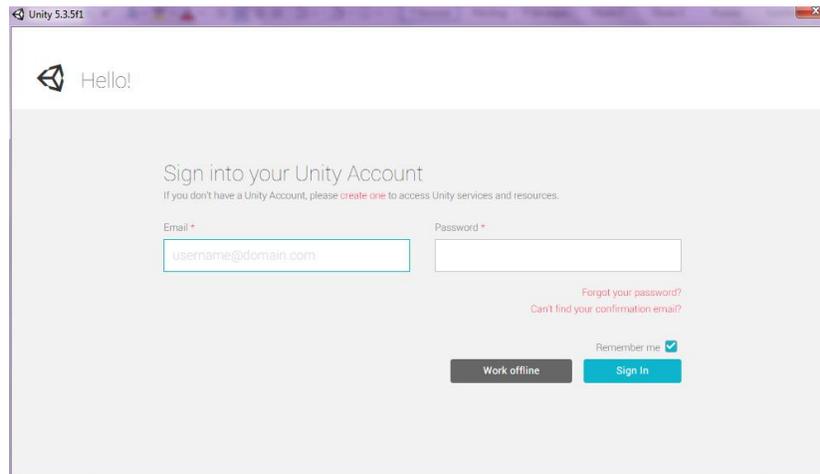


Figura 4.9. Iniciando con Unity.

## 4.5 Creación de nuevo proyecto

Al ingresar se mostrará la siguiente ventana, ver figura 4.10, la cual sirve para comenzar con un nuevo proyecto.

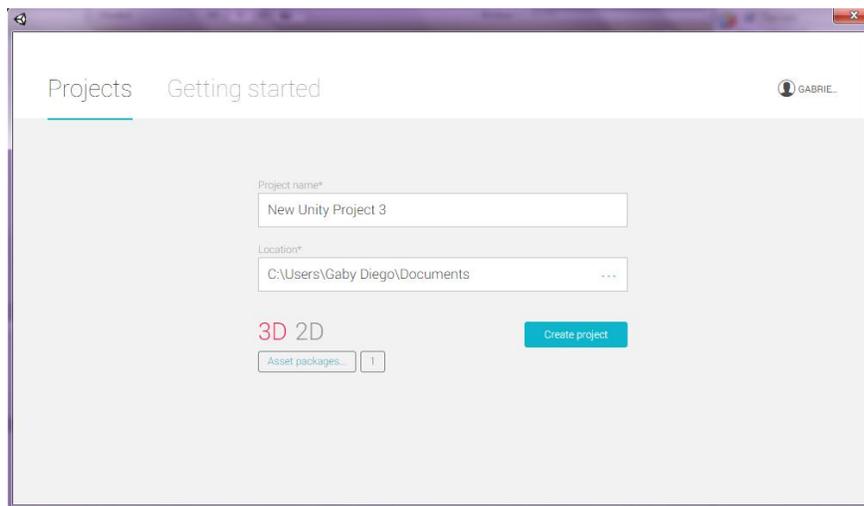


Figura 4.10. Pantalla principal de Unity.

Para crear un nuevo proyecto:

- Agregar un nombre de proyecto y la ruta en donde se guardará. Por defecto el nombre de proyecto será New Unity Project # y se guardará en la carpeta de Documentos.

- Seleccionar el tipo de proyecto, Unity puede trabajar con efecto 2D y 3D, en este proyecto se trabajará en 3D y es lo que se seleccionará, ver figura 4.11.

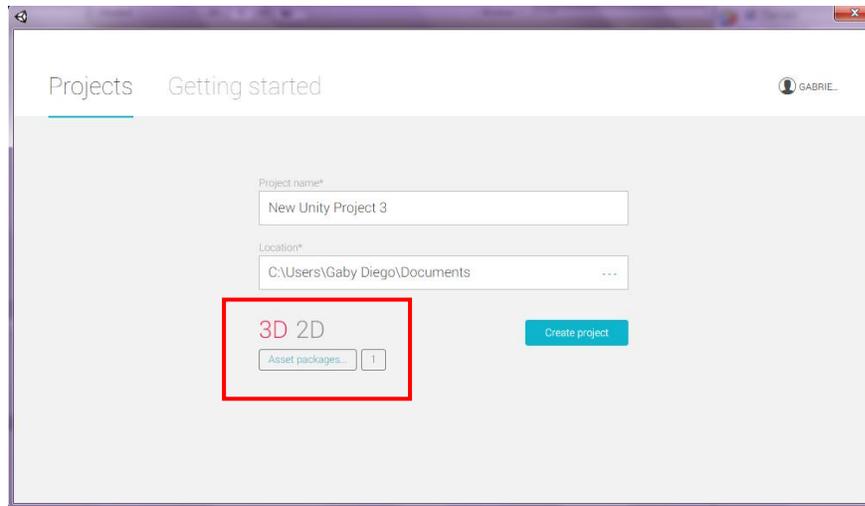


Figura 4.11. Seleccionando el tipo de proyecto en el que se trabajará.

- Para tener el proyecto con toda la paquetería del programa, se agregarán todos los paquetes (Asset Package) ver figura 4.12.

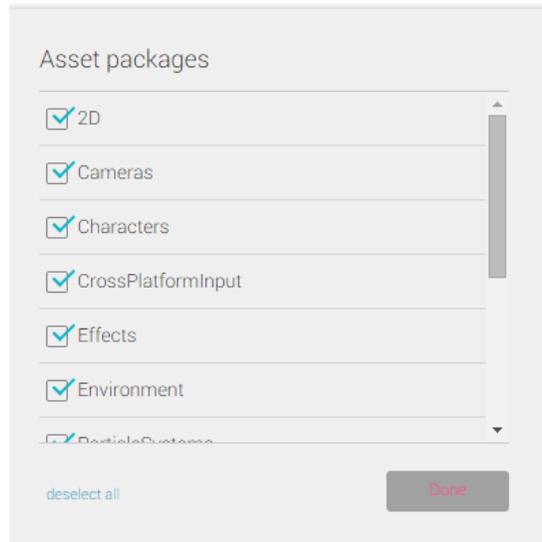


Figura 4.12. Seleccionando todos los paquetes del software Unity.

Una vez creado el nuevo proyecto, esta será la ventana principal del software. Ver Figura 4.13.

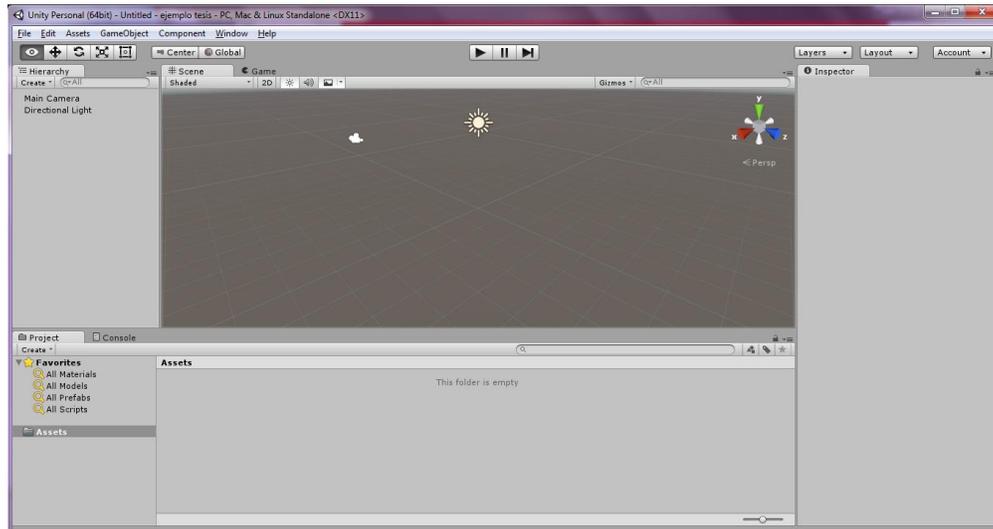


Figura 4.13. Pantalla principal del software Unity.

Para trabajar debe conocer las principales áreas de trabajo:

**Menú principal.** Es aquel menú que se encuentra en la parte superior de la ventana, contiene todas aquellas opciones de creación, edición, objetos de juego y configuraciones, ver figura 4.14.

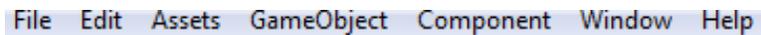


Figura 4.14. Menú principal del software Unity.

**Barra de navegación.** Es aquella parte del programa donde están las opciones de visualización de la estación de trabajo, ver figura 4.15.



Figura 4.15. Barra de navegación del software Unity.

**Área de trabajo.** Es aquella parte del programa donde se visualizará nuestro proyecto, ver figura 4.16.

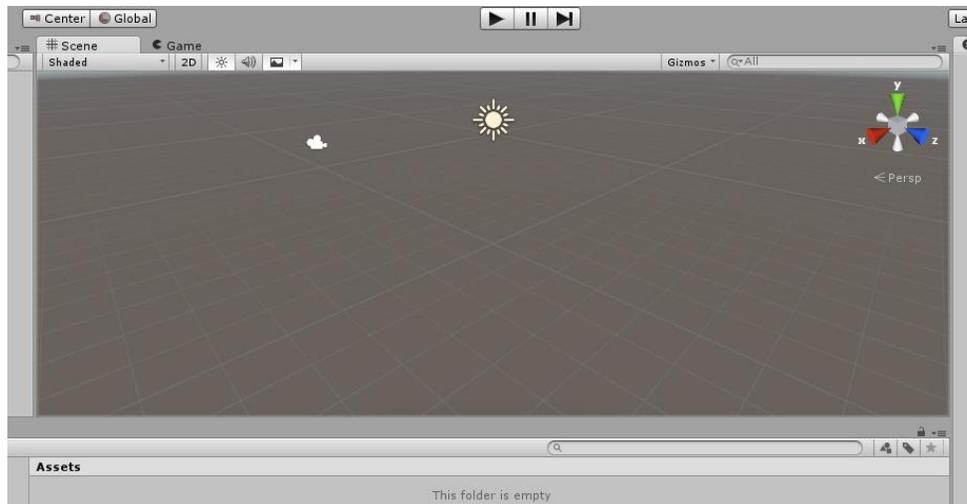


Figura 4.16. Área de trabajo del software Unity.

## 4.6 Construcción de un escenario

Para la construcción de un nuevo escenario, considérese lo siguiente:

## 4.6.1 Terreno

Para crear un nuevo escenario se empezará por crear el terreno que se utilizará, para ello se seleccionará del menú principal GameObject, después desplazaremos el cursor en 3D Object y dará clic en Terrain, ver figura 4.17.

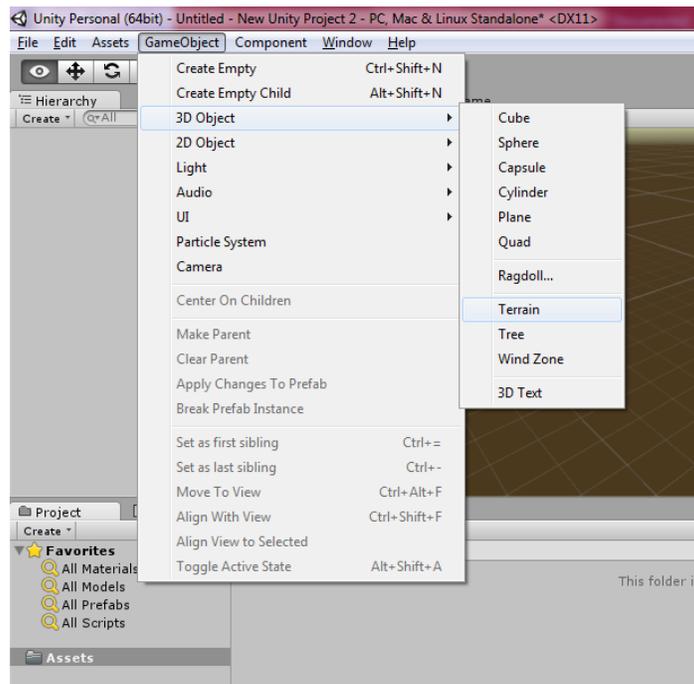


Figura 4.17. Creando terreno para escenario.

## 4.6.2 Características del terreno

Unity tiene por defecto una resolución de 2000 x 2000 píxeles, en este proyecto se tendrán un tamaño de **500 x 500 píxeles**. Esto se puede cambiar dando clic en Terrain que está ubicado del lado izquierdo del área de trabajo y del lado derecho se abrirá una pestaña que se llama Inspector dando detalles de ese objeto, ver figura 4.18.

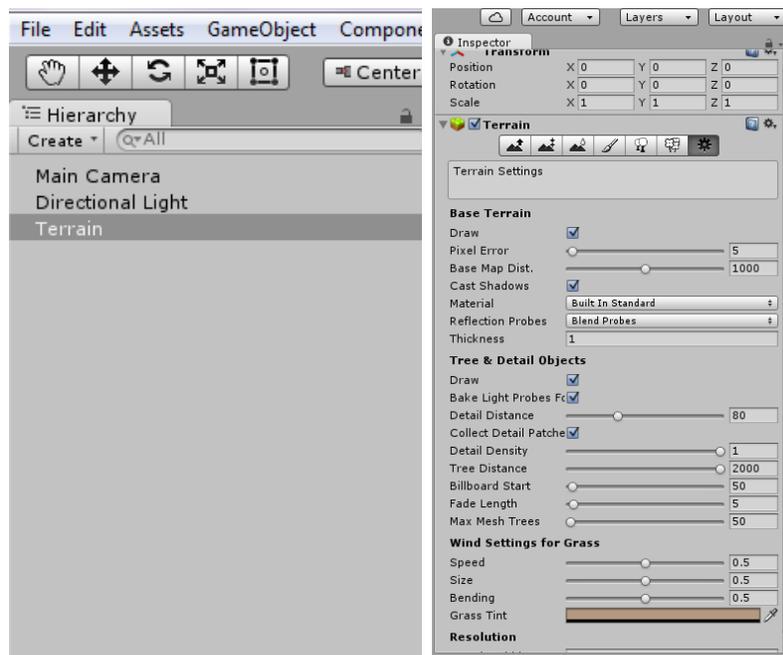


Figura 4.18. Cambiando tamaño de terreno.

Para realizar modificaciones en su tamaño hay un icono con forma de engrane, se seleccionará y se abrirán una serie de detalles del mapa, en

la parte inferior, se encuentra un apartado llamado Resolution, para modificar su tamaño solo se cambiará donde se menciona Terrain Width y Terrain Length (largo y ancho del terreno). Ver figura 4.19.

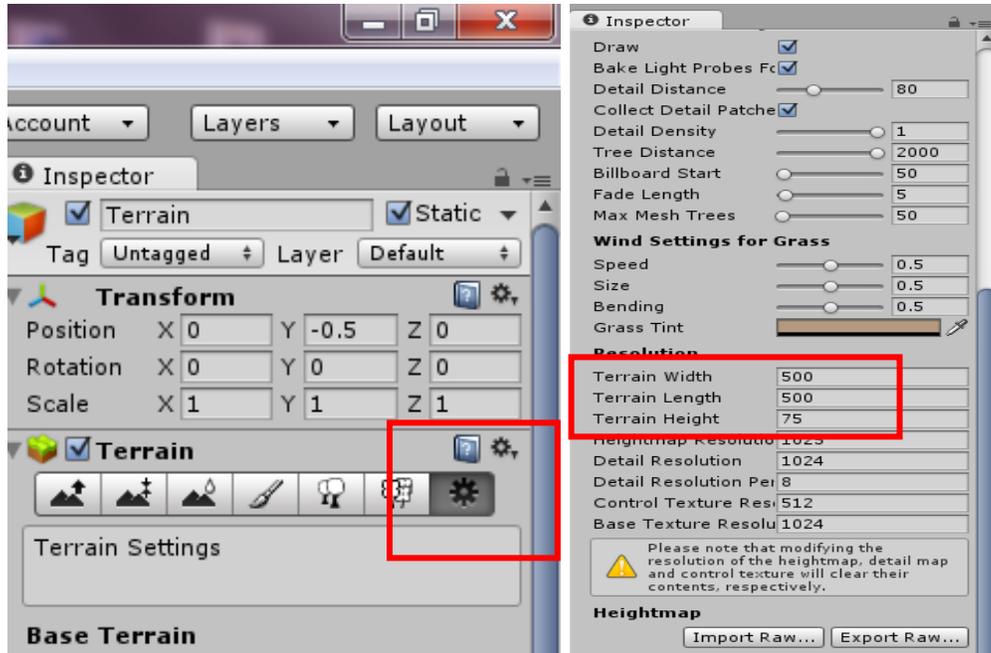


Figura 4.19. Modificando la resolución del escenario.

Una vez creado el terreno, es así como se visualizará en nuestra área de trabajo, ver figura 4.20.

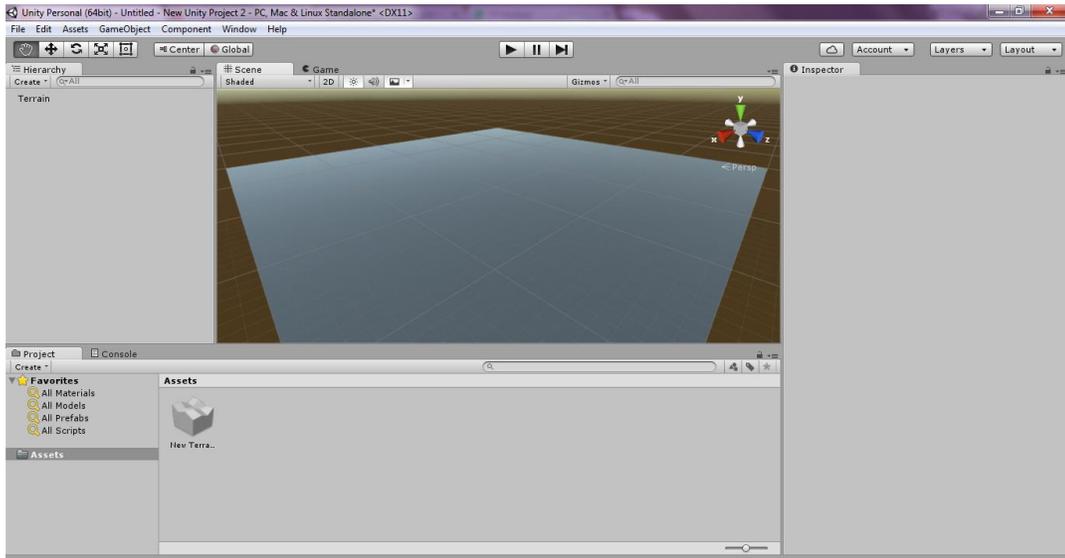


Figura 4.20. Vista del terreno.

### 4.6.3 Textura del terreno

Para que el terreno empiece a tener mayor realismo, se le va a añadir una textura.

Una textura en Unity son los elementos que brindarán color o detalle a un objeto, la plataforma cuenta con algunas texturas básicas, por ejemplo, pasto, tierra, arena, agua, madera, etc.

En este terreno se agregará una textura de tipo pasto con tierra; se logra de la siguiente manera:

- Seleccionamos Terrain y en Inspector hay un botón con un pincel, damos clic.
- Se mostrará un recuadro llamado Brushes y enseguida uno llamado Textures.
- Dará clic en Add Textures, se mostrará un pequeño recuadro que dice Add Texture, Edit Texture y Remove Texture, seleccionará Add texture, tal y como se muestra en la figura 4.21.

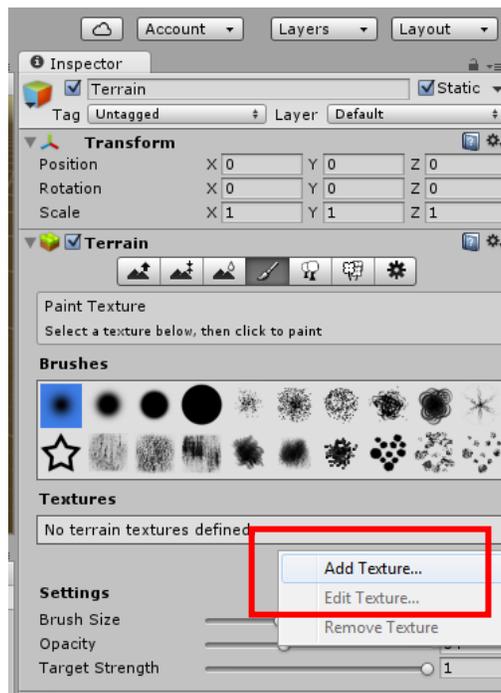


Figura 4.21. Añadiendo textura al terreno.

- Enseguida se mostrará una ventana pequeña del lado izquierdo con el nombre de Add Terrain Texture, se seleccionará el botón Select del primer recuadro, ver figura 4.22.

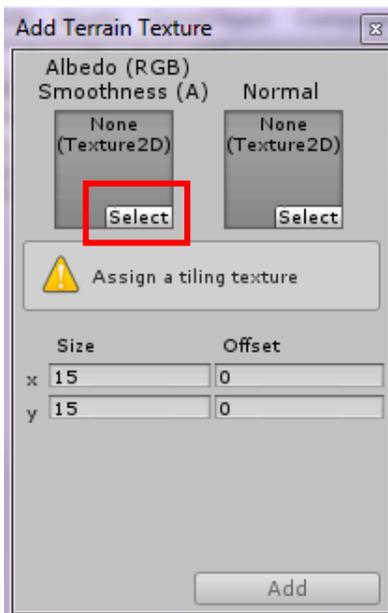


Figura 4.22. Selección de textura.

- Se abrirá otra ventana (Select Texture2D) en donde se mostrarán diversas texturas a seleccionar, se elegirá una con doble clic, ver figura 4.23.

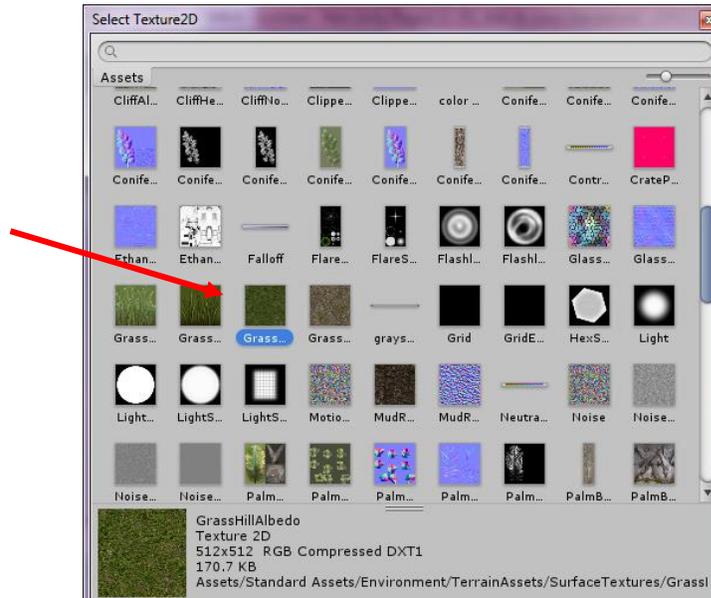


Figura 4.23. Añadiendo textura al terreno.

- Una vez seleccionada la textura se mostrará la ventana anterior con la textura que se eligió previamente y se dará clic en el botón Add, ver figura 4.24.



Figura 4.24. Añadiendo textura al terreno.

- Así es como lucirá el terreno una vez colocada la textura seleccionada, ver figura 4.25.

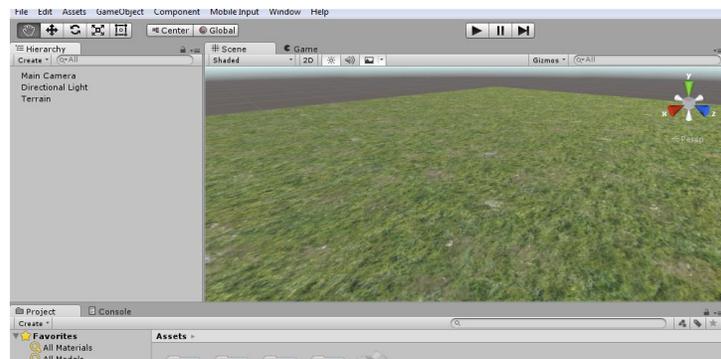


Figura 4.25. Vista del terreno con textura agregada.

#### **4.6.4 Relieve del terreno**

Para poder integrar relieve al terreno se utilizará un heightmap que también es conocido como mapa de altura. Su función principal es para crear elevaciones o profundidad en el terreno, por ejemplo: crear valles, montañas, ríos, lagos, etc.

##### **4.6.4.1 Creación de Heightmap**

Se caracterizan por ser imágenes a escala de blancos y negros en donde el color negro será la profundidad del mapa y los tonos más blancos, serán las elevaciones que tendrá (Wikimedia Foundation Inc., 2015).

Los mapas se pueden descargar directamente de internet o bien se pueden crear. Un ejemplo de heightmap de internet es como el de la figura 4.26.

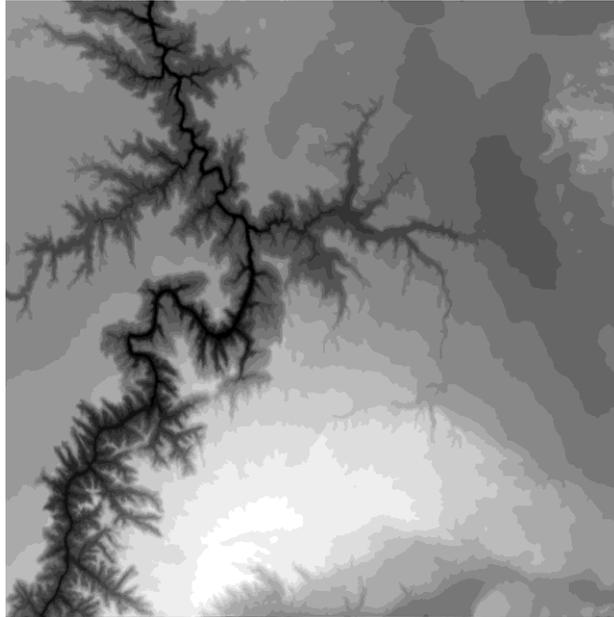


Figura 4.26. Diseño de heightmap descargado de internet.

Para realizarlos, se debe conocer el diseño que tendrá el mapa, esto son los ríos, lagos, montañas, valles etc., que se deseen colocar. También, se utilizará un editor de imagen que tenga la opción de guardado con la extensión .RAW o bien tener un programa que pueda convertir la imagen con formato .JPG o .PNG a .RAW.

Editores de imágenes (Opcional):

- Photoshop
- Corel Draw
- Paint.net

Conversores de imágenes (Opcional):

- XnConvert

Nota: para realizar el proceso de conversión revise Apéndice A.

Continuando con el ejemplo, se utilizará el editor de imagen Photoshop para crear un heightmap.

**Procedimiento en Photoshop:**

1. Se creará un nuevo documento y se le colocarán las siguientes medidas, estas medidas son necesarias para establecer el diseño de la imagen, ver figura 4.27.

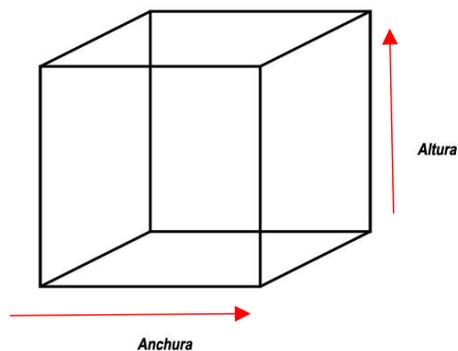


Figura 4.27. Ejemplo de anchura y altura.

- Anchura: 512 píxeles
- Altura: 512 píxeles
- Resolución: 72 píxeles/pulgada
- Modo de color: Escala de grises a 8 bits
- Contenido de fondo: blanco

Ver figura 4.28 para ejemplificar la anchura y altura del terreno.

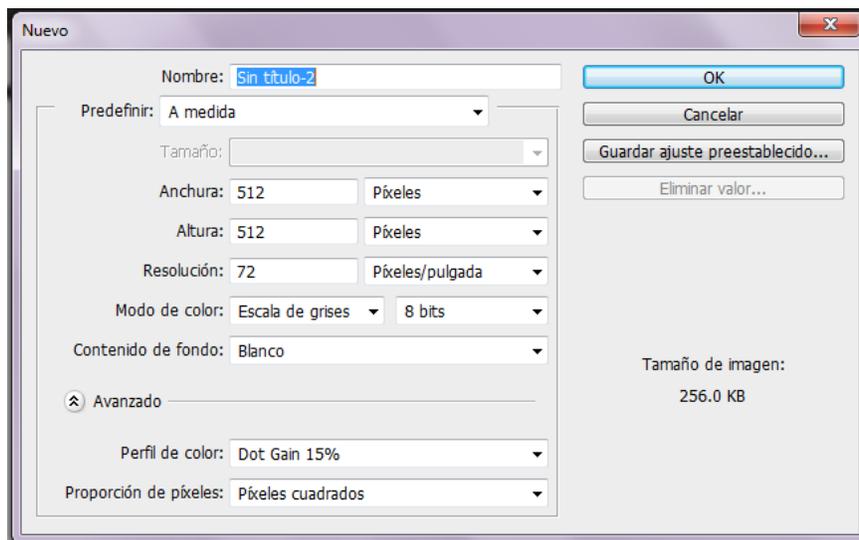


Figura 4.28. Medidas del documento para heightmap en Photoshop.

2. Dibujar en Photoshop el diseño que desee tener para su terreno o bien puede considerar el siguiente dibujo, ver figura 4.29.

- En el dibujo se observa que el color negro profundiza el relieve, en este caso se dibuja un río.
- El blanco es para generar la altura del terreno por ejemplo una montaña.
- El gris muestra la planicie que en esencia es lo que permite distinguir el suelo con la altura de la montaña.
- Los colores degradados que pasan de la tonalidad de gris claro a negro, forma diversos desniveles de **altura** en nuestro caso se está diseñando un lago o un océano.

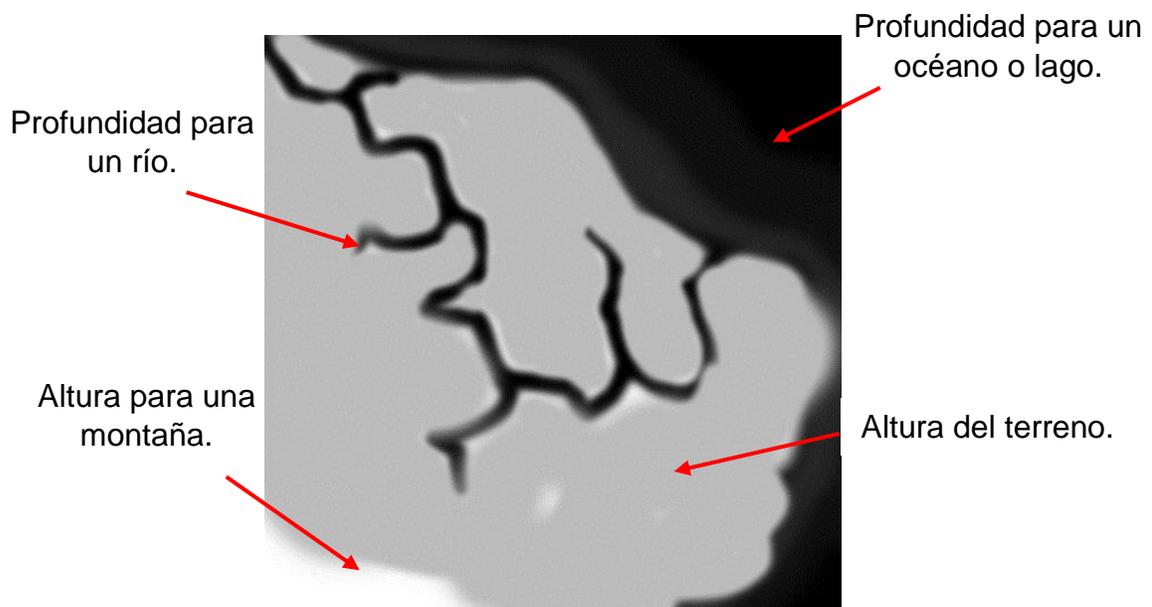


Figura 4.29. Diseño de heightmap en Photoshop.

3. Una vez creado el mapa (dibujo), se guarda con la extensión .RAW, esto se hace de la siguiente manera, ver figura 4.30.

- Dar clic en Archivo, seleccionar Guardar como.

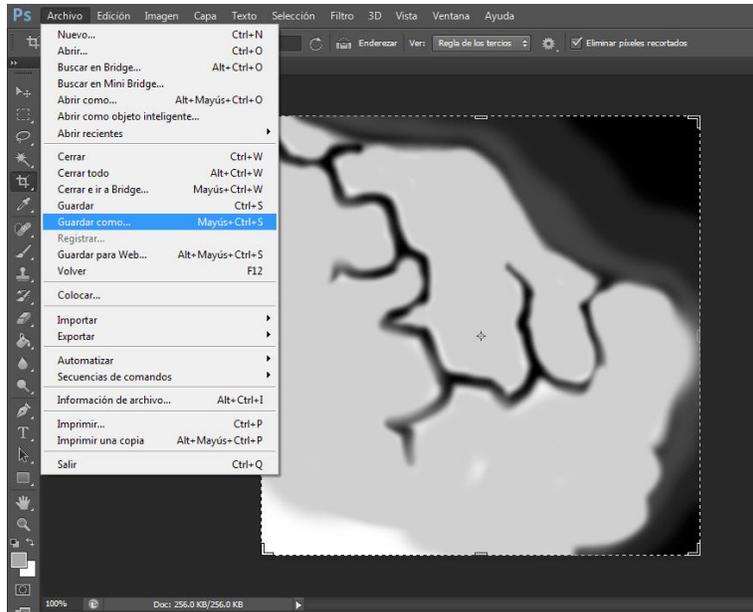


Figura 4.30. Guardando heightmap en Photoshop.

- En la siguiente ventana se da clic en formato y se seleccionará de la lista la extensión .RAW y después se dará clic en Guardar, ver figura 4.31.

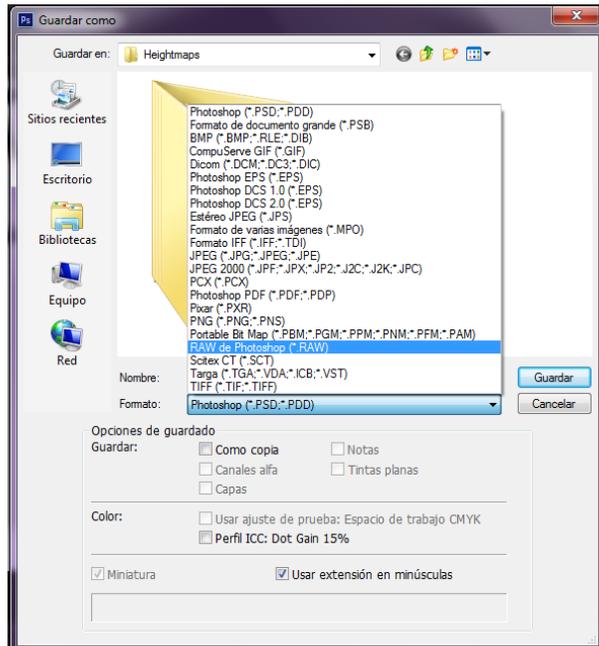


Figura 4.31. Guardando imagen con extensión .RAW.

4. Ya con la imagen en formato .RAW se insertará (importará) al escenario en Unity.

- Dar clic en el engrane.
- En Resolution se encuentra un botón llamado Import RAW, se dará clic, ver figura 4.32.

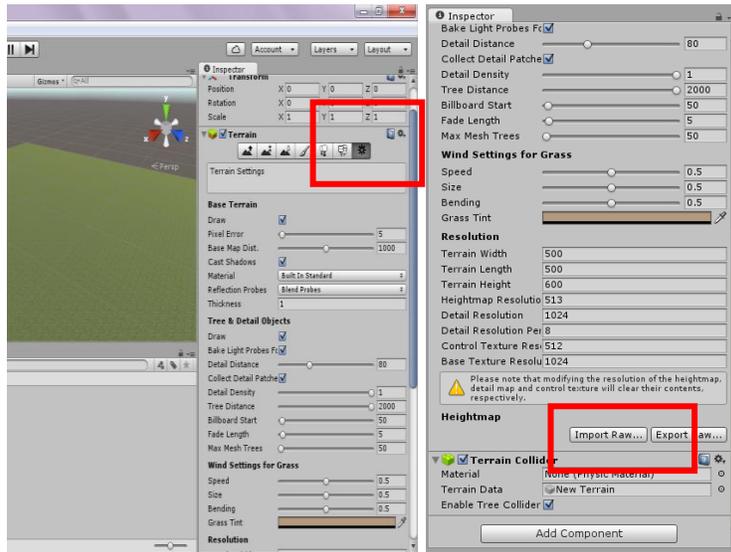


Figura 4.32. Importando heightmap a Unity.

- Posteriormente se genera la siguiente ventana, se busca la ruta donde se encuentra el archivo con extensión .RAW. para importarlo a nuestro proyecto original, ver figura 4.33.

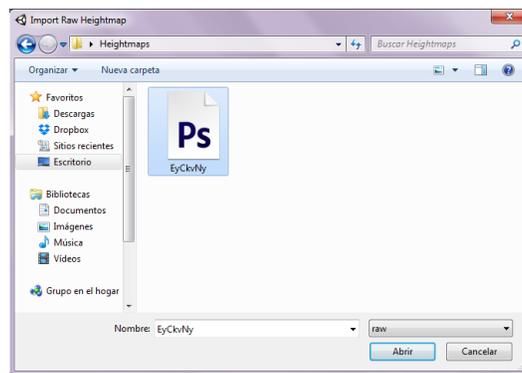


Figura 4.33. Importando el archivo .RAW.

- Después de agregar el archivo (importar), se muestra la siguiente ventana en donde deben especificarse el tamaño de la imagen, estos valores ya fueron establecidos como condición inicial en la sección donde se creó el heightmap, ver figura 4.34.

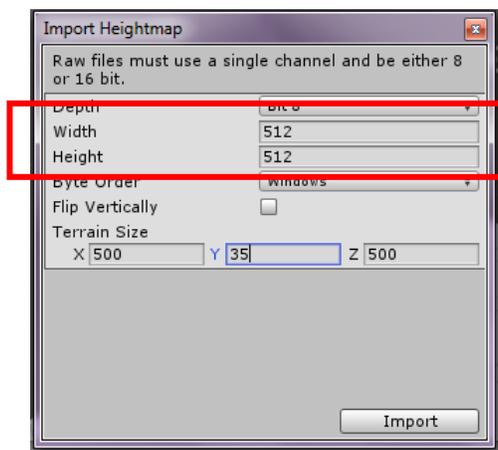


Figura 4.34. Importando el archivo .RAW II.

Nota: En caso de que el dibujo no se vea con la resolución adecuada es posible cambiar las mediciones de anchura, longitud y altura en Terrain Height, Terrain Length y Terrain Width, ver figura 4.35.

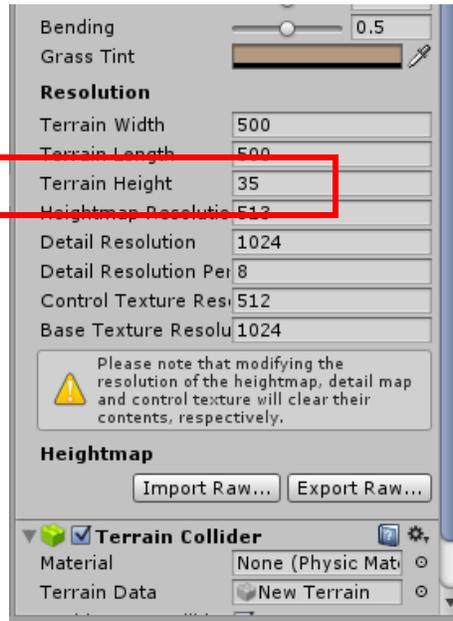


Figura 4.35. Modificando la altura del terreno.

- Finalmente la figura diseñada se muestra de la siguiente manera en el terreno, ver figura 4.36.

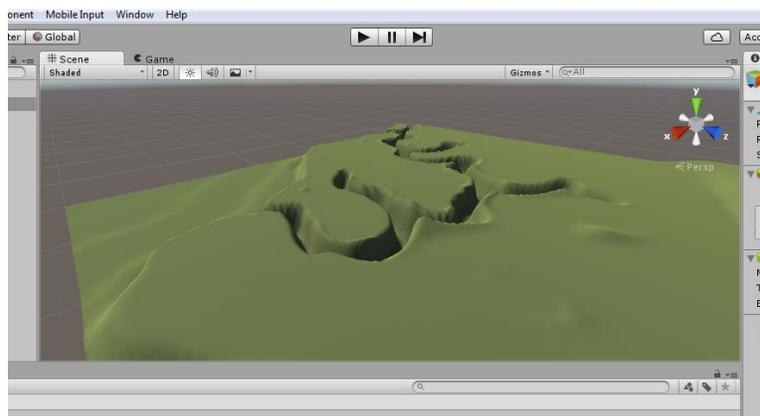


Figura 4.36. Visualización del terreno con el heightmap importado.

Para hacer modificaciones en el escenario en caso de que no haya quedado correctamente, se pueden usar estas herramientas que se encuentran en el Inspector al seleccionar el objeto Terrain, ver figura 4.37.

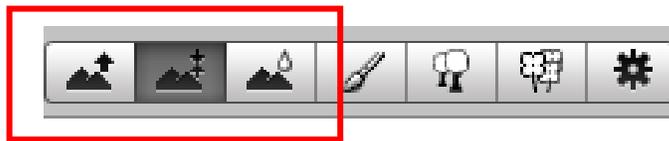


Figura 4.37. Botones que sirven para modificar las elevaciones del terreno.

- El botón con una montaña y la flecha apuntando arriba, sirve para alzar el mapa en un área específica.
- El segundo botón es para hundir el mapa.
- El tercer botón es para darle más suavidad a las elevaciones que se crearon.

Considerando el heightmap y continuando con el ejemplo, por la existencia de un río y un lago se necesita agua. Para agregar agua, que es otra textura se realiza de la siguiente manera:

## 4.6.5 Creación de océanos, lagos y ríos

Para agregar **agua** al terreno:

- Dentro de las texturas se encuentra el agua, para agregarla se busca como "Water" en el buscador de Unity, ver figura 4.38.

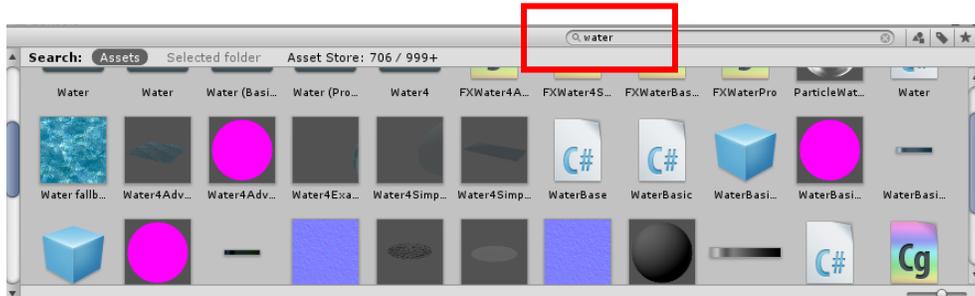


Figura 4.38. Buscando "agua" dentro del programa.

- Seleccionamos una de las texturas que aparecen, ver figura 4.39.

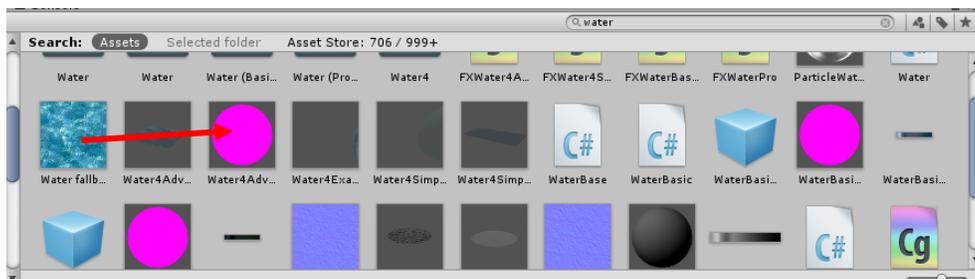


Figura 4.39. Seleccionando el agua que se va a insertar en el terreno.

- Agregamos la textura arrastrándolo hacia el terreno, ver figura 4.40.

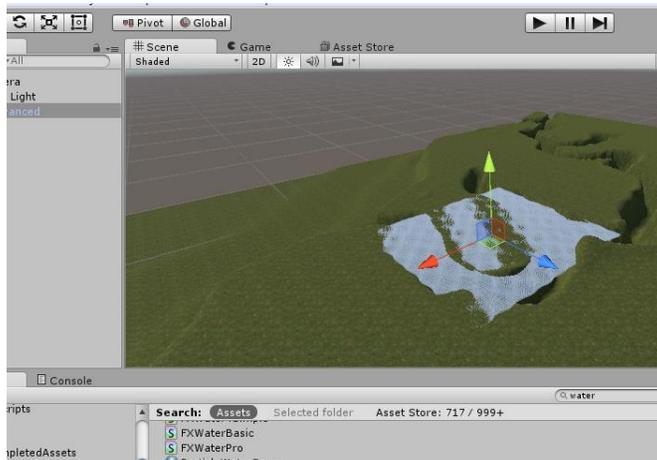


Figura 4.40. Añadiendo el agua al terreno.

- Para expandir el agua y que abarque el área deseada, se seleccionara el siguiente botón que se encuentra en la barra de navegación, ver figura 4.41.

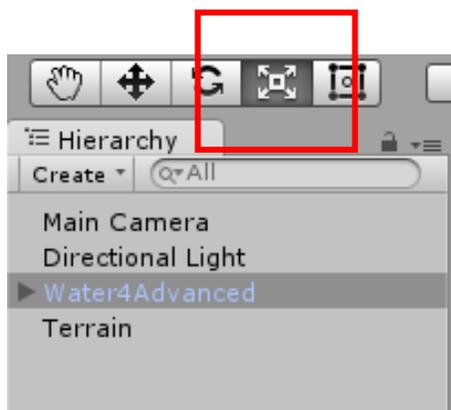


Figura 4.41. Botón para modificar el tamaño del agua en el terreno.

- Se da clic sostenido en la parte central de las flechas del plano que se muestra sobre el agua y al deslizarlo, el agua se hará más grande o más pequeño, ver figura 4.42.

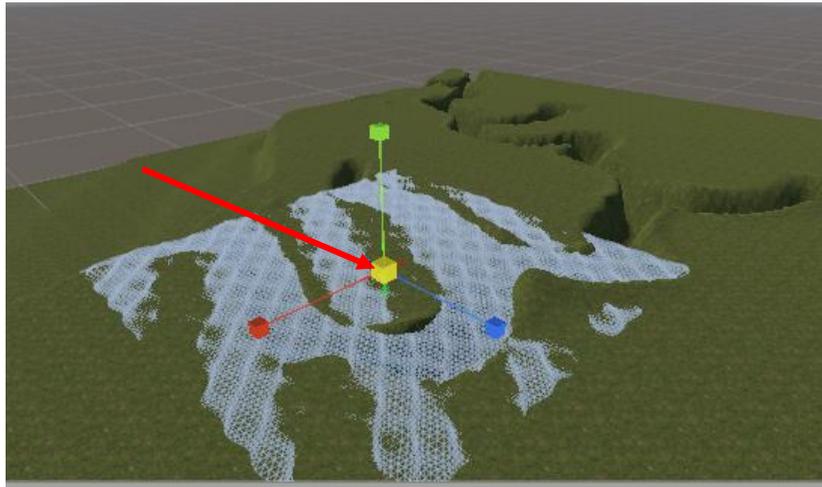


Figura 4.42. Modificando el tamaño del agua en el terreno.

- Para que el agua quede a ras de suelo o a la altura deseada, con el botón de desplazamiento moverá el agua arriba o abajo, derecha o izquierda, según sea el caso. Una vez terminado, el agua se muestra como en la siguiente figura, ver figura 4.43.

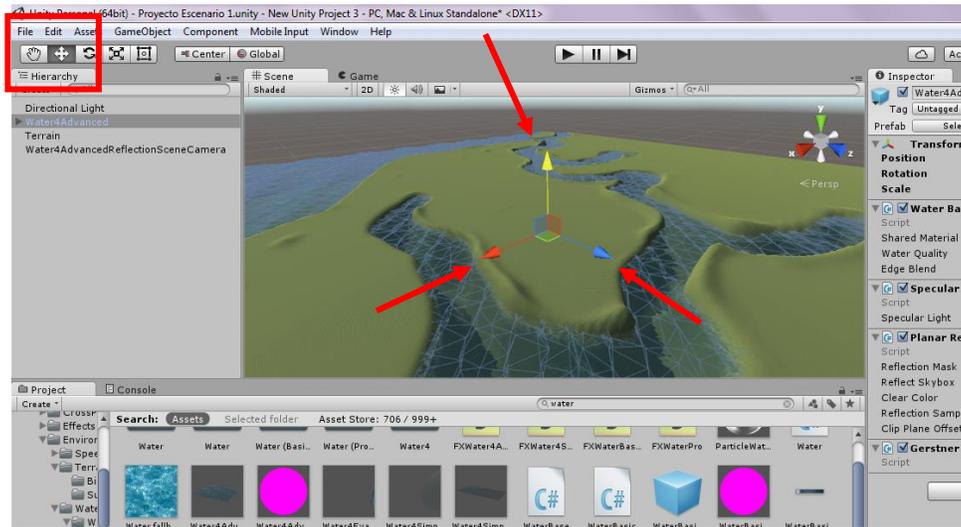


Figura 4.43. Selección de textura tipo agua.

#### 4.6.6 Creación de vegetación del terreno

Para agregar plantas, árboles, pasto, etc., al terreno:

- En el menú principal, se selecciona GameObject, después 3D Object y posterior Tree, ver figura 4.44.

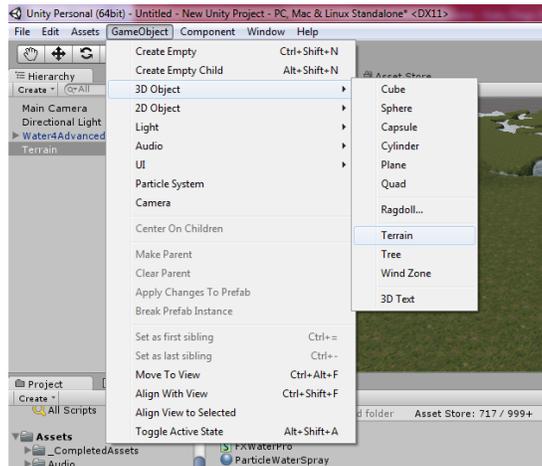


Figura 4.44. Agregando árboles al terreno.

- En el terreno solo se mostrará una especie de tronco en una de sus esquinas, esto no colocará arboles pero lo que sucede al colocar esto, es que todas las plantas que se coloquen después, tengan crecimiento y se vean afectadas por el tiempo como si fuesen una planta o un árbol en tiempo real, ver figura 4.45.



Figura 4.45. Forma del árbol añadido en el terreno.

- Para añadir los árboles, seleccionamos del lado izquierdo el objeto Terrain y en el Inspector seleccionaremos el botón con un árbol, posterior el botón Edit Trees y finalmente Add Tree, ver figura 4.46.

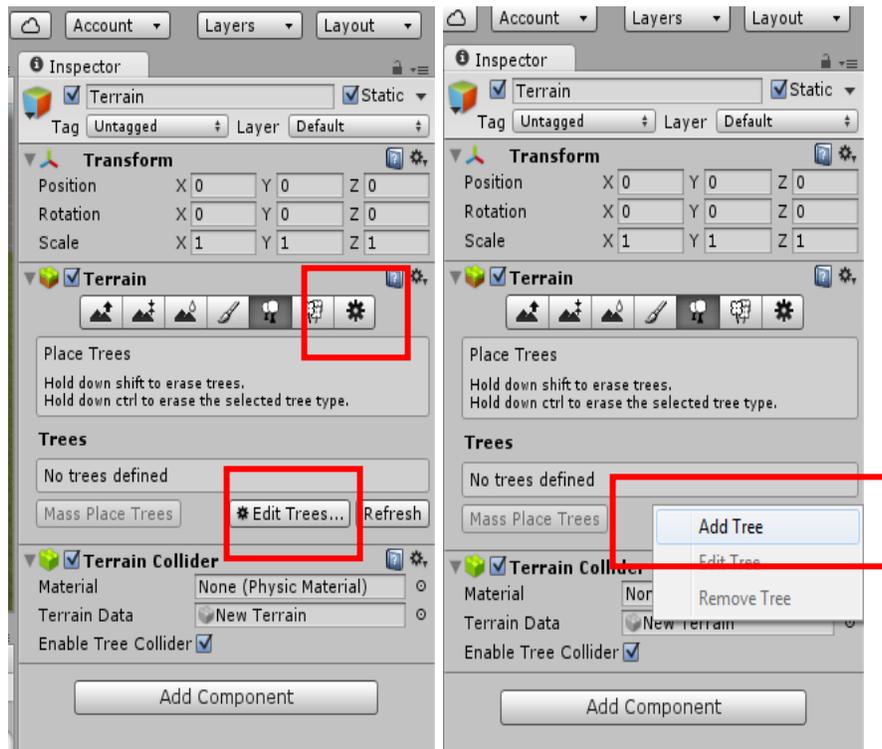


Figura 4.46. Agregando árboles al escenario.

- Se mostrará la siguiente ventana y en ella se da clic en el círculo que se encuentra en la parte posterior seguido de Tree Prefab, ver figura 4.47.

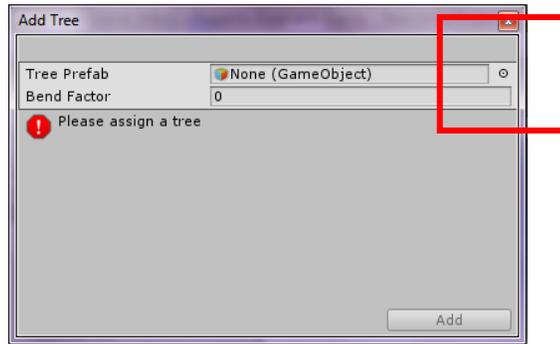


Figura 4.47. Agregando árboles al escenario II.

- Se seleccionará un árbol de la lista con doble clic, se agregará en la ventana anteriormente mostrada y para finalizar se da clic en Add, ver figura 4.48.

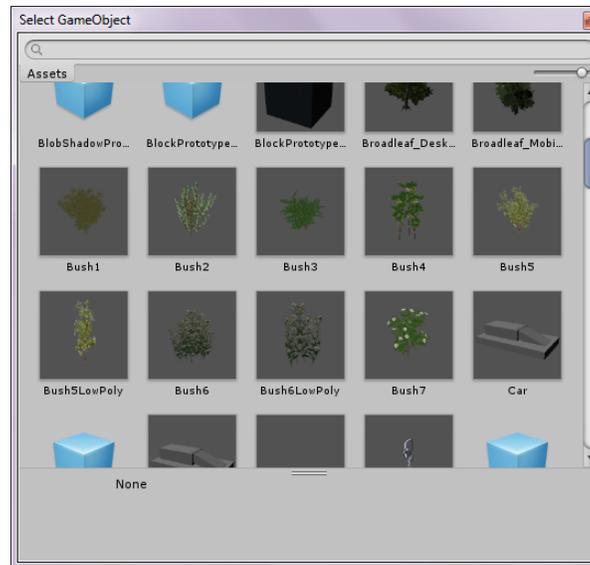


Figura 4.48. Seleccionando árbol como textura para escenario.

- En el Inspector se mostrará el árbol que se seleccionó, en el terreno se notará un círculo azul donde se posiciona el cursor del mouse, esa área es la que colocará el árbol seleccionado en el mapa al hacer clic, ver figura 4.49.

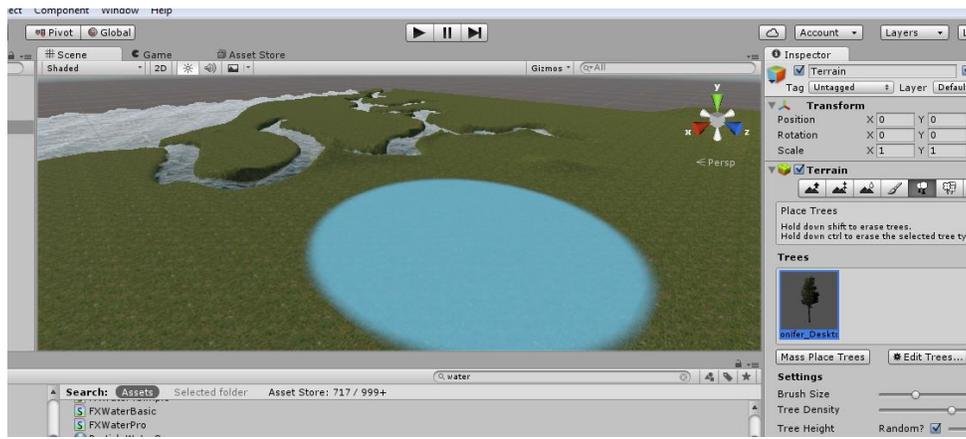


Figura 4.49. Agregando árboles al terreno.

- El tamaño del área de árboles a colocar puede modificar su tamaño, esto se hace desplazando el punto sobre la línea de Brush Size, La cantidad de árboles a colocar en Tree Density y la altura de los mismos en Tree Height, ver figura 4.50.

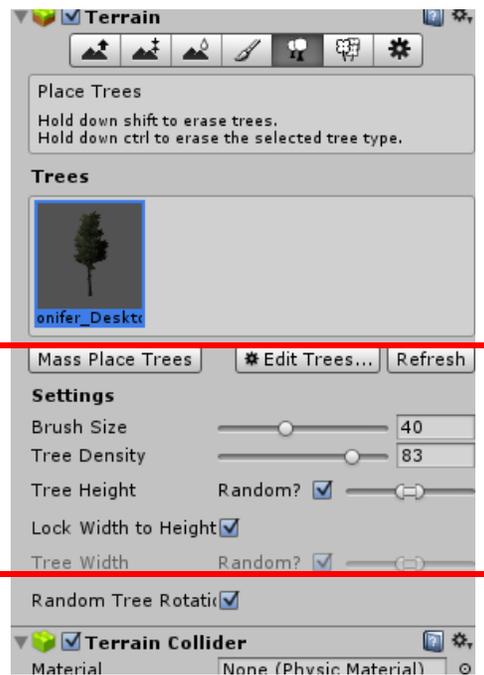


Figura 4.50. Modificando la cantidad y tamaño de brocha para agregar arboles al terreno.

- Una vez colocado los árboles en el escenario, se mostrará como la figura 4.51:

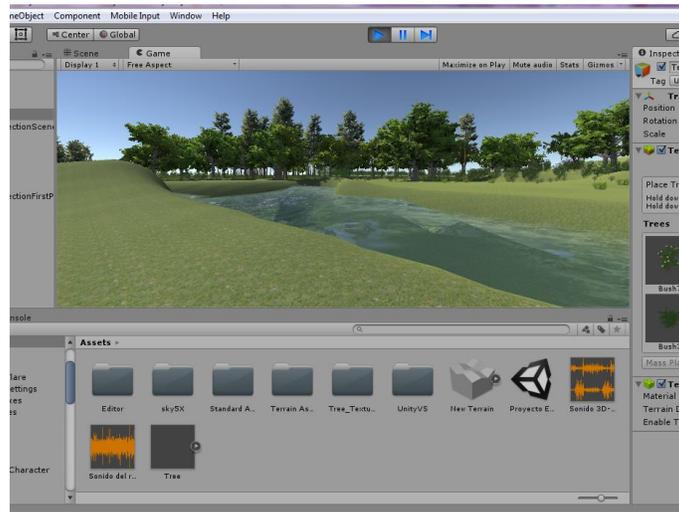


Figura 4.51. Vista del escenario con árboles en el terreno.

Para agregar pasto y/o arbustos:

- Seleccionando el objeto Terrain, en el Inspector podemos encontrar el siguiente botón, cuya función es agregar la textura de pasto y arbustos, según sea el caso, ver figura 4.52.



Figura 4.52. Botón que agregará la textura de pasto al escenario.

- Se mostrará un recuadro en la parte inferior, Details y Brushes, en el primero se muestran dos botones, Edit Details y Refresh, dará clic en Edit Details y enseguida en Add Grass Texture, ver figura 4.53.

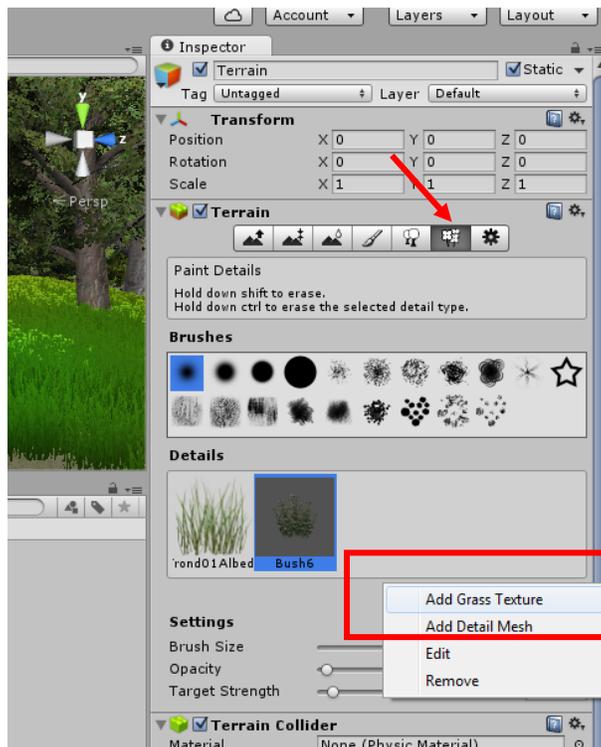


Figura 4.53. Agregando la textura pasto al escenario.

- Se mostrará la siguiente ventana, ver figura 4.54, se seleccionará el círculo que esta enseguida de Detail Texture.

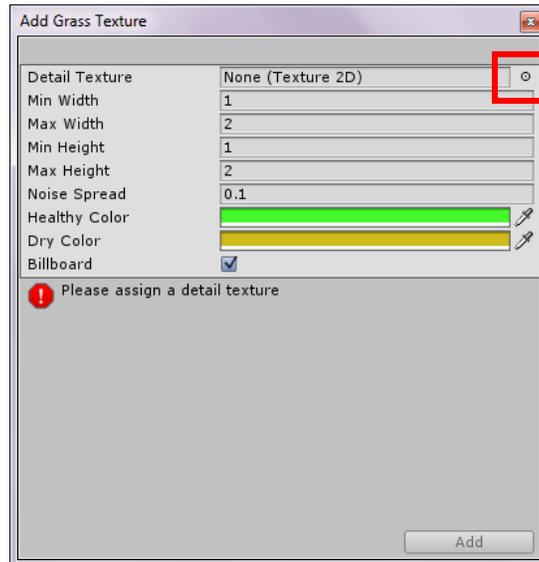


Figura 4.54. Asignando detalle de textura.

- En la siguiente ventana, se puede buscar la textura de pasto bajando con el mouse o bien escribir "grass" para que solo se muestren esas texturas y se seleccionará una de ellas, ver figura 4.55.

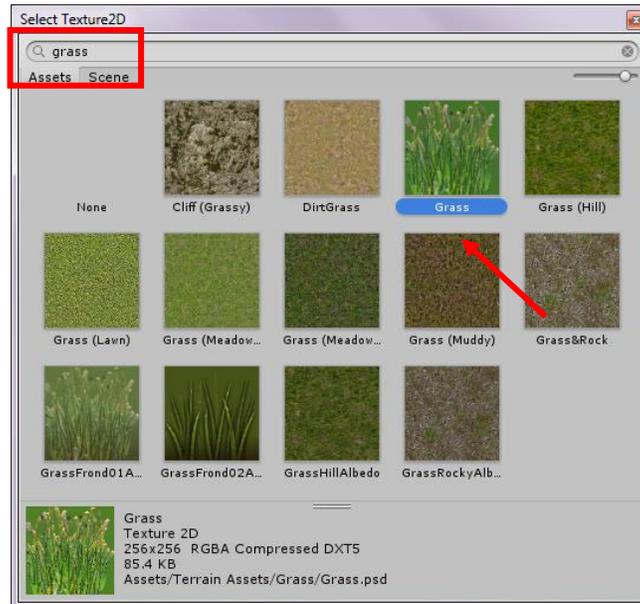


Figura 4.55. Selección de la textura pasto.

- Una vez seleccionada la textura se mostrará la siguiente, se dará clic en Add, ver figura 4.56.

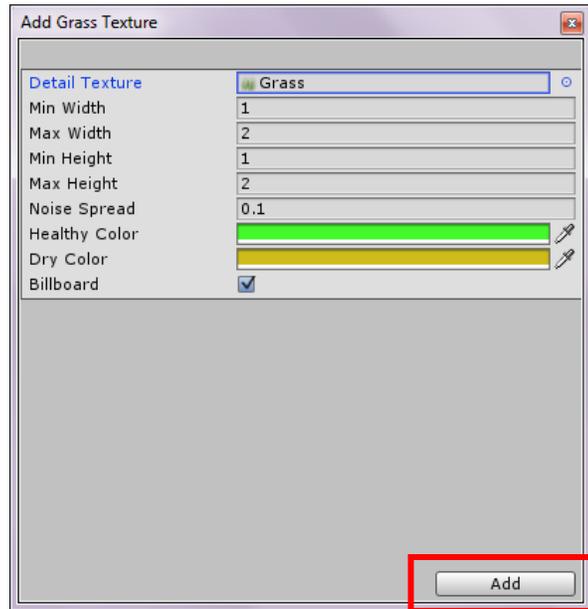


Figura 4.56. Agregando la textura de pasto al escenario.

- Se añadirá la textura que se seleccionó, el cursor cambiará a un círculo azul que indica el tamaño del área en la que se colocará el pasto. Esto se puede modificar al igual como los árboles que ya se explicó anteriormente. Una vez colocado el pasto, se visualizará de la siguiente manera, ver figura 4.57.

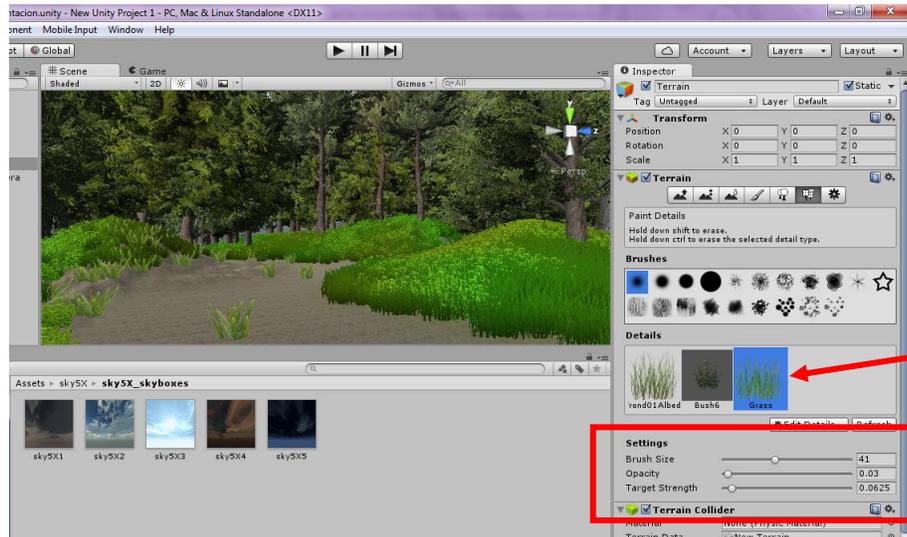


Figura 4.57. Vista del pasto en el escenario.

De igual manera, se puede agregar otro tipo de pasto, hierbas y arbustos, eso dependerá del tipo de escenario que se realice.

#### 4.6.7 Creación del cielo del terreno

Para agregar **nubes** o un **cielo** más real, Unity 3D proporciona un paquete llamado Skybox que contiene cinco tipos de cielo. Este será a elección dependiendo el diseño del escenario, en el ejemplo se manejará un cielo azul.

- En el buscador de Unity, se escribirá Skybox y dependiendo de la versión que se tenga, podrán aparecer directamente los tipos de

Skybox o bien una carpeta los contiene, en caso de ser la carpeta, se abrirá, ver figura 4.58.

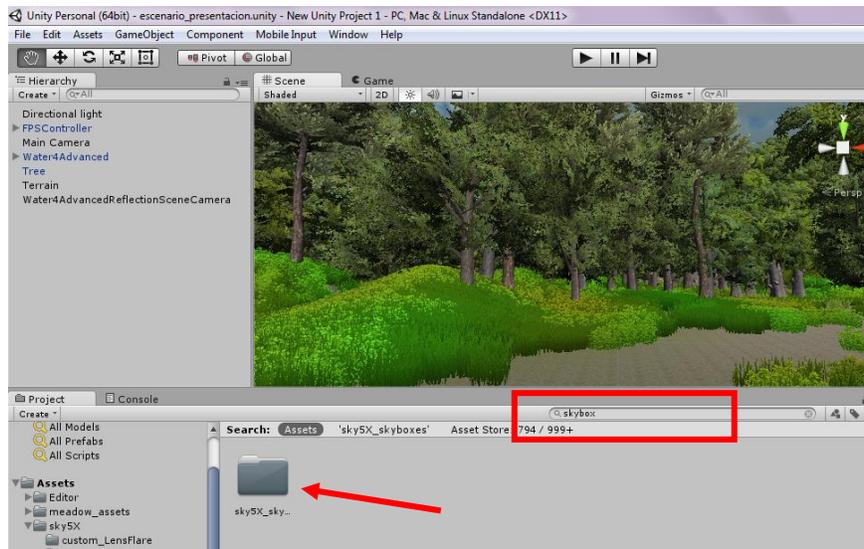


Figura 4.58. Diferentes tipos de Skybox en Unity.

- Al encontrar los Skybox proporcionados por Unity, se dará clic en uno de ellos y se arrastrará al escenario para agregarlo. Ver figuras 4.59 y 4.60.

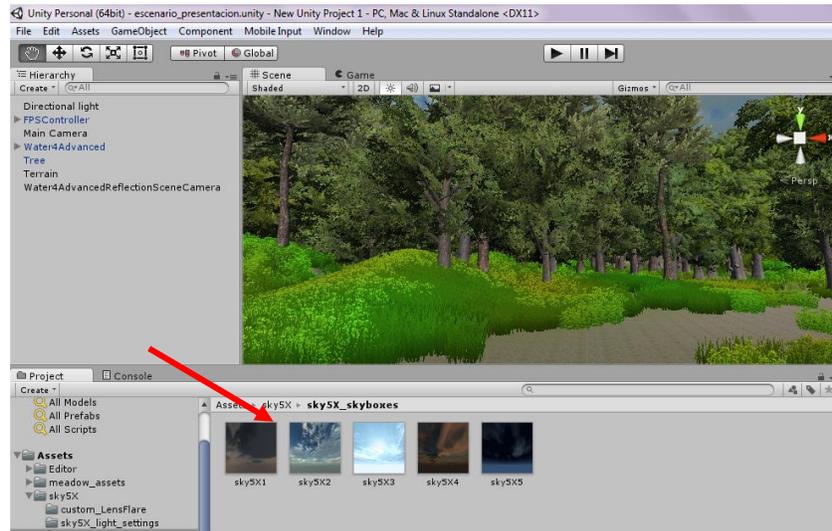


Figura 4.59. Selección de Skybox para el escenario.

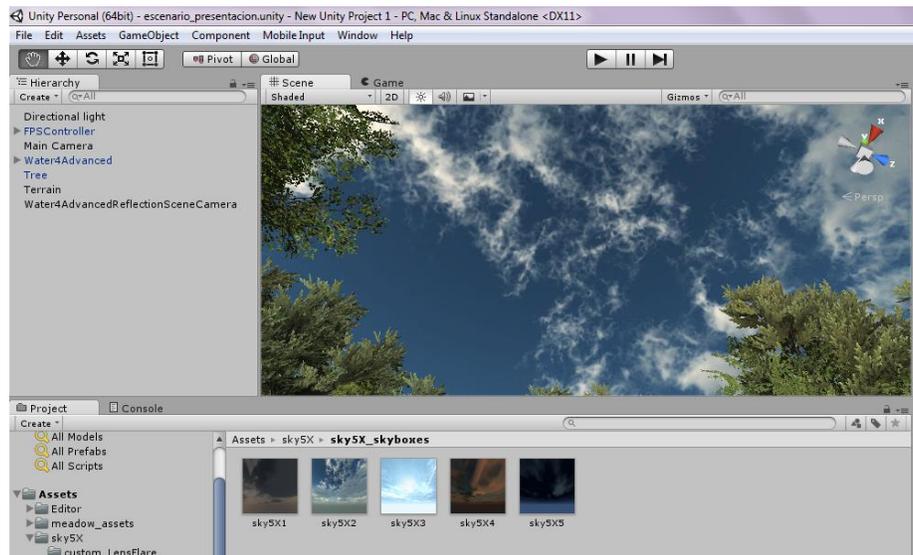


Figura 4.60. Resultado de agregar el Skybox al escenario.

## 4.6.8 Añadiendo personaje al terreno

Para visualizar el **funcionamiento del escenario** se requiere de agregar el robot o bien un personaje en primera persona. Como en este caso aún no se tiene el robot, se agregará un personaje en primera persona, brindando movilidad al escenario por medio del teclado y el mouse. Esto se hace de la siguiente manera:

- En el buscador, se escribirá “Character”, se mostrarán diversas carpetas y objetos, se seleccionará la que tenga el nombre “FirstPersonCharacter”, ver figura 4.61.

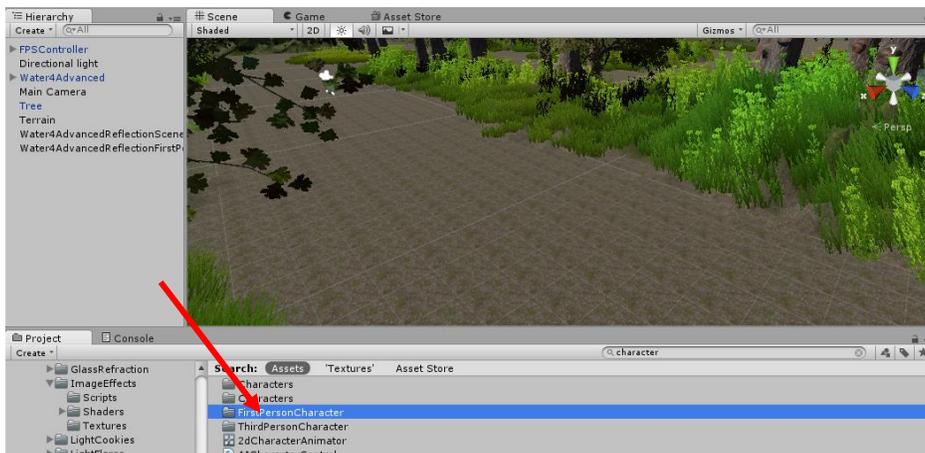


Figura 4.61. Selección de la carpeta de personaje en primera persona.

- Dentro de esa carpeta se encuentran más objetos, se seleccionará la carpeta con el nombre de Prefabs, ver figura 4.62.

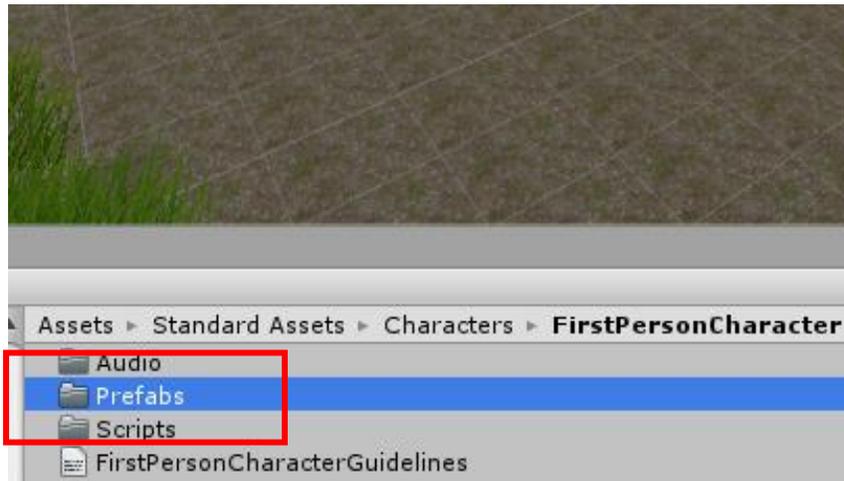


Figura 4.62. Selección de carpeta de personajes prefabricados.

- Se seleccionará el objeto con el nombre FPSControler<sup>5</sup> y se arrastrará con el mouse al escenario. Se mostrará un objeto que en este caso, es el personaje en primera persona, ver figura 4.63.

---

<sup>5</sup> Los videojuegos de disparos en primera persona, también conocidos como FPS (siglas en inglés de First-Person Shooter), son un género de videojuegos y subgénero de los videojuegos de disparos en los que el jugador observa el mundo desde la perspectiva del personaje protagonista.

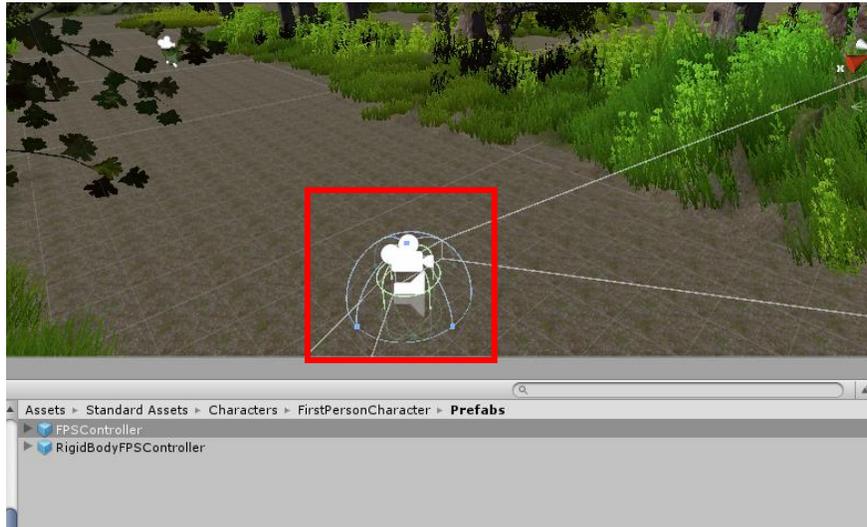


Figura 4.63. Añadiendo personaje en primera persona.

- Para no tener problema con el personaje y evitar que caiga al vacío, se debe alzar para que quede a ras de suelo en el terreno; se selecciona el personaje en el Hierachy y con el botón de desplazamiento, se acomodará a la altura requerida. Ver figura 4.64.

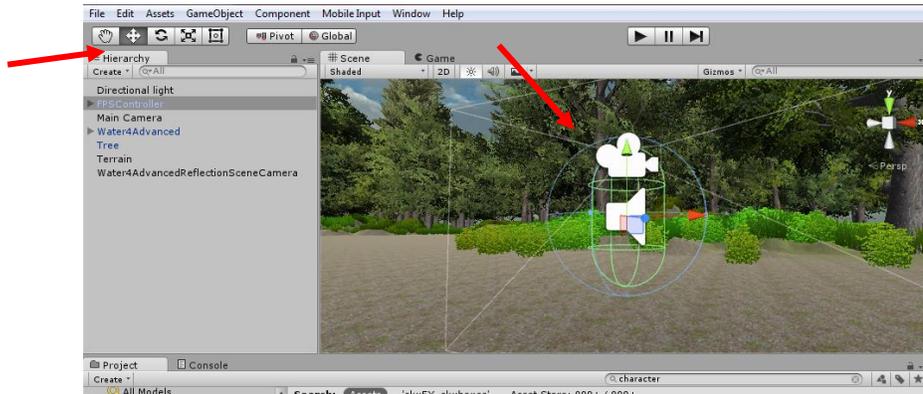


Figura 4.64. Ajustando altura del personaje en el escenario.

## 4.6.9 Ejecución del escenario

En el panel de navegación se encuentran tres botones, estos sirven para ejecutar el escenario, el primero es para dar Play al escenario, el segundo para Pausar y el tercero para cambiar de escena, este último es usado para aquellas personas que desean crear un videojuego con más de una escena, en este caso, solo se dará clic en Play, nótese que los botones pasarán de negro a color azul para indicar que está en ejecución, ver figura 4.65.

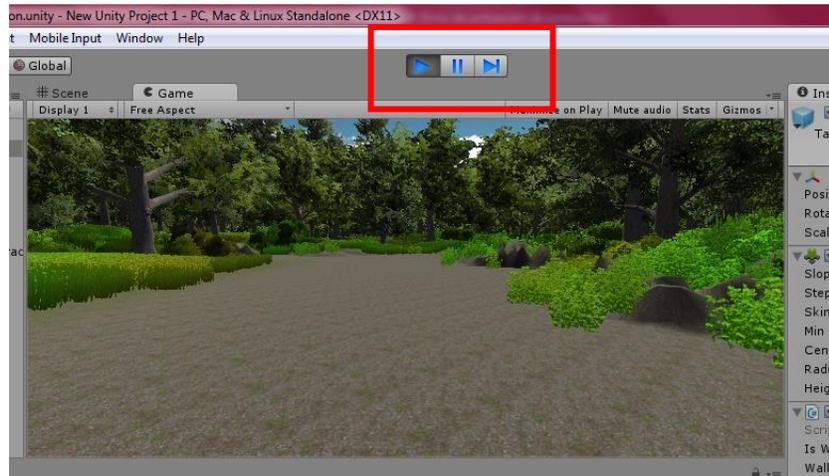


Figura 4.65. Ejecución del escenario.

La movilidad del personaje se realizará por medio del teclado y el ratón, tal y como se muestra en la figura 4.66.

<b>Control del escenario</b>	
<b>Avanzar</b>	↑
<b>Retroceder</b>	↓
<b>Izquierda</b>	←
<b>Derecha</b>	→
<b>Salto</b>	Barra espaciadora
<b>Vista o cámara</b>	Mouse o ratón

Figura 4.66. Control del escenario por medio del teclado.

Para detener la ejecución, se pulsará la tecla Esc, mostrará el cursor y se dará clic en Play pero esta vez para detener la ejecución del escenario.

#### 4.6.10 Compilación del proyecto

Para poder exportar el escenario al software educativo, se requiere compilar el proyecto, al hacer esto, generará un archivo de tipo .EXE y este será importado en el software educativo.

En la barra de tareas, se da clic en File y luego en Build & Run, ver figura 4.67.

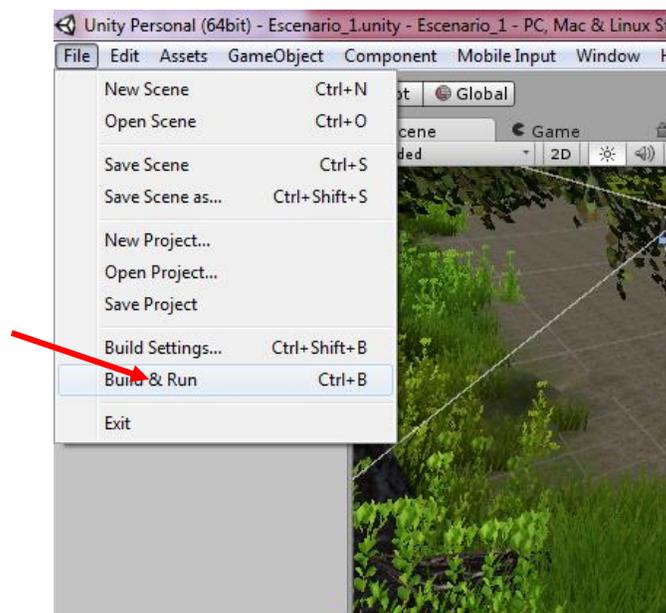


Figura 4.67. Instrucciones para llevar a cabo la compilación del proyecto.

En la siguiente ventana se seleccionará PC, Mac & Linux Standalone, Target Platform se seleccionará el tipo de sistema operativo según sea la máquina y el tipo de arquitectura, al finalizar se dará clic en Buil & Run. Esta operación puede tardar varios minutos, ver figura 4.68.

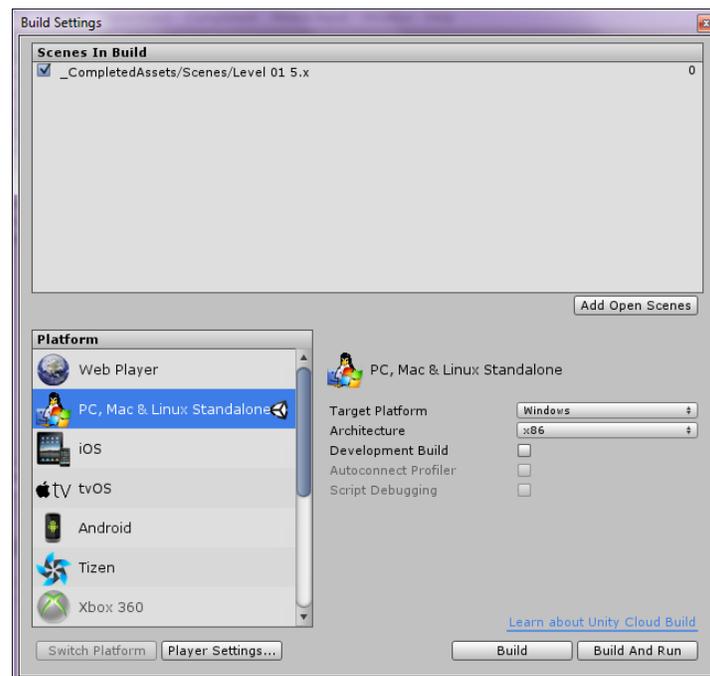


Figura 4.68. Compilando Proyecto.

Una vez finalizado, se mostrará la siguiente ventana, solo se dará clic en Play! Y el escenario se ejecutara fuera del software de Unity, en una ventana aparte, visualizándose en toda la pantalla, ver figura 4.69.

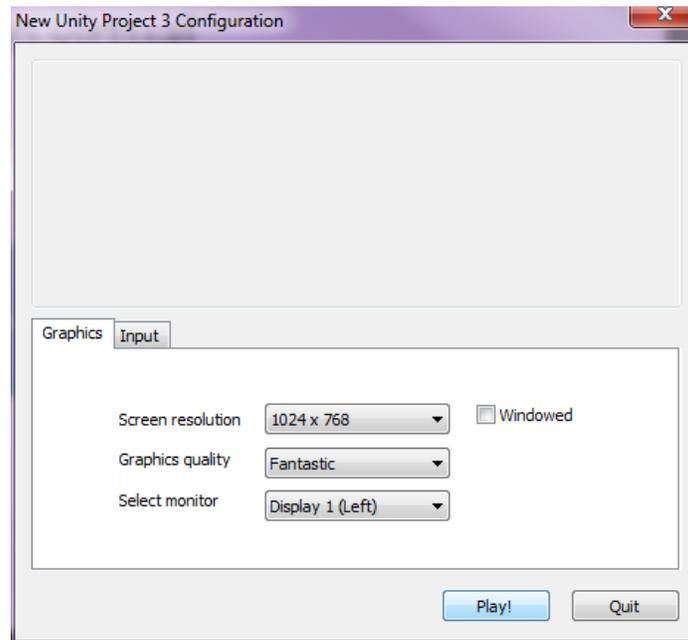


Figura 4.69. Ejecución del proyecto.

El escenario se mostrará de la siguiente manera, ver figura 4.70.



Figura 4.70. Visualización del escenario fuera de Unity 3D.

El control del escenario es igual a como se mencionó antes; para salir de la ejecución del proyecto, se teleará Alt + F4. El archivo .EXE se encuentra en la carpeta donde se guardó el proyecto y todo este será utilizado para importarlo al software educativo.

## **CAPÍTULO V**

### **RESULTADOS Y TRABAJO A FUTURO**

## **CAPITULO V: RESULTADOS**

Este desarrollo tiene como base anterior un proyecto de software educativo. Inicialmente se escoge un tipo de robot para armarse, posteriormente se elige de un catálogo uno de los escenarios en dos dimensiones “2D” que será el lugar donde el robot actuará para cumplir su objetivo.

La plataforma de diseño en la que se ejecutaría este proyecto sería Adobe Flash Player la cual, es soportada por la mayoría de los ordenadores, su lenguaje de programación es Action Script 1.0 y se realizaría con la plataforma de Flash Professional.

### **5.1 Descripción de la plataforma interactiva anterior**

La propuesta presentada para el desarrollo anterior se muestra en la figura 5.1, como se puede apreciar, la pantalla de usuario se encuentra integrada por las siguientes secciones; modelo, guía, zona de construcción y herramienta. Cada descripción de cada área se especifica de la siguiente manera:



Figura 5.1. Pantalla de usuario.

- **Título:** Contiene la información del desarrollo y las universidades que participan en el proyecto.
- **Herramientas:** En esta zona se encuentran el catálogo de piezas que pueden ser utilizadas para la construcción de un prototipo.
- **Zona de construcción:** Es el área disponible para armar el prototipo, el usuario selecciona la pieza disponible en su bloque de herramientas y la mueve a su área de construcción a través del mouse.
- **Modelo:** Dentro de esta zona se encuentra las diferentes morfologías de los robots; androide, zoomórfico, móvil, poliarticulado o híbrido, el usuario selecciona uno de los modelos ya preestablecidos para iniciar con su construcción en la zona correspondiente.

- **Guía:** En esta zona aparece paso a paso de manera gráfica, el inicio y fin del proyecto, el estudiante utiliza esta guía para la elaboración y ensamblado de su robot.

En la figura 5.2 se muestra el aspecto de la aplicación ya en ejecución.



Figura 5.2. Presentación del proyecto.

Al finalizar el ensamblado del prototipo, la aplicación lo lleva de manera inmediata a la ventana de escenarios virtuales e interactivos, el cual es el medio utilizado por el usuario para manipular en un entorno virtual la utilización y aplicación de los robots, dentro de este bloque se han predispuesto las siguientes secciones: escenario virtual, escenario, modelo y control.

En la figura 5.3, se muestra la estructura o plataforma interactiva del área de trabajo así como del escenario donde interactúa el robot dentro del proyecto.

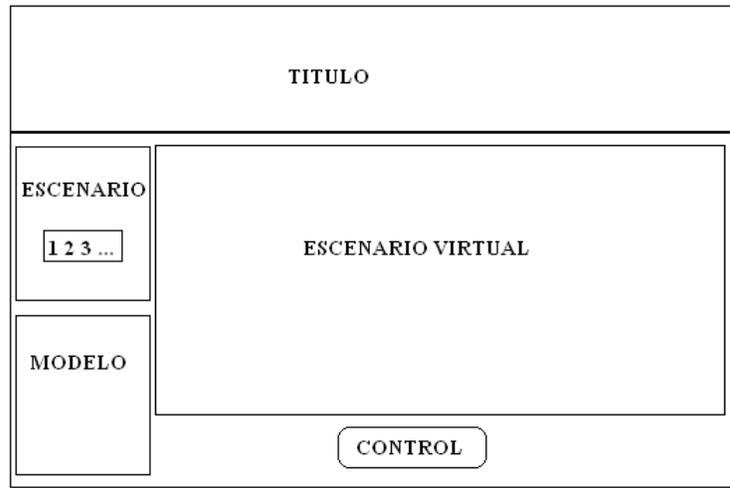


Figura 5.3. Estructura del escenario.

La descripción de cada área es la siguiente:

- **Título:** Contiene la información del desarrollo y las universidades que participan en el proyecto.
- **Escenario virtual:** Muestra el ambiente virtual donde trabajará el robot para cumplir su objetivo.
- **Control:** Esta área no diseñada en este proyecto se utilizará para realizar una programación lineal de secuencias establecidas por el usuario que dependen de la tarea que debe realizar el robot” trabajo futuro”

- **Escenario:** En esta área, el estudiante podrá elegir los escenarios disponibles para su robot, estos dependerán del campo de aplicación teniendo un ambiente lo mayor realista posible.
- **Modelo:** Se muestra el modelo de robot que se creó y que posteriormente será ingresado al escenario elegido.

Nota: El modelo de robot puede ser; androide, zoomórfico, móvil, poliarticulado o híbrido.

En la figura 5.4 se muestra el desarrollo realizado, se observa que la imagen del escenario es propiamente una foto o una imagen sin movimiento.



Figura 5.4. Idea de software educativo trabajado con escenarios en 2D.

Tomando en cuenta la figura anterior, considérese lo siguiente:

- **Escenario:** Dentro de esta zona el estudiante puede escoger diversos ambientes virtuales o escenarios diseñados para recrear el entorno real al campo de aplicación en la que puede trabajar el robot.
- **Audio:** El audio tiene la función de brindar apoyo auditivo para el usuario, ya sea para dar los objetivos del escenario o bien para desactivarlo.
- **Video:** Brinda el ejemplo de lo que el usuario debe realizar dentro del escenario con el robot.
- **Animación:** Permite al usuario realizar la ejecución de su robot hasta donde se haya trabajado por medio de fotogramas.
- **Modelo:** Muestra los diferentes modelos de robot con los cuales se puede trabajar dentro del escenario.
- **Control:** Esta área no diseñada en este proyecto se utilizará para realizar una programación lineal de secuencias establecidas por el usuario que dependen de la tarea que debe realizar el robot.

## 5.2 Descripción de la plataforma interactiva actual

El proyecto actual, está realizado con la plataforma de Unity 3D ya que el software que se utilizaría en el desarrollo anterior actualmente se puede considerar obsoleto, de igual manera, la forma de trabajo es

similar porque cuenta con elección de tipo de robot, armado y posterior a este, trabajará con el escenario pero ahora en tercera dimensión “3D”.

El trabajo es realizado por la institución Universidad Americana de Acapulco aunado con la UAM-Azcapotzalco por ser antecedente a este proyecto, ver figura 5.5.



Figura 5.5. Logos de las instituciones.

La ventana inicial del proyecto actual, muestra los tipos de robot disponibles y una vista previa del mismo el cual se puede rotar para ver el diseño completo, ver figura 5.6.



Figura 5.6. Pantalla inicial del software.

Habr  tres robots disponibles por cada tipo de robot, el usuario elegir  el que desee armar y dar  clic en el bot n de  rea de armado para pasar a la siguiente pantalla, una vez cargada se mostrar  lo siguiente, ver figura 5.7.



Figura 5.7. Pantalla de armado de robot.

La descripción de cada área es la siguiente:

- El **modelo** mostrará el modelo de robot que se eligió armar.
- La **guía** mostrará paso a paso de manera gráfica, el inicio y fin del proyecto, el estudiante utiliza esta guía para la elaboración y ensamblado de su robot.
- **Siguiente paso** mostrará las herramientas que siguen de acuerdo al orden de armado del robot.
- Las **herramientas**, son las piezas que se deben utilizar para el armado de robots.

Tiene la opción de rotar la pantalla para que el armado sea de manera sencilla y se podrá cambiar el color de pantalla para visualizar mejor las piezas. Una vez terminado de armar, pasará el robot al escenario.

La ventana que trabaja con los escenarios, cuenta con la misma estructura que el software anterior, solo cambió el escenario quedando de la siguiente manera, ver figura 5.8.



Figura 5.8. Pantalla del escenario virtual en 3D.

La descripción de cada área es la siguiente:

- El **escenario** mostrará un catálogo de escenarios disponibles por tipo de robot. Se desea ilustrar como pequeñas imágenes para que el usuario tenga una idea sobre qué campo de aplicación puede trabajar su robot, ver figura 5.9.



## *ESCENARIOS*

Figura 5.9. Visualización ejemplo del catálogo de escenarios.

- El **modelo** mostrará el tipo de robot que estará trabajando en el escenario.
- El área de **control** es aquella que brindará el manejo al robot pero que al igual que en software anterior, no está diseñada aún porque se realizará programación lineal de secuencias establecidas por el usuario, por lo tanto y de manera temporal funciona por medio del teclado y mouse del ordenador, ver figura 5.10.

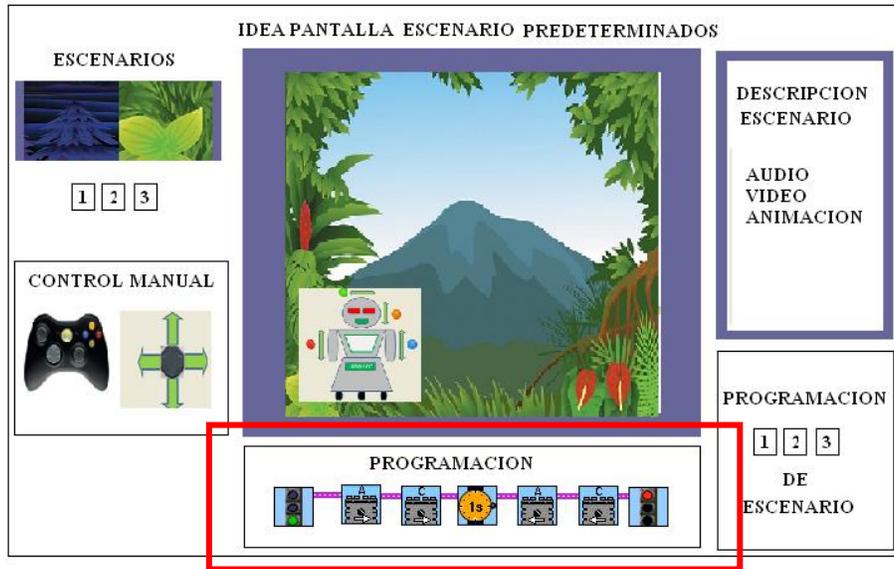


Figura 5.10. Idea de programación por bloques para robot.

El robot podrá desplazarse a través del escenario siendo este un personaje en tercera persona, tendrá movimiento similar a un robot físico y sonido de movimiento al igual que en el del escenario para brindar mayor realismo.

## **CAPÍTULO VI**

### **CONCLUSIÓN Y TRABAJO A FUTURO**

## **CAPÍTULO VI: CONCLUSIÓN Y TRABAJO A FUTURO**

El desarrollo de este trabajo cumple con los objetivos para los que fue diseñado, es más sencillo y fácil de entender, se aprenden conceptos básicos de robótica, su campo de aplicación y se impulsa la creatividad para diseñar. Es útil tanto para profesores como alumnos y es un buen complemento didáctico porque su contenido es de manera interactiva y no se requiere de manejar utilería costosa para aprender la materia.

Para trabajo a futuro se pretende que la programación hecha por el usuario para su robot, no solo quede de manera virtual sino que pueda descargarse a través de vía USB a un robot físico con apariencia similar o igual al que se diseñó que se creó en el software. Esta técnica es similar a otros programas como es el caso de Lego, pero la ventaja de nuestro proyecto es que además de trabajar la programación de manera virtual, se pueda utilizar también en un entorno real.

Con todo esto creado, se implementaría un proyecto más completo y seria de mucha utilidad para implementar la materia de programación de robots en las instituciones de México.

## **BIBLIOGRAFÍA**

## BIBLIOGRAFÍA

(2009). Obtenido de <http://escenariovirtual3d.blogspot.mx/2009/06/ques-es-escenario-virtual.html>

Autodesk. (2016). *Tinkercad*. Obtenido de <https://www.tinkercad.com/>

Autodesk Inc. (2016). *3DS MAX*. Obtenido de <http://www.autodesk.mx/products/3ds-max/overview>

Autodesk Inc. (2016). *Maya*. Obtenido de <http://www.autodesk.mx/products/maya/overview>

Blender. (2016). *Blender*. Obtenido de <https://www.blender.org/>

Cabri. (2009). *Cabri*. Obtenido de <http://www.cabri.com/es/cabri-3d.html>

*GeoEnciclopedia*. (s.f.). Obtenido de <http://www.geoenciclopedia.com/marte/>

Maxinez, D. G. (2012). Educational Robotics Constructivism and Modeling of Robots Using Reverse Engineering. *International Journal of Science and Engineering Investigations*, 2.

Technologies, S. (2016). *Smart Technologies*. Obtenido de <https://education.smarttech.com/es-es/products/notebook>

Technologies, U. (2016). *Unity Game Engine*. Obtenido de <https://unity3d.com/es>

Wikimedia Foundation Inc. (2015). Obtenido de <https://en.wikipedia.org/wiki/Heightmap>

Wikimedia Foundation Inc. (2016). *Wikipedia*. Obtenido de  
[https://es.wikipedia.org/wiki/Realidad\\_virtual](https://es.wikipedia.org/wiki/Realidad_virtual)

Winplot. (2011). *Winplot*. Obtenido de  
<http://math.exeter.edu/rparris/winplot.html>

## ÍNDICE DE FIGURAS

Figura 2.1. Realidad virtual aplicada en el área de medicina.....	9
Figura 2.2. Ejemplo de escenario virtual que muestra la interacción de robots en una fábrica. ....	10
Figura 2.3. Uso del software de SMART en el área de medicina.....	11
Figura 3.1. Robot poliarticulado trabajando para una ensambladora de coches. ....	17
Figura 3.2. Robot hibrido utilizado para la exploración de Marte. ....	19
Figura 3.3. Escenario donde se visualiza el suelo del planeta Marte....	20
Figura 3.4. Robot androide. ....	22
Figura 3.5. Robot zoomórfico.....	25
Figura 3.6. Robot móvil utilizado por la NASA para exploración de Marte.....	27
Figura 3.7. Robot poli articulado utilizado por empresas. ....	29
Figura 3.8. Robot hibrido utilizado para realizar diversas acciones. ....	30
Figura 4.1. Uso de 3D Studio Max para la creación de escenarios. ....	35
Figura 4.2. Uso de Blender para modelar y animar gráficos. ....	36
Figura 4.3. Uso de Autodesk Maya para la elaboración de animaciones y efectos especiales.....	37
Figura 4.4. Diseño de escenas para videojuegos en Unity 3D. ....	38
Figura 4.5. Software Unity y sus campos de aplicación.....	40
Figura 4.6. Tipos de licencia para Unity 3D. ....	42

Figura 4.7. Elección de licencia en Unity. ....	46
Figura 4.8. Descargando Unity Personal. ....	47
Figura 4.9. Iniciando con Unity.....	47
Figura 4.10. Pantalla principal de Unity. ....	48
Figura 4.11. Seleccionando el tipo de proyecto en el que se trabajará. ....	49
Figura 4.12. Seleccionando todos los paquetes del software Unity. ....	50
Figura 4.13. Pantalla principal del software Unity. ....	51
Figura 4.14. Menú principal del software Unity. ....	51
Figura 4.15. Barra de navegación del software Unity. ....	52
Figura 4.16. Área de trabajo del software Unity.....	52
Figura 4.17. Creando terreno para escenario. ....	53
Figura 4.18. Cambiando tamaño de terreno. ....	54
Figura 4.19. Modificando la resolución del escenario. ....	55
Figura 4.20. Vista del terreno.....	56
Figura 4.21. Añadiendo textura al terreno.....	57
Figura 4.22. Selección de textura. ....	58
Figura 4.23. Añadiendo textura al terreno.....	59
Figura 4.24. Añadiendo textura al terreno.....	60
Figura 4.25. Vista del terreno con textura agregada. ....	60
Figura 4.26. Diseño de heightmap descargado de internet. ....	62
Figura 4.27. Ejemplo de anchura y altura. ....	63
Figura 4.28. Medidas del documento para heightmap en Photoshop. ..	64

Figura 4.29. Diseño de heightmap en Photoshop. ....	65
Figura 4.30. Guardando heightmap en Photoshop. ....	66
Figura 4.31. Guardando imagen con extensión .RAW.....	67
Figura 4.32. Importando heightmap a Unity.....	68
Figura 4.33. Importando el archivo .RAW. ....	68
Figura 4.34. Importando el archivo .RAW II. ....	69
Figura 4.35. Modificando la altura del terreno.....	70
Figura 4.36. Visualización del terreno con el heightmap importado.....	70
Figura 4.37. Botones que sirven para modificar las elevaciones del terreno. ....	71
Figura 4.38. Buscando "agua" dentro del programa. ....	72
Figura 4.39. Seleccionando el agua que se va a insertar en el terreno.	72
Figura 4.40. Añadiendo el agua al terreno.....	73
Figura 4.41. Botón para modificar el tamaño del agua en el terreno. ...	73
Figura 4.42. Modificando el tamaño del agua en el terreno. ....	74
Figura 4.43. Selección de textura tipo agua.....	75
Figura 4.44. Agregando árboles al terreno. ....	76
Figura 4.45. Forma del árbol añadido en el terreno.....	77
Figura 4.46. Agregando árboles al escenario. ....	78
Figura 4.47. Agregando árboles al escenario II. ....	79
Figura 4.48. Seleccionando árbol como textura para escenario. ....	79
Figura 4.49. Agregando árboles al terreno. ....	80

Figura 4.50. Modificando la cantidad y tamaño de brocha para agregar arboles al terreno. ....	81
Figura 4.51. Vista del escenario con árboles en el terreno. ....	82
Figura 4.52. Botón que agregará la textura de pasto al escenario.....	82
Figura 4.53. Agregando la textura pasto al escenario.....	83
Figura 4.54. Asignando detalle de textura. ....	84
Figura 4.55. Selección de la textura pasto.....	85
Figura 4.56. Agregando la textura de pasto al escenario.....	86
Figura 4.57. Vista del pasto en el escenario. ....	87
Figura 4.58. Diferentes tipos de Skybox en Unity. ....	88
Figura 4.59. Selección de Skybox para el escenario. ....	89
Figura 4.60. Resultado de agregar el Skybox al escenario.....	89
Figura 4.61. Selección de la carpeta de personaje en primera persona. .....	90
Figura 4.62. Selección de carpeta de personajes prefabricados. ....	91
Figura 4.63. Añadiendo personaje en primera persona. ....	92
Figura 4.64. Ajustando altura del personaje en el escenario. ....	93
Figura 4.65. Ejecución del escenario.....	94
Figura 4.66. Control del escenario por medio del teclado.....	95
Figura 4.67. Instrucciones para llevar a cabo la compilación del proyecto. ....	96
Figura 4.68. Compilando Proyecto.....	97
Figura 4.69. Ejecución del proyecto.....	98

Figura 4.70. Visualización del escenario fuera de Unity 3D.....	99
Figura 5.1. Pantalla de usuario.....	102
Figura 5.2. Presentación del proyecto.....	103
Figura 5.3. Estructura del escenario.....	104
Figura 5.4. Idea de software educativo trabajado con escenarios en 2D. .....	105
Figura 5.5. Logos de las instituciones.....	107
Figura 5.6. Pantalla inicial del software.....	108
Figura 5.7. Pantalla de armado de robot.....	109
Figura 5.8. Pantalla del escenario virtual en 3D.....	110
Figura 5.9. Visualización ejemplo del catálogo de escenarios.....	111
Figura 5.10. Idea de programación por bloques para robot.....	112
Figura 7.1. Página Oficial de descarga de XnConvert.....	126
Figura 7.2. Ventana principal de XnConvert.....	127
Figura 7.3. Agregando archivo a convertir en XnConvert.....	128
Figura 7.4. Cambiando configuración de salida del programa XnConvert.....	129
Figura 7.5. Detalles de conversión de archivo en XnConvert.....	129
Figura 7.6. Archivo convertido con XnConvert.....	130

## ÍNDICE DE TABLAS

Tabla 2.1. Ejemplos de software en 3D aplicados en diferentes áreas de estudio. ....	13
Tabla 2.2. Descripción de cada tipo de robots existentes. ....	14
Tabla 4.1. Ventajas y desventajas de software dedicado a animación en 3D. ....	39
Tabla 4.2. Tipos de licencia para Unity 3D. ....	45

## APÉNDICE A

## **APÉNDICE A**

### **Conversor de Imagen por medio de XnConvert**

Para poder utilizar un heighthmap de internet, es necesario descargar la imagen que desee utilizar.

Una vez descargada la imagen es necesario realizar la conversión del tipo archivo imagen .JPG .JPEG a .RAW para esto se requiere descargar el software XnConvert, el cual, se puede hacer por medio de su página oficial que es la siguiente:

<http://www.xnview.com/en/xnconvert/>

Para descargar el software, en su sitio web se desplaza la página hacia abajo hasta encontrar la sección de descarga y se dará clic según el sistema operativo de su máquina, ver figura 7.1.

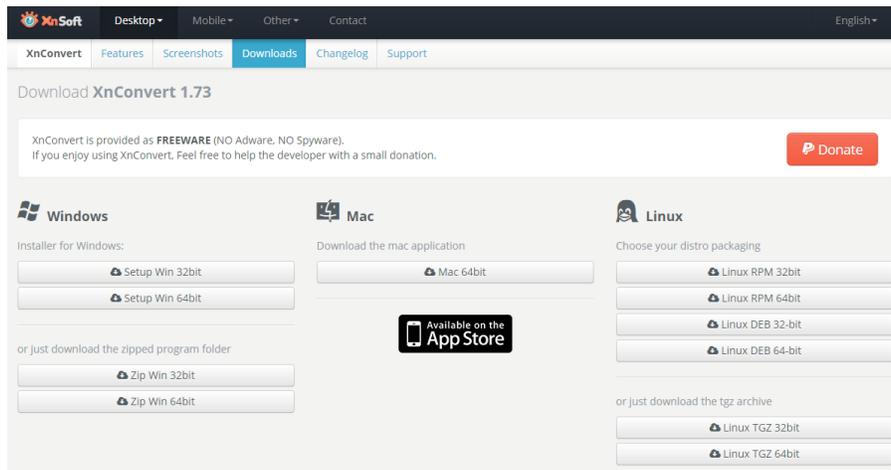


Figura 7.1. Página Oficial de descarga de XnConvert.

Una vez descargado el archivo, se instalará en su ordenador siguiendo las indicaciones del software.

Al ejecutar XnConvert se mostrará la siguiente ventana, ver figura 7.2.

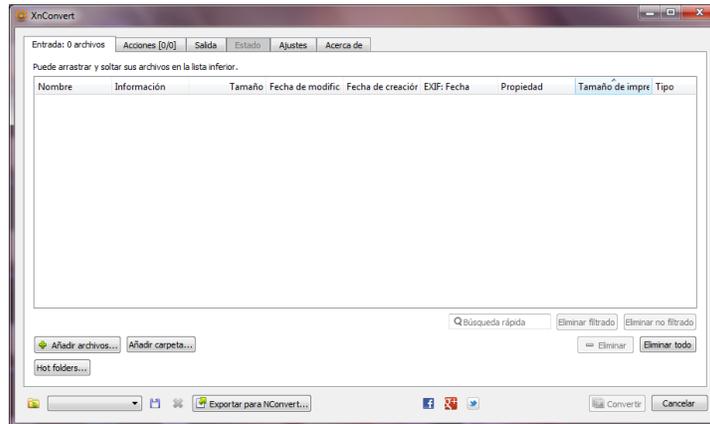


Figura 7.2. Ventana principal de XnConvert.

Para comenzar a convertir el archivo .JPG o .JPEG a .RAW es necesario saber dónde se encuentra su archivo, para este ejemplo, se tomará en cuenta que el archivo está en el escritorio del ordenador.

Una vez localizado el archivo, éste se arrastrará con clic sostenido a la ventana del programa y se agregará automáticamente, ver figura 7.3.

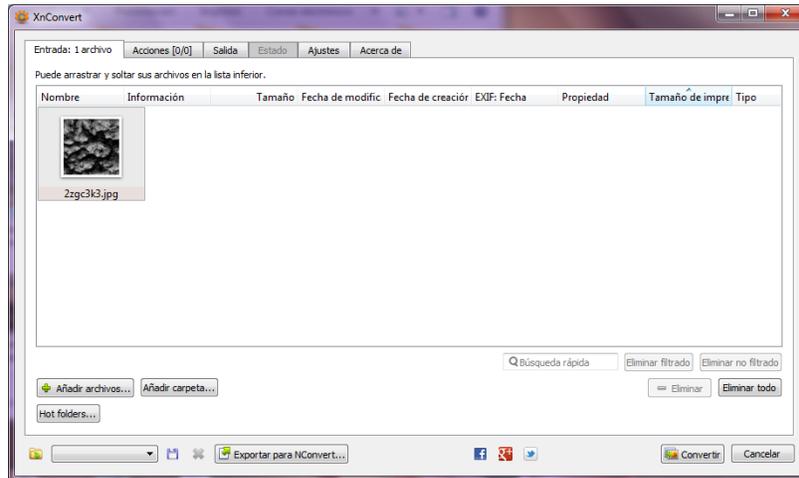


Figura 7.3. Agregando archivo a convertir en XnConvert.

Para que el programa realice el cambio es necesario cambiar la configuración de salida del mismo, esto se hace dando clic en el botón Salida, se seleccionará la carpeta o lugar de destino donde se guardará el archivo y se cambiará el tipo de formato al que se desea convertir y se da clic en Convertir, ver figura 7.4.

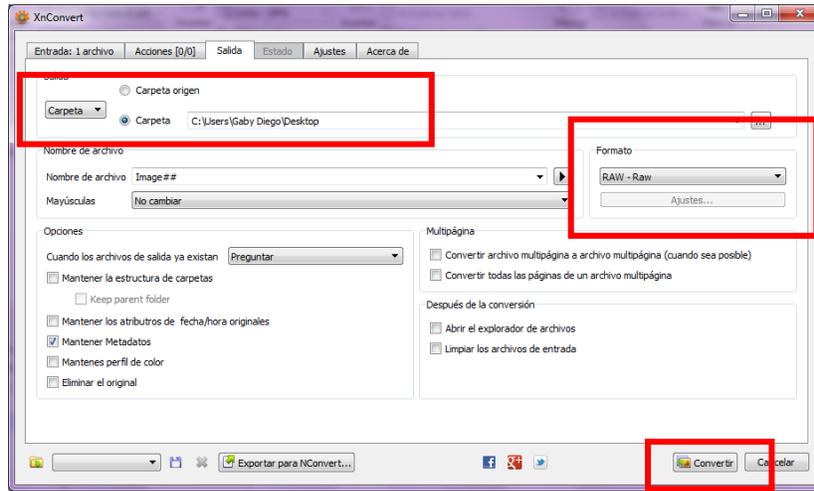


Figura 7.4. Cambiando configuración de salida del programa XnConvert.

Se mostrará la ventana siguiente, figura 7.5:

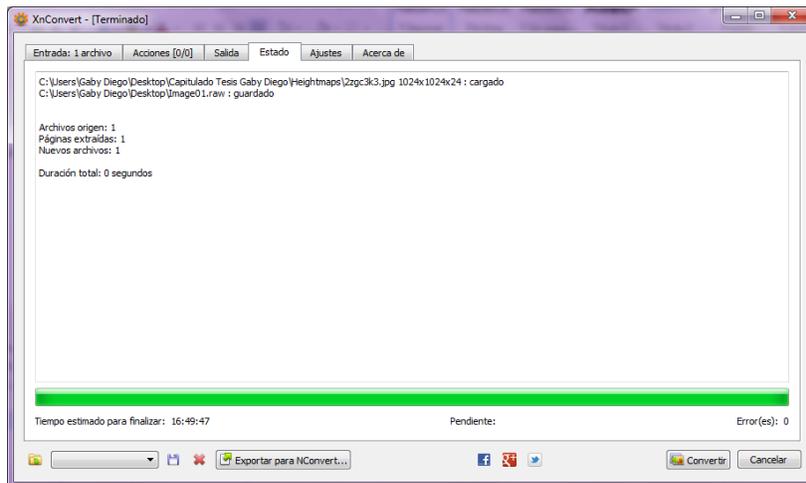


Figura 7.5. Detalles de conversión de archivo en XnConvert.

El archivo nuevo que se ha creado, aparecerá en el escritorio con el nombre de Image01 o Imagen#, según sea el caso del número de archivo que se haya convertido, ver figura 7.6.

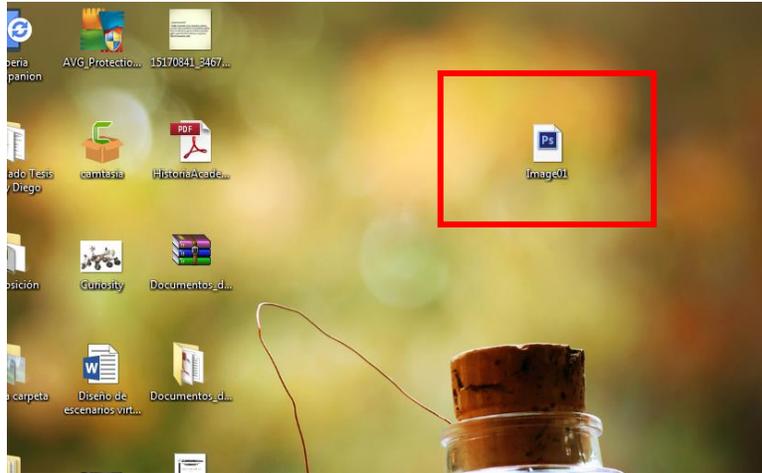


Figura 7.6. Archivo convertido con XnConvert.

Una vez convertido el archivo en .RAW este ya podrá ser utilizado en Unity 3D.