



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

# Diseño y desarrollo de una colección NFT a través de sistemas procedurales 3D

**Autor:**

Pablo Ecay Ripa

**Director:**

Sergio García el Hachimi

**Ponente:**

Dra. Eva Cerezo Bagdasari

Ingeniería en Diseño Industrial y Desarrollo de Producto  
Escuela de ingeniería y Arquitectura  
2021/2022

# Resumen

El proyecto consiste en la generación procedural de una colección de viviendas que será incorporada en el proyecto interno de la empresa Imascono llamado “Spaceship Project”. El resultado final está formado por una serie de imágenes de viviendas que servirán para ilustrar una colección NFT. En un futuro, los portadores de estas imágenes podrán acceder a un espacio virtual de la vivienda que contienen.

Para desarrollar este proyecto he diseñado un sistema de generación de la vivienda procedural. Este desarrollo se basa en la herramienta *Geometry Nodes* de Blender, encargada de seleccionar de forma aleatoria que objetos aparecen en la escena y sus posiciones. También he utilizado un *script* basado en Python para Blender que se encarga de automatizar el proceso de crear la colección de imágenes NFT de las viviendas. Además, genera un archivo de texto por cada imagen donde se describen los objetos que se encuentran en ella.

Una vez separada la vivienda en distintos módulos (Salón, dormitorio, teletransporte y laboratorio) y realizada una lista con todos los objetos y variantes de objetos que debía crear, comenzó el proceso de Modelar y texturizar todos los componentes. En este proceso se ha reducido la cantidad de los polígonos de los objetos lo máximo posible para facilitar la creación posterior del espacio virtual.

Se ha trabajado en la cohesión de todos los elementos, luces, variedad de colores y unidad en los objetos con el fin de conseguir una gran variedad de viviendas, pero siempre manteniendo un sentido estético y diferenciado entre ellas.

Por último, se ha creado la colección de imágenes de viviendas con sus respectivos archivos de texto describiendo los componentes que contienen.

# Índice

<b>1 Introducción</b>	4
1.1 Objeto y Alcance	5
1.2 Contexto	6
1.3 Metodología	7
1.4 Planificación	9
<b>2 Ideación</b>	10
2.1 Planteamiento de Módulos	11
2.2 Exploración Formal	12
2.3 Distribución de Módulos	13
2.4 Listado de Objetos	14
<b>3 Generación Procedural</b>	15
3.1 Geometry Nodes	16
3.2 Programación	17
<b>4 Desarrollo</b>	18
4.1 Metodología	19
4.2 Modelado	20
4.3 Texturizado	22
<b>5 Resultados</b>	27
<b>Conclusiones</b>	30
<b>Índice de Ilustraciones</b>	31
<b>Webgrafía</b>	33

## Anexos:

- 1 Introducción
- 2 Ideación
- 3 Generación Procedural
- 4 Programación
- 5 Modelado
- 6 Texturizado

01.

*Introducción*

# 1.1 Objetivo y Alcance

## ■ OBJETIVO:

El proyecto consiste en la generación procedural de una colección de viviendas que será incorporada en el proyecto interno de la empresa Imascono llamado “Spaceship Project”.

Cada vivienda debe ser única, conteniendo una combinación de objetos diferente y debe mantener una estética futurista acorde a la nave de Imascono en el proyecto *Spaceship*.

El resultado final constituye un conjunto de imágenes que forman una colección *NFT*. En un futuro, los portadores de estas imágenes tendrán acceso a una vivienda virtual con las características de la imagen.

## ■ ALCANCE:

Creación mediante herramientas de modelado 3D de distintos objetos y variantes de estos mismos objetos que compongan el espacio de cada vivienda.

Reducción de polígonos de cada objeto con el fin de facilitar la creación posterior de los espacios virtuales.

Texturizado de los objetos previamente modelados.

Diseño de un sistema automatizado que genere proceduralmente la colección de viviendas, así como la serie de imágenes de cada vivienda y sus respectivos archivos de texto que describan los objetos que contiene cada vivienda (ésto es necesario para la creación de la colección *NFT*).

Creación de la colección *NFT*.

No está incluida en el alcance la creación de los espacios virtuales que los usuarios finales podrán visitar.

# 1.2 Contexto

## ■ NFTS

Los NFT’s funcionan como un certificado que prueba la autenticidad de una obra digital.

Este concepto tiene una aplicación directa en el proyecto, ya que el portador de un NFT de las de la colección de viviendas, en un futuro podrá acceder a una vivienda virtual con esas características.

## 1.2 Contexto

### IMASCONO

Imascono es un estudio de tecnologías creativas especializado en realidad extendida. Está formado por un equipo de 25 personas que trabajan en una oficina de Zaragoza. Su principal ocupación es el desarrollo de aplicaciones, páginas web y sobre todo, espacios de realidad virtual y aumentada. Algunos ejemplos de espacios virtuales creados por Imascono son: la ofrenda de flores virtual (Figura 1) y el metaverso de Rafa Nadal Academy (figura 2).



Figura 1. Ofrenda de flores virtual (Fiestas del pilar 2020) Figura 2. Metaverso Rafa Nadal

### PROYECTO SPACESHIP

El proyecto “SpaceShip” de Imascono nace para celebrar los 10 años de la empresa. Esta nave espacial, es un espacio virtual al que cualquier persona con su teléfono móvil puede acceder escaneando un código QR. Una vez dentro, pueden desplazarse por el espacio con un avatar en tercera persona. Durante la experiencia se puede interactuar con la nave de Imascono, la cual representa la metodología que utiliza la empresa para abordar sus proyectos, desde el estudio de mercado y la ideación, hasta el lanzamiento del proyecto (figura 3). Una vez dentro de la nave, se puede acceder a distintos módulos donde se detalla el recorrido de la empresa y los proyectos más importantes que ha llevado a cabo (figura 4).

El proyecto está en desarrollo y se plantea implementar nuevas funciones. Una de ellas es vincular el proyecto con una colección *NFT*. De ahí **nace la idea de realizar viviendas personalizadas para los usuarios que entren en la nave.**

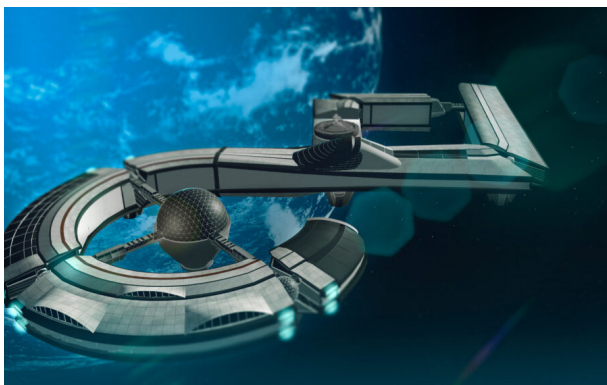


Figura 3. Nave de Imascono



Figura 4. Interior de la Nave de Imascono

# 1.3 Metodología

## ■ DESGLOSE DEL PROYECTO

El proyecto se ha desarrollado siguiendo 5 fases:

- **Ideación:**

- - \* Realización de Bocetos, moodboards, exploración formal, etc.
  - \* Elección de los módulos que aparecerán en las viviendas.
  - \* Distribuciones de los módulos.

- **Generación Procedural:**

- \* Aprendizaje de la herramienta Geometry Nodes de Blender
- \* Método para combinar proceduralmente los espacios.
- \* Generación automática de los metadatos de cada render.

- **Modelado:**

- \* Aprendizaje de Modelado con Blender
- \* Modelado de los módulos (salón, cocina, baño, etc.)
- \* Modelado de los *Assets* de cada módulo (sofá, mesilla, TV, etc.)
- \* Reducción de polígonos de los *Assets*.

- **Texturizado:**

- \* Aprendizaje de texturizado y creación de mapas de UV's.
- \* Texturizado de los *Assets*.

- **Generación de Imágenes:**

- \* Aprendizaje de Renderizado y composición en Blender.
- \* Realización de los renders de las habitaciones.
- \* Realización de una colección que incluya el número de renders deseado.

# 1.3 Metodología

## ■ PROGRAMAS UTILIZADOS

Se ha utilizado el programa Blender para abordar la mayoría de las tareas del proyecto, por lo que se ha requerido de un aprendizaje constante del programa paralelamente al desarrollo del proyecto.

Tareas realizadas:

- **Modelado Hard Surface:** La mayoría de los modelos 3D que se han creado son objetos rígidos donde predominan las líneas rectas y formas geométricas por lo que pueden crearse partiendo de formas primitivas como un cubo y realizando operaciones tales como extrusiones o movimiento de vértices. Sin embargo, también ha sido necesario aprender a modelar con subdivisiones para trabajar objetos con formas más redondeadas u orgánicas, como por ejemplo, una almohada o una manta.
- **Geometry Nodes:** Es una herramienta de Blender que permite modelar de forma procedural mediante un sistema de nodos. Esta herramienta de Blender está aún en desarrollo por lo que aún tiene bastantes fallos y sistemas incompletos. En éste proyecto se ha utilizado Geometry Nodes para modificar de forma procedural las distribuciones de las viviendas y los objetos que encontramos en cada una de ellas.
- **Unwrapping:** Todos los modelos 3D realizados en el proyecto se despliegan en un mapa de UV's para poder texturizarlos posteriormente. Éste proceso se ha realizado con las herramientas básicas de Blender.
- **Texturizado:** En un primer momento se planteó el texturizado de los objetos mediante texturas gratuitas de internet y aplicadas a los mapas de UV's en Blender. Sin embargo, durante el proyecto se han utilizado las herramientas de texturizado: *Substance Painter* y *Substance Designer* para esta tarea. También se ha utilizado la herramienta *Shader Nodes* de Blender para generar materiales fácilmente editables y manejables mediante nodos.
- **Programación:** Para poder generar los renders de las distintas viviendas automáticamente y para poder asociar cada una de ellas a metadatos que describan los objetos que tienen dichas viviendas, ha sido necesario utilizar la función de *scripting* de Blender.



# 1.4 Planificación

## PLANIFICACIÓN GENERAL

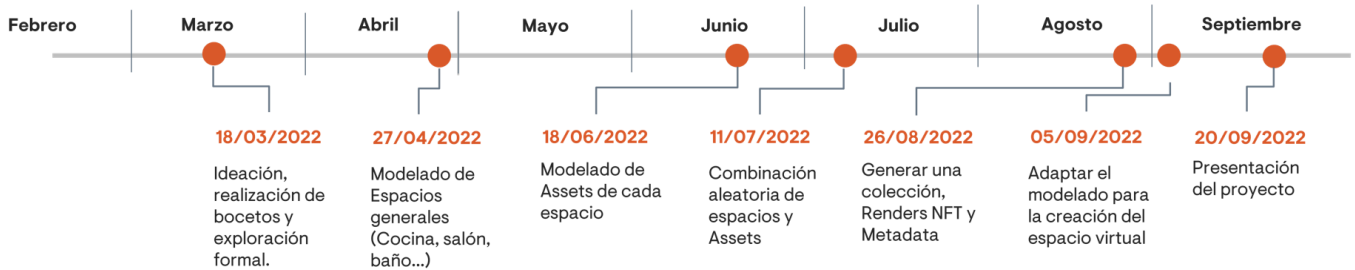


Figura 5. Esquema Planificación del proyecto

## APRENDIZAJE DE BLENDER

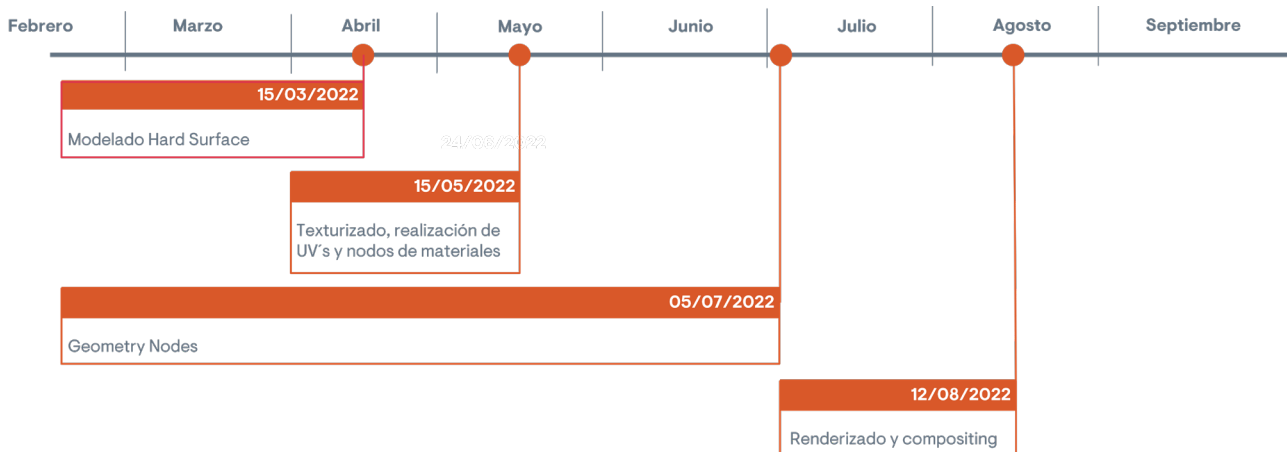


Figura 6. Esquema Aprendizaje de Blender

Se pueden encontrar más detalles de este apartado en los **anexos** **pág. 3-12**.

02.

*Ideación*

## 2.1 Planteamiento de Módulos

### ELECCIÓN DE MÓDULOS

Cada vivienda debe estar formada por distintos espacios a los que llamaremos módulos (salón, cocina, baño, etc.). Se ha utilizado la técnica creativa *Brain Storming* para plantear un gran número de posibles módulos. Posteriormente, se han filtrado eligiendo los módulos con más potencial. Se especifica el proceso hasta llegar a este resultado en los **anexos pág. 15 - 18**.

Módulos de cada vivienda:

- **Salón:** Espacio grande con elementos como un sofá, una televisión y una mesilla.
- **Dormitorio:** Espacio mediano con elementos como una cama y un escritorio.
- **Laboratorio:** Espacio mediano donde se accede al Creador de mundos (futuro proyecto planteado por Imascono).
- **Teletransporte:** Espacio pequeño por el que se accede a la vivienda.

## 2.2 Exploración Formal

### PANELES DE INFLUENCIA

Se han seleccionado imágenes de viviendas y objetos con una estética similar con el fin de servir como inspiración a la hora de crear los módulos (figura 7).

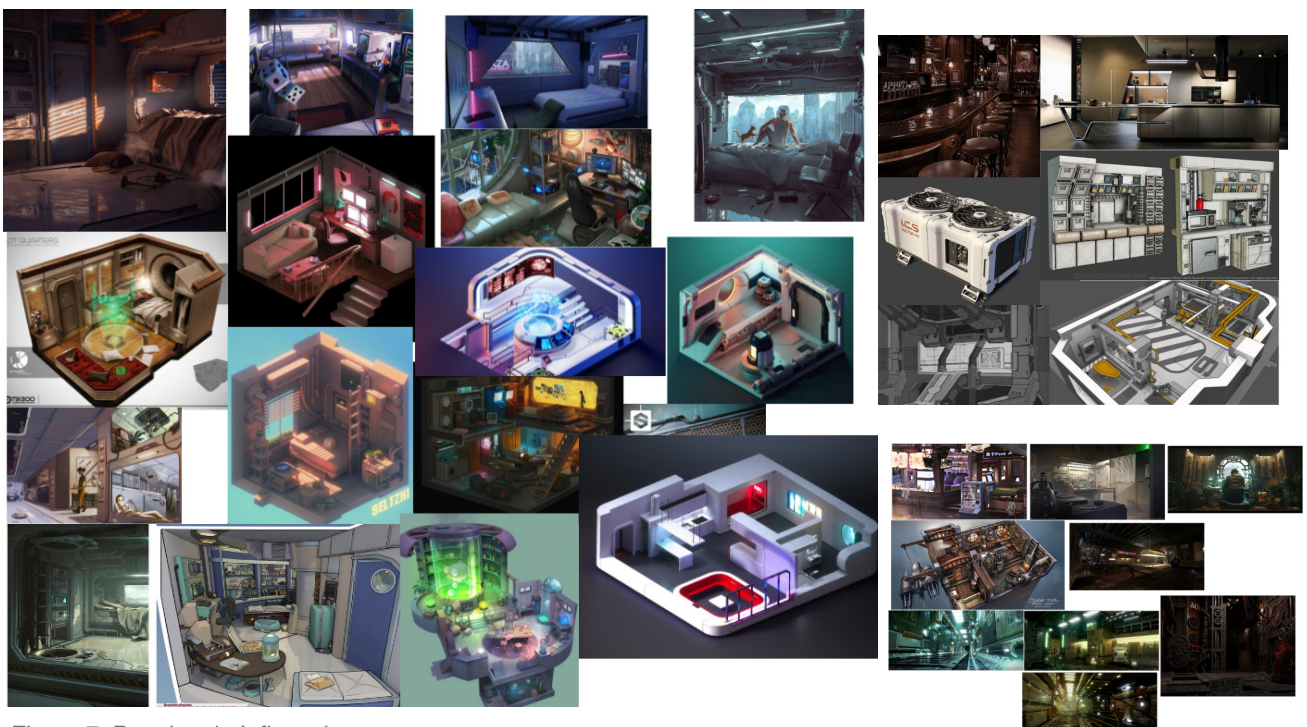


Figura 7. Paneles de Influencia

## 2.2 Exploración Formal

### ■ BOCETOS

Con el objetivo de visualizar distintas posibilidades en la distribución, se han realizado bocetos mezclando los cuatro módulos establecidos (figura 8). Además, se han realizado bocetos en 3D utilizando cubos y formas básicas en Blender (figura 9).

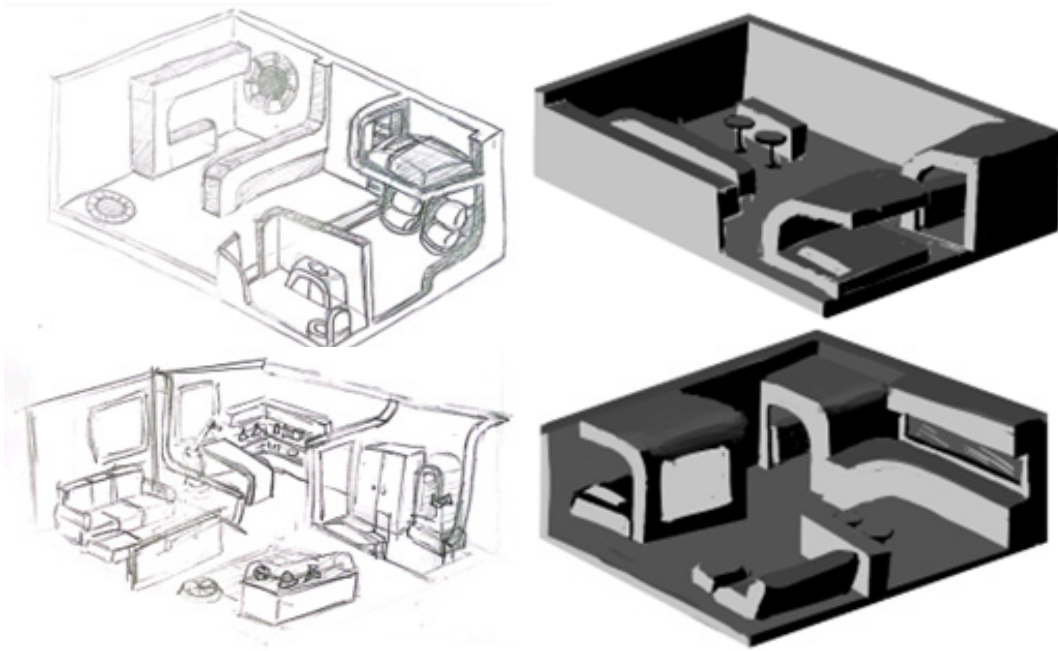


Figura 8. Bocetos realizados

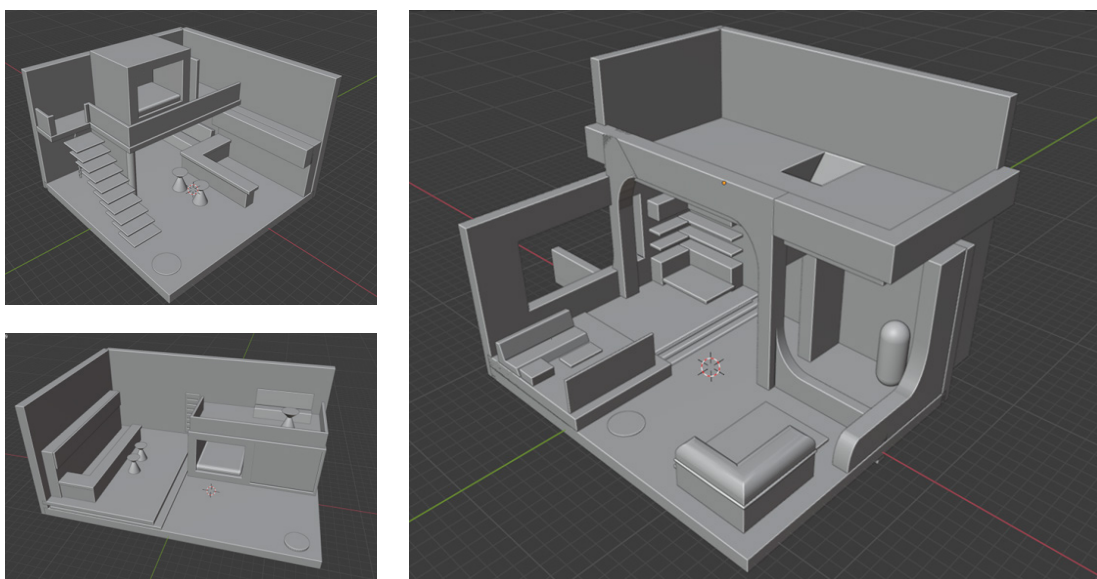
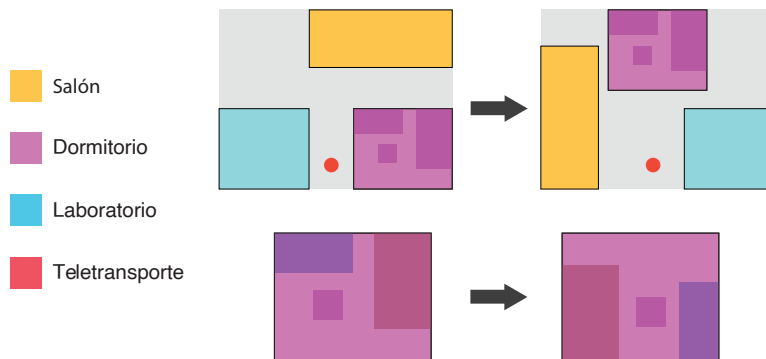


Figura 9. Bocetos realizados en 3D

## 2.3 Distribución de Módulos

### ■ FUNCIONAMIENTO DE LOS MÓDULOS

Al construir cada vivienda con módulos preestablecidos se consigue poder crear distintas distribuciones colocando los módulos en lugares diferentes. Además, la distribución de los objetos dentro de cada módulo puede variar sin alterar el resto del espacio.



En la figura 10 se ve como los módulos “Salón”, “Dormitorio” y “Laboratorio” pueden distribuirse de distintas formas. Además, cada módulo acepta distintas distribuciones de los objetos en su interior.

Figura 10. Explicación funcionamiento de Módulos

### ■ DISTRIBUCIÓN ELEGIDA

Distribución con forma de cuadrado en la que el salón ocupa una gran parte del espacio (figura 11).

El laboratorio y el dormitorio tienen el mismo tamaño, lo que permite intercambiar sus posiciones.

Se especifica el proceso hasta llegar a este resultado en los **anexos pág. 23 - 30**



Figura 11. Distribución elegida

## 2.3 Distribución de Módulos

### ■ TRANSICIÓN 2D - 3D

Para visualizar mejor la distribución elegida ha sido necesario recrearla en 3D. Para ello, se ha utilizado una cuadrícula con el fin de que los módulos y los objetos mantengan las mismas proporciones:

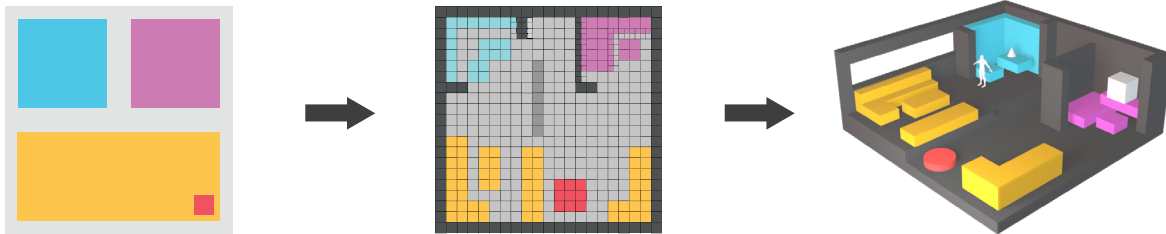


Figura 12. Transición 2D - 3D

## 2.4 Listado de Objetos

Se ha realizado una lista con todos los objetos que deberán aparecer en las viviendas.

### ■ MODELOS PRIMARIOS

Son objetos grandes e importantes que delimitan los espacios y sirven para ubicar otros.

#### Objetos del salón:

- Sofá
- Mesilla
- Televisión

#### Objetos del laboratorio:

- Mesa
- Estantería
- Brazo mecánico

#### Objetos del dormitorio:

- Escritorio
- Cama
- Silla

#### Objetos del teletransporte:

- Teletransporte

### ■ MODELOS SECUNDARIOS

Son objetos de tamaño más reducido que dependen de otros objetos para ubicarse

#### Objetos de la estantería:

- Destornillador, guantes, etc.
- Objetos al estilo "Reactor ARK"
- Libros

#### Objetos de mesilla salón:

- Mando de consola
- Planta
- Envase de noodles y lata

#### Objetos del escritorio:

- Ordenador
- Libros, apuntes
- Lámpara de lava
- Cascos de música

#### Objeto en pared del dormitorio: Objetos del salón:

- Póster
- Reloj
- Lámpara de Pie
- Alfombra
- Reloj

#### Objetos en la cama o sofá:

- Cojín
- Peluche

03.

*Generación Procedural*



## 3.1 Geometry Nodes

### INTRODUCCIÓN

Después de valorar diferentes posibilidades (**anexos pág. 34 y 35**), se ha optado por utilizar Geometry Nodes para la generación procedural. Geometry Nodes es una herramienta de Blender que permite crear y modificar geometría mediante un árbol de nodos. De esta manera se consigue un sistema de modelado procedural y no destructible.

El procedimiento de trabajo habitual de Geometry Nodes parte de utilizar una forma primitiva como puede ser un plano para después añadirle distintos nodos que modifiquen su geometría.

### APLICACIÓN AL PROYECTO

Se ha creado un árbol de nodos que realiza las siguientes funciones:

- Muestra una de las variantes de cada objeto. Esto quiere decir que si existen tres variantes distintas de sofás solo mostrará una.
- Modifica las posiciones de los objetos para que los módulos “Laboratorio” y “Dormitorio” puedan intercambiarse de sitio.

Para realizar estas funciones se han utilizado mayormente los nodos “Switch” (encargado de elegir que objetos mostrar en escena) y “Transform” (modifica la posición de los objetos) (figura 13).

Todo el proceso seguido hasta llegar al resultado final de Geometry Nodes se encuentra detallado en los **anexos pág. 34 - 43**

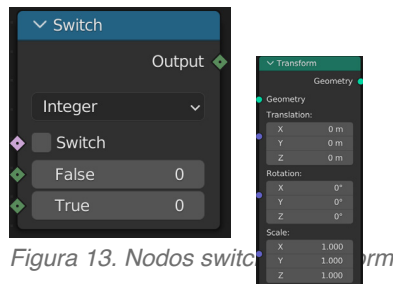


Figura 13. Nodos switch y transform

El resultado final consiste en una tabla de parámetros que, según reciba unos valores u otros, genera distintos resultados. Posteriormente se modelarán variantes de cada objeto (tres tipos de sofás, tres tipos de mesas, etc.) y dependiendo de los números que se reciban en la tabla de valores, se mostrarán unas variantes u otras. De esta manera solo falta crear una forma de cambiar los valores de la tabla automáticamente (figura 14)

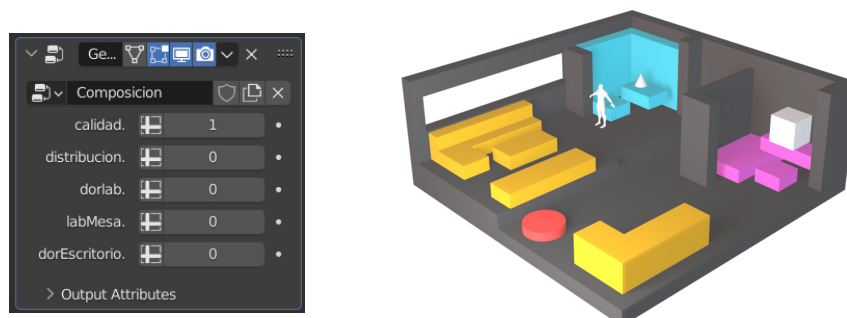


Figura 14. Parámetros Gemetry Nodes



## 3.2 Programación

### PLANTEAMIENTO

El resultado final del trabajo realizado en Geometry Nodes consiste en una tabla de parámetros en la que se introducen los valores manualmente. Sin embargo, es necesario automatizar el proceso para que se introduzcan valores de forma **automática**.

Blender ofrece la posibilidad de introducir *scripts* escritos en lenguaje de Python para realizar distintas tareas dentro del programa. El *script* utilizado ha sido escrito con la ayuda de la Python API para Blender.

### RESULTADO

El resultado final es un *script* formado por un bucle que se actualiza cada vez que el editor de vídeo de Blender cambia de fotograma. De esta manera, puedo crear un vídeo compuesto por “n” fotogramas y hacer que en cada fotograma utilice unos valores distintos en la tabla de parámetros de Geometry Nodes. Se puede visualizar el *script* completo y el desarrollo del mismo en los **anexos pág. 45-54**.

El *Script* también realiza otras 3 funciones:

**Rarezas:** Dentro de las variantes de un objeto, cada una tiene diferentes probabilidades de aparición en la colección. Por ejemplo, si se tiene un objeto dorado que solo se quiere que aparezca el 1% de las veces, si se crea una colección con 100 viviendas, aparecerá en una única vivienda.

**Metadatos:** En cada fotograma de vídeo se genera un archivo de texto “.json” que indica los objetos que aparecen en esa escena, así como sus rarezas de aparición y otras características. Este archivo funciona como una lista de todos los objetos que contiene la imagen y es necesario para la creación de la colección NFT (Figura 15).

**Variación de materiales:** Para modificar los materiales de los objetos desde el código, se ha accedido al editor de *Shaders* de Blender. Por cada objeto se han situado distintos materiales, sin embargo, solo se ha conectado a la salida del material uno de ellos. El código se encarga de cambiar la conexión de un material u otro a la salida (figura 16).

```
{
  "name": "Viviendas",
  "description": "Colección NFT de viviendas",
  "edition": 1,
  "attributes": [
    {
      "trait_type": "sofa",
      "value": "Sofá básico "
    },
    {
      "trait_type": "mesilla",
      "value": "mesilla1"
    },
    {
      "trait_type": "tv",
      "value": "cubo"
    },
    {
      "trait_type": "estanteria",
      "value": "cono"
    },
    {
      "trait_type": "brazo",
      "value": "cubo"
    }
  ]
}
```

Figura 15. Archivo de texto generado

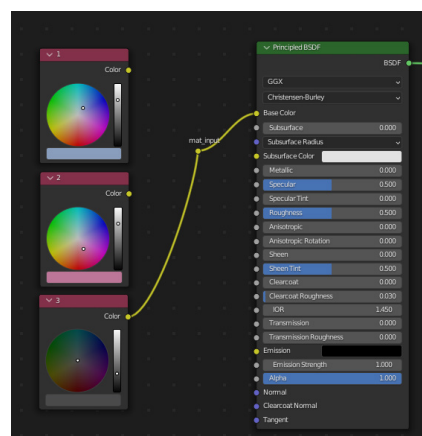


Figura 16. Funcionamiento del cambio de Material

## 3.3 Conclusiones

El resultado al que se ha llegado en el apartado de generación procedural es muy sencillo, simplemente es una tabla de valores creada en Geometry Nodes que cambia la distribución de la vivienda y que se controla mediante un *script*. Sin embargo, para llegar a este resultado, tuve que plantear una gran variedad de alternativas debido a que no conocía las posibilidades de Geometry Nodes y no sabía que podía ayudarme con un *script* para automatizar el proceso. Investigué otros trabajos de proyectos similares y la mayoría de ellos no utilizaban Geometry Nodes porque eran más sencillos y no lo necesitaban. Lo que me impulsó a seguir probando esta dirección fue que conocía otros proyectos parecidos al mío en el ámbito de la generación procedural y sabía que se habían realizado con Geometry Nodes (aunque no explicaban el proceso). Así que para llegar a un resultado muy sencillo en el que prácticamente utilicé dos nodos repetidos múltiples veces, tuve que asentar una base sólida de la herramienta.

Para realizar el *script* también tuve muchos problemas por la misma razón que con Geometry Nodes: no sabía cómo abordar los problemas, no tenía la certeza de estar siguiendo un buen camino y mis conocimientos de programación eran muy reducidos. Además, no conocía el lenguaje de programación utilizado (Python), lo que dificultó el proceso. Sin embargo, pude ayudarme con foros que explicaban conceptos básicos de Python para Blender y de la web Python API Blender.

04.

*Desarrollo de objetos*

## 4.1 Metodología

Se ha seguido una metodología dividida en dos fases (modelado y texturizado) para desarrollar los objetos con el fin de mantener un proceso eficaz y común para todos:

### ■ MODELADO

Se han utilizado las herramientas de modelado de Blender para crear cada objeto. Para modelar cualquier objeto se ha utilizado dos escenas de Blender. La primera consiste en una escena general que contiene todos los objetos representados con formas básicas (cilindros, cubos, etc.). Estas formas primitivas acotan el espacio que tiene que ocupar cada objeto dentro de la vivienda y su interacción con el resto de la escena. En la otra escena es donde se modelan los objetos partiendo de la forma básica. Una vez modelado, se coloca el objeto en su posición en la primera escena.

Cada objeto se ha modelado dos veces, la primera es un modelo con muchos polígonos y muy detallado. El segundo es un modelo con pocos polígonos y con formas simplificadas. Posteriormente se traspasarán los detalles del modelo muy detallado al de pocos polígonos en el proceso de texturizado. De esta manera se consigue un modelo que consume pocos recursos y que mantiene una gran cantidad de detalles.

Durante el modelado se ha seguido una metodología común para crear todos los objetos:

- Modelado manteniendo una topología compuesta por cuadrados y usando subdivisión para las zonas curvas.
- Creación de un modelo con una gran cantidad de polígonos y con muchos detalles.
- Eliminación de detalles y simplificación de la forma para crear un modelo con pocos polígonos.
- Triangulación del modelo para su posterior exportación.

### ■ TEXTURIZADO

Se ha seguido un único proceso para texturizar todos los objetos:

- Dentro de Blender se generan los mapas de *UV's* del modelo de bajo poligonaje y se empaqueta lo mejor posible.
- Posteriormente se exporta el modelo a Substance Painter donde se realiza un *Bake* para poder obtener los detalles del modelo de alta poligonización en el de bajo en forma de mapas de texturas.
- Finalmente se texturiza en Substance Painter y se exportan los mapas de texturas del material a Blender.

(Todo el proceso se explica detenidamente en el apartado “4.3 Texturizado”)

## 4.2 Modelado

### ■ TOPOLOGÍA

Una de las prioridades ha sido mantener una topología limpia basada en cuadrados, evitando triángulos y polígonos de más de cuatro vértices. Con esto se consigue evitar problemas de pinzamientos, mantener una homogeneidad en la superficie, etc.

Para crear los modelos de alta poligonización se ha utilizado modelado por subdivisiones, agregando muchos detalles en formas curvas. En este caso se ha tenido que prestar especial atención en evitar utilizar polígonos que no estén formados por cuatro vértices, debido a que la subdivisión trabaja especialmente mal con polígonos de estas características.

En los modelos de baja poligonización, se han utilizado técnicas de modelado poligonal manteniendo una topología reducida y con pocos detalles, dado que en un futuro estos modelos 3D tendrán que funcionar en página web.

Posteriormente, en el texturizado, se proyectarán los detalles de los modelos de alta poligonización sobre los modelos de baja poligonización.

**Número de polígonos:**  
4.128 cuadrados



**Número de polígonos:**  
852 cuadrados



*Figura 17. Silla de alta poligonización (izda.) - baja poligonización (dcha.)*

Como se puede observar (figura 17), el modelo con menos polígonos tiene menos detalles y no contiene redondeos sino aristas vivas que no se sienten realistas. Dicho modelo contiene los polígonos necesarios para que la silueta del modelo pueda verse con claridad, los demás detalles se podrán transferir desde el modelo de alto poligonaje. Una cantidad de polígonos de 4000 cuadrados, como en el primer caso, es inviable para poder visualizarlo en tiempo real (en una página web en el caso de este proyecto). Sin embargo, 800 polígonos es una cantidad mucho más manejable.

El proceso de modelado y las herramientas utilizadas se encuentran detalladas en los **anexos pág. 57-58**.

## 4.2 Modelado

### ■ LISTADO DE OBJETOS

Finalmente se han modelado 21 objetos primarios y 20 objetos secundarios. Además se incluyen los objetos base que permanecen estáticos en todas las variaciones de las viviendas. Estos son las paredes, el suelo, paredes secundarias y luces.

Los objetos “cubo de rubick”, “póster” y “cuchillo” han sido proporcionados por la empresa Imascono.

Los objetos modelados son:

#### **Modelos Primarios:**

- Sofá estándar
- Sofá con Chaise Longue
- Mesilla metálica
- Mesilla de madera
- Televisión con armario
- Televisión estándar
- Estantería estándar
- Estantería de madera
- ‘Brazo robótico
- Brazo robótico 2
- Mesa con pantalla
- Mesa colgante con pantalla
- Silla metálica
- Silla estilo gamer
- Cama empotrada
- Cama con armario
- Escritorio básico
- Escritorio de madera
- Teletransporte básico
- Teletransporte con forma de puerta
- Teletransporte de prismas

#### **Modelos secundarios:**

- Ordenador de sobremesa
- Cubo de rubick
- Libros y papeles desordenados
- Reloj de pared
- Póster 1
- Póster 2
- Póster 3
- Póster 4
- Póster 5
- Póster 6
- Lámpara de pie
- Planta
- Guitarra eléctrica
- Envase de noodles
- Cuchillo
- Pizza
- Fútbolín
- Piscina hinchable
- Robot mayordomo
- Robot Roomba

## 4.2 Modelado

## ■ MODELADO DE OBJETOS PRIMARIOS

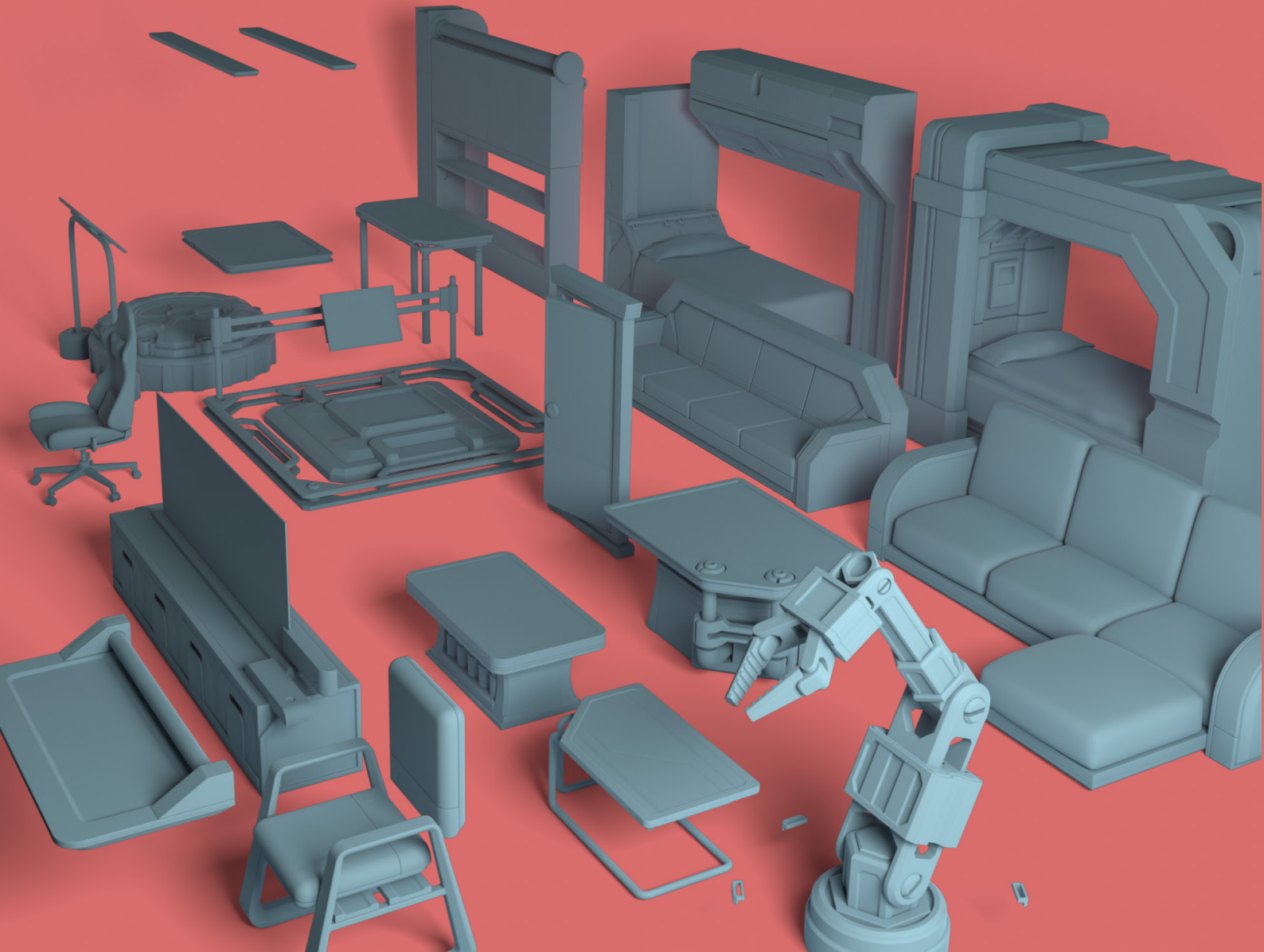


Figura 18. Render de modelos primarios

En esta imagen (figura 18) se pueden visualizar algunos de los modelos primarios generados. Ésta es su versión de alto poligonaje renderizada con un material básico. Como ya se ha comentado, cada uno de los modelos tiene una versión con menor cantidad de polígonos (las imágenes de los modelos por individual se encuentran en los anexos pág. 60-63).



## 4.2 Modelado

### ■ MODELADO DE OBJETOS SECUNDARIOS

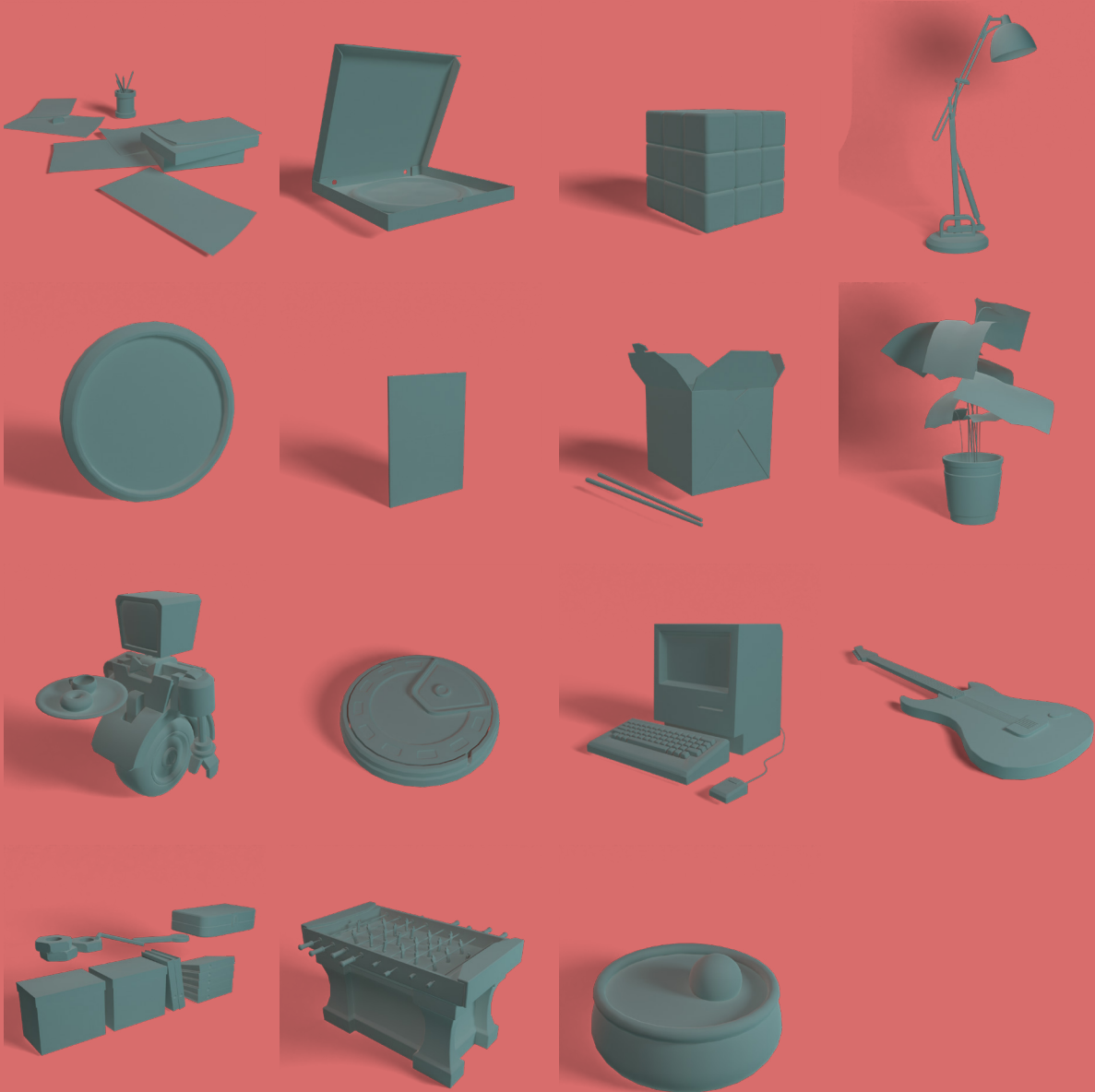


Figura 19. Render de modelos secundarios

Estos objetos han sido modelados reduciendo aún más la carga de polígonos. Esto es debido a que la mayoría son objetos pequeños poco importantes en la escena que no necesitan una gran cantidad de detalle.



## 4.2 Modelado

### ■ DIFICULTADES EN EL MODELADO Y CONCLUSIONES

El tipo de modelado utilizado en *softwares* como Blender, Maya o 3D Studio Max, es muy diferente al modelado 3D tipo *CAD* al que estoy acostumbrado con programas como Inventor o Solid Works. El dominio de este tipo de modelado ha llevado mucho más tiempo del planteado en un inicio. Mantener una topología adecuada y formada por cuadrados ha sido una tarea complicada que he ido perfeccionando durante el transcurso del proyecto. Al principio los modelos tenían problemas de pinzamientos, las superficies no se veían del todo lisas y tenía complicaciones con los objetos formados por superficies curvas (en estas superficies es donde se notan más los problemas en la topología). Por ello, al final de la etapa de modelado, he tenido que rehacer varios de los objetos previamente modelados que no cumplían las exigencias de topología.

Lo más costoso a la hora de crear los objetos ha sido el modelar dos versiones de cada objeto, una con alta y otra con baja poligonización. En un comienzo, la tarea era mucho más costosa. Sin embargo, durante el desarrollo, empecé a utilizar nuevos procesos que me permitían aprovechar la malla del modelo de alta poligonización para crear el de baja. Por ejemplo, en lugar de crear un redondeo en todas las aristas, utilicé un modificador que creaba el redondeo de forma no destructiva, por lo que más tarde podía simplemente quitar el modificador, reduciendo en gran parte la cantidad de polígonos sin esfuerzo.

Todo el proceso de modelado me ha servido para entender como funciona todo este mundo de la creación de objetos destinados a cine, videojuegos u otro tipo de plataformas. Se trata de una rama del modelado compleja y completamente diferente a la rama del modelado tipo CAD.

El proceso de modelado se realizó en varias fases. Primero se modeló y texturizó un único objeto con el fin de verificar que la estrategia seguida durante el modelado encajaba con la de texturizado. Posteriormente, modelé una cantidad reducida de objetos y los texturicé. Dividir el proceso de modelado me permitió visualizar problemas y fallos que contenían los modelos que no habría podido encontrar durante la etapa de modelado. Cuando tuve claro el proceso a seguir para todos los objetos y como evitar problemas durante el texturizado, procedí a modelar el resto de objetos.

Además de los modelos nombrados anteriormente, también se han realizado modelos que no se incluyen en el resultado inmediato de este proyecto. Sin embargo, si formarán parte de las viviendas una vez se cree el espacio virtual al que podrá acceder el usuario. Estos objetos son: Paredes, techo y vitrinas para visualizar NFT's de los usuarios. Más información de estos modelos en los **anexos pág. 65**.

El aprendizaje de modelado lo he realizado mediante cursos en la página de Blender Studio (<https://studio.blender.org/>) y mediante tutoriales en la plataforma de YouTube como los del canal Blendtuts-ES.

## 4.3 Texturizado

### ■ MAPEADO DE UV'S

El mapeado de UV's consiste en representar la superficie de un modelo 3D en un plano bidimensional con ejes "U" y "V". Para realizar el mapeado es necesario seleccionar unas costuras en el modelo por donde se va a partir para poder representarlo en 2D. Este proceso es necesario para texturizar cualquier modelo. La figura 20 es una fotografía en la que se visualiza cómo se representa la superficie de una forma 3D en un plano bidimensional.

Proceso seguido en Blender para realizar el mapeado UV de cada modelo 3D (figura 21):

- Seleccionar manualmente las **costuras** del objeto.
- **Desenvolver** y verificar con una textura de prueba que no sufra deformaciones y que la textura se ve homogénea en todas partes.
- Realizar el **empaquetado**, consistente en colocar los distintos objetos desenvueltos en un espacio cuadrado aprovechando lo mejor posible el espacio. El empaquetado se ha realizado automáticamente con las herramientas de Blender. Sin embargo, se ha necesitado empaquetar manualmente los objetos que no aprovechaban eficientemente el espacio con el proceso automatizado.



Figura 20. Fotografía explicativa de los UV's

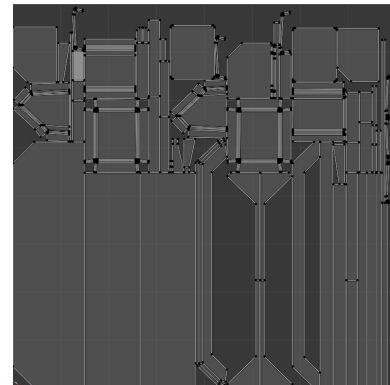
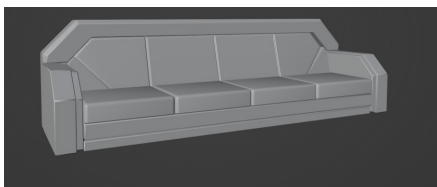
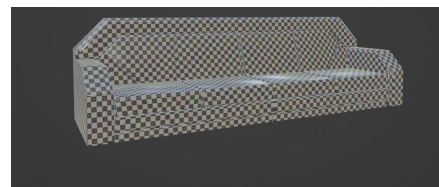


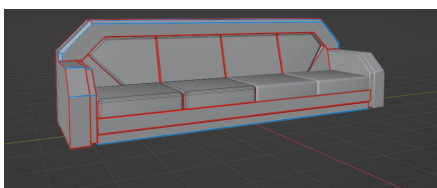
Figura 22. Mapa de UV's del sofá (Empaquetado)



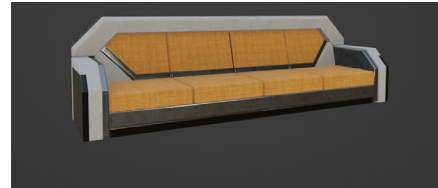
Modelo de baja poligonización



Textura de prueba para visualizar deformaciones



Selección de las costuras del objeto



Aplicación de la textura

Figura 21. Proceso Mapeado de UV's del sofá

## 4.3 Texturizado

### BAKE

El *bake* es el proceso en el que se proyecta información de un modelo 3D en un mapa de textura. La mayoría de las veces se utiliza la información de un modelo muy detallado con una gran cantidad de polígonos sobre otro con pocos polígonos. En este proceso es donde se han utilizado las dos versiones de los objetos previamente modeladas. Mediante el *bake* se obtienen mapas de texturas que una vez aplicados en el modelo de menos polígonos, hacen que se vea muy parecido al modelo con más polígonos (figura 24).

Los mapas de textura que se han utilizado en el *bake* son (figura 23):

- **Mapa de normales:** Guarda información sobre detalles de la superficie. De esta forma se modifica la manera en la que la luz interactúa con el objeto.
- **Mapa de oclusión ambiental (AO):** Guarda información de las zonas del modelo donde la luz entra con más dificultad y las oscurece.
- **Mapa de ID:** Guarda información sobre los materiales del modelo. Lo he utilizado para facilitar la tarea de texturizado utilizando distintos colores para cada parte separada de un modelo 3D.

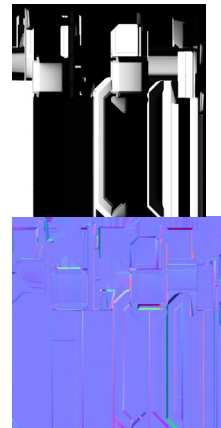


Figura 23. Mapa de AO y de Normales

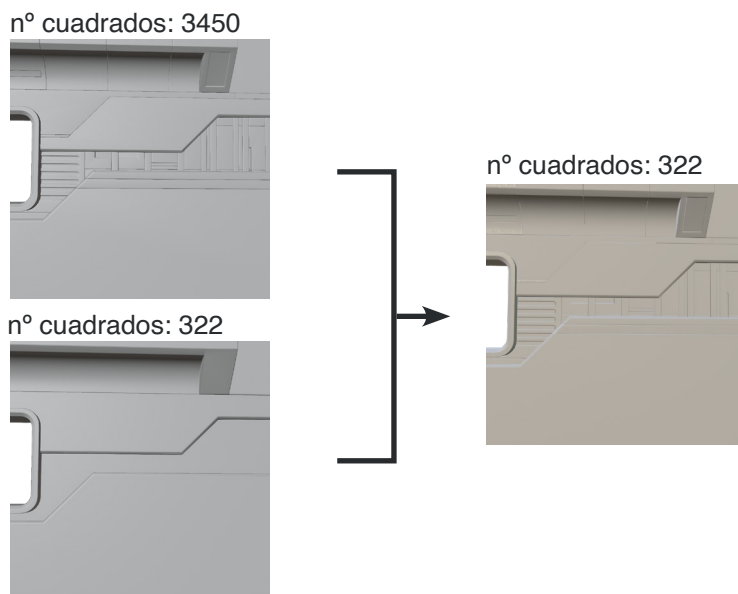


Figura 24. Proceso de bake en una pared de las viviendas

Como se puede ver (figura 24), aplicando los mapas de texturas del *bake*, el modelo con pocos polígonos tiene una calidad de detalle similar al modelo de alta poligonización.

## 4.3 Texturizado

### ■ HERRAMIENTAS UTILIZADAS

#### Substance Designer

El primer planteamiento fue texturizar todos los objetos mediante el editor de *shaders* de Blender y mediante texturas gratuitas de Internet. Sin embargo, se ha recibido una licencia de Substance Painter por parte de Imascono así que se ha realizado el texturizado en dicha herramienta.

También se ha podido utilizar el programa Substance Designer, donde se han creado algunas texturas para poder aplicarlas posteriormente en los modelos 3D en Substance Painter (figura 25). Proceso de utilización de Substance Designer se detalla en los **anexos pág. 69-71**.

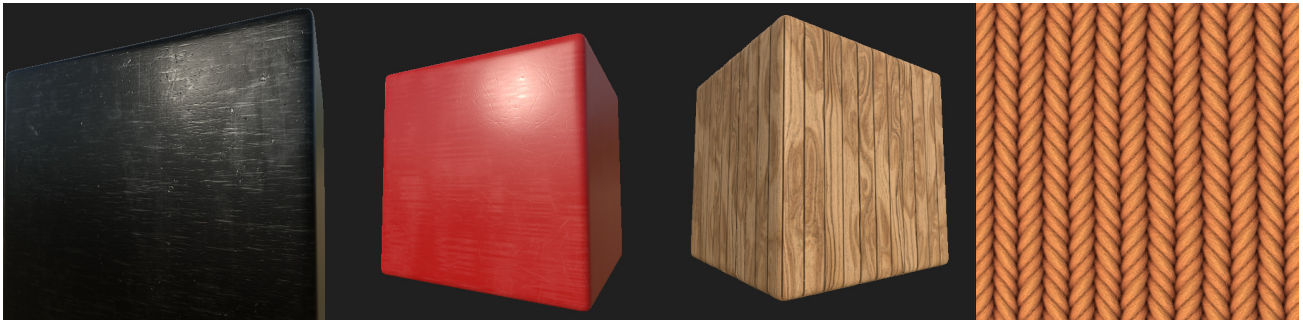


Figura 25. Materiales realizados en Substance Designer

#### Texturizado en Substance Painter

Para texturizar cada modelo se han utilizado los materiales creados en Substance Designer además de otros materiales que vienen por defecto en Substance Painter. También se han utilizado algunas máscaras inteligentes de Substance para agregar algunos detalles.

Para facilitar el proceso de texturizar, se ha utilizado un color diferente para cada pieza de los modelos mediante los mapas de ID. Posteriormente, se han usado esos colores como máscaras para agregar materiales a cada pieza con un único color (figura 26). Proceso de utilización de Substance Painter detallado en los **anexos, pág. 72-73**.

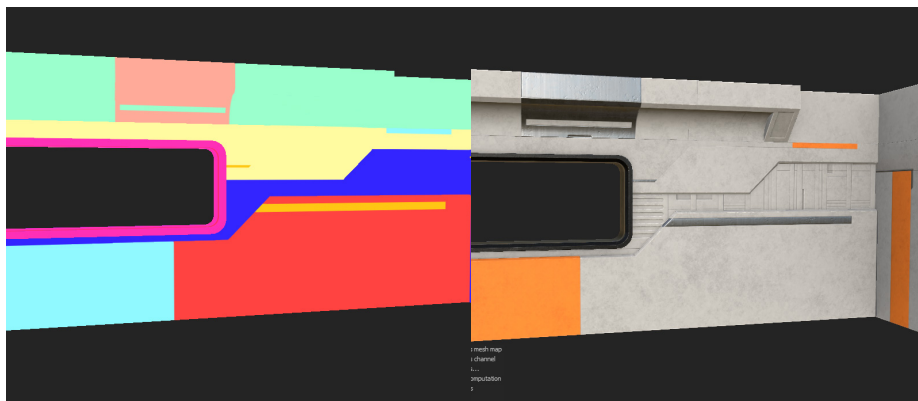


Figura 26. Texturizado utilizando Mapa de ID

## 4.3 Texturizado

### ■ EXPORTACIÓN A BLENDER

Después de texturizar, he exportado los mapas de texturas para poder visualizarlos en el modelo de bajo poligonaje de Blender.

Los mapas exportados son (figura 27):

- **Mapa de Color Base:** Guarda información del color de cada objeto.
- **Mapa de Oclusión Ambiental (AO):** Lo he utilizado mezclado con el color base para oscurecer concavidades o zonas interiores.
- **Mapa de Normales:** Modifica la superficie creando detalles sin tener que modificar la geometría.
- **Mapa de Rugosidad:** Guarda información sobre cómo rebota la luz en la superficie.
- **Mapa Metálico:** Se utiliza para indicar cuando un material es metálico o no.

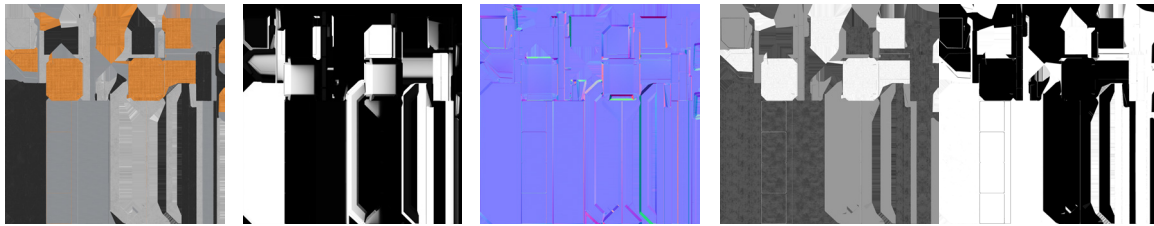


Figura 27. Mapas de Base color - AO - Normales - Rugosidad - Metálico



## 4.3 Texturizado

### ■ TEXTURIZADO DE MODELOS PRIMARIOS



Figura 28. Texturizado de modelos primarios

## 4.3 Texturizado

### ■ TEXTURIZADO DE MODELOS SECUNDARIOS



Figura 29. Texturizado de modelos secundarios

## 4.3 Texturizado

### ■ TEXTURIZADO DE MODELOS SECUNDARIOS



Figura 29. Texturizado de modelos secundarios

Para ahorrar recursos en cuanto a la utilización de texturas en los modelos secundarios, se han utilizado texturas de menos resolución que en los modelos primarios. Además, se han aprovechado algunas texturas en más de un modelo. Por ejemplo, la mayoría de plásticos contienen la misma textura.



05.

*Resultados*

# Resultados



Figura 30. Render de una combinación posible

La figura 30 es una de las viviendas que se pueden generar combinando los distintos objetos modelados y texturizados. Con las variantes de objetos creadas, existen **12.386.304 posibles combinaciones**.

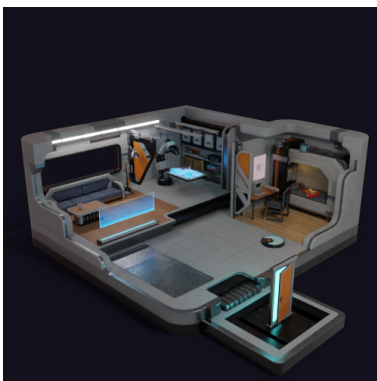
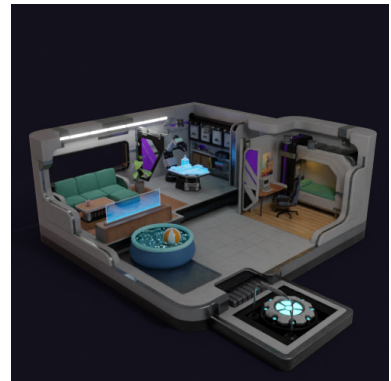
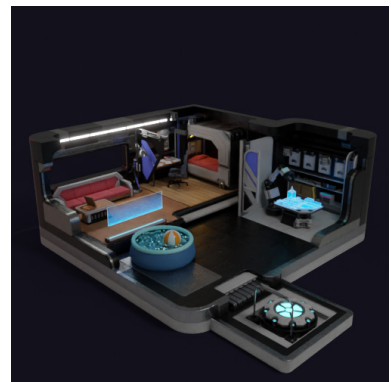
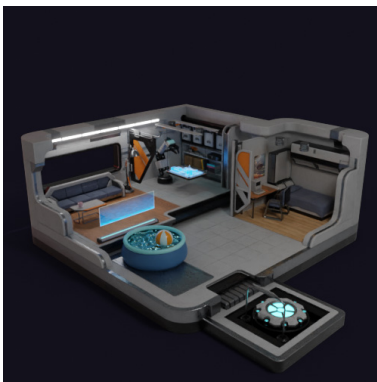
# Resultados

## COLECCIÓN DE VIVIENDAS

Debido a que no se dispone de un ordenador de elevadas prestaciones no se ha podido realizar una colección de viviendas de gran tamaño. Sin embargo, se ha realizado una colección de 20 imágenes que demuestran que la generación procedural y automática diseñada funciona correctamente.

La colección cumple los porcentajes de rarezas de cada objeto y cada imagen contiene sus metadatos con la lista de objetos que aparecen. Más detalles en los **anexos** **pág. 86**.

El proyecto está listo para que en el momento que Imascono lo decida, se genere una colección del número de viviendas deseada y se convierta en una colección NFT subiéndola a una plataforma de venta de NFT's como "OpenSea".



# Resultados

## COLECCIÓN DE VIVIENDAS

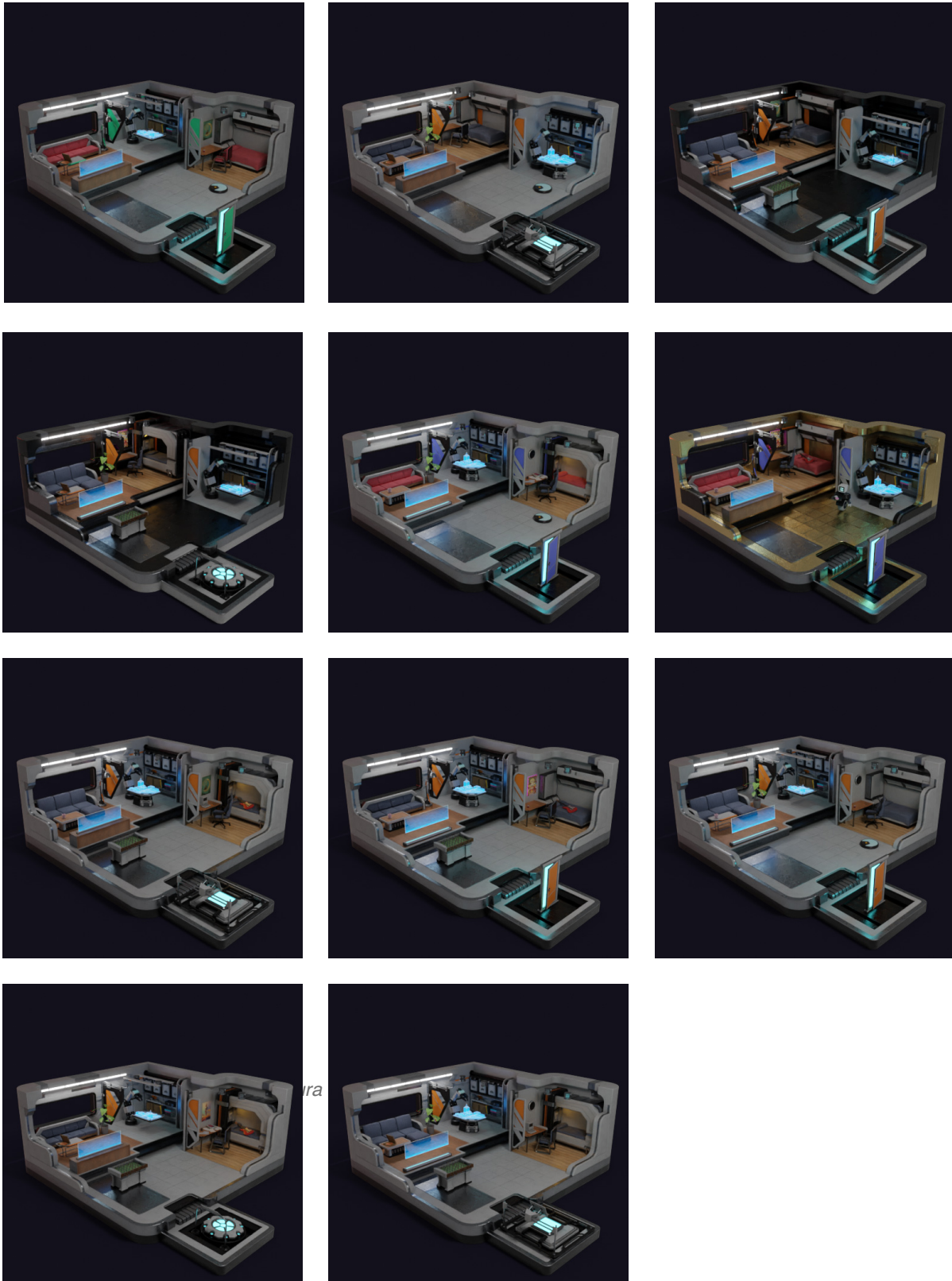


Figura 31. Colección de Viviendas



# Conclusiones

El mayor valor que ha tenido este trabajo para mí ha sido el **aprendizaje** realizado. La mayor parte del tiempo lo he dedicado a aprender distintos conceptos que he utilizado en el proceso. Me he encontrado con muchísimos obstáculos debido a que no conocía los programas utilizados, el *workflow* para modelar y texturizar objetos, etc. Entre las cosas que he tenido que aprender se encuentran: Modelado en Blender, Geometry Nodes de Blender, texturizado mediante nodos en Blender, renderizado, iluminación, composición, creación de materiales en Substance Designer, backeado en Substance Painter, texturizado en Substance Painter, programación en Python, etc.

Una de las lecciones que he aprendido es a acotar el alcance de un proyecto, debido a que se trataba de un proyecto con un enfoque abierto y con pocas restricciones. Cuando estaba en la etapa de ideación y tenía que decidir que módulos y que objetos aparecerían en las viviendas, surgieron ideas de una gran variedad de módulos y objetos. Sin embargo, tuve que reducir el número de módulos al mínimo debido a que añadir un módulo más, requería diseñar nuevos objetos, de cada uno de esos objetos crear varias variantes y de todas esas variantes crear un modelo de alta poligonización, otro de baja poligonización, realizar un *bake* y texturizarlo. En definitiva, la cantidad de trabajo crecía exponencialmente por el hecho de añadir un único módulo. El crear únicamente cuatro módulos, me ha ayudado a centrarme en crear más variaciones de los objetos y, en general, a centrarme más en los detalles.

He creado una metodología con la que generar las viviendas proceduralmente. Sin embargo, esta es solo la primera meta del proyecto de Imascono. En la etapa de ideación del proyecto, se diseñaron variantes de la vivienda con más módulos y con una segunda planta. Más adelante, se desarrollarán estos módulos ideados desde un inicio y se enfocarán las distribuciones de las viviendas siguiendo la jerarquía de bronce, plata y oro descrita en la fase de ideación. Por último, se desarrollarán los espacios virtuales de las viviendas a los que podrán acceder los usuarios que contengan un NFT de una de las viviendas.

# Índice de Ilustraciones

<i>Figura 1. Ofrenda de flores virtual</i>	6
<i>Figura 2. Metaverso Rafa Nadal</i>	6
<i>Figura 3. Nave de Imascono</i>	6
<i>Figura 4. Interior de la Nave de Imascono</i>	6
<i>Figura 5. Esquema Planificación del proyecto</i>	9
<i>Figura 6. Esquema Aprendizaje de Blender</i>	9
<i>Figura 7. Mood Board</i>	11
<i>Figura 8. Bocetos realizados</i>	12
<i>Figura 9. Bocetos realizados en 3D</i>	12
<i>Figura 10. Explicación funcionamiento de Módulos</i>	13
<i>Figura 11. Distribución elegida</i>	13
<i>Figura 12. Transición 2D - 3D</i>	14
<i>Figura 13. Nodos switch y transform</i>	16
<i>Figura 14. Parámetros Gemetry Nodes</i>	16
<i>Figura 15. Archivo de texto generado</i>	17
<i>Figura 16. Funcionamiento del cambio de Material</i>	17
<i>Figura 17. Sila low poly - high poly</i>	20
<i>Figura 18. Render de modelos primarios</i>	21
<i>Figura 19. Render de modelos secundarios</i>	22
<i>Figura 20. Fotografía explicativa de los UV's</i>	23
<i>Figura 21. Proceso Mapeado de UV's del sofá</i>	23
<i>Figura 22. Mapa de UV's del sofá</i>	23
<i>Figura 23. Mapa de AO y de Normales</i>	24
<i>Figura 24. Ejemplo de bake</i>	24
<i>Figura 25. Materiales realizados en Substance Designer</i>	25
<i>Figura 26. Texturizado utilizando Mapa de ID</i>	25
<i>Figura 27. Mapas de Base color - AO - Normales - Rugosidad - Metálico</i>	26
<i>Figura 28. Texturizado de modelos primarios</i>	27
<i>Figura 29. Texturizado de modelos secundarios</i>	32
<i>Figura 30. Render de una combinación posible</i>	34
<i>Figura 31. Colección de Viviendas</i>	36

# Webgrafía

Python API de Blender [En línea] [Consulta: 03 / 07 / 2022]  
<https://docs.blender.org/api/current/index.html>

Teoría del color [En línea] [Consulta: 23 / 06 / 2022]:  
<https://color.adobe.com/es/create/color-wheel>

Paneles de influencia [En línea] [Obtenido: 08 / 03 / 2022]:  
<https://www.pinterest.es/>

Aprendizaje de modelado en Blender [En línea] [Obtenido: 12 / 03 / 2022]:  
<https://www.youtube.com/c/Blendtutses>

Aprendizaje de Modelado y texturizado en Blender [En línea] [Consulta: 23 / 04 / 2022]:  
<https://studio.blender.org/>

Texturas de la comunidad de Substance Painter [En línea] [Consulta: 28 / 06 / 2022]:  
<https://substance3d.adobe.com/community-assets>

HDRI's para el renderizado [En línea] [Obtenido: 26 / 08 / 2022]:  
<https://polyhaven.com/hdri>

Texturas descargadas [En línea] [Obtenido: 13 / 07 / 2022]:  
<https://polyhaven.com/textures>