

Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

Unidad Lerma

División de Ciencias Sociales y Humanidades

Licenciatura en Arte y Comunicaciones Digitales

Proyecto de Investigación Creación Terminal

Organismos Inexistentes: Biomorfismo Procedural 3D

Presenta

Hasem Anzurez Baeza

Asesor de tesina

Dr. Hugo Solís García

03 de noviembre de 2023

Índice

1. Introducción. ... 3

2. Marco artístico histórico: El papel del Biomorfismo procedural generativo en las prácticas del arte digital. ... 5

2.1. Strand Tower ... 6

2.2. Tuboid ... 6

2.3. Aggregation ... 6

2.4. Malwarez y Spam Plants ... 7

2.5. Artificial Ecology ... 7

3. Marco teórico conceptual ... 9

3.1. Orígenes del biomorfismo y sus características. ... 9

3.2. Vida artificial ... 10

4. Proyecto Creativo: Organismos Inexistentes: Biomorfismo Procedural 3D ... 11

4.1. Descripción ... 11

4.2. Experiencia del público ... 13

4.3. Organismos Inexistentes: Biomorfismo Procedural 3D como una propuesta creativa de arte digital ... 15

4.4. Diagrama de flujo de la propuesta creativa. ... 16

5. METODOLÓGIA DEL PROCESO CREATIVO: ... 16

5.1. Los Primeros Pasos ... 17

El terror y los primeros acercamientos al biomorfismo en entornos virtuales e inmersivos. ... 17

Animación: Nociones de inmersividad y composición para el desarrollo de animaciones enfocadas a generar múltiples estímulos sensoriales ... 18

Biomorfismo y Jakob Stolte ... 20

5.2. Pre-Producción: El Enfoque de la Investigación y la Experimentación. ... 21

El Método de Jakob Stolte. ... 23

Mi método. ... 26

Corales ... 33

5.3. Producción: Programación y Animación ... 36

Programando el Add-on. ... 36

Haciendo la animación ... 42

5.4. Post Producción ... 48

6. Conclusiones finales ... 52

7. Bibliografía ... 53

8. Anexos 54

1. Introducción.

Este escrito documenta el proceso de desarrollo de mi proyecto de investigación y creación para el arte o de uso artístico, el cual se basa en el desarrollo de un add-on o herramienta para el software Blender que está enfocado en el desarrollo de objetos biomorfos 3D creados de forma procedural y generativa. Además, como parte del proceso creativo se desarrolló una animación en la que se presenta una serie de figuras biomorfas desarrolladas con la misma herramienta haciendo una demostración y ejemplo de uno de los múltiples usos que se le pueden dar.

Esta investigación es importante, ya que fomenta el desarrollo de herramientas para el apoyo e innovación en el ámbito artístico, tecnológico y comunitario, pues pretende ayudar en los procesos de creación artística de sus usuarios, permitiendo además que los más experimentados puedan modificar el código fuente de la herramienta para crear sus propias versiones del add-on, ya que este es de código abierto. Otro enfoque de esta investigación está en el desarrollo de conceptos artísticos, tecnológicos y estéticos basados en proponer un acercamiento entre lo natural, lo orgánico, lo artificial y lo digital, en donde se plantea y se explora el desarrollo de cuatro especies de organismos inexistentes generados por computadora que buscan evocar a elementos de la naturaleza tales como flores, plantas, corales o simulación de estructuras óseas, y figuras orgánicas que hagan alusión a carne, órganos y demás figuras abstractas de la naturaleza en un sentido más experimental.

La parte de la propuesta creativa que se enfoca en la producción de una animación también está en constante diálogo sobre buscar que el espectador tenga un acercamiento con la vida artificial, como se dijo antes, explorando organismos inexistentes generados por computadora, pero esta vez desde una experiencia más contemplativa e inmersiva, en la cual se muestra un recorrido donde se pueden apreciar las texturas de cada objeto, su movimiento, transformaciones, evoluciones y estructuras, además de los sonidos que producen en su entorno, con la intención de crear un espacio inmersivo, desde la composición del video y su parte sonora, que busca ser envolvente.

Otro punto importante de esta investigación es que puede servir de ayuda a quienes estén interesados en el desarrollo de herramientas o add-ons para Blender, ya que cada paso y proceso que me llevó el desarrollar mi propia herramienta se encuentra documentado , así como la explicación de distintas nociones básicas del funcionamiento del software Blender y las herramientas que el mismo nos brinda para este tipo de desarrollos, además de la explicación de cada uno de los elementos o modificadores usados en el proceso de creación a través de la interfaz gráfica de dicho software y también cómo se ejecutan dichos elementos a partir de un código de Python, el lenguaje de programación en el que Blender basa su funcionamiento y por consiguiente sus complementos (add-ons).

Como mencioné anteriormente, el objetivo de este proyecto de investigación es modificar el software Blender para generar una nueva herramienta (add-on) que pueda facilitar el desarrollo de figuras biomorfas 3D de forma procedural y generativa. Con esta nueva herramienta se produjo una obra con una serie de figuras biomorfas modeladas en 3D de forma procedural, la cual consta de una animación o vídeo renderizado con la intención de ser proyectado. Para poder lograr estos objetivos primero hubo que responder algunas preguntas que se plantearon como parte del problema de investigación y que se fueron resolviendo a lo largo de la misma:

¿Cómo usar el software Blender para generar elementos con una estructura y estética de forma procedural y generativa que se basen en el biomorfismo?

Esta pregunta se desarrolló para poder establecer una base en esta investigación y para adquirir y buscar el conocimiento acerca del funcionamiento y las herramientas que ofrece el software de Blender y además conocer las características estéticas del biomorfismo para facilitar su desarrollo y tener cierto control en cuanto la experimentación para crear estos organismos generados por computadora.

Seguido de esto último, otra pregunta que se planteó fue

¿Qué serie de pasos son necesarios para que un elemento adquiera una forma biomorfa? Teniendo el conocimiento de la pregunta anterior, hace más fácil responder esta pregunta, ya que solo hará falta entender el procedimiento jerárquico que se utiliza para el desarrollo de estas figuras, lo que nos servirá para poder experimentar de una forma más

segura con este método, dando la oportunidad de crear variaciones en las figuras generadas por el add-on o cambiando el código fuente de este mismo.

Mi propuesta creativa se basa en generar una herramienta que sea de fácil uso y entendimiento para artistas, diseñadores, ilustradores y modeladores 3D, que ayude en su proceso de desarrollo artístico o para cualquier medio o tipo de expresión sin importar su nivel de experiencia usando este tipo de herramientas. Además, busca ser accesible (de código abierto) y funcional para cualquier computadora capaz de ejecutar Blender, esto gracias a que es un software de código libre que también permite la elaboración de add-ons usando este programa como fuente, ya que cuenta con un editor de textos que está hecho para programar usando el mismo lenguaje (Python). En dicho editor de textos es en donde se programan los add-ons desde su funcionalidad hasta la interfaz, permitiendo así que cualquiera pueda entrar al código fuente de la herramienta y realizar los cambios que desee, ya sea para mejorarla, hacer su propia versión o simplemente experimentar con ella.

Para llegar al desarrollo de mi investigación y propuesta creativa me centré en la experimentación con el software, la investigación en base a la documentación y manuales de Blender, además de adquirir información de una entrevista que formulé hace un par de años a Jakob Stolte, un artista que trabaja con temas de biomorfismo generado de forma procedural y que es parte de mis referentes en esta tesina.

En cuanto a la estructura de este documento, primero hablaré sobre el marco artístico e histórico que existe alrededor del tema del biomorfismo, además de explicar la relación que tienen las obras presentadas con mi proyecto de investigación creación. Posteriormente hablaré del marco teórico y conceptual que sustenta mi investigación, en el que se exploran conceptos vinculados con el desarrollo de mi propuesta creativa.

2. Marco artístico histórico: El papel del Biomorfismo procedural generativo en las prácticas del arte digital.

En la contemporaneidad la práctica del Biomorfismo y la codificación orientada a la creación de elementos biomorfos se ha visto reflejada en el trabajo de varios artistas, diseñadorxs y arquitectxs. A continuación, hablaré de algunos proyectos que tratan el

Biomorfismo generado por computadora de forma procedural y finalmente cómo estos se relacionan con mi investigación y propuesta creativa.

2.1. Strand Tower

Por un lado, los arquitectxs, Testa y Weisser en 2006 crearon Strand Tower. Strand Tower consta de una serie de imágenes renderizadas con el software Weaver, en donde por medio de código simularon el tejido de una fibra, diseñando por medio de la codificación su deshilachamiento, agrupamiento y el tejido (Casey Reas, 2010, pág. 24).

En esta obra se puede observar cómo los arquitectos aprovecharon el software Weisser para poder diseñar una estructura con características propias del biomorfismo a pesar de que el software está orientado a otro tipo de producciones no artísticas. Las imágenes creadas por el software muestran un conjunto de redes o fibras deshilachadas siguiendo patrones curvos, algunos entrelazándose entre ellos y yendo a direcciones aleatorias.

2.2. Tuboid

Los artistas Erwin Driessens y Maria Verstappen realizaron un software para computadoras llamado Tuboid (2000) que crea membranas tubulares (semejantes a estalactitas), ya que se van formando por una capa pequeña en la unta y progresivamente las capas van creciendo hacia abajo. El comportamiento y acomodo de cada capa es diseñado a través de un algoritmo genético que cambia rotaciones y longitudes. Cada forma tiene una construcción única, la cual se puede imprimir y en versiones más recientes se puede navegar a través del modelo 3D gracias al software (Verstappen, 2000).

Se podría decir que los artistas Erwin Driessens y María Verstappen desarrollaron una herramienta a partir de la investigación para el arte, yendo desde la parte escultórica o plástica a la digital. El propio software usa una modificación genética estructural programada, en donde tal como en la vida orgánica, se programan una serie de algoritmos evolutivos que posteriormente construyen un genoma propio y único de cada estructura.

2.3. Aggregation

Andy Lomas es un artista con formación matemática, lo cual lo orilló a estudiar la relación entre las reglas matemáticas simples y su capacidad de crear formas orgánicas semejantes a corales. Aggregation (2005) es una serie de imágenes que muestran estructuras 3D, las cuales fueron creadas a partir de la programación de semillas, las cuales van creciendo con el tiempo, simulando trayectorias de partículas que terminan formando los corales (Lomas, 2005) (Casey Reas, 2010, pág. 166)

Similar a Tuboid, el artista Andy Lomas programó una semilla que se vale de operaciones matemáticas para la creación de estas estructuras únicas y complejas, simulando sistemas de partículas moviéndose en el espacio por un campo de fuerzas.

2.4. Malwarez y Spam Plants

Malwarez es una serie de imágenes renderizadas en 3D que se asemejan a virus y bacterias. El artista visual Alex Dragulescu realizó ingeniería inversa a virus informáticos, códigos spyware y troyanos para encontrar patrones, llamadas API y subrutinas para alimentar un algoritmo que haga crecer los organismos artificiales 3D. (Dragulescu, Alex Dragulescu, 2007).

Dragulescu tomó valores ASCII dentro de mensajes spam para determinar la estructura de una serie de objetos 3D semejantes a flores o plantas, parecido a lo que hizo con Malwarez. (Dragulescu, Alex Dragulescu, s.f.). Alex Dragulescu pudo realizar estas obras biomorfas a partir del arte generativo, recabando datos informáticos de distintos tipos de malwares (tanto en Malwarez y Spamplants), mostrando así otra forma de tratar el biomorfismo a partir de la codificación, o como es este caso, a partir de la decodificación.

2.5. Artificial Ecology

Jakob Stolte es un artista visual que estudia el acercamiento de la naturaleza, lo artificial y el mundo digital a través del biomorfismo, yendo desde la visualización de gráficos 3D, escultura, pinturas y técnicas mixtas con Inteligencias artificiales. Centrándonos en sus trabajos de modelado 3D rápidamente nos podremos percatar de la diversidad de organismos artificiales con los que trabaja, yendo desde la simulación de órganos, corales y de cierto modo, imágenes abstractas que fácilmente nos evocan a una naturaleza desconocida y completamente artificial (Stolte, Art by Jakob Stolte, s.f.).

Me gustaría enfatizar que en este breve recorrido por el biomorfismo y su codificación pudimos ver cómo es que los artistas y arquitectos han tratado este tema usando distintas estrategias, herramientas y métodos que dan como resultado la creación de formas biomorfas generadas proceduralmente, yendo desde aprovechar softwares diseñados para ciertas tareas específicas, pero que nos pueden ayudar a nuestros objetivos, tal como es el caso de Strand Tower, de Testa y Weisser. También podemos hablar de la codificación de algoritmos genéticos o matemáticos generativos, como la obra Tuboid, de los artistas Erwin Driessens y Maria Verstappen, o como lo es Agreggation, de Andy Lomas, que dan como resultado estructuras y figuras biomorfas con un código único. Recabar datos, la reinterpretación y la decodificación para alimentar un algoritmo que cree plantas o virus en 3D es otra forma de tratar el biomorfismo, como lo vimos en las obras Malwarez y Spam Plants de Alex Dragulescu. Finalmente queda hablar de Jakob Stolte, que se enfoca al biomorfismo desde varios ámbitos artísticos con el propósito de acercar a la naturaleza, lo digital y lo artificial en uno mismo. Estas obras y proyectos abordan temas y métodos similares a mi proyecto de investigación/creación y está en constante diálogo con temas de decodificación y reinterpretación de datos, las relaciones entre lo natural y lo artificial, la relación entre lo orgánico y lo digital, además del mismo uso de herramientas y técnicas dentro del desarrollo del proceso creativo.

En cuanto a la exploración de la relación entre lo natural y lo artificial, varios de los proyectos se centran en la relación entre lo natural y lo artificial, el cual es un tema clave en mi investigación. Artificial Ecology de Jakob Stolte, por ejemplo, se enfoca en explorar a la naturaleza desde lo digital y lo artificial haciendo uso de distintas herramientas como el modelado procedural 3D en Blender (como es mi caso). Jakob Stolte no solo se enfoca en desarrollar este tipo de modelos en 3D, sino que también busca desarrollar una estética que evoque a la naturaleza que permita que el espectador reflexione sobre la vida artificial generada por computadora tal cuál como una parte de mi propuesta creativa que está enfocada a la exploración de organismos generados por computadora. De la mano con este tema está la relación entre lo orgánico y lo digital y cómo estos pueden llegar a compartir algunas características, como por ejemplo en la decodificación de algoritmos genéticos, matemáticos, extracción de datos informáticos para desarrollar herramientas procedurales que den resultados similares a la creación y desarrollo de la naturaleza. Ejemplos de esto

son las obras como Aggregation, Tuboid, Spam Plants y Malwarez, en donde se busca desarrollar vida a partir de procedimientos similares al desarrollo genético y su evolución en la naturaleza, en donde (como es el caso de Tuboid y Aggregation) el algoritmo desarrollado se basa en la evolución natural de la genética a través del desarrollo de una serie de algoritmos matemáticos para crear objetos similares a plantas, corales y estructuras creadas por la naturaleza con la característica de que cada una fue creada proceduralmente, dándoles un código genético artificial único. Si bien mi investigación no busca desarrollar un algoritmo matemático o un algoritmo genético, sí se inspira en estos métodos para la creación procedural de figuras biomorfas.

Decodificación, programación y experimentación para el desarrollo de software o herramientas de creación son algunos métodos y temas que dialogan con mi investigación, como el caso de Strand Tower en el que se experimentó con software de arquitectura para desarrollar figuras biomorfas en donde el proceso de investigación involucró la reinterpretación de un software para lograr un objetivo artístico y explorar nuevas posibilidades de desarrollo.

3. Marco teórico conceptual

De que trata esto.

3.1. Orígenes del biomorfismo y sus características.

El artículo “Lo biomorfico como construcción poética de las estructuras de la vida” (Morokawa, 2022, pág. 5) habla de cómo el término “biomorfismo” ha existido en el mundo del arte moderno desde principios del siglo XX, cuando Geoffrey Grigson, un crítico de arte usó la palabra “biomorfismo”. Geoffrey Grigson hizo uso de esta palabra para poder marcar la diferencia entre dos tipos de abstracciones: Las geométricas y las no geométricas, o biomorfas, como el las designaba (Pastoriza, 2019); “Las abstracciones son de dos tipos: las geométricas, las que conducen a la muerte inevitable; y biomórfico. Las abstracciones biomórficas son el comienzo de la siguiente fase central en el progreso del arte” (Grigson, 1935). Geoffrey Grigson en la revista “Axis” (1935) habla de biomorfismo como una “abstracción” que se asemeja o evoca a la vida apartándose de figuras

geométricas complejas o rigurosas y que el biomorfismo vive en parte del arte abstracto en donde “las abstracciones biomórficas son el comienzo de la siguiente fase centra en el progreso del arte” (Grigson, 1935). Es importante mencionar que Grigson se inspiró de los textos de un profesor de zoología llamado Alfred Cort Haddon, quien es la primera persona en proponer el término “biomorfismo” en su libro “Evolution in Art: as illustrated by the Life Histories of Designs”, en donde a partir del arte de Nueva Guinea británica trata de hacer una distinción o clasificación de los motivos decorativos de este arte, en donde a objetos o elementos artificiales los llama “Skeuomorphs” y los derivados de las transformaciones de objetos naturales, los cuales se dividen en “Fisicomorfos”, que se refieren a fenómenos físicos y los “Biomorfos”, que representan seres vivos (Maldonado G. , 2019, págs. 11,12). Por otro lado, el crítico de arte y director del Museo de arte Moderno (MOMA) Alfred H. Barr utilizó el término “biomorfismo” para referirse a elementos curvilíneos, decorativos en vez de estructurales, orgánica en lugar de geométrica, además Alfred Barr organizó la exposición "Cubismo y Arte Abstracto”, en donde se presentaron artistas biomorfos como Jean Arp y Yves Tanguy. Esta exposición ayudó a definir y popularizar el biomorfismo como corriente artística. (Pastoriza, 2019, pág. 12).

En retrospectiva y pensando en las primeras veces en las que el término “biomorfismo” fue mencionado, podemos hacer una relación en varios aspectos más que nada de cómo se compone estéticamente un objeto biomorfo; generalmente se habla del uso de curvas, figuras no geométricas, abstractas, orgánicas en representación a los seres vivos, etc. Con el paso del tiempo, el biomorfismo en el arte ha ido evolucionando poco a poco hasta rozar con temas que pueden tratar sobre la vida artificial y generada por computadora, en donde se busca una relación entre lo orgánico o lo natural con lo digital.

3.2. Vida artificial

En los años noventa, un biólogo evolutivo llamado Tom Ray realizó una investigación que trataba sobre la evolución y el mundo digitales. *Tierra*, como Tom Ray lo llamó, es un modelo que pretendía simular características de la biología con procesos evolutivos basados en conceptos de evolución y replicación. “Ray liberó un programa autorreplicante dentro de un caldo primordial de información que había programado y produjo lo que denominó un "ecosistema", en el cual las "poblaciones" de los "organismos

digitales" podían evolucionar" (Helmreich, 2023). A medida que su investigación avanzaba comenzó a asociar términos y conceptos como "vida" y "mundo artificial", ya que, "Para Ray, los programas de cómputo que se autorreplican -como los virus en las computadoras- pueden considerarse como nuevas formas de vida" (Helmreich, 2023).

En la investigación del biólogo Tom Ray hace un acercamiento entre elementos de la naturaleza, como lo son los ecosistemas u organismos con elementos que son parte de la digitalidad y el desarrollo computacional, pero ¿Qué es la vida artificial? Pues un artículo perteneciente a la "Revista Colombiana de Filosofía de la Ciencia" (2001) se basa en la definición de Christopher Langton, uno de sus fundadores (De la vida artificial) en donde menciona que "la Vida Artificial, es una ciencia que busca comprender a la vida no tal y como es, sino tal y como podría ser" (Maldonado C. E., 2001, pág. 3). Más adelante se sigue mencionando que el desarrollo y la investigación de la vida artificial se es posible gracias al uso de computadoras, con las que buscan desarrollar algoritmos genéticos para crear vida artificial, a los cuales se les suele llamar semillas (Maldonado C. E., 2001, págs. 3,4).

El desarrollo, investigación y simulación de vida artificial por medio de métodos computacionales está en constante diálogo con este tipo de proyectos, en donde se busca traer un fragmento de la naturaleza a la digitalidad. Si bien mi proyecto de investigación creación no se enfoca como tal en desarrollar ecosistemas o códigos genéticos evolutivos, sí busco este acercamiento de lo digital, lo artificial y lo natural en el desarrollo de objetos tridimensionales desarrollados por computadora. Los organismos que estoy creando se rigen en gran medida por este tipo de conceptos y técnicas.

4. Proyecto Creativo: Organismos Inexistentes: Biomorfismo Procedural 3D

4.1. Descripción

Organismos Inexistentes: Biomorfismo procedural 3D consta de una investigación para las artes, de la cual deriva una herramienta que facilite la creación de figuras biomorfas de forma procedural y automática a través del software de modelado Blender. La

herramienta producto de esta investigación permitirá realizar una animación usando modelos creados por dicha herramienta.

Blender es un software de modelado 3D de código abierto que usa el lenguaje de programación Python como código base, lo que permite a "los usuarios más avanzados emplear la API de Blender para secuencia de comandos Python para personalizar la aplicación y escribir herramientas especializadas" (About — Blender.org, 2023)

Este tipo de herramientas se les conoce como "Add-Ons". Mi investigación culminará en la elaboración de un add-on y algunas variaciones de este que estén enfocados en la creación de figuras biomorfas de forma automatizada y aleatoria usando la técnica procedural. Para llegar a este resultado se realizará una serie de experimentaciones sobre un "primitive mesh" de Blender, el cual será transformado con una serie de "operaciones automáticas que alteran la geometría de un objeto" (Modifiers — Introduction, 2023), llamadas "modificadores", los cuales ya vienen integrados en la interfaz gráfica del usuario (GUI) de Blender. El primitive mesh en cuestión será un cubo, ya que, al tener una geometría básica y nada compleja, nos servirá como punto inicial para aplicar los modificadores pertinentes para transformarlo en una figura biomorfa.

Una vez seleccionados los modificadores que van a alterar la malla de nuestro cubo inicial habrá que convertirlos a un código en Python que automatice su funcionamiento, desde la creación del *primitive mesh* hasta la modificación de su geometría de forma procedural y aleatoria. Es necesario convertir cada modificador a código, ya que estos modificadores se deben activar de forma manual desde la GUI del software y del mismo modo, los parámetros de cada modificador deben ser alterados de forma manual, caso contrario a lo que queremos hacer para el funcionamiento automático del add-on y sus variaciones. El siguiente paso de esta parte de la investigación sería programar y exportar una interfaz para el add-on que nos permita ejecutar el código generador de figuras biomorfas en cualquier computadora en la que se instale el add-on de Blender y sea capaz de ejecutar el software.

Cuando el add-on y sus variaciones estén funcionando se utilizarán para generar una serie de figuras biomorfas, las cuales se manipularán para realizar una animación en la que se explorarán las geometrías, texturas y transformaciones de cada "organismo inexistente".

Cuando las figuras que estarán en la animación hayan sido seleccionadas, habrá que separarlas por categorías con respecto a la estructura con la que se han generado. A partir de eso, se animará el movimiento y transformación de cada figura accediendo a los modificadores creados por el add-on, además de la creación de texturas para cada organismo, iluminaciones y movimientos de cámara que nos permitan explorar de forma inmersiva sobre cada figura biomorfa.

El renderizado de la animación se realizaría en el mismo software de Blender, usando la técnica de renderizado en base al trazado de rayos con el que cuenta Blender (cycles), el cual proporciona un resultado gráfico más realista y detallado en cada iluminación, sombreado y superficie de cada textura. Finalmente, se elaboraría un código a base de nodos en el software TouchDesigner que se encargue de generar una serie de sonidos sinusoidales reactivos a los valores RGBA (Red, Green, Blue y Alpha) de la animación ya renderizada, de la cual se van a extraer dichos datos para determinar la amplitud o la frecuencia de cada sonido.

Finalmente, el audio resultado de la programación de TouchDesigner pasará a post producción en el software de edición de audio Reaper para añadir múltiples efectos, como el paneo y la reverberación, para posteriormente integrar el audio final a la animación, que será presentada al público por medio de un proyector.

4.2. Experiencia del público

El principal objetivo de este add-on es garantizar que esté al alcance de todos los usuarios que utilicen el software Blender. Se ha diseñado específicamente como una herramienta de código abierto y gratuita, lo que significa que no importa si la utiliza un artista, programador, diseñador, modelador 3D o simplemente alguien interesado en el desarrollo de add-ons; el usuario de la herramienta tendrá acceso al código fuente, por lo que esta accesibilidad al add-on deja a cualquiera colaborar el proceso de producción y mejora del add-on, permitiendo a la comunidad de Blender el participar en su desarrollo y creación de versiones propias.

El diseño minimalista es una característica fundamental de este add-on, ya que su interfaz muestra un solo botón que se encarga de ejecutar el código automáticamente, lo

que significa que no hay necesidad de tener conocimientos avanzados para poder utilizarlo. Además, para garantizar que cualquier usuario, independientemente de su nivel de experiencia, pueda sacar el máximo provecho de esta herramienta, se ha incluido un pequeño apartado de instrucciones ubicado directamente en la interfaz. Esto asegura que cualquiera pueda entender y utilizar el add-on de manera efectiva.

La aleatoriedad es un pilar importante para el funcionamiento de este add-on. Cada vez que se ejecuta el botón se generará una figura completamente diferente. La razón detrás de esto es que cada modificador ha sido programado para utilizar valores aleatorios en cada ejecución, por lo que el usuario no tendrá que preocuparse en tener que modificar parámetros de entrada o cosas técnicas ni avanzadas.

En resumen, este add-on busca que cualquier persona pueda aprovecharlo, independientemente de su nivel de habilidad, además de experimentar con la generación de figuras aleatorias. La combinación de su accesibilidad, diseño minimalista y libre uso están inspirados en la visión que tiene Blender para con sus usuarios: “Todo el mundo debería tener libertad para crear contenidos CG en 3D, con medios de producción técnicos y creativos gratuitos y libre acceso a los mercados” (About — Blender.org, 2023).

La elaboración de la animación está enfocada en que el espectador pueda tener una experiencia inmersiva en un entorno que presente una experiencia visual similar a la de un recorrido o viaje para conocer los organismos creados por el add-on. Estos *organismos inexistentes* son los que cobran protagonismo, ya que por la composición del video se busca enaltecerlos para que el espectador pueda explorarlos visualmente, desde sus texturas, luces, movimientos y transformaciones. La animación de la cámara es la que hace posible este recorrido, ya que está en constante movimiento alrededor de los *organismos* o los mantiene siempre en foco para captar todo tipo de cambio en las figuras que se presentan en pantalla.

La experiencia también está diseñada para que el espectador pueda escuchar los constantes cambios de la animación, ya que el audio también pretende ser inmersivo, contando con una composición sonora basada en la transformación de las imágenes, además cuenta con efectos de paneo de audio que ayudan a dar la sensación de movimiento de las figuras. Otro efecto con el que cuenta es con el de la reverberación, el cual da la

sensación de que el audio que se reproduce rebota en toda la sala dando una impresión de inmensidad del entorno en el que se reproduce.

La propuesta creativa de este video busca generar en el público una experiencia de Inmersividad, asombro y curiosidad por los organismos biomorfos. La inmersión, total en este mundo desconocido y la combinación de elementos visuales y sonoros buscan crear una experiencia impactante que puede inspirar una mayor apreciación y comprensión de la vida en todas sus formas.

4.3. Organismos Inexistentes: Biomorfismo Procedural 3D como una propuesta creativa de arte digital

Se puede considerar que este proyecto tiene una propuesta creativa propia del arte digital, ya que contiene elementos de creación de contenido artístico en base a la experimentación visual, la inmersión audiovisual, el uso de herramientas digitales, además de promover la investigación para la creación de herramientas artísticas en las que se exploren conceptos y técnicas como el arte procedural y arte generativo.

La creación de contenido artístico en base a la experimentación visual implica la creación y manipulación de formas y organismos que se generan proceduralmente, los cuales pueden seguir siendo manipulados para añadirles más características como en el texturizado, el cómo se comportan estos elementos con las iluminaciones y añadiendo nuevas modificaciones para transformar las figuras, por mencionar algunas. La idea de que en cada ejecución del add-on se produzca una figura completamente diferente pretende fomentar la experimentación y la creatividad en la generación de arte con esta herramienta.

También se puede considerar que es una propuesta de arte digital, ya que se exploran elementos de inmersión y experiencia sensorial a través del uso de herramientas audiovisuales que se pueden considerar como elementos esenciales del arte digital en este tipo de producciones. El diseño de la animación y el audio se combinan para proporcionar una experiencia artística completa y envolvente que enaltecen la experiencia de la animación.

El uso de herramientas digitales para esta propuesta artística se basa en el uso de software de modelado y animación en 3D, como lo es Blender en este caso. El uso de este

tipo de herramientas es fundamental en la creación de arte digital, ya que permite la manipulación y la representación de conceptos de una manera que sería difícil o imposible de lograr en otro tipo de medios y permite explorar conceptos más enfocados al arte digital como los antes mencionados.

La accesibilidad y participación comunitaria está en el énfasis del código abierto y gratuito, que promueve e invita a la participación de la comunidad en la creación y mejora del add-on. Esto se alinea con la idea de que el arte digital puede ser colaborativo y accesible para una amplia gama de artistas, investigadores y desarrolladores. También es importante recordar que el desarrollo de esta herramienta usa de base la creación artística procedural y generativa, en la que no es necesaria una intervención directa a la hora de crear las figuras, si no que dicha creación se basa en algoritmos, procedimientos, reglas específicas y la automatización, siendo estos elementos una variedad de técnicas utilizadas en el desarrollo de este tipo de arte.

4.4. Diagrama de flujo de la propuesta creativa.

5. METODOLÓGIA DEL PROCESO CREATIVO:

El desarrollo de esta investigación y su propuesta artística han pasado por múltiples etapas de desarrollo para llegar a lo que es ahora, incluso siendo propuestas diferentes en cada cierto periodo de tiempo, o por situaciones que se fueron resolviendo para llegar al proyecto que es ahora. Esta investigación fue planteada al principio sobre los entornos virtuales orientados al terror y lo grotesco, en donde se buscaba crear un entorno virtual de apreciación y exploración estética sobre elementos abstractos enfocados al terror.

Otra de las etapas que estuvieron presentes en mi proceso creativo fue una en la que se planteó realizar una animación para un planetario en dónde el tema abordado por la investigación y la propuesta artística estaban definidos por conceptos que tenían que ver con el cómo se comporta el tiempo en lo virtual. Para mí es importante documentar estas etapas y distintas propuestas que tuvo mi proyecto, ya que, aunque cada propuesta inicial

fue distinta una de la otra eso fue justo lo que ayudó a que concretara lo que ahora es esta investigación y propuesta artística.

5.1. Los Primeros Pasos

5.1.1. El terror y los primeros acercamientos al biomorfismo en entornos virtuales e inmersivos.

Al momento de plantearme esta investigación mi enfoque no estaba en el biomorfismo o el desarrollo de una herramienta para el uso artístico, más bien estaba orientada al desarrollo de un entorno virtual que permitiera explorar el tema del terror y lo grotesco como experiencias estéticas en este tipo de medios, los cuales generalmente son representados en videojuegos.

Lo que proponía como hipótesis era que los espacios virtuales pueden ser medios generadores que potencian las experiencias estéticas basadas en el terror, en donde la obra que iba a realizar utilizaría a los espacios virtuales como medio, ya sean entornos virtuales explorables o un art games, con el propósito de que el usuario/espectador sea libre de explorar dichos entornos, los cuales estarían construidos desde la estética del terror en cuanto a la arquitectura del entorno, los modelos 3D presentados (Ya sean de monstruos, figuras abstractas o siniestras) y la ambientación en general, para así realizar un acercamiento del usuario a una experiencia estética. En resumen, lo que proponía era crear un espacio artístico para crear experiencias estéticas aterradoras y dejar que los usuarios o espectadores las exploren.

La importancia de esta primera propuesta en mi proceso creativo está en la forma en la que abordaba la parte artística en cuanto a su estética. Las figuras que presentaba en el entorno, desde el bocetaje hasta las primeras versiones del proyecto, siempre buscaban ser semejantes a objetos orgánicos, en su mayoría parecidos a carne y figuras abstractas. Otro punto de relevancia de este primer proyecto es el hecho de que para poder crear estas figuras terroríficas era necesario que experimentara con las herramientas que me proporcionaba Blender, el software en donde también modelaba en 3D los elementos de mi entorno virtual.

De algún modo y casi sin saberlo, estaba empezando a experimentar y crear elementos que se pueden considerar como biomorfismo. En el proceso de creación de estas figuras me di cuenta de que varias compartían elementos en común en su proceso de creación en Blender, como lo es en los modificadores que llegaba a ocupar; solo hacía falta hacer cambios en ciertos parámetros de dichos modificadores para lograr distintos resultados en las figuras. Si bien, en ese momento no pensé en la creación de un add-on que facilité la creación de estas figuras, si pude ver una relación entre este tipo de figuras y la programación procedural y cómo podrían pasar a ser objetos creados a través de la programación generativa.

Más tarde, este primer proyecto tuvo un cambio abrupto en la investigación y propuesta artística en el que todas las ideas anteriores, incluyendo conceptos y diseño fueron desechados para tomar una propuesta que se basaba en la elaboración de una animación inmersiva para un planetario. Aunque la idea de hacer una obra artística inmersiva seguía en pie, la forma de hacerlo iba a cambiar, incluso el medio en el que se iba a presentar, así que aquí vino el primer cambio u adaptación.

5.1.2. Animación: Nociones de inmersividad y composición para el desarrollo de animaciones enfocadas a generar múltiples estímulos sensoriales

El cambio del proyecto se debió a que a mí y a una compañera nos ofrecieron elaborar y presentar una animación para un planetario, en el que se nos permitía presentar el trabajo como proyecto terminal. Después de platicarlo, aceptamos el proyecto y empezamos con su desarrollo.

El papel que me tocó desempeñar fue el de realizar la parte de la propuesta creativa. Como el proyecto para el planetario estaba planeado para presentarse a un público en el que niñas y niños predominaban, la idea de hacer una animación enfocada a generar experiencias estéticas basadas en el terror (como mi primer proyecto) no iba a ser lo más apropiado, así que se decidió que íbamos a trabajar con los conceptos del tiempo y la virtualidad, y cómo estos dos se comportan el uno con el otro.

Lo importante de la propuesta artística de este proyecto radica en su inmersión, con el objetivo de aprovechar al máximo el planetario. El planetario proyectaría la animación en

su semiesfera con un proyector especial para hacerlo en este tipo de espacios en los que, por la estructura del planetario, es fácil emular una sensación de tridimensionalidad en las imágenes proyectadas y en donde el papel del desarrollo del contenido que se pretendía proyectar jugaba un papel importante, ya que tenía que seguir ciertos criterios tanto en la animación y el renderizado para poder lograr estos efectos, además la proyección iría acompañada de un sonido multicanal o audio 5.1, mejor conocido como audio envolvente, comúnmente utilizado en espacios como salas de cine, en donde se usa un aproximado de 6 canales de audio para hacer u circuito cerrado y crear un efecto 3D en donde se percibe la sensación de que el audio viaja de un punto a otro en la sala.

El planetario se convierte en el epicentro en donde se enfocan las experiencias multisensoriales propulsadas por la Inmersividad que pretendíamos generar con el proyecto, por lo que acudimos al planetario para ver su funcionamiento y aprender de algunas de las técnicas usadas en las animaciones que comúnmente se proyectan en estos espacios. Una de mis primeras impresiones al ver las animaciones que fueron proyectadas era que la mayoría, si no es que, en todas, pretendían hacer sentir a los espectadores que estaban en un viaje o se desplazaban a través del video realizando una especie de recorrido a través de ruinas egipcias o siguiendo el viaje de una montaña rusa, como, por ejemplo. Además, cada animación era presentada en una perspectiva de primera persona, como si nosotros fuéramos la cámara captando todo a nuestro alrededor. Otro punto importante que noté en estas proyecciones es que cuando hay un solo objeto proyectado al cuál seguir era más fácil percibir esa sensación inmersiva y para esto regularmente se proyecta un objeto sobre un fondo negro, como es el caso del viaje en una montaña rusa antes mencionada. En el caso de escenarios más completos, como en el de las ruinas egipcias, esa sensación inmersiva se potenciaba cuando la cámara (o nosotros, en este caso) se paseaba por dentro de las estructuras y nos dejaban ver todo lo que había alrededor, pero siempre en constante movimiento.

Con todo esto en mente comencé a trabajar en propuestas conceptuales para la animación, en donde se haría un recorrido sobre múltiples elementos, como estructuras, objetos, entornos múltiples y espacios abstractos, pero siempre siguiendo la regla de estar en constante movimiento y en medida de lo posible hacer un recorrido desde lo micro hasta

lo macro. Con todos estos elementos y parámetros técnicos establecidos a partir de la observación de las animaciones que se nos presentaron en un principio, comenzamos con el primer y único boceto que se realizó de esta animación en donde al principio se presentaba un universo (muy cliché para un planetario) mientras que la imagen hace un zoom out de a poco para dejar ver que este universo se encontraba dentro de un reloj de cucú como si este universo le sirviera de mecanismo para poder funcionar. El movimiento de la cámara en zoom out no paraba, lo que nos permitiría ver que este reloj de cucú pertenecía a la casa de un pequeño ratón, casa que se ubicaba dentro de un agujero que pertenecía a la pared de una recámara más grande mientras que la cámara seguiría haciendo un recorrido en retroceso por una serie de escenarios hasta llegar al espacio y mostrar el universo con el que comenzó la animación.

Lamentablemente las ideas de animación y la nueva propuesta artística no se ajustaban completamente con la idea original de la investigación, así que, como los White Stripes, nuestro equipo de trabajo se terminó disolviendo. Finalmente tenía que tomar la importante decisión de si seguir con el proyecto de la animación para el planetario tomando en cuenta que tendría que empezar todo desde cero, o de volver a mi antiguo proyecto el cual tenía semanas de retraso. Terminé optando por seguir con la animación del planetario, ya que en mi mente tenía frescas las ideas sobre ese proyecto y además veía un mundo de posibilidades a futuro con ese proyecto.

Ahora el problema del millón era encontrar un proyecto que se pudiera adaptar a la situación en la que me encontraba, en donde de cierta forma tenía una parte resuelta de la propuesta artística pero no una investigación detrás de ello. Estuve repasando trabajos personales anteriores en los que pudiera basarme para formular una nueva investigación, hasta que me encontré con un par de modelados 3D en los que tiempo atrás estuve experimentando para hacer figuras semejantes a carne, corales y estructuras óseas.

5.1.3. Biomorfismo y Jakob Stolte

Mi interés por hacer este tipo de modelados no surgió de la nada, sino que tiempo atrás me había topado con el trabajo de Jakob Stolte, un artista que trabajaba modelando figuras semejantes a corales u organismos marinos, entre muchas otras figuras abstractas. Por medio de un mensaje en su página de Facebook le pregunté si me podría explicar los

métodos que usaba para para producir sus obras y el accedió a darme una breve explicación del método que usaba (el cual explicaré un poco más adelante). Desde entonces y en breves periodos de tiempo comencé a experimentar con este tipo de figuras.

Tomando en cuenta mis primeras propuestas artísticas, de investigación y analizando mis experiencias en proyectos pasados, me planteé realizar una investigación sobre el proceso de desarrollo procedural de figuras orgánicas creadas por computadora en donde después se le permitiera al espectador conocer estos organismos por medio de una animación diseñada para ser proyectada en planetario aprovechando la sensación de inmersión multisensorial que produce este tipo de espacios. Usando esta premisa, mi proyecto de investigación ha ido avanzando y evolucionando hasta convertirse en lo que es ahora.

5.2. Pre-Producción: El Enfoque de la Investigación y la Experimentación.

La base de este proyecto está en la experimentación. Gracias a la orientación que el artista Jakob Stolte me proporcionó tiempo atrás pude comenzar con mis propios experimentos en Blender. Sin embargo, antes de pasar a la explicación de mis primeros experimentos para crear figuras biomorfas, es necesario hacer un repaso por algunos conceptos utilizados en Blender y la mayoría de los softwares de modelado en 3D, con el fin de comprender los siguientes apartados.

Nociones Básicas

En Blender "la creación de una escena 3D necesita al menos tres componentes clave: modelos, materiales y luces" (Introduction — Blender manual, 2023). Por ahora solo explicaré un apartado, el de los modelos:

Mallas.

Los modelos generalmente constan de una estructura conocida como *malla*, la cual es en principio un objeto primitivo, como un cubo, un cilindro, un cono, etc. Estas mallas primitivas, o *primitive mesh* pueden ser modificadas posteriormente para crear estructuras más complejas en su malla (Meshes — Introduction, 2023).

Un cubo, por ejemplo, está conformado por doce aristas, seis caras y ocho vértices, por lo que es considerado como un objeto tridimensional (Primitives — Blender Manual, 2023). Así como el cubo, cada *primitive mesh* u objeto complejo tridimensional se rige por la misma estructura: aristas, caras y vértices.

Los vértices son la parte más importante de una malla, ya que son los que conectan a los demás elementos, por lo que entre más vértices existan en el modelo, la malla será más compleja. En cambio, los bordes son la unión de dos vértices mediante una línea recta. Los bordes se utilizan para crear una cara, la cual el manual de estructuras de Blender la define como "el área entre tres (triángulos), cuatro (cuadrángulos) o más (n-gonos) vértices, con un borde a cada lado. Las caras suelen abreviarse como *tris*, *quads* y *n-gons*". El mismo manual de estructuras explica que la diferencia entre los triángulos y los cuadrángulos es que los triángulos son siempre planos y para la computadora es más fácil calcularlos. Los cuadrángulos son más difíciles de calcular, ya que tienen mayor facilidad de deformación, por lo que es más común que estos sean utilizados en proyectos de modelado complejo y animación (Structure — Blender Manual, 2023).

1.1.1.1. Modificadores

Ahora bien, estas mallas pueden ser alteradas con una serie de herramientas llamadas modificadores. Existen cuatro tipos de modificadores: los de modificación, generativos, de deformación y físicas. Por ahora hablaré de los modificadores del tipo generativos, siendo estos los cuales nos permiten cambiar la forma física de nuestro objeto. Dicho de otro modo: "Los modificadores son operaciones automáticas que afectan la geometría de un objeto de forma no destructiva. Con los modificadores, puede realizar muchos efectos automáticamente que, de otro modo, serían demasiado tediosos de realizar manualmente (como subdivisión de superficies) y sin afectar la geometría base de su objeto" (Modifiers — Introduction, 2023).

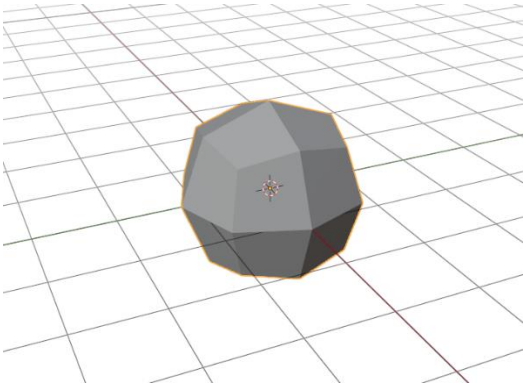
Otra característica de los modificadores es que se pueden cambiar los parámetros de estos antes de ser aplicados de forma definitiva a la geometría del objeto. Esta característica se puede comparar con una máscara o conjunto de máscaras que se pueden sumar una a una para deformar a un *primitive mesh*, como lo sería un cubo, el cual por debajo seguirá

teniendo la geometría de un cubo hasta que estas máscaras sean aplicadas definitivamente aplicando también las deformaciones a la malla de un cubo. Más adelante continuaré explicando el funcionamiento de algunos modificadores generativos y algunos de deformación como son el *subdivision surface* o el *displace*, que mencionaré en breve.

5.2.1. El Método de Jakob Stolte.

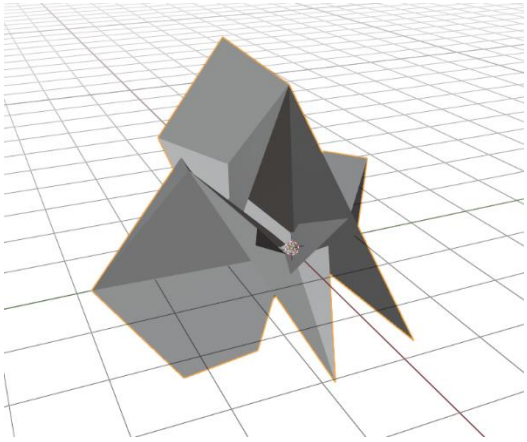
Volviendo a Jakob Stolte, él me explicó que el método que utiliza se basaba en usar objetos con una malla primitiva a los cuales les agrega modificadores de desplazamiento de malla, luego subdivide la malla para que esta tenga más vértices y nuevamente desplazar esa malla subdividida para hacer varios pliegues en el objeto, por último, la malla se subdivide para que el objeto tenga más detalle: "Básicamente, el primer método que se me ocurrió fue desplazar objetos simples con texturas, subdividirlos y luego volver a desplazarlos, hasta que el objeto se pliega sobre sí mismo; si subdivides eso, el resultado será irregular y estará lleno de agujeros" (Stolte, 2020). A continuación, ejemplificaré con imágenes y numeración el proceso del artista Jakob Stolte:

1. Primero se añade un objeto primitivo, en este caso se añadió un cubo al cuál se le aplicó un modificador *subdivision surface* con un valor de 1 para que la malla se subdividiera un poco más generando una geometría un poco más compleja.



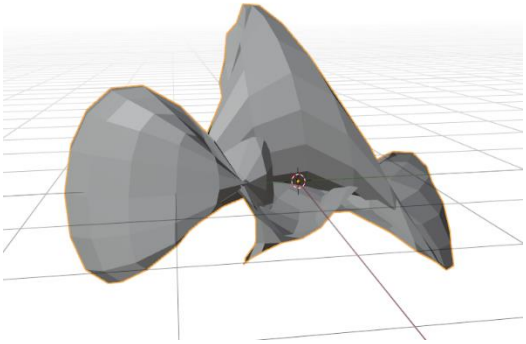
(Stolte, Desarrollo de Organismos Generados por Computadora, 2020).

2. Ahora se agrega un modificador del tipo *displace* que, como su nombre lo dice, desplaza la malla de la geometría. Este desplazamiento lo hace a partir de texturas que pueden ser modificadas. En este ejemplo se usó el modificador *displace* con una textura del tipo *distorted noise*, al cual fueron modificados los parámetros de *amount*, *size* y *nabla* para poder adquirir la geometría presentada en el siguiente ejemplo.



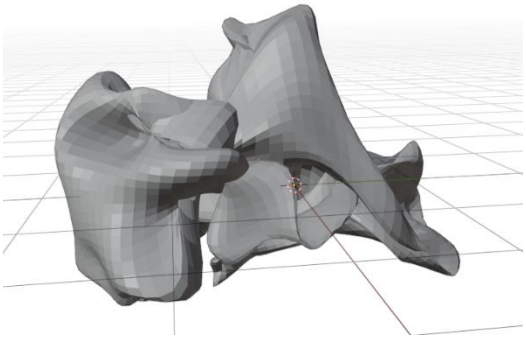
(Stolte, Desarrollo de Organismos Generados por Computadora, 2020)

3. Posteriormente se añade otro *subdivision surface* al objeto para que la malla esté más subdividida y por consiguiente el modelado adquiera mejor calidad en su geometría.



(Stolte, Desarrollo de Organismos Generados por Computadora, 2020)

4. Finalmente se añade una subdivisión más y un *displace* para que la geometría de la figura se vuelva a plegar sobre sí misma. En este último paso se pueden añadir más subdivisiones para que la malla esté más suavizada, pero es importante recordar que entre más subdivisiones existan, la geometría tendrá aún más vértices y será más difícil calcular la malla del objeto para el procesador de la computadora.



(Stolte, Desarrollo de Organismos Generados por Computadora, 2020)

Para lograr estos resultados es necesario que cada modificador esté ordenado uno después del otro, ya que al irlos añadiendo al objeto estos se van sumando uno tras otro haciendo una jerarquía entre ellos. Alterar el orden de esta jerarquía puede modificar la geometría del objeto obteniendo resultados distintos a los que queremos llegar, aunque esto no es del todo malo, ya que también se puede experimentar con la jerarquía para obtener resultados interesantes.

Jakob explica que después empezó a experimentar con modificadores de físicas, en donde usaba sistemas de partículas y de fluidos para poder dar una mejor topología a los objetos: “Más tarde comencé a experimentar con simulaciones, como volver a mallar sistemas de partículas. Luego descubrí que, si dejas que una simulación fluida siga la geometría de un objeto generado, puede ser una forma de darle una mejor topología” (Stolte, Desarrollo de Organismos Generados por Computadora, 2020). A pesar de que no mostró más ejemplos de este último método, para fines prácticos de mis propios

experimentos en Blender no es necesario usar este método, además de que esto traería complicaciones en el desarrollo del add-on que explicaré más adelante.

5.2.2. Mi método.

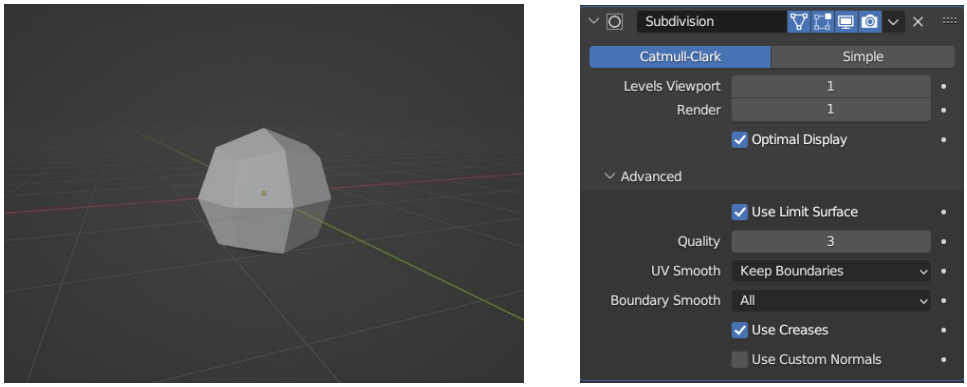
Influenciado por el método de Jakob Stolte por fin comencé con mis propios experimentos. Primero hablaré de los métodos y modificadores que utilicé con las primeras pruebas y posteriormente ejemplificaré y profundizaré en cada una de las etapas.

Primero opté por usar un cubo como base para empezar a agregarle modificadores. El primero de ellos fue un *subdivision surface* para que la geometría del objeto pasara a ser parecida a una esfera con una malla de baja resolución. Después comencé a jugar con un modificador que desaparece las caras de una malla, pero que convierte todos los bordes en polígonos de cuatro lados, por lo que el resultado es similar a una estructura de alambres; el *wireframe*. Ahora la geometría de nuestro objeto se basaba en su misma estructura conformada por los bordes que unen cada vértice. Si a esta nueva estructura se añade otro modificador de subdivisión (*subdivision surface*), nuestro objeto tiende a suavizarse y a adquirir una estética más orgánica en donde cada abertura cuadrada se transforma a un agujero circular, el problema es que la geometría en la estructura del objeto aún se percibe demasiado cuadrada. Para resolver esto y que la geometría comience a parecer a algo más orgánico añadí un modificador que lo que hace es reducir los cuatro bordes y vértices de una cara cuadrada a solo tres, por lo que la malla original pasa a ser más poligonal. Este modificador se llama *decimate* y es preciso que esté antes del *wireframe* para que la estructura de la geometría se muestre irregular.

Se podría decir que llegado a este punto ya tenemos nuestro primer prototipo de figura biomorfa. A esta primera figura le podemos seguir modificando parámetros para ir obteniendo variaciones en la geometría. Ahora bien, si añadimos un modificador de desplazamiento más, la malla de nuestra figura empezará a deformarse con formas más complejas y orgánicas, el problema con esto es que puede que la resolución en la malla se pierda, así que recomiendo poner un *subdivision surface* después del *displace* para que a la hora de editar el desplazamiento de la malla esta se vea suavizada. Antes de pasar a los ejemplos y explicaciones, recapitularé el orden de la jerarquía de los modificadores que le

añadimos a nuestro cubo: *subdivision surface, decimate, wireframe, subdivision surface* y finalmente un *displace* seguido de un *subdivision surface*.

1. Después de añadir el cubo, añadimos un modificador *subdivision surface*, el cual nos permitirá trabajar con una geometría con más definición que el cubo, pero sin llegar a ser tan compleja. El modificador *subdivision surface* "se utiliza para dividir las caras de una malla en caras más pequeñas, dándole una apariencia suave. Le permite crear superficies lisas complejas mientras modela mallas simples de vértices bajos" (Subdivision Surface Modifier — Blender Manual, 2023). Dependiendo del nivel de subdivisiones que se agreguen al cubo este evolucionará su geometría hasta convertirse en una esfera. Se recomienda que este nivel de subdivisiones no exceda de cuatro, ya que si esto pasa es probable que a la computadora le cueste procesar la geometría del objeto al que se la aplicó el modificador. Para estas pruebas, la primera subdivisión tendía el valor de 1, por lo que el cubo adquiere más vértices y evoluciona a una especie de esfera simple, exactamente como en el paso 1 del método de Jakob Stolte.

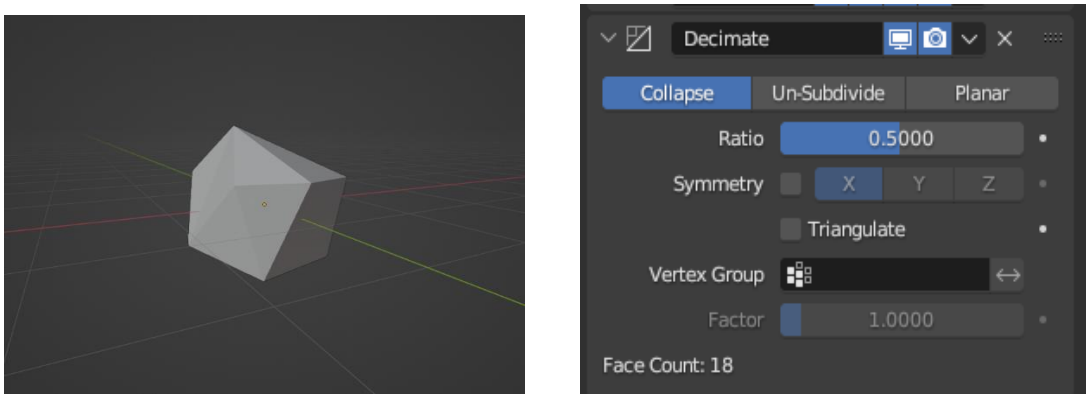


Fuente: Elaboración propia.

Otro punto importante es que el parámetro de *levels viewport* y *render* sean los mismos, ya que en *levels viewport* podemos ver una previsualización de lo que le

hace el modificador a nuestro objeto, en cambio, el valor del parámetro *render* es el que le aplicará estos cambios a la malla a la hora de renderizar una imagen o animación.

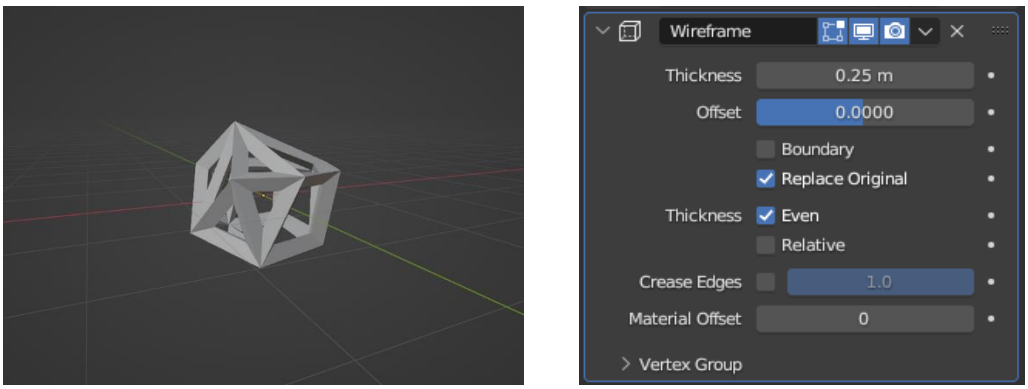
2. Posteriormente se añade un modificador *decimate* para que la geometría del objeto se deforme un poco y su estructura sea irregular dando una sensación de ser un objeto más orgánico. *Decimate* es un modificador que "permite reducir el número de vértices/caras de una malla con cambios mínimos de forma" (Decimate Modifier — Blender Manual, 2023). Para hacer esto, el modificador *decimate* convierte una cara cuadrada (de cuatro vértices) a una triangular, por lo que la malla se ve afectada pasando a ser tener una estructura poligonal, pero tratando de mantener la geometría original. A este proceso se le llama *colapsar* o *collapse*, en el cual se puede cambiar la relación (*ratio*) de este colapso que va del 0 al 1, en donde en el valor 1 no existe ninguna deformación de la malla, pero a medida que este número va disminuyendo (en números decimales) la malla va reduciendo vértices, convirtiendo caras cuadradas a triangulares.



Fuente: Elaboración propia.

En este caso, el valor asignado al parámetro *ratio* será de 0.500, permitiendo que nuestra geometría no se deforme extremadamente, pero si lo suficiente para cambiar la estructura original.

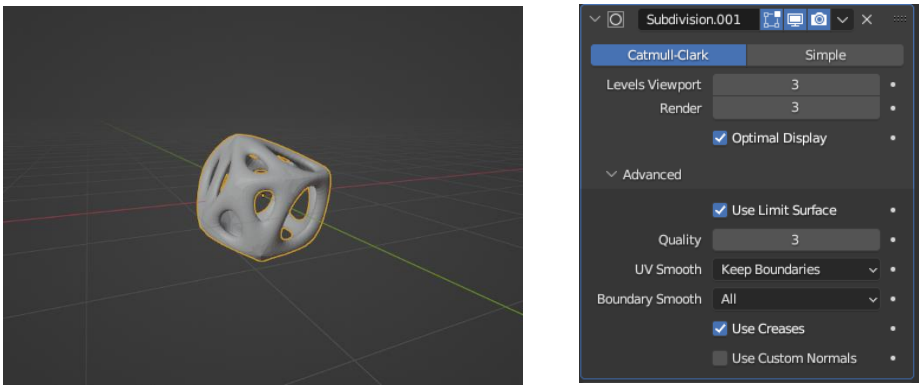
3. Ahora es turno del *wireframe*, que como se dijo con anterioridad, hace que la geometría de nuestro objeto ahora parezca una estructura de alambres. Según el manual de Blender, "el modificador *wireframe* transforma una malla en una estructura alámbrica iterando sobre sus caras, recopilando todos los bordes y convirtiéndolos en polígonos de cuatro lados. Tenga en cuenta el hecho de que su malla debe tener caras para poder estructurarlas. Puede definir dinámicamente el grosor, el material y varios otros parámetros de la estructura alámbrica generada a través de las opciones de modificación dadas" (Wireframe Modifier — Blender Manual, 2023). Este modificador contiene múltiples opciones para poder manipularlo, de las cuales solo nos centraremos en dos; *Thickness* y *offset*. El *thickness* es una opción del modificador que nos permitirá cambiar el tamaño o grosor de la estructura "alámbrica" de nuestra geometría. Por otro lado, el *offset* nos sirve para cambiar la orientación hacia la que se genera la estructura, es decir, determina si la estructura aparece por dentro o por fuera de la estructura, yendo de un valor de -1 a 1. Si el valor del *offset* es igual a 0 entonces la estructura se creará alrededor de la malla original (Wireframe Modifier — Blender Manual, 2023).



Fuente: Elaboración propia.

Para este ejemplo se usó un *thickness* de 0.25 y un *offset* de 0.0, pero estos valores pueden variar dependiendo del resultado al que se quiera llegar.

4. Ahora se añade otro modificador de *subdivision surface* para que la geometría de nuestra figura tenga más definición. En este paso ya tenemos una primera versión semejante a una figura biomorfa. Es importante recordar que los parámetros de cada modificador pueden ser alterados para mostrar distintos resultados, sin olvidar que la jerarquía de estos modificadores también influye en el proceso de deformación de la malla.

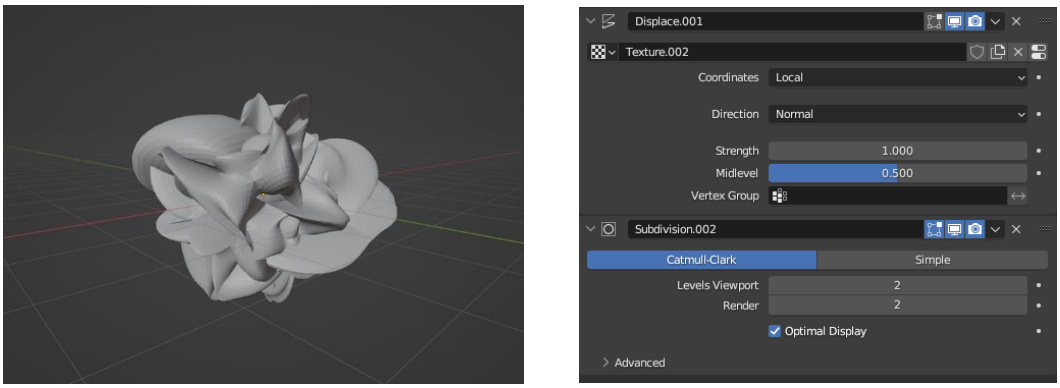


Fuente: Elaboración propia.

Es importante entender que la diferencia entre el primer *subdivision surface* y este es que el primero se añadió para "darle cuerpo" a nuestro cubo. Este segundo *subdivision surface* es para aumentar la definición de la malla de nuestra geometría. Es por eso por lo que el valor de nuestro *subdivision surface* es de 3.0. Hasta este punto, nuestra figura tiene un valor de 4 subdivisiones

5. Para cumplir con este primer prototipo y darle más complejidad a la geometría hace falta añadir un *displace*, el cual desplaza los vértices de la malla de nuestro objeto. Para hacer esto, el *displace* extrae los valores de una textura llamada *displacement*

map para realizar la deformación de la malla. Para este ejemplo se estará utilizando una textura de desplazamiento del tipo *musgrave*. No hay que olvidar que hay que agregar un *subdivision surface* para poder ver más suavizada la malla de nuestra figura, en este caso el valor es de 2 subdivisiones.

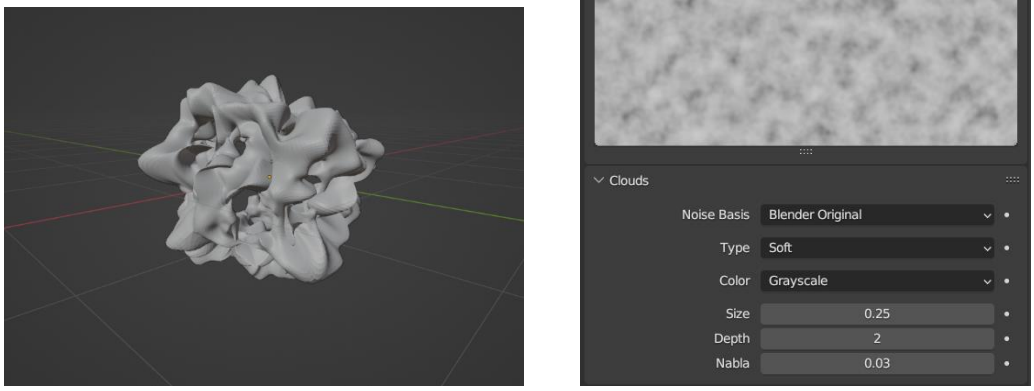


Fuente: Elaboración propia.

Como mencioné, el modificador displace funciona gracias a una textura que determinará el desplazamiento de la malla. Estas texturas, también conocidas como *displacement maps* vienen en distintos tipos, los cuales son; *blend, clouds, distorted noise, image or movie, magic, marbe, musgrave, noise, stucci, voronoi y wood*. Un *displacement map* consta de una imagen en escala de grises que se utiliza para modificar la forma o la altura de una superficie tridimensional. Esta imagen en escala de grises contiene información que indica cuánto desplazar o elevar cada punto de la superficie 3D en una dirección específica. La escala de grises se interpreta de la siguiente manera: los tonos más claros (con valor de 1.0) representan desplazamientos hacia arriba, mientras que los tonos más oscuros (valor de 0.0) representan desplazamientos hacia abajo. También es posible realizar esas transformaciones de desplazamiento gracias al RGB de una textura, en donde estos canales de color se pueden utilizar para desplazar los ejes X, Y y Z de los vértices en direcciones locales (Displace Modifier — Blender Manual, 2023).

Cada textura va a afectar a la malla de nuestra figura de formas diferentes, además de que cada textura tiene propiedades que van variando, dependiendo de qué tipo de mapa

de desplazamiento se esté utilizando en el modificador. Por ejemplo, si cambiamos la textura *musgrave* a *clouds* de nuestro *displace* a nuestro modificador del paso cinco, la geometría de nuestra figura tendrá un resultado diferente:



Fuente: Elaboración propia

Gracias a la experimentación con estos modificadores logré comprender su uso y comportamiento para poder lograr mi objetivo de crear figuras biomorfas, así que con este primer prototipo comencé a explorar los múltiples resultados que podía obtener cambiando los parámetros de los modificadores de *decimate*, *wireframe*, *displacement* y sus *displacement maps*. Con esto ya estaba listo para empezar a seleccionar los distintos tipos de organismos que pasarían a convertirse parte del add-on.

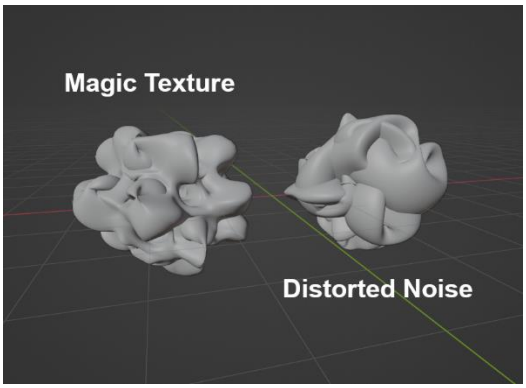
Producción.

Me aventuré a cambiar la jerarquía que había propuesto al principio para lograr resultados distintos en la geometría de mis objetos. Al final logré concretar cuatro tipos de figuras biomorfas diferentes; Organismos semejantes a plantas o flores, organismos semejantes a corales y (en este caso) componentes de organismos como lo pueden ser órganos, carne o estructuras óseas. Finalmente hay un organismo que no se clasifica en ninguno de los tres

tipos de figuras, ya que este organismo comparte múltiples características de los organismos antes mencionados. A continuación, explicare de forma breve en qué consisten estos cuatro tipos de organismos o figuras biomorfas:

Flores o plantas.

Los organismos con una estructura similar a flores o plantas se construyen a partir del primer prototipo con ligeras variaciones que se encuentran en los modificadores de *wireframe* y *displace* en su apartado de texturas en donde estas llegan a variar. Para este tipo de organismos se utiliza una *magic texture* o *distorted noise* en el modificador *displace* para que la geometría de la figura genere pliegues semejantes a pétalos.



Fuente: Elaboración propia

Corales.

Para este tipo de estructuras, se han diseñado dos estilos. Una en donde el *primitive mesh* inicial no es un cubo, sino más bien un cilindro el cual tendrá una jerarquía de modificadores distinta a las demás, ya que se compone de un *subdivision surface* con valor de una subdivisión, un *decimate,* un *wireframe*, otro *subdivision surface con el valor de tres subdivisiones* y finalmente un *simple deform* con la propiedad de *twist* en un ángulo de *360°*. Por otro lado, el segundo estilo de las figuras tipo coral se realiza cambiando ciertos parámetros específicos en la estructura de los modificadores del primer prototipo; La

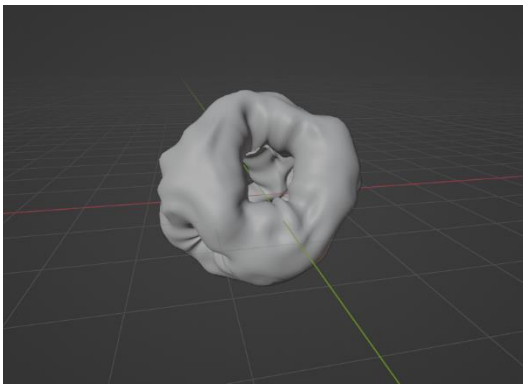
primera subdivisión del cubo tiene que ser mayor a 1 (en este caso es 2), mientras que el *thickness* de nuestro *wireframe* debe estar en un rango de entre 0.1 y 0.5 para que la geometría de la malla no se distorsione de más perdiendo sentido de la figura. Finalmente, nuestra segunda subdivisión debe tener un valor mayor a 2 subdivisiones (en este caso es de 4) y nuestro *displace* debe contar con una textura de desplazamiento del tipo *clouds* en la que su tamaño puede ir variando de un rango entre .10 y 2.0.



Fuente: Elaboración propia

Componentes de organismos.

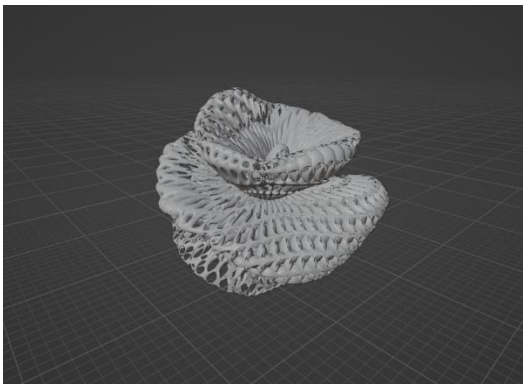
Este tipo de figuras que tratan de simular carne u órganos se componen de 5 modificadores: *decimate, wireframe, subdivision surface, displace* y un segundo *subdivision surface*. Las variaciones más importantes se encuentran en el primer *subdivision surface*, en el que el valor de subdivisiones debe ser mayor a 2 (en este caso es de 3), mientras más subdivisiones tenga mayor rugosidad y pliegues tendrá la geometría del objeto. La segunda variación importante está en que la textura que debe usar el *displace* tiene que ser específicamente la de *magic texture*, ya que es la que se apega mejor al tipo de figura que se busca presentar.



Fuente: Elaboración propia

Organismo sin clasificación.

Este organismo no tiene clasificación como tal, ya que comparte características de distintas clasificaciones. Por ejemplo, la geometría es similar a la de u coral, pero la forma que esta puede llegar a crear se asemeja a la figura de una flor, .En cuanto a la composición de modificadores, la jerarquía quedaría de la siguiente manera; *subdivision surface* con un valor mayor a 4 (como en este ejemplo), un *displace* con una textura del tipo *marble*, un *wireframe* con un *thickness* que lleve un rango de entre 0.001 y 0.200, posteriormente un *subdivision surface* y finalmente un *displace* con una textura que puede variar entre la *voronoi texture* o *clouds texture* (en este ejemplo se usa *clouds texture*, pero el resultado es bastante parecido en ambos casos).



Fuente: Elaboración propia

5.3. Producción: Programación y Animación

5.3.1. Programando el Add-on.

Ahora es turno de escribir un código de Python que ejecute cada acción para crear estas figuras. El primer paso sería escribir un código simple en el que estén programadas las siguientes acciones básicas:

```
#Importar librerías#

#Módulo API Python de Blender de nivel superior#
import bpy

# Añadir primitive mesh

#Se añade un primitive mesh (cubo) a la escena con una escala
uniforme de 1 metro y una posición en 0#

bpy.ops.mesh.primitive_cube_add(location=(0,  0,  0),  scale=(1,
1, 1))

# Añadir Modificadores

#Se añade un modificador de subdivisión surface#
bpy.ops.object.modifier_add(type='SUBSURF')

#Se añade el modificador wireframe#
bpy.ops.object.modifier_add(type='WIREFRAME')

#Se añade el modificador decimate#
bpy.ops.object.modifier_add(type='DECIMATE')

#Se añade el modificador subdivision dos veces para ser sumadas#
bpy.ops.object.modifier_add(type='SUBSURF')

bpy.ops.object.modifier_add(type='SUBSURF')
```

Ahora bien, en base a este primer código el siguiente paso sería poder acceder a parámetros específicos de los modificadores para poderlos manipular mediante código, además esto nos ayudará a reducir la repetición de ciertos modificadores.

```
import bpy
import mathutils
from random import randint

#Añadir primitive mesh
```

```
#Añade un cubo
bpy.ops.mesh.primitive_cube_add(location=(0, 0, 0))

# Ahora se asigna el cubo (Objeto activo) a una variable llamada
Cube para poderla llamar más adelante y aplicarle los
modificadores
cube = bpy.context.active_object


# Añadir Modificadores

# Subdivision Surface

# Agrega el modificador Subdivision Surface y establece el nivel
de subdivisión en 2
subdivision_modifier  =  cube.modifiers.new(name="Subdivision",
type='SUBSURF')

#Establece el nivel de subdivisiones (Del render y el viewport)
(No pasar de 1 al ser la primer subdivision)
subdivision_modifier.levels = 1
subdivision_modifier.render_levels=1



# Decimate

# Añade el modificador de decimate y se establece un radio (que
puede variar)
decimate_modifier     =     cube.modifiers.new(name="Decimate",
type='DECIMATE')
decimate_modifier.ratio = 0.5



# Wireframe

# Crea el modificador de wireframe y se guarda en una variable
wireframe_modifier = cube.modifiers.new(name = "Wireframe", type
= 'WIREFRAME')

# Accede al parámetro thickness y cambia el valor original por
un valor que puede variar
wireframe_modifier.thickness = .35



# Subdivision Surface

# Segunda Subdivision (para darle forma más suabe, no exceder
de 3)
subdivision_modifier  =  cube.modifiers.new(name="Subdivision",
type='SUBSURF')
subdivision_modifier.levels = 3
subdivision_modifier.render_levels= 3
```

```
# Displace

# Se crea un modificador displace y se guarda en una variable
displace_modifier = cube.modifiers.new(name="Displace", type =
'DISPLACE')

# Ahora creamos una textura para el displace de tipo “Clouds” y
la guardamos en una variable.
textura_clouds      =      bpy.data.textures.new(name="Clouds",
type='CLOUDS')

# Se modifica la escala de la textura (el valor puede variar)
textura_clouds.noise_scale = 0.94

# Ahora guardamos nuestra textura modificada para que nuestro
displace lea sus datos
displace_modifier.texture = textura_clouds

# Configuramos la fuerza del modificador
displace_modifier.strength = 1



# Subdivision Surface

    # Tercer Subdivision (para darle forma más suabe, no exceder
de 1)
subdivision_modifier  =  cube.modifiers.new(name="Subdivision",
type='SUBSURF')
subdivision_modifier.levels = 1
subdivision_modifier.render_levels= 1



 #Shade Smooth

 #Finalmente se le hace un Shade Smooth a la geometría de nuestra
figura para que la figura sea lisa
bpy.ops.object.shade_smooth()
```

Ahora que ya logramos acceder a las propiedades de nuestros modificadores habrá que hacer que cada número tenga un valor aleatorio, para que cada vez que se ejecute el código este cree figuras distintas, además de hacer que tengan una posición aleatoria en el espacio.

```
#Importar librerías#

#En este caso se importarán librerías que nos ayuden a calcular
valores aleatorios para poder asignarlos a las variables que
determinarán los cambios en algunos de los modificadores, ya que
habrá otros en los que el random ya lo tengan integrado#
 import bpy
 import mathutils
```

```
import random
from random import randint

# Se crean tres variables de posición aleatoria entre un
rango de -3 a 3 para los ejes X, Y y Z de nuestro cubo
px = randint (-3 ,3)
py = randint (-3 ,3)
pz = randint (-3 ,3)

# Se crea una variable aleatoria para el ratio del
modificador displace
random.ratio = random.uniform (.3, 1)

# Se crean variables aleatoria para el thickness, el offset
del modificador wireframe
random.thicknness = random.uniform (.10, 1)
random.offset = random.uniform (.1, 1)

# Finalmente se crean variables aleatorias para deformar el
texture clouds
random.noise_scale = random.uniform (.30, 2)
random.noise_depth = randint (0, 24)

# Y para deformar el displace
random.strength = random.uniform (.5, 2.2)
random.midlevel = random.uniform (.45, .5)


#Añade un cubo

# Añadir un cubo con una posición inicial aleatoria
bpy.ops.mesh.primitive_cube_add(location=(px, py, pz))

# Asignar el objeto activo a una bariable llamada Cube
cube = bpy.context.active_object



# MODIFICADORES

# Subdivision Surface

# Agrega el modificador Subdivision Surface
subdivision_modifier =
cube.modifiers.new(name="Subdivision", type='SUBSURF')

#Establece el nivel de subdivisiones (En este caso se
asignará un valor aleatorio de subdivisiones en un rango de
1 a 2)
subdivision_modifier.levels = randint (1,2)


# Decimate

# Añade el modificador de decimate y se establece un ratio
con un valor aleatorio en cada ejecución
decimate_modifier = cube.modifiers.new(name="Decimate",
type='DECIMATE')
```

```
decimate_modifier.ratio = random.ratio


# Wireframe

# Crea el modificador de wireframe y guárdalo en una
variable
wireframe_modifier = cube.modifiers.new(name = "Wireframe",
type = 'WIREFRAME')
# Accede al parámetro thickness y cambia el valor original
por una variable con valor aleatorio (Lo mismo para el
offset)

wireframe_modifier.thickness = random.thicknness
wireframe_modifier.offset = random.offset



# Subdivision Surface

# Segunda Subdivision (para darle forma mas suabe, no
ecceder de 3)
subdivision_modifier =
cube.modifiers.new(name="Subdivision", type='SUBSURF')
subdivision_modifier.levels = 3



# Displace

displace_modifier = cube.modifiers.new(name="Displace", type
= 'DISPLACE')
textura_clouds = bpy.data.textures.new(name="Clouds",
type='CLOUDS')

# Se eforman los valores del scale y depth del texture
clouds para hacer una textura aleatoria en cada ejecución.
textura_clouds.noise_scale = random.noise_scale
textura_clouds.noise_depth = random.noise_depth
displace_modifier.texture = textura_clouds

# Configura la fuerza del modificador
displace_modifier.strength = random.strength

#Shade Smooth

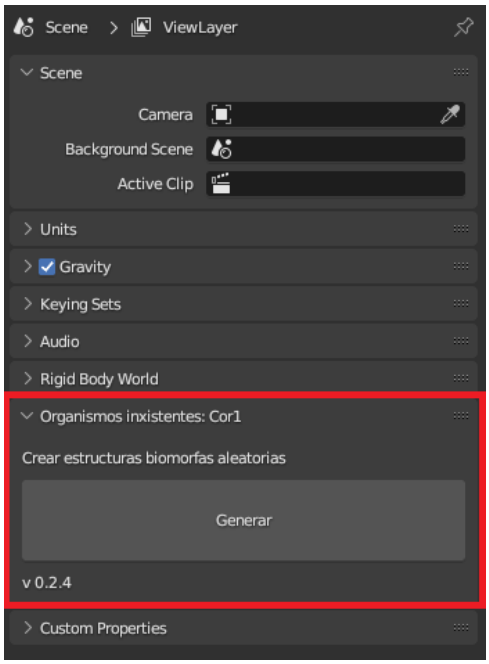
#Se le hace un Shade Smooth para que la figura sea lisa
bpy.ops.object.shade_smooth()
```

Este código es la base para nuestro add-on, ya que con este código cada figura que queramos crear va a ser aleatoria. También este código nos va a servir de ayuda para poder modificarlo y así desarrollar el código de cada

clasificación de figuras, en este caso se hizo el código para las figuras biomorfas tipo corales.

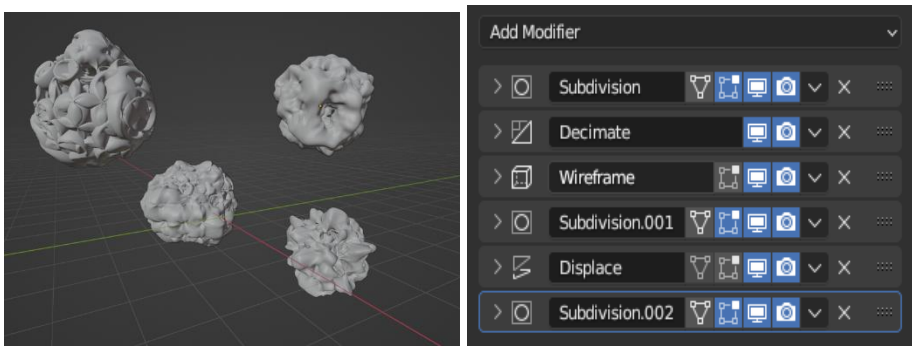
Aunque este código funcione, aún no hemos terminado de desarrollar nuestro add-on, ya que falta hacer una interfaz para Blender que nos permita ejecutar este código desde un botón y no desde la terminal. Para esto hice uso de un código hecho para funcionar como base de interfaces para add-ons. El código fue desarrollado por *SoyKhaler* (2022), un usuario de Blender que desarrolló esta plantilla para el uso libre. El código final con el código base para add-on de *SoyKhaler* estará en el apartado de anexos (***Anexo 1***).

Una vez instalado el add-on nos aparecerá un nuevo botón en la interfaz de Blender en el apartado de *scene*.



Fuente: Elaboración propia.

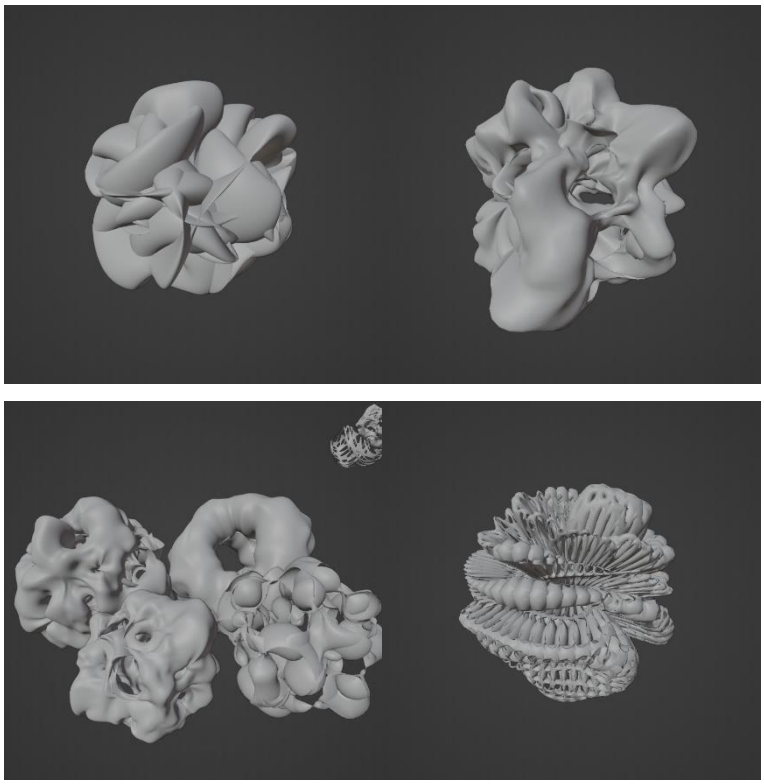
Este botón se encargará de crear una figura biomorfa cada vez que este sea presionado. Por otro lado, cada figura aún tendrá disponible la opción de que cada modificador de estas pueda seguir siendo manipulado y alterado por el usuario.



Fuente: Elaboración propia

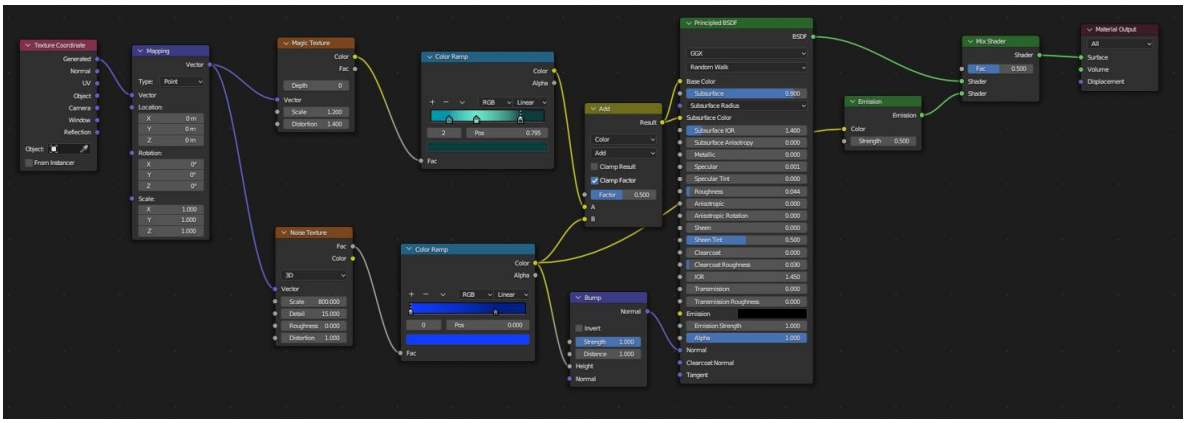
5.3.2. Haciendo la animación

Ahora es turno de desarrollar una animación como parte final de la propuesta creativa. Para poder generar el contenido de la animación utilicé el add-on que desarrollé y variaciones de este para poder hacer los distintos tipos de figuras que se mostraron a lo largo de este documento; Plantas o flores, corales o estructuras óseas, orgánicos, semejantes a carne y biomorfismo abstracto.



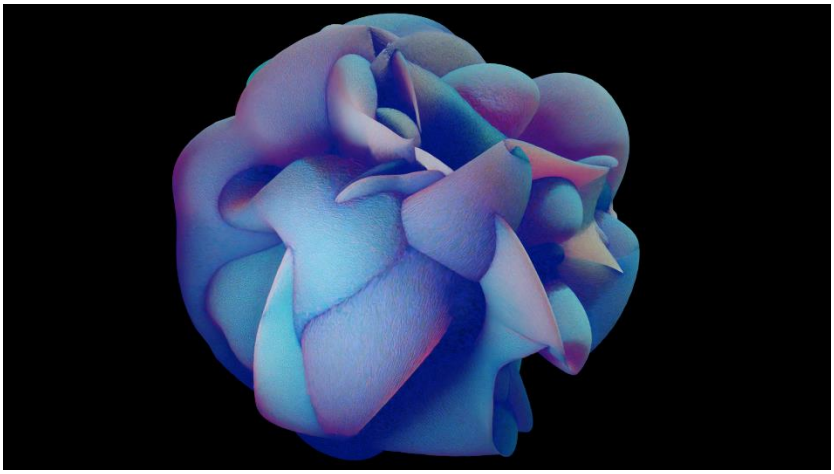
Fuente: Elaboración propia.

Ahora toca darle un texturizado a cada una de estas figuras, pero no cualquier tipo de texturizado, si no uno que también nos remita a la naturaleza, por lo que comencé a desarrollar texturas con degradados suabes, rugosidades, transparencias y patrones irregulares de color en la textura como suele pasar en algunas plantas o flores. Para hacer esto hice uso de una herramienta de texturizado propia de Blender que se basa en Nodos para crear múltiples tipos de materiales. Para la primera figura realicé este árbol de nodos el cual trata de rescatar todas las propiedades mencionadas con anterioridad:



Fuente: Elaboración propia.

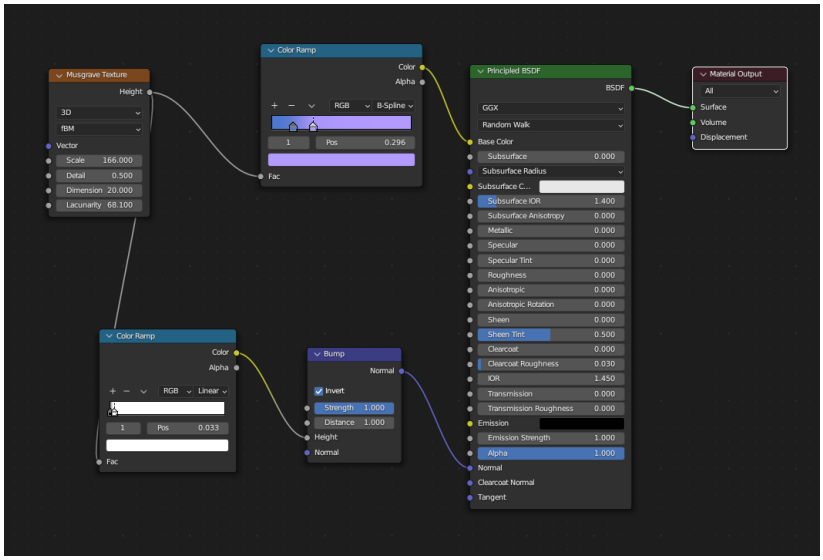
Dando un resultado en el material de la figura de esta forma:

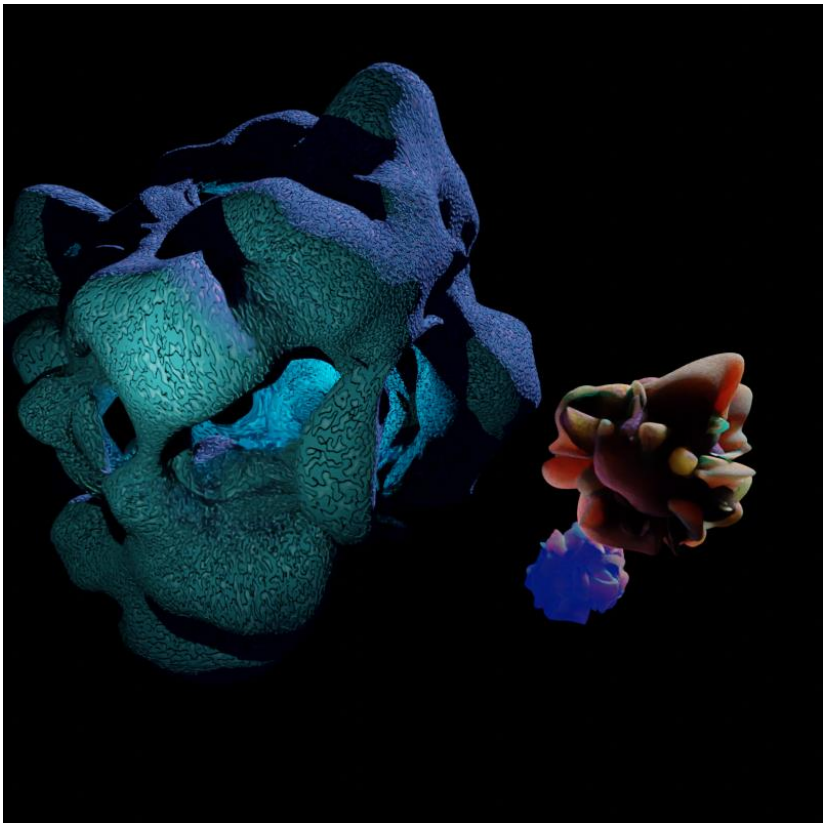


Fuente: Elaboración propia

A partir de aquí cada textura o material se basaría en una lógica parecida para desarrollar este tipo de texturas, algunas más simples que otras, pero cumplen con su

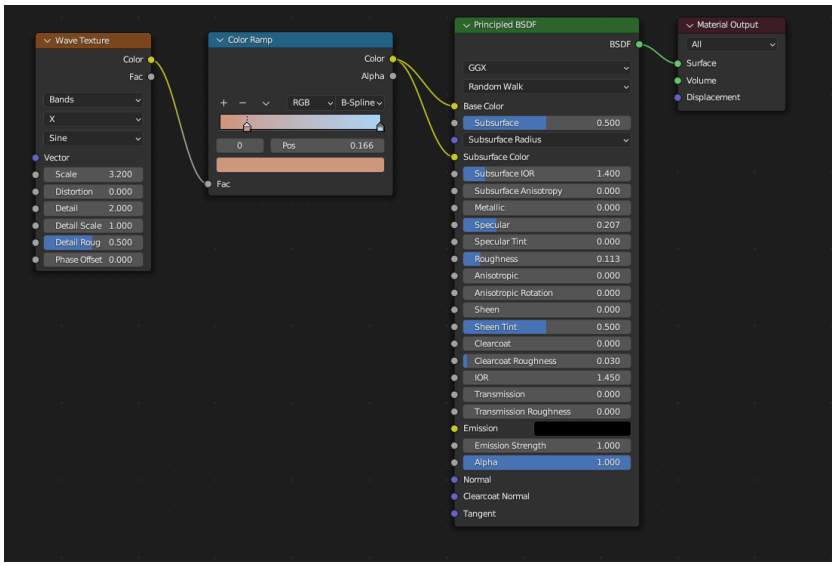
cometido, como es el caso de los corales, en donde no hay gran complejidad la composición de su texturizado pero el resultado trata de ser fiel a la realidad:

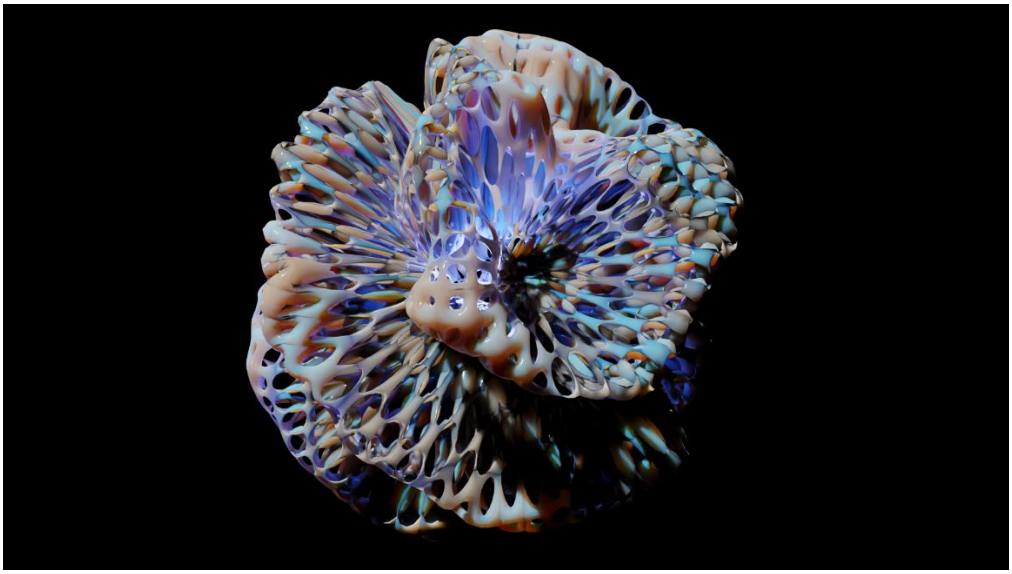




Fuente: Elaboración propia

Otro caso parecido al anterior es el de una figura presentada que no tiene clasificación como tal y que es la que hace más alusión a un biomorfismo abstracto. Esta figura tiene un material que cuanta con un degradado sencillo de dos colores el cual sigue un patrón de ondas, pero su material es ligeramente traslúcido, lo que permite el paso de la luz provocando que se creen distintas tonalidades en la figura. A continuación, la composición de nodos y el resultado de la figura:

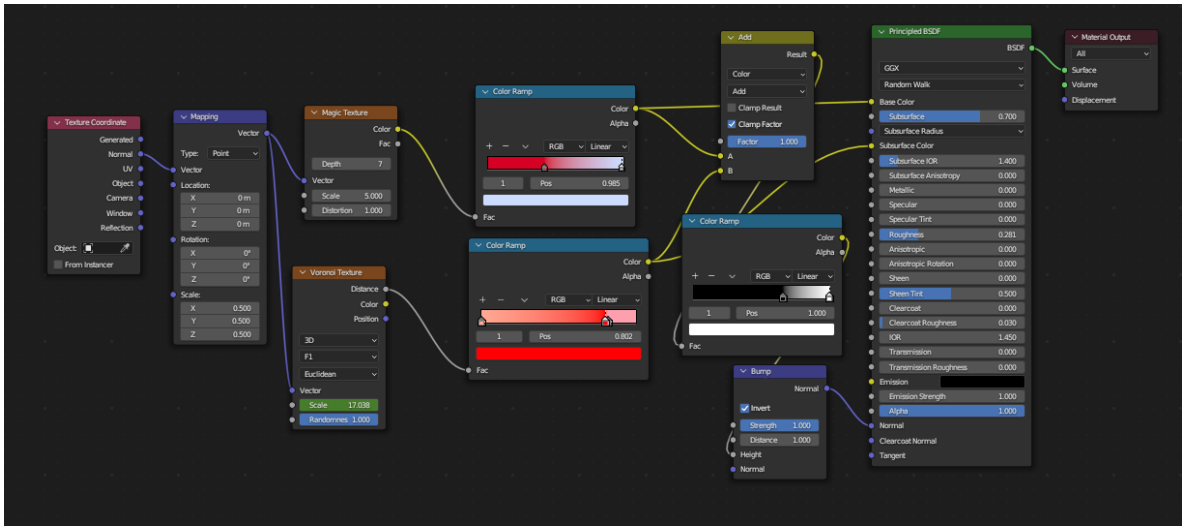




Fuente: Elaboración propia.

Finalmente, las texturas de carne se basan en un solo mapa de nodos, pero cada una tiene variaciones en algunas texturas aplicadas y rugosidades. Con estas figuras se buscaba

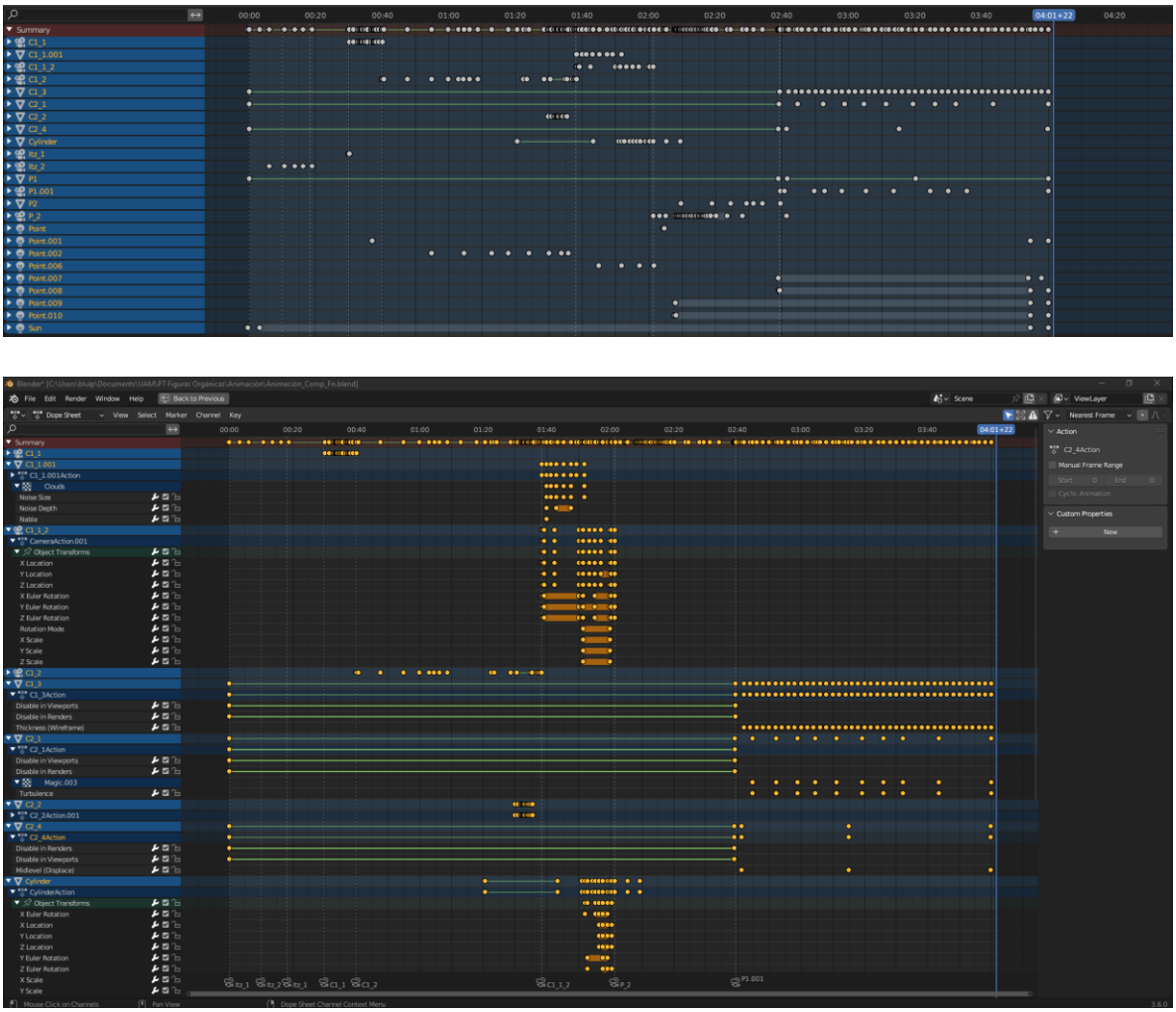
que el material fuera ligeramente traslúcido, con tonos rosas o corales y de momentos con tonos rojizos y con bastantes rugosidades.





Fuente: Elaboración propia.

Teniendo todo el texturizado, el siguiente paso sería animar cámaras, agregar luces a la escena, animar parámetros de los modificadores de las figuras para que estas se muevan e incluso animar las texturas. Finalmente se animaron alrededor de 25 objetos para hacer un total de 7250 frames, lo que son aproximadamente cuatro minutos de animación a una taza de 30 frames por segundo.



Fuente: Elaboración propia

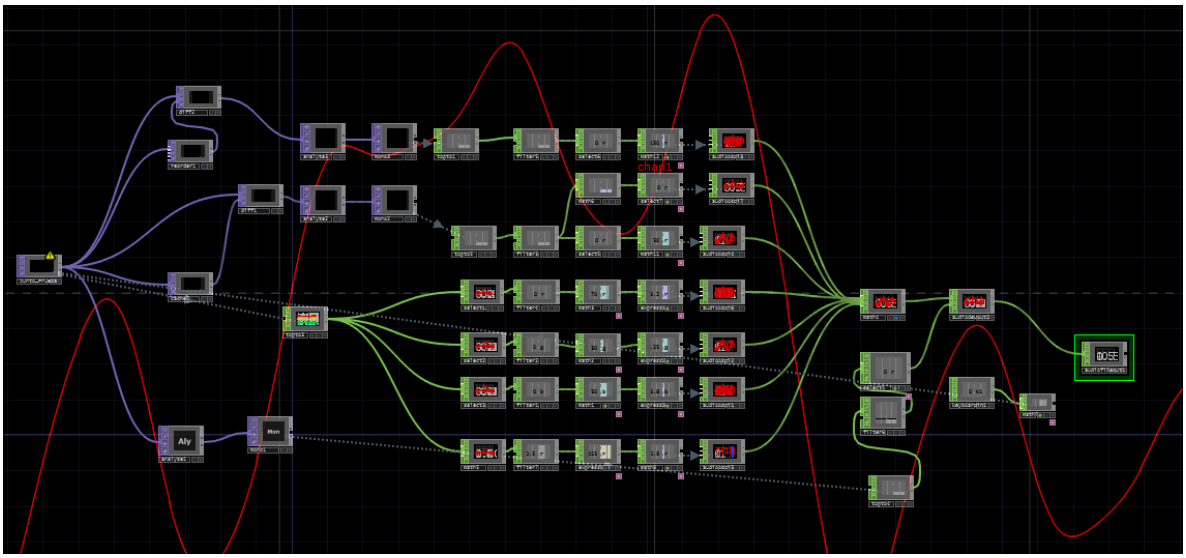
Esta animación fue renderizada usando el motor Cycles que viene implementado en Blender, el cual da resultados más realistas en la animación ya que basa su iluminación el trazado de rayos, el cual hace que cada rayo emitido por una fuente de luz rebote en cada superficie tal cual y como se hace en la vida real. Para esto la animación tuvo que ser

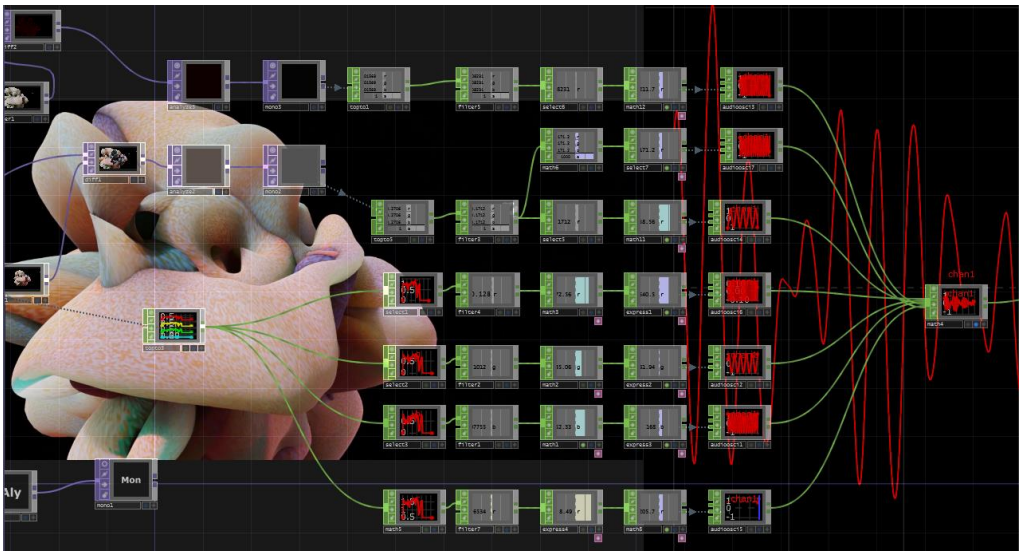
renderizada en tres computadoras, por lo que la animación se fragmentó en tres partes (una para cada computadora) y posteriormente estas se unirían en la post producción.

Cada equipo contaba con dos tarjetas gráficas NVIDIA GeForce RTX 3090, un procesador 12th Intel Core i9-12900k y 128 GB de RAM instalada, por lo que renderizar esta animación fue relativamente más fácil que hacerlo en una computadora con especificaciones más bajas e imposible con un ordenador común.

5.4. Post Producción

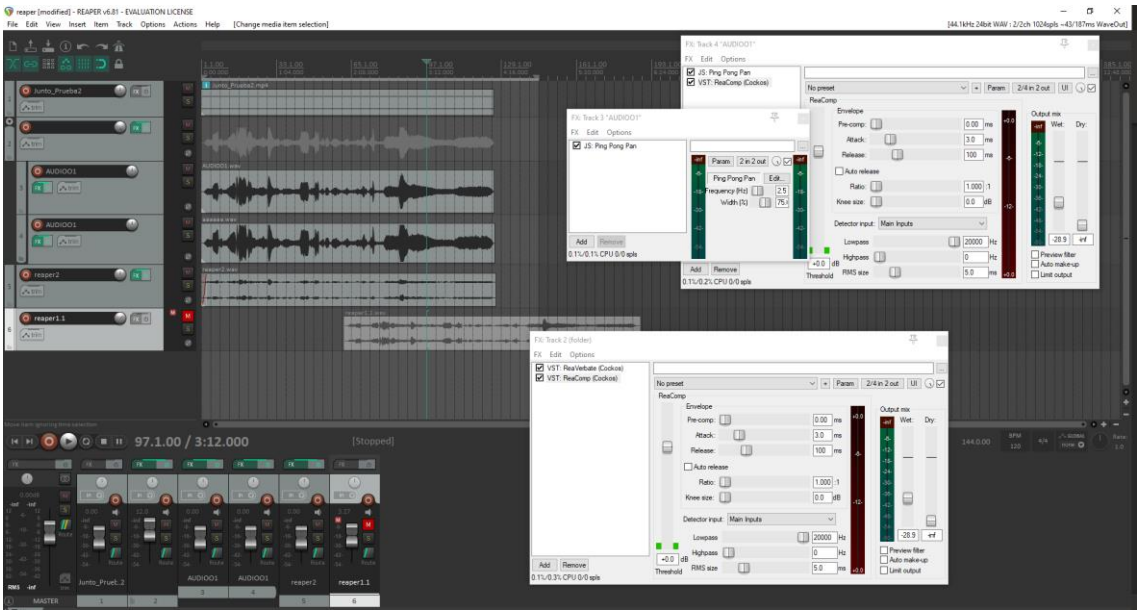
Con la animación ya renderizada solo hace falta unirla con un programa de edición de video y corregir detalles para dejar limpia la imagen. Esto es necesario ya que la animación pasará por un código hecho en TouchDesigner, el cual extraerá los datos del RGB de la animación para convertirlos en valores numéricos que pasarán a ser frecuencias de sonios sinusoidales, haciendo que estos estén en constante cambio dando una sensación de movimiento de la imagen y espacialidad del sonido (el cual también será retocado más adelante). A continuación, el árbol de nodos desarrollado para TouchDesigner:





Fuente: Elaboración propia

Para finalizar con la parte sonora solo haría falta exportar el archivo y pasarlo por un editor de audio para añadirle efectos de reverberación, delay, ecualizadores y paneos de bocina.



Fuente: Elaboración propia

Finalmente, solo haría falta montar el audio con el video en el editor de video y finalmente exportarlo para pasar al montaje, el cuál se haría de dos formas: Una proyección y en formato de video en web (en este caso montado en YouTube).

El montaje de la animación se realizó en la exposición de proyectos terminales de Arte Digital EAD 08 realizada en el Museo Zanbatha de Lerma de Villada, Estado de México.



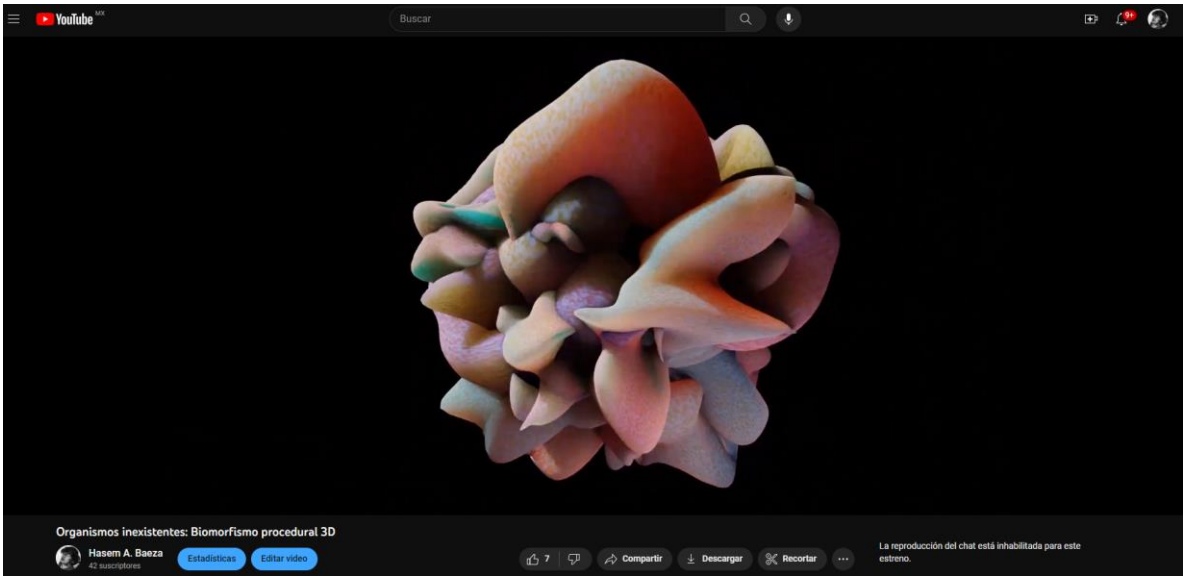
Fuente: Elaboración propia

El espacio en el que estuvo montada la obra contaba con un proyector que mostraba la animación, además de dos bocinas conectadas en estéreo para que los efectos de paneo pudieran ser percibidos por el público espectador.



Fuente: Elaboración propia

El público podía entrar a la sala de proyecciones y ver el video de forma libre, además este contaba con un código QR que dirigía a la animación montada en YouTube que se estrenó el mismo día de la inauguración de la exposición:



https://www.youtube.com/watch?v=MGAdG6s5xB4

Fuente: Elaboración propia, sacado de YouTube

6. Conclusiones finales

Concluyendo con este trabajo, se espera que con esta herramienta desarrollada se impuse y se de apoyo en el desarrollo de artistas, diseñadores y/o modeladores 3D en su proceso de creación, además de incentivar el desarrollo de más herramientas de este tipo y de código abierto para que puedan estar al alcance de todos. Organismos Inexistentes: Biomorfismo Procedural 3D es una investigación que se realizó pensando en la comunidad y en el desarrollo de investigaciones para las artes o de uso artístico explorando temas que tienen que ver con la creación procedural y generativa a partir de procesos por computadora.

Se invita a que se tome el código fuente del add-on que desarrollé y sea manipulado, reprogramado, mejorado o adaptado a nuevas herramientas para la comunidad artística. La animación se introdujo como parte de la propuesta creativa para poder mostrar cómo se pueden usar este tipo de herramientas para crear expresiones artísticas, como lo fue en este caso, tratando de hacer un acercamiento de la vida artificial, la naturaleza y la digitalidad por medio de procesos generativos y procedurales.

7. Bibliografía

Blender Foundation. (2023). *About — Blender.org*. Obtenido de https://www.blender.org/about/

Blender Foundation. (2023). *Decimate Modifier — Blender Manual*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html

Blender Foundation. (2023). *Displace Modifier — Blender Manual*. Obtenido de https://docs.blender.org/manual/es/3.6/modeling/modifiers/deform/displace.html

Blender Foundation. (2023). *Introduction — Blender manual*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/introduction.html

Blender Foundation. (2023). *Meshes — Introduction*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/introduction.html

Blender Foundation. (2023). *Modifiers — Introduction*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/modifiers/introduction.html

Blender Foundation. (2023). *Primitives — Blender Manual*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/meshes/primitives.html

Blender Foundation. (2023). *Structure — Blender Manual*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/meshes/structure.html

Blender Foundation. (2023). *Subdivision Surface Modifier — Blender Manual*.

Blender Foundation. (2023). *Wireframe Modifier — Blender Manual*. Obtenido de https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/wireframe.html

Casey Reas, C. M. (2010). *Form Code, in design, art, and architecture.* New York: Princeton Architectural Press.

Dragulescu, A. (2007). *Alex Dragulescu*. Obtenido de Malwarez: https://sq.ro/malwarez.htm

Dragulescu, A. (s.f.). *Alex Dragulescu*. Obtenido de Spam Plants: https://sq.ro/spam-plants.htm

Grigson, G. (Julio de 1935). A Quarterly Review Of Contemporary "Abstract" Painting & Sculpture. (M. Evans, Ed.) *Axis, No. 3*, 8 - 10.

Helmreich, S. (29 de Octubre de 2023). *La vida artificial*. Obtenido de Desacatos: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1607-050X2000000300008

Lomas, A. (2005). *Andy Lomas Morphogenetic Creations*. Obtenido de Aggregation: https://www.andylomas.com/aggregationImages.html

Maldonado, C. E. (2001). *La heurística de la vida artificial*. Obtenido de Redalyc.org: https://www.redalyc.org/articulo.oa?id=41400504

Maldonado, G. (2019). *Biomorfismo 1920 - 1950.* Madrid, España: Galería Guillermo de Osma.

Morokawa, R. L. (29 de Agosto de 2022). O BIOMÓRFICO COMO CONSTRUÇÃO POÉTICA DA SÉRIE ESTRUTURAS DA VIDA. *Revista Da Fundarte, 50*(50). doi:10.19179/rdf.v50i50.1106

Pastoriza, F. R. (28 de Octubre de 2019). *Periodistas en Español* . Obtenido de Al rescate del biomorfismo: https://periodistas-es.com/al-rescate-del-biomorfismo-135431

SoyKhaler. (5 de Septiembre de 2022). *GitHub - Soykhaler/Addon_PlantillaBlender: plantilla base para hacer addons sin conocimientos de programacion en Python*. Obtenido de GitHub: https://github.com/soykhaler/Addon_PlantillaBlender

Stolte, J. (27 de 10 de 2020). Desarrollo de Organismos Generados por Computadora. (H. A. Baeza, Entrevistador)

Stolte, J. (s.f.). *Art by Jakob Stolte*. Obtenido de Artificial Ecology: https://www.art-by-jakob-stolte.de/

Verstappen, D. &. (2000). *Driessens & Verstappen*. Obtenido de Tuboid: https://notnot.home.xs4all.nl/tuboid/tuboid.html

8. Anexos

Anexo 1. Código base del Add-On:

```
bl_info = {
    "name": "Organismos Inexistentes: Cor1",
    "author": "Hasem A. Baeza",
    "version": (0,2, 4),
    "blender": (3, 6, 0),
    "location": "Render",
    "description": "Herramienta para crear estructuras y formas biomorfas",
    "warning": "",
    "doc_url": "",
    "category": "Add Mesh",
}

import bpy
import mathutils
import random
from random import randint

#funciones

    #bpy.context.object.modifiers["Wireframe"].thickness = 0.035



def main(context):
#-----------------------------------------------------SCRIPT AQUI--------------------
-----------------------------------------
    px = randint (-3 ,3)
    py = randint (-3 ,3)
    pz = randint (-3 ,3)
    random.thicknness = random.uniform (.10, 1)
```

```
    random.ratio = random.uniform (.3, 1)
    random.offset = random.uniform (.1, 1)
    random.noise_scale = random.uniform (.30, 2)
    random.noise_depth = randint (0, 24)
    random.strength = random.uniform (.5, 2.2)
    random.midlevel = random.uniform (.45, .5)

#Añade un cubo

    # Añadir un cubo
    bpy.ops.mesh.primitive_cube_add(location=(px, py, pz))
    # Asignar el objeto activo a una bariable llamada Cube
    cube = bpy.context.active_object

# MODIFICADORES

    # Subdivision Surface

    # Agrega el modificador Subdivision Surface y establece el nivel de subdivisión
en 2
    subdivision_modifier = cube.modifiers.new(name="Subdivision", type='SUBSURF')
    #Establece el nivel de subdivisiones (No pasar de 1 al ser la primer
subdivision)
    subdivision_modifier.levels = randint (1,2)

    # Decimate

    # Añade el modificador de decimate y se establece un radio (que puede variar)
    decimate_modifier = cube.modifiers.new(name="Decimate", type='DECIMATE')
    decimate_modifier.ratio = random.ratio


    # Wireframe

    # Crea el modificador de wireframe y se guardalo en una variable
    wireframe_modifier = cube.modifiers.new(name = "Wireframe", type = 'WIREFRAME')
    # Accede al parametro thickness y cambia el valor original por un valor que
buede variar
    wireframe_modifier.thickness = random.thicknness
    wireframe_modifier.offset = random.offset

    # Subdivision Surface

    # Segunda Subdivision (para darle forma mas suabe, no ecceder de 3)
    subdivision_modifier = cube.modifiers.new(name="Subdivision", type='SUBSURF')
    subdivision_modifier.levels = 3

    # Displace

    displace_modifier = cube.modifiers.new(name="Displace", type = 'DISPLACE')

    textura_clouds = bpy.data.textures.new(name="Clouds", type='CLOUDS')
    #textura_clouds para una segunda version, mientras se desactiva#
    #textura_clouds.noise_type = 'HARD_NOISE'#
    textura_clouds.noise_scale = random.noise_scale
    textura_clouds.noise_depth = random.noise_depth
    displace_modifier.texture = textura_clouds
```

```
# Configura el tamaño del modificador
    displace_modifier.strength = random.strength

# Tercer Subdivision (para darle forma más suabe, no exceder de 3)
    subdivision_modifier = cube.modifiers.new(name="Subdivision", type='SUBSURF')
    subdivision_modifier.levels = 1


    #Shade Smooth

    #Se le hace un Shade Smooth para que la figura sea lisa
    bpy.ops.object.shade_smooth()



#------------------------------------------------------------INTERFAZ GRÁFICA-----------
-----------------------------------------

class SimpleOperator(bpy.types.Operator):
    """Tooltip"""
    bl_idname = "object.simple_operator"
    bl_label = "Generar"

    def execute(self, context):
        main(context)
        return {'FINISHED'}

def menu_func(self, context):
    self.layout.operator(SimpleOperator.bl_idname, text=SimpleOperator.bl_label)

# Register and add to the "object" menu (required to also use F3 search "Simple
Object Operator" for quick access)
def register():
    bpy.utils.register_class(SimpleOperator)
    bpy.types.VIEW3D_MT_object.append(menu_func)


def unregister():
    bpy.utils.unregister_class(SimpleOperator)
    bpy.types.VIEW3D_MT_object.remove(menu_func)


if __name__ == "__main__":
    register()

    # test call
    #bpy.ops.object.simple_operator()

#interfaz grafica


class LayoutDemoPanel(bpy.types.Panel):
    """Creates a Panel in the scene context of the properties editor"""
    bl_label = "Organismos inxistentes: Cor1"
    bl_idname = "SCENE_PT_layout"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
```

```
    bl_context = "scene"

    def draw(self, context):
        layout = self.layout

        scene = context.scene



        # Big render button
        layout.label(text="Crear estructuras biomorfas aleatorias")
        row = layout.row()
        row.scale_y = 3.0
        layout.label(text="v 0.2.4")
        row.operator("object.simple_operator")



def register():
    bpy.utils.register_class(LayoutDemoPanel)


def unregister():
    bpy.utils.unregister_class(LayoutDemoPanel)


if __name__ == "__main__":
    register()
```