



# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Tesis Doctoral

DISEÑO GENERATIVO ASISTIDO POR COMPUTADOR.  
DESARROLLO DE UNA HERRAMIENTA INNOVADORA  
DE AYUDA A LA EXPLORACIÓN DE SOLUCIONES  
CONCEPTUALES EN PRODUCTOS.

Programa de Doctorado en Diseño, Fabricación y Gestión  
de Proyectos Industriales.

Doctorando: Gonzalo Rafael Acosta Zazueta

Director: Jorge Alcaide Marzal

Valencia, enero de 2024



## AGRADECIMIENTOS

Expreso mi profundo agradecimiento a mi familia, en especial a mi padre y madre, por su apoyo incondicional, enseñanzas valiosas y creer en mí de manera constante. A mi hermana, Eira, le agradezco por ser mi ejemplo de fortaleza y por demostrarme cómo enfrentar con determinación todos los desafíos que se nos presentan.

Agradezco a Manuel por su constante apoyo, disposición para escuchar y valiosas opiniones. Quiero expresar mi profundo agradecimiento a Carlos, quien no solo ha sido una guía excepcional, sino también un amigo invaluable cuya amistad ha marcado positivamente mi trayectoria. También, a Ricardo por compartir sus sabios consejos, que han sido de gran utilidad en mi camino.

Quiero dedicar un reconocimiento especial a mi director, Jorge Alcaide Marzal, cuya perseverancia, colaboración y paciencia infinita han sido fundamentales en este viaje. Agradezco su guía constante y sus sabios consejos a lo largo de los años que ha abarcado este doctorado.

Agradezco al Consejo Nacional de Ciencia y Tecnología de México (CONACYT) por su valioso respaldo económico, el cual ha sido fundamental para la realización de esta tesis doctoral.

## RESUMEN.

En el transcurso de la presente tesis doctoral, se llevó a cabo la tarea de analizar el desarrollo del diseño generativo a lo largo del tiempo y en diversos campos de aplicación. Durante esta exploración, se identificó una carencia significativa de herramientas CAD de apoyo en la etapa temprana del diseño conceptual. La ausencia de recursos para facilitar la creación de diseños conceptuales básicos y la generación de diversas soluciones de diseño se presenta como un problema evidente en este contexto. El objetivo principal de la investigación es desarrollar una herramienta que aborde eficientemente esta carencia y proporcione asistencia al diseñador durante estas tareas cruciales del proceso creativo.

La metodología empleada incluyó una exhaustiva búsqueda bibliográfica para comprender el desarrollo del diseño generativo en sus distintas etapas y campos de aplicación. Se identificaron los componentes principales que componen la estructura de objetos y se propuso una clasificación de estructuras básicas que abarca una amplia variedad de productos. Además, se estableció un flujo de trabajo que sirvió como guía para la implementación de la estructura básica, volúmenes, subestructuras y elementos adicionales.

Los resultados destacan la implementación exitosa de un algoritmo en Rhino-Grasshopper, un lenguaje de programación visual, que permitió la generación de soluciones de diseño variadas para una diversidad de objetos. El diseño se enriqueció mediante componentes programados en Python, facilitando la implementación de subestructuras y elementos adicionales, tanto esenciales como no esenciales.

En las conclusiones, se subraya la eficacia de la herramienta en la mejora del diseño generativo, atendiendo factores transversales como la estética, el aprovechamiento del uso del ordenador en la etapa temprana del diseño y la estabilidad de las soluciones propuestas. La contribución de esta investigación se refleja en la creación de una herramienta robusta y eficaz, aportando un nuevo enfoque en el ámbito de las herramientas de diseño en la etapa conceptual.

Palabras clave: Diseño Generativo, herramientas CAD, diseño conceptual, estructuras básicas, soluciones de diseño.

## RESUM.

En el transcurs de la present tesi doctoral, es va dur a terme la tasca d'analitzar el desenvolupament del disseny generatiu al llarg del temps i en diversos camps d'aplicació. Durant aquesta exploració, es va identificar una manca significativa d'eines \*CAD de suport en l'etapa primerenca del disseny conceptual. L'absència de recursos per a facilitar la creació de dissenys conceptuals bàsics i la generació de diverses solucions de disseny es presenta com un problema evident en aquest context. L'objectiu principal de la investigació és desenvolupar una eina que aborde eficientment aquesta manca i proporcione assistència al dissenyador durant aquestes tasques crucials del procés creatiu.

La metodologia emprada va incloure una exhaustiva cerca bibliogràfica per a comprendre el desenvolupament del disseny generatiu en les seues diferents etapes i camps d'aplicació. Es van identificar els components principals que componen l'estructura d'objectes i es va proposar una classificació d'estructures bàsiques que abasta una àmplia varietat de productes. A més, es va establir un flux de treball que va servir com a guia per a la implementació de l'estructura bàsica, volums, subestructures i elements addicionals.

Els resultats destaquen la implementació reeixida d'un algorisme en \*Rhino-\*Grasshopper, un llenguatge de programació visual, que va permetre la generació de solucions de disseny variades per a una diversitat d'objectes. El disseny es va enriquir mitjançant components programats en \*Python, facilitant la implementació de subestructures i elements addicionals, tant essencials com no essencials.

A les conclusions, se subratlla l'eficàcia de l'eina en la millora del disseny generatiu, atenent factors transversals com l'estètica, l'aprofitament de l'ús de l'ordinador en l'etapa primerenca del disseny i l'estabilitat de les solucions proposades. La contribució d'aquesta investigació es reflecteix en la creació d'una eina robusta i eficaç, aportant un nou enfocament en l'àmbit de les eines de disseny en l'etapa conceptual.

Paraules clau: Disseny Generatiu, eines CAD, disseny conceptual, estructures bàsiques, solucions de disseny.

## ABSTRACT.

In the course of this doctoral thesis, the task was undertaken to analyze the development of generative design over time and in various fields of application. During this exploration, a significant lack of CAD support tools in the early stage of conceptual design was identified. The absence of resources to facilitate the creation of basic conceptual designs and the generation of diverse design solutions presents itself as an evident problem in this context. The main objective of the research is to develop a tool that efficiently addresses this deficiency and provides assistance to the designer during these crucial tasks of the creative process.

The methodology employed included a thorough literature review to understand the development of generative design in its different stages and fields of application. The main components composing the structure of objects were identified, and a classification of basic structures covering a wide variety of products was proposed. Additionally, a workflow was established as a guide for the implementation of the basic structure, volumes, substructures, and additional elements.

The results highlight the successful implementation of an algorithm in Rhino-Grasshopper, a visual programming language, which allowed the generation of varied design solutions for a diversity of objects. The design was enriched through components programmed in Python, facilitating the implementation of both essential and non-essential substructures and elements.

In the conclusions, the effectiveness of the tool in improving generative design is emphasized, addressing cross-cutting factors such as aesthetics, leveraging computer usage in the early stage of design, and the stability of the proposed solutions. The contribution of this research is reflected in the creation of a robust and effective tool, providing a new approach in the field of design tools in the conceptual stage.

Keywords: Generative Design, CAD tools, conceptual design, basic structures, design solutions.

## ÍNDICE.

Introducción General. ....	1
1. El Diseño Generativo. ....	4
1.1. Gramáticas de formas ..... 6	6
1.2. Mejoramiento de forma..... 13	13
1.3. Algoritmos genéticos..... 14	14
2. El Uso de Aplicaciones para el Diseño Generativo..... 18	18
3. Planteamiento e Hipótesis. .... 26	26
4. Desarrollo. .... 31	31
4.1 Estructura del producto. .... 36	36
4.2 Subestructuras. .... 37	37
4.3 Elementos adicionales..... 38	38
4.4. La creación de los componentes..... 38	38
4.5. Selección del Software CAD ..... 40	40
5. Diseño de la Herramienta ..... 44	44
5.1 El algoritmo ..... 50	50
5.2 El grupo de componentes ..... 54	54
5.2.1 Componentes subestructurales. .... 54	54
5.2.2 Componentes de generación de formas..... 57	57
6. Aplicación Práctica y Resultados..... 63	63
7. Conclusiones..... 81	81
8. Contribuciones Científicas..... 83	83
9. Bibliografía ..... 85	85
Anexo A: Código Python de Componentes ..... 96	96
Anexo B: Manual de Uso de Componentes..... 110	110

## ÍNDICE DE FIGURAS.

Figura 1. Factores transversales. Fuente: Elaboración propia. ....	29
Figura 2. Cuerpo principal (izquierda), secundario (centro) y de conexión (derecha). ....	32
Figura 3. Elemento extensional (izquierda), superficial (centro) y masivo (derecha).....	33
Figura 4. A la izquierda detalles esenciales de una cocina, a la derecha los no esenciales. ....	33
Figura 5. Vertical extensional. ....	45
Figura 6. Vertical superficial. ....	45
Figura 7. Vertical masivo ....	46
Figura 8. Extensional añadido. ....	46
Figura 9. Superficial añadido. ....	47
Figura 10. a) Masivo añadido. b) Masivo compacto ....	47
Figura 11. a) Horizontal masivo. b) Horizontal cóncavo. ....	48
Figura 12. a) Masivo añadido. b) Masivo compacto. ....	49
Figura 13. Masivo complejo. ....	49
Figura 14. Cóncavo parcial. ....	50
Figura 15. Diagrama de la clasificación. Fuente: Elaboración propia. ....	50
Figura 16. Flujo de trabajo. Fuente: Elaboración propia. ....	51
Figura 17. Creación de línea auxiliar a partir de estructura básica. ....	55
Figura 18. Inserción de punto auxiliar. ....	55
Figura 19. Array points a través del mismo componente. ....	56
Figura 20. Curvas proyectadas sobre volúmenes. ....	56
Figura 21. Reorientación de geometrías conforme a las superficies del objeto. ....	58
Figura 22. Colocación de una geometría volumétrica sobre un objeto. ....	58
Figura 23. Creación de botones mediante un perfil dibujado por el usuario. ....	59
Figura 24. Generación de botones ....	60
Figura 25. Pomos generados. ....	60
Figura 26. Pantallas o vaciados. ....	61
Figura 27. Creación de bordes en contornos. ....	62
Figura 28. Patrón sobre superficies. ....	62
Figura 29. Flujo de trabajo en el algoritmo. Fuente: Elaboración propia. ....	63
Figura 30. Estructura básica de un modelo de cafetera. ....	64
Figura 31. Estructura básica de una cinta de correr. ....	64
Figura 32. Clúster para generar volúmenes. ....	65
Figura 33. Clúster de generadores tridimensionales. ....	66
Figura 34. Base de cafetera. ....	67
Figura 35. Cuerpo superior de la cafetera. ....	67
Figura 36. Cuerpo de conexión. ....	68
Figura 37. Perfiles para la creación del volumen. ....	68
Figura 38. Selector de perfiles. ....	69
Figura 39. Puntos de referencia (subestructuras). ....	69
Figura 40. Componentes de puntos auxiliares. ....	70
Figura 41. Detalles esenciales. ....	71
Figura 42. Ejemplos de detalles no esenciales. ....	71
Figura 43. Componentes generadores de botones. ....	72
Figura 44. Componente para generar las perillas. ....	72

Figura 45. Componente generador de pantallas. ....	73
Figura 46. Componente para bordes. ....	73
Figura 47. Ejemplos de soluciones de diseño generadas. ....	74
Figura 48. Estructura básica de la cinta de correr. ....	75
Figura 49. Volumen creado a partir de un perfil seleccionado. ....	75
Figura 50. Volúmenes extensionales. ....	76
Figura 51. Creación de subestructura para panel de control. ....	76
Figura 52. Creación de volumen en la subestructura. ....	77
Figura 53. Implementación de elementos adicionales. ....	77
Figura 54. Implementación de controles. ....	78
Figura 55. Ejemplos de soluciones de diseño para una cinta de correr. ....	78

# INTRODUCCIÓN GENERAL.

Para introducir el tema del diseño asistido por ordenador mediante el diseño generativo, es importante tener en cuenta una definición básica del diseño. Esta actividad implica la preconcepción sistematizada de la forma y las demás características de un producto, teniendo en cuenta diversos aspectos como los sociales, tecnológicos, estéticos, psicológicos, anatómicos y fisiológicos. En resumen, se trata de crear un modelo detallado del objeto antes de su realización (Gay & Samar, 2007).

En la historia reciente, en el proceso de diseño de prácticamente cualquier objeto, se han aprovechado los avances tecnológicos, y en lo que respecta a los avances informáticos se destaca el diseño asistido por ordenador (CAD). Mediante el uso de herramientas informáticas se crean representaciones gráficas en dos o tres dimensiones de los objetos físicos. El CAD es ampliamente utilizado para diseñar productos físicos en una amplia gama de industrias, donde el software realiza los cálculos para determinar la forma y el tamaño óptimos para una variedad de productos y aplicaciones de diseño industrial.

Sin embargo, cada día se presentan diferentes y más exigentes características y/o necesidades en el diseño de estos objetos, en donde la gestión subjetiva de todos los parámetros que afectan un diseño se vuelve muy complicada e incluso humanamente imposible en brindar una solución aceptable. Ante esto, el diseño generativo provee una alternativa innovadora en el diseño de elementos con representación de geometrías cada vez más complejas.

El diseño generativo se puede definir como una herramienta de innovación en el diseño dentro de áreas como la ingeniería y la arquitectura. (Sivam Krish, 2011) lo define como "La transformación de la energía computacional en energía de exploración creativa, capacitando a diseñadores humanos para explorar un mayor número de posibilidades de diseño dentro de restricciones modificables".

Así que se puede decir que es un método para generar formas automáticamente a partir de la modificación de las variables que definen el propio objeto. Durante la modificación de estas variables, en ocasiones intervienen diferentes tipos de algoritmos para crear

estas variaciones que permiten acceder de un modo inteligente a una infinidad de formas con tan solo indicar los nuevos parámetros, es decir, las nuevas necesidades del diseño.

Este proceso provee a los diseñadores de una mayor variedad de soluciones factibles, gestionando eficientemente las opciones de diseño probando las configuraciones y aprendiendo de cada iteración lo que funciona y lo que no, debido a que el diseño generativo se ha inspirado en los procesos del diseño natural, por lo que los diseños se desarrollan como variaciones genéticas por medio de mutaciones y cruces.

Entre los objetivos de esta investigación, se encuentra el replanteamiento acerca del nivel en el cual el diseño generativo como herramienta CAD se puede ver involucrado para potenciar los resultados en el diseño. Esto mediante la aplicación práctica en casos de estudio para la generación de soluciones alternativas, analizar las ventajas y desventajas de este tipo de herramientas con respecto al diseño tradicionalmente usado en la industria. También estudiar el grado de contribución que brinda el diseño generativo en la creatividad humana para búsqueda e implementación de soluciones efectivas y convincentes.

Sumado a lo anterior, se pretende buscar métodos de diseño generativo mediante la estructuración de modelos para optimizar los resultados en la selección de soluciones de diseño, minimizando o filtrando de una mejor manera los diseños propuestos. Además de explorar la combinación de los mejores atributos entre métodos de diseño generativo y los algoritmos genéticos en las diferentes áreas de oportunidad.

En este contexto, el uso de redes neuronales ha emergido como una herramienta potencialmente revolucionaria para la creación y optimización de formas y estructuras. Las redes neuronales ofrecen la capacidad de aprender patrones complejos a partir de conjuntos de datos (Cunningham, Shu, Simpson, & Tucker, 2020), lo que las hace prometedoras para la generación automática y refinada de diseños. La confluencia entre diseño generativo y redes neuronales abre nuevas posibilidades para la exploración de soluciones de diseño avanzadas y altamente adaptativas.

A pesar de las ventajas potenciales, la integración de redes neuronales en herramientas de diseño asistido presenta desafíos significativos (Regenwetter, Nobari, & Ahmed, 2022). La complejidad inherente de las redes neuronales, tanto en términos de entrenamiento como de implementación, puede resultar abrumadora para los diseñadores, especialmente aquellos que no tienen experiencia en programación avanzada o en el manejo de modelos de aprendizaje automático. Esta complejidad podría dificultar la adopción generalizada de estas herramientas en entornos de diseño.

En la concepción de la herramienta de diseño asistido desarrollada en esta investigación, se tomó la decisión de no incorporar directamente redes neuronales. Esta elección se fundamenta en la necesidad de crear una herramienta eficiente y de fácil uso, especialmente dirigida a diseñadores que pueden no estar familiarizados con aspectos técnicos avanzados. La simplicidad y accesibilidad fueron consideraciones clave para

garantizar una adopción más extendida y una aplicación práctica de la herramienta en entornos de diseño industrial.

A pesar de la decisión de no utilizar redes neuronales directamente en la herramienta principal, se reconoce la importancia de ofrecer opciones avanzadas para usuarios que buscan diseños más complejos. Por lo tanto, los algoritmos en Grasshopper tiene la flexibilidad de conectarse con plugins especializados que hacen uso de redes neuronales. Esto proporciona una vía para aquellos diseñadores que desean explorar funcionalidades más avanzadas y aprovechar las capacidades específicas de las redes neuronales en la generación de geometrías más intrincadas. Esta modularidad ofrece una solución equilibrada entre accesibilidad y potencial avanzado en el diseño generativo.

# 1. EL DISEÑO GENERATIVO.

La etapa de diseño conceptual se conoce comúnmente como principalmente creativa. Muchos autores han estudiado los procesos que conducen a la creación exitosa de conceptos de productos, así como los obstáculos que dificultan las tareas del diseñador en esta etapa. Un campo de investigación típico se centra en la relación entre el dibujo (como representación de la forma del producto) y la creatividad (Do, Gross, Neiman, & Zimring, 2000; Goel, 1995; Lubart, 2005; Obrenovic, 2009; Schon & Wiggins, 1992; Suwa & Tversky, 1997).

El diseño generativo (DG) constituye una forma diferente de romper bloqueos creativos al dejar que el ordenador genere alternativas, permitiendo al diseñador explorar una gama mucho más amplia de posibles diseños de productos en un tiempo comparativamente menor. Esta disciplina es relativamente nueva en el diseño industrial, no obstante, está algo más extendido en el diseño de dentro de la arquitectura, donde la naturaleza algorítmica de la geometría generada encaja perfectamente en la estética de los edificios modernos.

Para dar paso al desarrollo de este trabajo, primero se debe recoger las múltiples aportaciones de esta disciplina en el campo del diseño con el fin de establecer una visión amplia de su papel e identificar sus posibles aplicaciones en otras áreas menos exploradas. Además, es importante conocer las ventajas y desafíos que pueden surgir de la aplicación de las diferentes herramientas que proporciona el diseño generativo.

Según (Krish, 2011), la investigación en lo que ahora se conoce como diseño generativo fue iniciada por Frazer a principios de la década de 1970. Al inicio de su desarrollo los investigadores académicos se enfocaron principalmente en la teoría del diseño, sin el avance de las metodologías de implementación; en consecuencia, los métodos formales para el diseño generativo no fueron inmediatos.

El término diseño generativo se define de muchas maneras en la literatura, algunas definiciones orientadas a cada segmento de interés de investigación, pero en general las definiciones de Sivam Krish y Kristina Shea retratan el objetivo esencial del diseño generativo.

Sivam Krish, un experto con experiencia internacional en investigaciones que abarcan varias disciplinas vinculadas al diseño, la ingeniería y la tecnología. Se ha especializado en la creación de métodos de enseñanza. En 2013, definió el DG como *“the transformation of computational energy into creative exploration energy empowering*

*human designers to explore greater number of design possibilities within modifiable constraints*” [ la transformación de la energía computacional en energía de exploración creativa que permite a los diseñadores humanos explorar un mayor número de posibilidades de diseño dentro de restricciones modificables]

Kristina Shea, una destacada docente e investigadora en el campo de la ingeniería de diseño y diseño generativo, es reconocida por sus logros en el desarrollo de sistemas de diseño generativo y sus contribuciones a la investigación en diseño de productos y procesos. En 2005, Shea lo describió como *“Generative design systems are aimed at creating new design processes that produce spatially novel yet efficient and buildable designs through exploitation of current computing and manufacturing capabilities”* [Los sistemas de diseño generativo tienen como objetivo crear nuevos procesos de diseño que produzcan diseños espacialmente novedosos pero eficientes y construibles, a través de la explotación de las capacidades informáticas y de fabricación actuales]

La investigación bibliográfica de este estado del arte se ha realizado utilizando las bases de datos ScienceDirect, Scopus, Mendeley, ProQuest, SAGE Journals, SpringerLink y Google Scholar. Es importante señalar que, con esta búsqueda encontramos varias metodologías cuyo objetivo es generar diseños mediante la aplicación de una metodología DG de forma individual o articulada con algunas otras técnicas de diseño.

En este contexto, se puede mencionar un abanico de metodologías de diseño generativo de mayor a menor presencia como son: gramáticas de forma, algoritmos genéticos, semántica, optimización de topología, conjunto fuzzy, redes neuronales, L-Systems, modelado paramétrico, inteligencia artificial, recocido de formas, autómatas celulares, estrategia evolutiva, optimización multiobjetivo, creación de prototipos de diseño urbano computacional, recocido simulado, gramática de descripción, diseño evolutivo, optimización de forma, computación evolutiva, reglas de forma, método de combinación de formas, aprendizaje por refuerzo, lenguajes de programación, técnica de diseño generativo de llenado de espacio, análisis y optimización de inicialización, algoritmos de enrutamiento, sistema de ingeniería Kansei híbrido, codificador automático variacional, semántico-topológico-geométrico, método de integración CAD/CAE, enfoque de diseño generativo, machine learning, optimización evolutiva multicriterio, síntesis de texturas, deep learning.

Dentro de este abanico de metodologías, se pueden clasificar en familias de técnicas en función de sus características generales. Aquí, se propone una clasificación general en cinco grandes grupos, que se muestran en la siguiente tabla (Tabla 1). Para dar mención en este estado del arte a los grupos con mayor importancia, se abordarán los tres grupos con mayor presencia en la literatura: Gramáticas, Algoritmos y Optimización.

Tabla 1. Clasificación de técnicas de DG.

Gramática	Algoritmos / IA	Semántica	Optimización	Paramétrico
<ul style="list-style-type: none"> <li>- Shape Grammars.</li> <li>- L-Systems.</li> <li>- Shape Annealing.</li> <li>- Simulated Annealing.</li> <li>- Description Grammar.</li> <li>- Shape Rules.</li> </ul>	<ul style="list-style-type: none"> <li>- Genetic Algorithms.</li> <li>- Neural Networks.</li> <li>- Artificial Intelligence.</li> <li>- Cellular Automata.</li> <li>- Evolutionary Strategy.</li> <li>- Computational Urban Design Prototyping (CUDP).</li> <li>- Evolutionary Design.</li> <li>- Evolutionary Computing.</li> <li>- Reinforcement Learning.</li> <li>- Programming Languages.</li> <li>- Space-filling-Generative Design Technique (Sf-GDT).</li> <li>- Routing Algorithms.</li> <li>- Variational autoencoder (VAE).</li> <li>- Machine Learning.</li> <li>- Multi-criteria Evolutionary.</li> <li>- Optimization</li> </ul>	<ul style="list-style-type: none"> <li>- Semantic.</li> <li>- Fuzzy Set.</li> <li>- Semantic-Topological-Geometric (STG).</li> </ul>	<ul style="list-style-type: none"> <li>- Topology optimization.</li> <li>- Multi-objetive optimization.</li> <li>- Shape Optimization.</li> <li>- Initialization, Analysis, and Optimization (IAO).</li> <li>- Taching-Learning-Based Optimization (TLBO).</li> <li>- Hybrid Kansei Engineering System (HKES).</li> </ul>	<ul style="list-style-type: none"> <li>- Parametric Modeling.</li> <li>- Shape-blending Method.</li> <li>- Parametric Design Procedures.</li> <li>- CAD/CAE Integration Method.</li> <li>- Generative Design Approach (GDA).</li> <li>- Knowledge Based Engineering (KBE).</li> </ul>

Fuente: Elaboración propia.

### 1.1. Gramáticas de formas

Las Shape Grammars o gramáticas de forma en español, se puede definir como un método para representar la geometría en forma generativa mediante la aplicación de una lógica geométrica del objeto (Krish, 2011). Por lo tanto, mediante la aplicación de un conjunto de reglas, que conforman esta lógica, se basan en las características inherentes del objeto y que comienzan con una forma inicial para transformar sucesivamente (McCormack, Cagan, & Vogel, 2004).

Esta técnica merece una mención especial dentro de las técnicas del DG, ya que es notoriamente la más utilizada y la metodología empleada en el planteamiento de esta tesis. El trabajo seminal de Stiny y Gips (George Stiny & Gips, 1972) sentó las bases para este formalismo generativo; en él se presenta el caso para la generación de pinturas y esculturas, aunque los autores destacan la posibilidad de producir algorítmicamente otro tipo de objetos como piezas musicales, compuestos químicos o frases del lenguaje. Más tarde, Stiny introdujo gramáticas de formas paramétricas para la generación de diseños de respaldos de sillas estilo Hepplewhite (George Stiny, 1980), documentando la aplicación de esta técnica para el diseño de productos.

A lo largo del tiempo, han surgido varios tipos de gramáticas de forma. En el trabajo de Sara García “Classifications of Shape Grammars” (Garcia, 2017) se encuentra una descripción completa de los diferentes enfoques de gramáticas y una categorización ordenada cronológicamente. También, entre las investigaciones relevantes en el campo del diseño de productos, se encuentra el trabajo titulado “Evaluation of a 3D Shape Grammar Implementation” (Chau, Chen, McKay, & de Pennington, 2004), ofrece un cuadro comparativo utilizado como referencia en este documento para ilustrar la implementación del diseño generativo. En su trabajo Chau, Chen, McKay y Pennington presentan una propuesta que combina algoritmos genéticos y gramáticas de formas para diseñar una botella de Coca-Cola. Además, proporcionan información detallada sobre las implementaciones de las gramáticas de formas desde 1975 hasta 2005.

Desde esas publicaciones iniciales, surgieron muchas aplicaciones de este enfoque formal. Inicialmente, los campos de crecimiento natural de las gramáticas de formas fueron las artes y la arquitectura (Khemlani, 1995; Liebich, 1994). La revista "Environment and Planning B: Planning and Design" constituye un buque insignia en la publicación de trabajos de gramática de forma, la mayoría de ellos del lado arquitectónico. La extracción de un estilo gramatical para su posterior replicación es un tema común en los trabajos de (Buelinckx, 1993; Chiou & Krishnamurti, 1995; Downing & Flemming, 1981; U Flemming, 1981; Ulrich Flemming, 1987; Herbert, Sanders, & Mills, 1994; Koning & Eizenberg, 1981; G Stiny & Mitchell, 1978; Yan, 1992).

En (Çağdaş, 1996) se da una descripción completa de la gramática de casas adosadas en Reino Unido. También, Seebohm y Wallace abordan la aplicación de reglas para el diseño arquitectónico (Seebohm & Wallace, 1998) y presentan una interpretación tridimensional de las gramáticas de forma de Stiny de 1980. Por otro lado, se pueden encontrar diferentes aproximaciones al diseño estructural en el trabajo de (Shea, Aish, & Gourtovaia, 2005; Shea & Cagan, 1999), en estas investigaciones es posible apreciar la aplicación de gramáticas de forma con otras técnicas de diseño nombradas como recocido de forma y recocido simulado. Por su parte (Smyth & Edmonds, 2000) analizan el uso de gramáticas de forma en la etapa conceptual del diseño arquitectónico y reconocen el potencial del soporte computacional para la exploración de soluciones.

La amplia adopción de la gramática de forma en el campo del diseño arquitectónico se refleja en la cantidad significativa de trabajos relacionados. En trabajos más recientes, como el de (Stouffs, 2018), se destacan numerosas aplicaciones que emplean la descripción gramatical como una guía fundamental en el proceso generativo. Asimismo, algunos años atrás, (José P. Duarte, 2005), presentó el uso de la descripción gramatical para procesar los datos proporcionados por el usuario y el emplazamiento en la construcción de las casas Malagueira diseñadas por Álvaro Siza. Estos ejemplos resaltan la importancia y la continua evolución de la gramática de forma en la arquitectura contemporánea.

Otros trabajos destacables en el diseño arquitectónico y el uso de la gramática de forma son el sistema de brackets chino de (Wu, 2005), la conexión con la semántica en un sistema para el modelado de las zonas urbanas vernáculas del sureste de China propuesto en (Liu, Xu, Pan, & Pan, 2006), la personalización de planos de apartamentos en (Veloso, Celani, & Scheeren, 2018). En el campo del diseño urbano, la investigación de (Mandić & Tepavčević, 2015) presenta un análisis de varios enfoques para la aplicación de la gramática de formas en el diseño urbano. Por otro lado, en una investigación un poco más reciente (Ozdemir & Ozdemir, 2018) produce, con gramática de forma utilizando dos de las técnicas de toma de decisiones multicriterio (MCDM), variaciones de alternativas de planos de tiendas.

Si bien las gramáticas de forma como técnica de diseño generativo experimentaron un crecimiento significativo en el ámbito arquitectónico, su adopción también se extendió a otros campos, incluyendo el diseño mecánico e industrial. Uno de los primeros trabajos encontrados en la literatura relacionada con el diseño de productos es The generation

of Hepplewhite-style chair-back designs (Knight, 1980), que toma como referencia el trabajo de (George Stiny, 1977) para el desarrollo de una gramática de formas paramétricas.

Otros trabajos sobre diseño mecánico se pueden encontrar en (Flasiński, 1995) o procesos de fabricación en (Brown, McMahon, & Sims Williams, 1996), aquí los autores hacen uso de gramáticas de forma y la técnica de diseño llamada “semántica”. En (M Agarwal & Cagan, 1998) se destaca como el primero en utilizar gramáticas de formas en el diseño de productos. En su investigación, Agarwal y Cagan exploraron cómo la funcionalidad del producto puede generar estructuras formales alternativas. Resulta interesante destacar que los autores reconocen las limitaciones que este enfoque impone al proceso creativo, especialmente en relación con los cambios radicales en el diseño.

Al igual que Brown, McMahon y Sims Williams, en (S.-W. Hsiao & Chen, 1997) también se combinan gramáticas de formas con un enfoque semántico para definir estructuras de formas significativas para una silla de oficina, que luego se modifican mediante varias reglas de regulación. Este enfoque fue más tarde utilizado en (S.-W. Hsiao & Wang, 1998) para producir diseños de automóviles, pero agregó la técnica de conjuntos difusos. Ambos ejemplos usaron un sistema CAD 3D conectado a una base de datos semántica. Es importante destacar que en ese momento el uso de ordenadores para el desarrollo de soluciones basadas en gramáticas de formas era bastante limitado. Un estudio del estado del arte realizado en 1999 por Gips (Gips, 1999) proporciona una visión general de las implementaciones de software de gramáticas de formas en esa época. Cuatro años más tarde, en (S.-W. S. W. Hsiao & Huang, 2002), vuelve a utilizar el ejemplo de las sillas de oficina para construir un sistema experto de diseño emocional basado en redes neuronales. Puede predecir la evaluación emocional de diferentes alternativas de sillas, generada mediante un procedimiento simplificado de gramática de formas.

Durante los años siguientes, las gramáticas de formas fueron aplicadas a una variedad de productos en varios estudios destacados. Uno de ellos es el realizado por (McCormack et al., 2004), en el cual se emplearon gramáticas de formas para capturar la identidad de la marca Buick y generar alternativas de diseño basadas en ella. Esta investigación, al igual que el trabajo de (Chen & McKay, 2004), se enfoca en revisar la identidad de marca. En este estudio se examinan las botellas de cuidado personal con el objetivo de analizar sus principios básicos de diseño y codificarlos dentro de una gramática de formas paramétricas.

Podemos encontrar una extensa investigación sobre identidad de marca en (Chau, 2002), aquí a través del enfoque gramatical, se pretende preservar la identidad de marca, y también se presentan algunos casos de estudio como la gramática de Dove. También en (Pugliese & Cagan, 2002) se discute la captura de marca con el fin de modelar una motocicleta Harley-Davidson con una gramática de forma de motocicleta. Además, en (Garcia & Menezes Leitão, 2018) se presenta una gramática de silla

polivalente a través de la implementación de gramáticas de forma, especialmente gramáticas de conjuntos.

Además de los estudios mencionados anteriormente, existen otros trabajos que exploran la aplicación de esta técnica de diseño generativo en el contexto de la identidad de marca. Por ejemplo, (Chau, 2002) lleva a cabo una investigación exhaustiva sobre este tema, utilizando el enfoque gramatical para preservar la identidad de marca. En su estudio, presenta casos de estudio como la gramática de Dove. Por otro lado, en (Pugliese & Cagan, 2002), se discute la captura de la marca para modelar una motocicleta Harley-Davidson mediante el uso de una gramática de formas específica para motocicletas. Además, en (Garcia & Menezes Leitão, 2018), se presenta una gramática de formas para el diseño de una silla polivalente, implementada mediante el uso de las gramáticas de forma, incluyendo gramáticas de conjuntos. Estos estudios amplían aún más el campo de aplicación de las gramáticas de formas en el diseño de productos y en la preservación de la identidad de marca.

En (Orsborn, Cagan, Pawlicki, & Smith, 2006), se exploró el uso de gramáticas de formas para cuantificar las diferencias entre clases de vehículos, lo que permitió la generación de nuevas formas de vehículos con una tendencia cruzada. Por otro lado, en un estudio sobre el diseño de máquinas de café (Manish; Agarwal, Cagan, & Constantine, 1999), se analizaron los costos asociados. Además, en esa época comenzaron a surgir trabajos que conectaban la generación automática de objetos con la fabricación automática (J.P. Duarte & Simondetti, 1997; Sass, 2008; Yufei Wang & Duarte, 2001). Estos estudios pioneros sentaron las bases para la integración de la generación y fabricación automatizadas en el diseño de productos.

La personalización masiva de vajillas de cerámica dio un paso significativo con la introducción del sistema basado en la gramática de formas propuesto por (Castro E Costa & Duarte, 2013). Este enfoque pionero permitió la generación automática de elementos de vajillas personalizados, sentando las bases para la implementación de un sistema completo de personalización masiva. Según Duarte, el desarrollo del sistema de diseño es el primer paso fundamental para adentrarse en la implementación exitosa de la personalización masiva.

En otros estudios relacionados, (Barros, Duarte, & Chaparro, 2011, 2015) se presenta un modelo para la personalización masiva en la industria del mueble, centrándose específicamente en el diseño y la implementación de un sistema basado en la gramática de formas para las sillas Thonet. Además de la gramática de formas, se menciona el uso de otras técnicas de diseño generativo, como el Shape Annealing y el Parametric Modeling, ampliando así las posibilidades de personalización y exploración de variaciones en el diseño de las sillas.

Estos avances en la personalización masiva basada en gramáticas de formas y otras técnicas generativas son prometedores en diversos campos del diseño. La capacidad de generar productos únicos y adaptados a las preferencias individuales de los usuarios ofrece nuevas oportunidades tanto para los consumidores como para las empresas. La

implementación exitosa de sistemas de diseño personalizado requiere un enfoque integral que combine tecnología, automatización y un profundo entendimiento de las necesidades y deseos del público objetivo.

Sin duda, la introducción del sistema de personalización masiva basado en la gramática de formas en el ámbito de las vajillas de cerámica, así como el modelo propuesto para las sillas Thonet, demuestran el potencial y la viabilidad de la personalización masiva en el diseño de productos. Estos enfoques permiten a los usuarios participar activamente en el proceso de diseño y obtener productos que reflejen su estilo y preferencias individuales. A medida que se avanza en la investigación y desarrollo de sistemas de diseño personalizado, se abren nuevas posibilidades para la innovación y la diferenciación en la industria del diseño.

En algunas investigaciones recientes, como se evidencia en (S. W. Kieharova, Pradujphongphet, & Bohez, 2014; Somlak Wannarumon Kieharova, Pradujphongphet, & Bohez, 2015), las gramáticas de formas y la estrategia evolutiva están siendo aplicadas al desarrollo de productos en campos específicos, como la joyería. Estos enfoques brindan apoyo a los diseñadores al explorar nuevas y innovadoras formas que este ámbito puede ofrecer. Al combinar las capacidades generativas de las gramáticas de formas con las estrategias evolutivas, se fomenta la generación de diseños únicos y atractivos que desafían los límites tradicionales.

Por otro lado, en (Ruiz-Montiel et al., 2013), se presenta un análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) del uso de las gramáticas de formas. Este análisis destaca las fortalezas de esta metodología, como su versatilidad para producir cualquier forma deseada y su capacidad de proporcionar un método intuitivo para la definición de formas. Además, las gramáticas de formas son valoradas por su capacidad para generar diseños complejos y formas inesperadas a partir de un conjunto reducido de reglas. Un aspecto clave es que las gramáticas de formas se pueden automatizar, permitiendo la generación rápida de múltiples alternativas de diseño, lo que aumenta la eficiencia y la capacidad de exploración de soluciones en el proceso de diseño.

Estas investigaciones demuestran el potencial de las gramáticas de formas y las estrategias evolutivas en diversos campos del diseño, como la joyería, donde se buscan formas distintivas y atractivas. Al aprovechar estas herramientas, los diseñadores pueden expandir su creatividad y explorar nuevas posibilidades de diseño. Además, el análisis DAFO proporciona una visión completa de las fortalezas y debilidades de las gramáticas de formas, permitiendo una evaluación informada de su aplicabilidad en diferentes contextos de diseño. En resumen, estas investigaciones contribuyen al avance y la mejora continua de los métodos de diseño generativo, enriqueciendo la disciplina del diseño con enfoques innovadores y eficientes.

Sin embargo, es importante destacar las debilidades asociadas al uso de las gramáticas de formas. Entre ellas, se menciona la dificultad para crear y controlar las reglas de la gramática de forma de manera efectiva, lo cual puede dificultar la consecución de los objetivos de diseño deseados. Además, la ejecución de las gramáticas de formas puede

requerir recursos computacionales considerables, lo que podría limitar su aplicación en entornos con recursos limitados.

A pesar de estas limitaciones, existen oportunidades prometedoras en el horizonte. El continuo avance en las capacidades de diseño asistido por ordenador abre nuevas posibilidades para mejorar y ampliar el uso de las gramáticas de formas. A medida que los ordenadores se vuelven más poderosos y sofisticados, se espera que se incremente la capacidad para generar y manipular diseños complejos mediante el uso de gramáticas de formas. Esto podría llevar a una mayor inclusión de estas herramientas en el proceso de diseño, potenciando así la creatividad y eficiencia en la generación de soluciones innovadoras.

No obstante, también es necesario abordar una posible amenaza. Existe cierta reticencia por parte de algunos diseñadores a adoptar plenamente el uso de las gramáticas de formas en su práctica profesional. Esto puede atribuirse a la falta de familiaridad con la metodología o a la percepción de que las gramáticas de formas limitan la libertad creativa. Superar esta barrera requerirá una mayor difusión y educación sobre los beneficios y el potencial de estas herramientas, así como ejemplos exitosos de su aplicación en el campo del diseño.

Un problema que se ha identificado en el enfoque tradicional de las gramáticas de formas es que, para obtener soluciones válidas, a menudo se reduce el espacio de alternativas. Esto puede limitar la exploración de diseños novedosos y originales (Ruiz-Montiel et al., 2013). Para abordar este desafío, se han propuesto enfoques alternativos, como el uso de gramáticas de grafos. Un ejemplo interesante de esto se encuentra en (McDermott, 2012), donde se utiliza un sistema 3D evolutivo interactivo basado en gramáticas de grafos. Este enfoque presenta una perspectiva intrigante, ya que combina la capacidad generativa de las gramáticas con la estructura y las relaciones definidas por los grafos. Esto puede permitir una mayor flexibilidad y expresividad en el diseño, abriendo nuevas posibilidades en la aplicación de las gramáticas en el diseño de objetos tridimensionales.

En otro enfoque, (Krish, 2011) presenta el Método de Diseño Generativo como una alternativa para el diseño conceptual generativo de productos. Este método se implementa mediante el uso de Genofom, un software que genera variaciones aleatorias en los parámetros del producto. Es importante destacar que Genofom se integra con conocidos programas de diseño como Solidworks o Rhinoceros, lo que permite una mayor flexibilidad y compatibilidad con los sistemas ya establecidos en la industria. No obstante, se abordará con más detalle el uso de estos softwares en el contexto de este trabajo en secciones posteriores.

Una forma de simplificar la complejidad de las gramáticas de formas es mediante el enfoque de las gramáticas ingenuas (Ruiz-Montiel et al., 2013). Las gramáticas ingenuas son menos restrictivas, lo que permite un mayor número de alternativas, pero también pueden generar soluciones inválidas. Para abordar este problema, se utilizan procedimientos de aprendizaje aplicados al sistema gramatical.

Otro enfoque para simplificar las gramáticas de formas es el uso de gramáticas de conjuntos. Según (Granadeiro, Pina, Duarte, Correia, & Leal, 2013), las gramáticas de conjuntos son una forma más sencilla y restringida de las gramáticas de formas, donde el reconocimiento de subformas, la aparición de formas y la aplicación de reglas iterativas no se realizan de manera libre y descontrolada. Este enfoque permite una mayor precisión en la generación de formas y evita soluciones indeseadas.

En (Scott C Chase, 2002), se presenta un sistema generativo interactivo que merece mención. El autor discute la posibilidad de generar automáticamente gramáticas, un tema de gran importancia para lograr un sistema eficiente de diseño generativo. Además, en (S C Chase, 2005), se aborda el uso de funciones gráficas directas en lugar de reglas gramaticales codificadas en texto.

En varios trabajos realizados en estos años, se ha utilizado una combinación de gramáticas de formas y algoritmos evolutivos (AE) en diferentes contextos. Algunos ejemplos incluyen el diseño de marcas de Coca Cola (McKay, Ang, Chau, & Pennington, 2006), el bordado étnico Zhuang (Cui & Tang, 2013) y una revisión de las casas de la pradera de Frank Lloyd Wright (Granadeiro et al., 2013).

En relación con el modelado paramétrico como herramienta creativa, (Turrin, Von Buelow, & Stouffs, 2011) resume varios enfoques que han sido utilizados. Se propone el uso de algoritmos genéticos para controlar la generación de alternativas, lo cual ofrece nuevas posibilidades para la exploración y creación de diseños.

En una serie de trabajos pioneros, (H. C. Lee, Herawan, & Noraziah, 2012) y (H. C. Lee & Tang, 2009) han explorado la combinación de gramáticas de formas y algoritmos evolutivos para el diseño de productos. Estos enfoques, conocidos como "evolución gramatical", representan una disciplina específica que utiliza las gramáticas como sistemas descriptivos y los algoritmos evolutivos como generadores de variaciones en el diseño.

La gramática evolutiva se ha convertido en una forma de "algoritmo generativo de diseño paramétrico", permitiendo la creación de soluciones de diseño mediante la generación y evolución de variaciones dentro de un conjunto de reglas gramaticales. Este enfoque combina la flexibilidad de las gramáticas de formas con la capacidad de los algoritmos evolutivos para buscar soluciones óptimas o satisfactorias dentro del espacio de diseño.

La evolución gramatical ha demostrado ser una metodología efectiva para la exploración y generación de diseños en diversos campos. Al utilizar la capacidad de las gramáticas para describir formas y estructuras, y la capacidad de los algoritmos evolutivos para generar variaciones y evaluar soluciones. La revisión de diversos estudios bibliográficos ha revelado la capacidad de obtener resultados innovadores y personalizados en el diseño de productos. Estos trabajos han demostrado consistentemente que es posible alcanzar soluciones creativas y adaptadas a través de enfoques específicos en el proceso de diseño.

## 1.2. Mejoramiento de forma.

En el ámbito del diseño generativo y la optimización topológica, la búsqueda de soluciones óptimas y eficientes es un objetivo clave. Para lograr esto, se han integrado diferentes procedimientos que se adaptan bien a las metodologías de diseño generativo. Estos enfoques combinan diversas técnicas, como algoritmos genéticos y optimización de topología, para explorar un amplio espacio de soluciones y encontrar la mejor configuración de distribución de material y conectividad dentro de un dominio de diseño (Deaton & Grandhi, 2014).

Un ejemplo destacado es el trabajo de (Ikonen & Sóbester, 2018), quienes utilizaron la combinación de L-System, Algoritmos Genéticos y Optimización de Topología para maximizar la frecuencia natural fundamental de un panel de aluminio con refuerzo integral. Este estudio demuestra cómo la convergencia de estas metodologías puede resolver problemas específicos de diseño de manera efectiva.

Otro enfoque interesante es el propuesto por (Yedong Wang et al., 2018), quienes desarrollaron un método de diseño generativo para estructuras de celosía. Su enfoque consta de tres etapas: generación automática de la estructura, análisis de elementos finitos para evaluar el rendimiento y optimización rápida para mejorar la distribución de materiales en las paredes de la estructura.

En el campo de la optimización multiobjetivo, (Shieh, Li, & Yang, 2018) utilizaron el algoritmo evolutivo multiobjetivo (MOEA) y el sistema de ingeniería Kansei para obtener alternativas óptimas en el diseño de productos. Su estudio se enfocó en el diseño de un jarrón, demostrando cómo la optimización puede abordar tanto los parámetros de diseño como la validación de soluciones. En (Khan & Gunpinar, 2018) aplicaron la Optimización Basada en el Aprendizaje-Enseñanza (TLBO) en la mejora de soluciones de diseño. Este método desarrollado en (Rao, Savsani, & Vakharia, 2011) se basa en un enfoque poblacional de profesor-estudiante, donde un profesor seleccionado influye en los estudiantes para mejorar el rendimiento general de la clase. Este estudio destaca la importancia de la optimización inteligente en el diseño generativo.

En cuanto a la comparación de enfoques, (Tyflopoulos, Flem, Steinert, & Olsen, 2018) proporcionan un estado del arte del diseño generativo y la optimización topológica, junto con una interesante comparación de diferentes enfoques de optimización de topología. También sugieren el uso de algoritmos de optimización de topología para determinar los límites óptimos de adaptación estructural en el diseño de una fijación de esquí.

Estos ejemplos demuestran cómo la combinación de enfoques de diseño generativo y técnicas de optimización topológica ha permitido obtener soluciones más eficientes y personalizadas en diferentes áreas de diseño. Estos avances son fundamentales para impulsar la innovación y la optimización en el desarrollo de productos y estructuras en ingeniería.

### 1.3. Algoritmos genéticos

Los Algoritmos Genéticos (AG) son una técnica de optimización que emula el proceso de selección natural. Una población inicial de individuos se define como cadenas, luego se seleccionan los individuos y se combinan sus cadenas para crear nuevos individuos en un proceso aleatorio que tiene en cuenta su aptitud para una función objetivo determinada (Ruiz-Montiel et al., 2013). En este sentido, el número especial de Design Studies sobre diseño generativo titulado “Computational Making” recoge varios puntos de vista sobre el proceso creativo algorítmico.

Los AG se han utilizado por separado o combinados con gramáticas de forma o redes neuronales para producir sistemas de diseño generativo. Algunos trabajos tempranos se pueden encontrar en los años 90 (Apineda, 1993; Gero, 1996; Hybs & Gero, 1992; Pollalis, 1994).

Una de las primeras aplicaciones de AG en la literatura es GENE\_ARCH, un sistema de diseño generativo basado en AG que se utiliza para desarrollar soluciones arquitectónicas sostenibles y ecoeficientes (Caldas, 2008).

Se puede observar que, al igual que las gramáticas de forma, el diseño generativo mediante algoritmos genéticos y enfoques similares ha encontrado aplicación en la arquitectura, especialmente en investigaciones más recientes. Por ejemplo, en el trabajo de (Medjdoub & Bi, 2018), se utiliza la programación basada en restricciones para determinar la ruta de los conductos en sistemas fancoil instalados en los techos de los edificios. Por su parte (Miao et al., 2018) han desarrollado un método denominado Computational Urban Design Prototyping (CUDP) para la creación rápida de prototipos de diseño urbano. En el estudio de (Sun & Taplin, 2018), se emplea el algoritmo COmputational Urban Layout Design basado en algoritmos genéticos para generar planos residenciales. Además, se utiliza el Autómata Celular como una herramienta generativa para explorar formas alternativas en edificios y espacios urbanos.

La inteligencia artificial también está teniendo un impacto significativo en el diseño generativo. Esto se puede observar en el trabajo realizado por (Heusler & Kadija, 2018), quienes se centran en la creación de diseños avanzados de fachadas complejas utilizando técnicas de inteligencia artificial. Asimismo, (Pantazis & Gerber, 2018) investigan y generan diseños de fachadas utilizando un Sistema Multi Agente, el cual es considerado un ejemplo de inteligencia artificial distribuida en el campo del diseño. Además, (Francalanza, Fenech, & Cutajar, 2018) utilizan inteligencia evolutiva y artificial para diseñar módulos de enlace entre las articulaciones de un robot articulado. Estos ejemplos demuestran cómo la inteligencia artificial está siendo aplicada de manera innovadora en el diseño generativo.

Existen numerosas revisiones exhaustivas sobre el estado del arte de la Computación Evolutiva (CE) que se pueden consultar para obtener una visión completa de esta área. Una de estas revisiones se encuentra en el trabajo (Kicing, Arciszewski, & De Jong, 2005), donde se ofrece un análisis detallado de la CE y una clasificación cronológica de

sus aplicaciones en ingeniería estructural. Aunque requeriría una evidente actualización, esta revisión proporciona una amplia perspectiva sobre cómo la CE ha sido utilizada en el campo de la ingeniería estructural a lo largo del tiempo.

En cuanto a la mejora de los modelos generativos en el aprendizaje automático y el uso de técnicas de aprendizaje en el diseño, se puede encontrar una revisión relativamente reciente en el trabajo de (Bidgoli & Veloso, 2018). Este estudio examina las mejoras que se han logrado en los modelos generativos mediante el uso de técnicas de aprendizaje automático, así como la aplicación de técnicas de aprendizaje en el proceso de diseño. Esta revisión ofrece una visión actualizada de cómo el aprendizaje automático está influyendo en el diseño generativo y cómo se están utilizando técnicas de aprendizaje para mejorar la generación de diseños.

Ambas revisiones son recursos valiosos para comprender el estado actual de la Computación Evolutiva y el uso de técnicas de aprendizaje en el diseño generativo, proporcionando información relevante sobre las aplicaciones y avances más recientes en estos campos.

La investigación de (Neill et al., 2010) destaca la potente combinación entre las gramáticas de forma y la gramática evolutiva, utilizando una nueva herramienta de diseño evolutivo para generar diseños conceptuales de refugios. Esta fusión de técnicas permite la creación de soluciones innovadoras y adaptativas en el campo del diseño arquitectónico.

Por otro lado, (Renner & Ekárt, 2003) ofrecen una descripción interesante sobre las contribuciones y aplicaciones de los Algoritmos Genéticos (AG) en el diseño. En su trabajo, mencionan diversos estudios e investigaciones que resaltan el impacto positivo de los AG en el ámbito del diseño, mostrando su versatilidad y capacidad para generar soluciones eficientes y creativas.

En su investigación, (Cluzel, Yannou, & Dihlmann, 2012) emplean armónicos de Fourier para describir las siluetas de automóviles y utilizan un algoritmo genético para derivar alternativas de diseño. En un estudio comparativo entre las gramáticas de forma y los algoritmos genéticos, se señala que las gramáticas de forma, como método de codificación, presentan una mejor integridad, robustez y consistencia en comparación con otros métodos. Sin embargo, tienen una baja extensibilidad y adaptabilidad a los algoritmos genéticos.

Por otro lado, los algoritmos genéticos son capaces de generar todo tipo de formas con un mayor nivel de detalle, lo que puede provocar sentimientos y emociones en el usuario. Además, la codificación utilizada por los algoritmos genéticos cubre un espacio mucho más amplio de posibles curvas 2D o siluetas 2D, lo que les confiere una mayor flexibilidad y capacidad de exploración en comparación con las gramáticas de forma.

En resumen, el enfoque de Cluzel, Yannou y Dihlmann combina armónicos de Fourier y algoritmos genéticos para el diseño de siluetas de automóviles. Aunque las gramáticas de forma destacan por su integridad y consistencia, los algoritmos genéticos ofrecen una

mayor versatilidad y capacidad de generar formas detalladas, lo que puede tener un impacto emocional en los usuarios. Esta comparación destaca las fortalezas y características distintivas de ambos enfoques en el diseño generativo.

En su trabajo, (Strug, Grabska, & Ślusarczyk, 2014) proponen una combinación de algoritmos genéticos y representaciones hipergráficas, usando un diseño de tabla como ejemplo. En este enfoque, se utiliza una representación hipergráfica para modelar los operadores genéticos, estableciendo una estructura jerárquica que captura las relaciones entre los elementos del diseño.

En su investigación, (Ji, 2016) introduce un algoritmo basado en el componente de secuencias de comandos Grasshopper VB para la generación automatizada de patrones de juntas en forma de "T" en el diseño de celosías chinas con formas irregulares y estructura independiente. Este enfoque permite producir formas geométricas personalizadas y optimizadas, facilitando la exploración de diseños únicos en el contexto de las celosías chinas.

En el ámbito del diseño de joyas, se ha explorado la aplicación de las Gramáticas de Forma (GF) en combinación con estrategias evolutivas. Por ejemplo, el trabajo de (S. W. Kielarova et al., 2014) propone un sistema de diseño interactivo-generativo que utiliza un híbrido de GF y estrategia evolutiva. En otra investigación de (Somlak Wannarumon Kielarova et al., 2015), se demuestra cómo el uso de algoritmos evolutivos incrementa significativamente las soluciones de diseño de pendientes. Además, en un estudio un poco más reciente realizado por (Sansri & Kielarova, 2018), se emplea el algoritmo evolutivo "Strength Pareto Evolutionary Algorithm" (SPEA-II) para la optimización multiobjetivo, considerando la relación de forma y el peso en el diseño de joyas. Estas investigaciones destacan la utilidad de las GF y los algoritmos evolutivos en el diseño de productos relacionados con la joyería.

En el campo del diseño de productos, Sansri y Kielarova exploran la aplicación de algoritmos genéticos en el diseño de un frasco de perfume. En su estudio del año 2016 (Somlak Wannarumon Kielarova & Sansri, 2016) proponen un sistema de diseño generativo que combina un algoritmo genético interactivo con un algoritmo genético multiobjetivo para optimizar la forma del frasco. Este enfoque permite encontrar soluciones que cumplen con múltiples objetivos y mejora el proceso de diseño.

La aparición inesperada en el diseño puede manifestarse de diferentes maneras. En la investigación de (Krish, 2011), se emplea la variación aleatoria de parámetros de diseño dentro de límites predefinidos para generar una variedad de opciones de solución. Estas soluciones se filtran posteriormente a través de restricciones para garantizar su viabilidad. Por otro lado, en el trabajo de (Strug et al., 2014), se reconoce que el uso del diseño evolutivo permite la introducción de soluciones nuevas e inesperadas. (Gero, 1996) también reconoce la importancia de permitir la introducción o modificación de valores de los parámetros de diseño para fomentar la aparición de soluciones novedosas e inesperadas. En su investigación, presenta diversos modelos de procesos de emergencia basados en nociones de variables aditivas y sustitutivas.

La investigación de (Albers, Rovira, Aguayo, & Maier, 2008) destaca cómo las primeras actividades en los algoritmos genéticos permitieron la creación de formas informáticas que simulan una actuación creativa. Por otro lado, en el trabajo más reciente de (Gunpinar & Gunpinar, 2018), también se emplean algoritmos genéticos para generar variaciones de forma en productos CAD, con el objetivo de proponer formas innovadoras a diseñadores y consumidores.

## 2. EL USO DE APLICACIONES PARA EL DISEÑO GENERATIVO.

Si bien el diseño generativo y el uso del ordenador se han vuelto intrínsecos, vale la pena documentar algunas de las diferentes aplicaciones que ha tenido a lo largo de los años. Revisando algunos trabajos existentes en la literatura es posible encontrar estudios que abordan la implementación del ordenador desde muchos campos y variados enfoques. Estos pueden variar desde diseño de productos, arquitectura, ingeniería, arte, etc. Se creó una gran base de datos, pero solo se mencionan algunos en los que el uso del ordenador es relevante.

Por ejemplo, para el diseño de productos, en (Chen & Owen, 1998) se propone un sistema basado en la modificación de primitivas mediante diferentes reglas de descomposición, unión o redimensión de formas simples a otras más complejas que representen el producto completo. (S.-W. Hsiao & Wang, 1998) presentan otro sistema basado en información semántica, que se utiliza para derivar formas de productos a través de b-splines y manipular interactivamente los resultados.

En el campo de la arquitectura se pueden citar los siguientes trabajos. Hemberg & O'Reilly propusieron Genr8, que es una herramienta de software para arquitectos (Hemberg & O'Reilly, 2004). Esta propuesta genera superficies en el entorno Maya de la herramienta CAD a partir del modelo de crecimiento generativo basado en gramática. Las superficies generadas son creadas por un Hemberg Extended Map L-System (HEMLS), que es una versión más compleja del Lindenmayer System (L-system). En otra investigación, se implementó EduDesign, un sistema generativo interactivo para el diseño de arquitectura, como complemento de 3D Studio (Dincer, Tong, & Cagdas, 2012). En la investigación de Prats, Earl, Garner, & Jowers se presenta un modelo para la exploración de formas en nuevos diseños en etapas tempranas (M Prats, Earl, Garner, & Jowers, 2006). Además de esto, la tarea de detección de formas se aborda en (Miquel Prats, Lim, Jowers, Garner, & Chase, 2009) mediante el estudio de bocetos y la transformación de formas mediante la técnica de reglas de forma.

Es posible encontrar trabajos de investigación que señalan algunas necesidades y desafíos que trae la aplicación informática en las diferentes etapas del diseño. Por ejemplo, Smyth & Edmonds apuntan al uso de gramáticas de formas tridimensionales como una necesidad (Smyth & Edmonds, 2000). De manera similar, S. Chase denota que el desarrollo de herramientas de diseño, como los sistemas generativos, ha sido lento debido a la falta de una buena interacción entre el usuario y el sistema (Scott C Chase,

2002). En el trabajo de (Lin & Lee, 2013) se analizan métodos y herramientas de usabilidad en diseño generativo a través del software Rhinoceros y el plugin Grasshopper.

Por su parte, (Tsai, Chen, Tsai, & Hung, 2010) combinan los principios de la teoría de conjuntos difusos, el método de combinación de formas y algoritmos genéticos para ayudar al diseñador en las primeras etapas del diseño en un entorno asistido por ordenador. Es en este entorno que Talbott en su trabajo de 2007, señaló varios problemas causados por la interacción diseñador-ordenador (Talbott, 2007).

Si bien el uso del ordenador no es un elemento de uso común en etapas conceptuales, tiene un alto potencial, como se puede apreciar en el trabajo de (Neill et al., 2010) donde se utilizó un complemento interactivo para Blender 3D, convirtiendo el uso del ordenador en una herramienta para aumentar la creatividad humana.

El uso del diseño asistido por ordenador se utiliza principalmente en la etapa de diseño detallado, más que en la etapa de diseño conceptual, como se mencionó anteriormente. Así lo señala el trabajo de (Somlak Wannarumon Kielarova et al., 2015), en esta investigación se explora una aplicación en joyería para llevar los beneficios del CAD a las primeras etapas del proceso de diseño creativo a través de un sistema GD.

Además de las investigaciones mencionadas, se pueden encontrar otras que también abordan el uso del ordenador en el diseño generativo con diferentes propósitos. Un ejemplo relevante es el trabajo realizado por Rubaiat Habib Kazi, quien se enfoca en cerrar la brecha entre el diseño conceptual y el diseño detallado utilizando CAD (Kazi, Grossman, Cheong, Hashemi, & Fitzmaurice, 2017). En su investigación, presentan DreamSketch, una interfaz de diseño 3D que combina las características del dibujo a mano alzada y el diseño asistido por ordenador con algoritmos generativos. Además, Kazi hace referencia a otras soluciones tecnológicas que abordan este tema, como 3D6B Editor (Kallio, 2005), el lienzo mental (Dorsey, Xu, Smedresman, Rushmeier, & McMillan, 2007), Insitu (Paczkowski, Kim, & Morvan, 2011) y Tilt Brush, la innovadora aplicación de dibujo en realidad virtual desarrollada por Google. En una etapa similar de diseño, enfocada en la exploración conceptual de la forma, se encuentra el trabajo de Kwon, Gross, & Yi-Luen, quienes proponen el sistema interactivo ArchiDNA como una solución (Kwon, Gross, & Yi-Luen Do, 2009).

En el trabajo realizado por (Grasl & Economou, 2011), se introduce un complemento llamado GRAPE, el cual fue implementado en Rhinoceros. Este complemento ofrece funcionalidades adicionales y amplía las capacidades del software en el contexto del diseño generativo. En investigaciones más recientes, se ha observado el desarrollo de complementos para software como Revit, Autocad y Rhino5, los cuales se encuentran en diferentes etapas de desarrollo. Estos complementos buscan mejorar la integración de algoritmos generativos y facilitar el proceso de diseño en estas plataformas de software ampliamente utilizadas en la industria.

En el estudio realizado por (Sansri & Kielarova, 2018), se utiliza un complemento de diseño generativo llamado Octopus. Este plugin está instalado en Grasshopper, el cual

se ejecuta dentro del programa Rhinoceros. El caso de estudio abordado se centra en el modelado de joyería con un broche de clip, utilizando las capacidades generativas del complemento para explorar y generar diseños innovadores en este ámbito.

Por otro lado, en la investigación reciente llevada a cabo por (L. Wang, Chen, Janssen, & Ji, 2020), se presentan una serie de herramientas para el diseño arquitectónico. Estas herramientas tienen como objetivo ayudar en la exploración de alternativas de diseño a través del enfoque evolutivo. El estudio se enfoca en la aplicación de algoritmos generativos y técnicas evolutivas para generar soluciones arquitectónicas eficientes y adaptativas, permitiendo a los diseñadores explorar un amplio espacio de diseño y descubrir soluciones innovadoras.

En ambos casos, se evidencia el uso de complementos y herramientas específicas para el diseño generativo en el ámbito de la joyería y la arquitectura respectivamente, lo cual demuestra la continua investigación y desarrollo de herramientas que facilitan la generación de diseños creativos y personalizados en diferentes disciplinas. Con esta búsqueda bibliográfica también se ha encontrado que el software Rhinoceros y Grasshopper es ampliamente utilizado en diversas investigaciones y aplicaciones de diseño. Algunos ejemplos destacados son el estudio realizado en (Castro E Costa & Duarte, 2013), en el que se adopta Grasshopper en el diseño de vajillas de cerámica. El uso de este entorno de programación visual permite explorar y generar formas y patrones variados en la vajilla, brindando opciones creativas y personalizadas.

Otro caso es el trabajo de (Abdullah & Kamara, 2013), quienes aplican Grasshopper en el diseño de edificios. Aprovechando sus capacidades generativas, utilizan este software para explorar diferentes opciones arquitectónicas y optimizar la eficiencia espacial.

En el trabajo de (L. C. J. Lee & Lin, 2014) se utiliza Grasshopper para el desarrollo de un estuche móvil. Mediante el enfoque generativo, logran generar una geometría adaptable y personalizada que se ajusta a las necesidades y preferencias del usuario.

En otra investigación, (Huang & Li, 2014) exploran el diseño de superficies de forma libre utilizando Grasshopper. Gracias a esta herramienta, pueden crear formas complejas y orgánicas, ampliando las posibilidades de diseño en la arquitectura y el diseño industrial.

Sumado a las investigaciones anteriores, se ha aprovechado Grasshopper como una herramienta versátil para la exploración y generación de diseños creativos en diversos campos. Por ejemplo, (Somlak Wannarumon Kielarova & Sansri, 2016) lo aplicaron en el diseño de frascos de perfume, lo que les permitió explorar una amplia variedad de formas y estilos para crear diseños únicos y atractivos en el mundo de la perfumería.

Por su parte, (Ji, 2016) empleó Grasshopper en la creación de celosías chinas, utilizando técnicas generativas para generar patrones ornamentales complejos y elegantes. Esta aplicación demuestra cómo la herramienta puede ser utilizada para diseñar elementos arquitectónicos ornamentales de gran belleza y complejidad.

En el ámbito del diseño urbano, (Koenig, Miao, Knecht, Buš, & Mei-Chih, 2017) optaron por Grasshopper como una herramienta para explorar y generar soluciones urbanas innovadoras. Esto resalta la capacidad de Grasshopper para abordar desafíos en el

diseño urbano y contribuir a la creación de entornos urbanos más eficientes y estéticamente atractivos.

En conjunto, estas investigaciones ponen de manifiesto la versatilidad y el potencial de Grasshopper en diversos campos del diseño, impulsando la creatividad y la exploración de nuevas soluciones en cada uno de ellos.

En la búsqueda de herramientas generativas, se ha observado que muchas de ellas se enfocan en la optimización de formas, lo cual es ciertamente una de las aplicaciones clave del Diseño Generativo (DG). Sin embargo, es importante destacar que el DG va más allá de la mera optimización y ofrece un enfoque más amplio en la creación de formas mediante la aplicación de restricciones. Estas restricciones son indicadas al ordenador, que se encarga de buscar y generar formas que cumplan con ellas. El DG permite explorar nuevas posibilidades y encontrar soluciones que se ajusten a criterios específicos, ya sean funcionales, estéticos o técnicos. Al utilizar restricciones como guías creativas, el DG ofrece un enfoque más intuitivo y colaborativo entre el diseñador y la máquina, permitiendo la co-creación de diseños que se adaptan a las necesidades y requerimientos específicos del proyecto. Así, el DG se posiciona como una herramienta poderosa para la generación de formas y la materialización de ideas innovadoras en diversos campos del diseño.

La optimización topológica y el diseño generativo son dos enfoques diferentes en el campo del diseño. A menudo, la optimización topológica se confunde con el DG, por lo que es importante comprender las diferencias clave entre estas herramientas.

La optimización topológica se centra en mejorar y optimizar un diseño existente realizado mediante un sistema CAD tradicional. Utilizando procesamiento algorítmico, la optimización topológica busca eliminar material del objeto, aligerándolo sin comprometer sus características de rendimiento. En este caso, el diseño ya tiene una morfología establecida y la optimización se realiza dentro de esos límites.

Por otro lado, el DG parte de la definición de restricciones del objeto a diseñar. A diferencia de la optimización topológica, el ordenador no tiene una idea preconcebida de la forma del objeto. En cambio, genera cientos o miles de soluciones de diseño basadas en las restricciones establecidas. El DG es un proceso creativo que permite explorar una amplia gama de soluciones de diseño, sin estar limitado por una forma preexistente. Además, el DG puede incluir un componente de optimización si se establecen restricciones para mejorar aún más el formulario resultante.

En resumen, mientras que la optimización topológica se enfoca en mejorar y optimizar un diseño preexistente, el DG se centra en la creación de nuevas formas y soluciones de diseño basadas en restricciones establecidas. Ambos enfoques son valiosos en el proceso de diseño, pero es importante reconocer sus diferencias y comprender cuál es el más adecuado para cada situación.

Por esta ventaja que aporta el diseño generativo, ha llevado a empresas líderes en el campo del modelado 3D, como Autodesk, Dassault Systèmes y Siemens, a dedicar recursos significativos a la investigación y desarrollo de software especializado en esta área.

Estas empresas reconocen las ventajas que ofrece el diseño generativo en términos de generación automática de múltiples soluciones de diseño y exploración de opciones creativas. Han trabajado arduamente para desarrollar herramientas y plataformas que permitan a los diseñadores aprovechar al máximo el potencial del diseño generativo en sus proyectos.

El objetivo de estas empresas es proporcionar a los diseñadores herramientas intuitivas y potentes que les permitan definir restricciones y objetivos de diseño, y luego generar automáticamente opciones de diseño basadas en esos parámetros. Estos softwares especializados utilizan algoritmos avanzados para generar y evaluar múltiples soluciones, lo que ahorra tiempo y permite a los diseñadores explorar un amplio espectro de posibilidades.

Además, estas empresas han incorporado capacidades de simulación y análisis en sus herramientas de diseño generativo, lo que permite a los diseñadores evaluar el rendimiento y la viabilidad de las diferentes soluciones generadas. Esto brinda una mayor confianza en la toma de decisiones de diseño y facilita la iteración y optimización del diseño.

En el catálogo de Autodesk, se encuentra una amplia gama de software que brinda capacidades de diseño generativo para diversos sectores. Estas herramientas se han desarrollado para abordar diferentes necesidades y aplicaciones dentro del campo del diseño.

Entre los softwares destacados se encuentran Inventor y Fusion 360, que se especializan en la optimización general de topología. Estas herramientas permiten a los diseñadores generar y explorar múltiples opciones de diseño, optimizando la forma y estructura de los objetos para mejorar su rendimiento y eficiencia.

Maya, por otro lado, se enfoca en el campo de la animación y ofrece capacidades de diseño generativo para la creación de personajes y entornos animados. Los diseñadores pueden utilizar algoritmos y restricciones para generar automáticamente movimientos y comportamientos realistas.

Para el sector de la construcción, Autodesk ofrece Revit y Dynamo Studio. Estas herramientas permiten a los arquitectos y diseñadores generar diseños generativos para edificios y estructuras, optimizando la distribución espacial, la eficiencia energética y otros aspectos importantes en el proceso de diseño.

Netfabb es otro software de Autodesk que se centra en la optimización de celosías y superficies. Los diseñadores pueden utilizar esta herramienta para crear estructuras optimizadas que maximicen la resistencia y reduzcan el peso de los objetos.

Por último, Within se especializa en la optimización de superficies y está dirigido específicamente al diseño de implantes médicos. Esta herramienta permite a los diseñadores generar y optimizar formas y estructuras que se adapten mejor a las necesidades de cada paciente.

En resumen, Autodesk ofrece una amplia gama de software especializado en diseño generativo para diversos sectores, como la ingeniería, la arquitectura, la animación y la medicina. Estas herramientas permiten a los diseñadores explorar nuevas soluciones de

diseño, optimizar la forma y la estructura de los objetos, y mejorar la eficiencia y el rendimiento en sus proyectos.

Dassault Systèmes, reconocido por su experiencia en software de diseño asistido por ordenador, ha integrado módulos de diseño generativo en sus productos como SolidWorks y CATIA. Estos módulos ofrecen herramientas avanzadas para la creación y optimización de diseños generativos.

En SolidWorks, Dassault Systèmes ha implementado la herramienta de estudio de topología conocida como TOSCA. Esta herramienta permite a los diseñadores explorar diversas configuraciones de diseño y optimizar la topología de los objetos para mejorar su rendimiento y eficiencia. Además, SolidWorks también cuenta con módulos como Driven Generative Designer y Generative Shape Design, que permiten generar geometrías complejas y formas orgánicas mediante algoritmos y restricciones personalizadas.

Por su parte, Siemens ha desarrollado capacidades de optimización de topología y creación de geometría avanzada en su software NX y Solid Edge. Estas herramientas permiten a los diseñadores explorar una amplia gama de soluciones de diseño generativo, optimizando la forma y la estructura de los objetos para cumplir con requisitos específicos. Los módulos de diseño generativo de Siemens utilizan algoritmos avanzados para generar geometrías complejas y eficientes, maximizando la resistencia y reduciendo el peso de los objetos.

Tanto Dassault Systèmes como Siemens han incorporado módulos de diseño generativo en sus productos de software, como SolidWorks, CATIA, NX y Solid Edge. Estas herramientas brindan a los diseñadores capacidades avanzadas para explorar y optimizar diseños generativos, permitiéndoles generar geometrías complejas, mejorar la eficiencia y obtener resultados más innovadores en sus proyectos de diseño.

En el ámbito del diseño generativo, además de las soluciones ofrecidas por Autodesk, Siemens y Dassault Systèmes, existen otras herramientas informáticas especializadas que también merecen destacarse.

Frustum Inc. ha desarrollado Generate TrueSOLID, una herramienta de diseño generativo enfocada en el diseño mecánico y de productos. Esta herramienta permite a los diseñadores explorar una amplia gama de opciones de diseño, generando geometrías optimizadas y eficientes para diversas aplicaciones industriales.

Para la animación y el modelado, Houdini y Blender 3D son herramientas populares y ampliamente utilizadas. Estos programas brindan a los artistas y diseñadores una amplia gama de herramientas para la creación de animaciones y modelos 3D de alta calidad.

SolidThinking Inspire, desarrollado por Altair, se especializa en la optimización de superficies. Esta herramienta permite a los diseñadores explorar diferentes configuraciones de diseño y optimizar la topología y las formas de los objetos para lograr un rendimiento óptimo.

Para la fabricación avanzada, herramientas como nTopology y Elemento ofrecen capacidades de diseño generativo y fabricación aditiva. Estas herramientas permiten a

los diseñadores crear geometrías complejas y optimizadas para su producción mediante técnicas avanzadas de fabricación.

Otras herramientas destacadas incluyen Creo de PTC Corporation, MSC Apex Generative Design, Onshape y Paramatters, que ofrecen funcionalidades de optimización de topología y fabricación avanzada.

En el campo de la arquitectura, Archicad y Microstation son herramientas especializadas que ofrecen funcionalidades específicas para la creación de diseños arquitectónicos.

Es importante mencionar que algunas de estas herramientas pueden integrarse con los softwares de diseño asistido por ordenador ofrecidos por Autodesk, Siemens y Dassault Systèmes, permitiendo a los diseñadores aprovechar las capacidades de múltiples programas y obtener resultados más completos e integrados en sus proyectos de diseño generativo.

En el ámbito del diseño generativo, también existen otras herramientas experimentales y de recursos libres que merecen mención.

Automake Software es una herramienta en línea que combina sistemas generativos con conocimiento artesanal y tecnologías de producción digital. Este software permite la creación de objetos en 3D mediante el diseño generativo, ofreciendo una experiencia interactiva y creativa para los diseñadores.

NodeBox, por otro lado, es una aplicación especializada en el diseño de patrones y gráficos 2D. Esta herramienta permite a los usuarios crear diseños gráficos complejos y generativos mediante la programación visual, ofreciendo flexibilidad y control en la creación de patrones personalizados.

Además, hay una versión gratuita de Dynamo, una herramienta de programación gráfica de código abierto diseñada para el diseño en Autodesk Revit. Dynamo permite a los diseñadores crear algoritmos y scripts visuales para generar y modificar elementos arquitectónicos y estructurales en un entorno de modelado paramétrico.

Por último, pero no menos importante, cabe destacar Grasshopper for Rhinoceros. Grasshopper es una plataforma de programación visual que se integra con Rhinoceros, un software ampliamente utilizado en la industria del diseño. Grasshopper permite a los diseñadores explorar y desarrollar algoritmos generativos para crear geometrías complejas y formas innovadoras.

En resumen, estas herramientas experimentales y de recursos libres, como Automake Software, NodeBox, Dynamo y Grasshopper, ofrecen a los diseñadores alternativas para explorar el diseño generativo y expandir sus posibilidades creativas en diferentes contextos, desde la creación de objetos tridimensionales hasta el diseño de patrones y gráficos. A continuación, se muestra una tabla (Tabla 2) que destaca los software más representativos en el campo.

Tabla 2. Software para Diseño Generativo.

<u>Desarrollador</u>	<u>Software</u>	<u>Enfocado en</u>	<u>Plugin o Modulo</u>
Altair	SolidThinking Inspire	Optimización de topología	
Autodesk	Fusión 360	Optimización de topología y diseño de producto	
Autodesk	Netfabb	Optimización de procesos y superficies.	
Autodesk	Dynamo Studio	Industria de la construcción	
Autodesk / Mode Lab	Revit	Industria de la construcción	Dynamo
Autodesk	Inventor	Optimización de estructuras	Shape Generator / integración con Fusion 360
Autodesk	Within	Optimización de superficies e implantes médicos.	
Autodesk	Maya	Animaciones	Hypergraph
Bentley	Microstation	Arquitectura e Ingeniería	GenerativeComponents
Dassault Systèmes	SolidWorks	Optimización de superficies	Topology Study and Tosca
Dassault Systèmes	CATIA	Optimización de superficies	Driven Generative Designer and Generative Shape Design
Experimental Media Research Group	NodeBox	Patrones	
Frustum Inc.	Generate	Diseño mecánico y diseño de producto.	TrueSolid
Graphisoft	Archicad	Arquitectura	Grasshopper
McNell & Associates	Rhinoceros	Optimización de topología y diseño de productos.	Grasshopper
MSC Software	MSC Apex Generative Design	Optimización de topología	
nTopology	Element	Optimización de estructuras	
nTopology	nTopology	Diseño mecánico y diseño de producto.	
Onshape Inc.	Onshape	Optimización de topología	
Open-source	Blender 3D	Animación y modelado.	BY-GEN generative design addon and
PTC Corporation	Creo	Optimización de topología	Extensión de Diseño Generativo Creo, Extensión de Modelado y Optimización de Topología
SideFX	Houdini	Animaciones	
Siemens	NX (Unigraphics)	Optimización de topología	
Siemens	Solid Edge ST10	Optimización de topología	
University College Falmouth, Dr Justin Marshall	Automake	Diseños de manualidades/objetos.	

Fuente: Elaboración propia.

### 3. PLANTEAMIENTO E HIPÓTESIS.

Tras llevar a cabo una búsqueda exhaustiva en la bibliografía especializada, se han identificado factores de relevancia universal que abarcan la mayoría de las técnicas de diseño generativo. Entre estos factores se encuentran la estética, la utilización del potencial informático y la estabilidad de las soluciones propuestas. Basándonos en esta comprensión, se ha concebido una herramienta asistencial para el diseño conceptual de productos, diseñada para ampliar las capacidades de los diseñadores, sin importar su nivel de experiencia. La hipótesis de esta investigación es que al desarrollar una herramienta integral para el diseño conceptual de productos, se puede proporcionar a los diseñadores una solución que les permita explorar y generar conceptos de diseño estéticamente distintivos, aprovechando el potencial informático disponible y garantizando al mismo tiempo su viabilidad.

Como ya se ha mencionado en capítulos anteriores, los factores transversales cuentan con una relevancia en la etapa del diseño conceptual. Por ello, en este capítulo se va a profundizar la selección de los factores transversales incorporados en la herramienta asistencial, que se basó en aspectos relevantes en la fase conceptual del proceso de diseño y las oportunidades identificadas en investigaciones anteriores. (Reich, 1993) destacó la importancia de la estética en el diseño, subrayando que, a pesar de que a menudo se pasa por alto en estudios relacionados con sistemas de soporte de diseño computacional, la estética no debe ser ignorada por completo. Este enfoque resalta la relevancia de considerar la apreciación estética como un elemento fundamental en el diseño de productos.

Además, tanto (Stephen N. Smyth, 2000) como (Yanagisawa & Fukuda, 2005) hicieron hincapié en la importancia de concebir diseños conceptuales que logren una integración eficaz entre la forma y la función desde las etapas iniciales del proceso de diseño, aprovechando el potencial de sistemas CAD. Esto demuestra el reconocimiento del valor del uso del ordenador en la generación de soluciones de diseño desde las etapas conceptuales.

(Robertson & Radcliffe, 2009) y (Chang, Chien, Lin, Chen, & Hsieh, 2016) abordaron el impacto del uso del ordenador y las herramientas CAD en la creatividad del diseñador en las etapas iniciales del diseño, un aspecto también reconocido por (Krish, 2011) y

(Rosenman, 1997). Estas investigaciones han señalado la importancia de la viabilidad de las soluciones generadas y su relevancia en el proceso de diseño.

La estética se reconoció como un factor fundamental, ya que influye en la apreciación y aceptación de los productos diseñados. Aunque la evaluación estética es subjetiva y no existen mecanismos objetivos para validar las soluciones propuestas, se consideró importante abordar este aspecto en la etapa conceptual del diseño. Se enfocó en la composición general del producto y en la incorporación de elementos adicionales que enriquecieran su estética inicial.

Aquí se puede destacar también que el factor estético es uno de los más directamente abordables en la fase conceptual. La apariencia del producto es una de las primeras cosas que se definen, incluso estando sujeta a modificaciones en etapas posteriores.

El uso del ordenador se destacó como otro factor transversal relevante. Si bien su aplicación en el diseño generativo es ampliamente reconocida, su uso en la etapa conceptual del diseño no está tan difundido. Sin embargo, se reconoció su potencial para facilitar la generación de nuevas formas y se planteó desarrollar una herramienta que aprovechara estas capacidades, especialmente dirigida a diseñadores con poca experiencia.

La estabilidad de las soluciones generadas también se consideró un factor importante. No basta con generar formas novedosas, sino que estas deben ser viables y cumplir con requisitos específicos. Para garantizar que las soluciones propuestas sean útiles y confiables, se tomó en cuenta este factor en el desarrollo de la herramienta asistencial.

Se tiene que justificar la estética, el uso del ordenador en la etapa conceptual de diseño y la estabilidad de soluciones en el planteamiento del desarrollo de la herramienta asistencial basada en un algoritmo con componentes desarrollados en Python.

Estos factores transversales constituyen la base sobre la cual se establecieron los requisitos en el desarrollo de una herramienta de asistencia para el diseño de productos en la etapa conceptual. Estos elementos clave, presentes de manera consistente en el proceso de diseño, son fundamentales para garantizar la estética, la eficacia en el uso del potencial del ordenador y la estabilidad en los resultados de las soluciones generadas por la herramienta.

La incorporación de estos factores transversales en la formulación del estudio garantiza un enfoque integral para abordar los desafíos y objetivos comunes en el diseño generativo. Esto confiere a la herramienta una aplicabilidad versátil y adaptabilidad a diversos proyectos y contextos de diseño, proporcionando a los diseñadores una herramienta versátil y adaptable en la fase conceptual de desarrollo de productos.

El primer factor transversal identificado es la estética de las soluciones generadas en el diseño generativo. Debido a su naturaleza subjetiva, es difícil contar con mecanismos objetivos para validar o evaluar dichas soluciones (Alcaide-Marzal, Diego-Mas, & Acosta-Zazueta, 2020). Incluso cuando se utilizan técnicas generativas como algoritmos

genéticos que permiten una validación más precisa, aún se requiere de la interacción humana de forma continua para evaluar la estética (Yanagisawa & Fukuda, 2005).

Dado el reconocimiento de esta complejidad, en este trabajo se ha decidido abordar la estética de las soluciones, teniendo en cuenta las limitaciones inherentes a la etapa conceptual del diseño. En esta etapa temprana del proceso de diseño, los detalles específicos aún no están completamente definidos. Por lo tanto, se ha centrado en la composición general del producto, incorporando elementos adicionales que enriquezcan la estética inicial de los objetos.

Es importante destacar que la exploración de la estética en la etapa conceptual del diseño tiene sus limitaciones, pero proporciona una base sólida para guiar y enriquecer el proceso de diseño. Aunque no se puede lograr un nivel de detalle exhaustivo en esta etapa, se busca establecer una dirección estética general que inspire y oriente el desarrollo posterior del diseño.

En otras palabras, este factor se aborda principalmente en la composición general del objeto durante la etapa conceptual, con el propósito de agregar un mayor nivel de detalle a los elementos principales que lo conforman. Estos detalles pueden tomar la forma de elementos adicionales que contribuyen a la composición principal, los cuales se describirán con mayor precisión más adelante. Es importante destacar que el objetivo estético en esta etapa del diseño no radica en profundizar en particularidades específicas, sino en enriquecer la imagen general del objeto diseñado al incorporar elementos que ayuden a identificar su forma.

Por otro lado, otro de los factores transversales que merece atención se encuentra relacionado con el uso del ordenador en el diseño generativo. Se ha documentado ampliamente que el ordenador es un elemento destacable y poderoso para la generación de soluciones de diseño. Sin embargo, en la etapa del diseño conceptual, su utilización no está tan extendida como se podría esperar.

La falta de uso generalizado del ordenador en esta etapa se debe principalmente a que puede convertirse en un factor limitante en la aparición de ideas creativas. El diseño conceptual es un proceso en el que se busca explorar y desarrollar conceptos novedosos, y el uso excesivo de herramientas digitales puede restringir la libre generación de ideas, volviendo el proceso más mecánico y menos intuitivo.

Conscientes de esta limitación, se ha tomado la iniciativa de abordar esta problemática y buscar soluciones que permitan aprovechar al máximo el potencial del ordenador en la etapa del diseño conceptual. Para ello, se ha desarrollado una serie de componentes instalables, integrados en una herramienta asistencial, que facilitan la generación de nuevas formas de objetos y brindan apoyo en este proceso creativo.

Estos componentes instalables están diseñados para ser intuitivos y de fácil manejo, permitiendo a los diseñadores explorar diferentes opciones de diseño y experimentar con diversas configuraciones. Además, se han incorporado mecanismos que fomentan

la interacción y retroalimentación entre el diseñador y la herramienta, de manera que se estimule la creatividad y se promueva la generación de ideas innovadoras.

En la generación de soluciones de diseño, no solo basta con la concepción de formas, sino que estas formas también deben cumplir con ciertas características y requisitos para considerarlas como soluciones válidas. Estos requisitos pueden incluir aspectos técnicos, funcionales, ergonómicos, estéticos o cualquier otra restricción específica del proyecto.

Es importante destacar este factor transversal en el contexto de nuestro estudio, ya que si queremos lograr una vinculación efectiva entre el uso del ordenador y el diseño conceptual, las soluciones generadas en esta etapa deben ser relevantes y útiles para el diseñador. Esto implica que las formas generadas no solo deben ser formas novedosas, sino también funcionales y factibles de implementar en la realidad.

Para abordar este desafío, la herramienta asistencial, la cual está basada en algoritmos desarrollados en Python, incorpora funcionalidades con la capacidad de establecer restricciones específicas, lo que permite que las soluciones propuestas cumplan con los requisitos definidos por el propio usuario. Con esa capacidad se ofrece la opción para ajustar y refinar las soluciones generadas, brindando al diseñador la flexibilidad necesaria para encontrar el equilibrio adecuado.

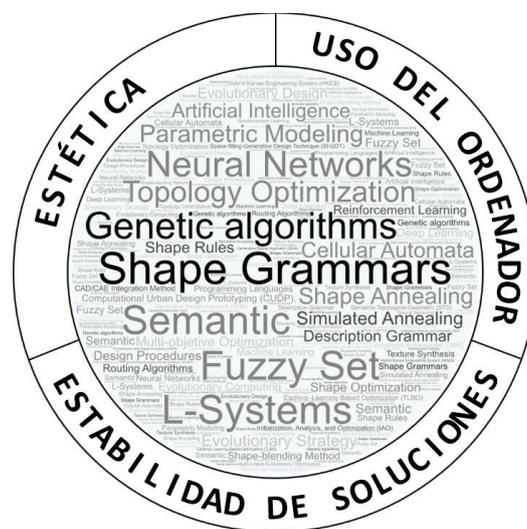


Figura 1. Factores transversales. Fuente: Elaboración propia.

A pesar de la búsqueda bibliográfica realizada, no se encontró una herramienta existente que aborde los tres factores transversales (Fig. 1. Estética, uso del ordenador y estabilidad de las soluciones) de manera integrada en la etapa del diseño conceptual. Si bien existen herramientas y técnicas que se centran en uno o dos de estos aspectos, no se encontró una solución completa que abarque los tres de manera simultánea y coherente.

Esta brecha en el panorama actual resalta la importancia y la necesidad de desarrollar una herramienta (o metodología) que aborde estos factores transversales en conjunto.

Al combinar la estética, el uso del ordenador y la estabilidad de las soluciones, se busca proporcionar a los diseñadores una herramienta integral que les permita explorar y generar soluciones de diseño conceptuales que sean estéticamente identificativas, utilicen eficientemente el potencial del ordenador y sean viables desde el punto de vista técnico y funcional.

El desarrollo de una herramienta que aborde estos tres factores transversales juntos podría brindar a los diseñadores una mayor eficiencia y creatividad en la etapa conceptual del diseño. Al contar con una herramienta que integre estas capacidades, los diseñadores podrían explorar un amplio espectro de soluciones de diseño, evaluar rápidamente su viabilidad y obtener resultados novedosos.

En resumen, la falta de una herramienta existente que aborde de manera conjunta los factores transversales identificados en la etapa del diseño conceptual resalta la oportunidad de desarrollar una solución integral que satisfaga estas necesidades. La creación de una herramienta asistencial basada en algoritmos y componentes desarrollados en Python, que tenga en cuenta la estética, el uso del ordenador y la estabilidad de las soluciones, podría brindar a los diseñadores una poderosa herramienta para la generación de soluciones de diseño en la etapa conceptual.

## 4. DESARROLLO.

El análisis de los objetos a construir fue el primer paso en el desarrollo de la herramienta de asistencia. Se inició observando la estructura o esqueleto que define la forma básica de cada objeto. Se identificaron elementos comunes en la morfología de distintos objetos ordinarios con el fin de establecer una estructura elemental que serviría de base para la creación de otros objetos.

El objetivo de este análisis fue encontrar características básicas en la morfología de los objetos y utilizarlas como puntos de partida para la generación de nuevas formas. Al identificar similitudes estructurales entre diferentes objetos, se pudo flexibilizar y reutilizar estas estructuras para la creación de nuevos objetos que compartieran características morfológicas.

Este enfoque permitió simplificar el proceso de diseño al establecer una estructura inicial que serviría como punto de partida para la generación de soluciones de diseño. Al aprovechar las similitudes morfológicas estructurales, se logró agilizar el proceso de creación de nuevos objetos, brindando al diseñador una base sólida sobre la cual desarrollar y explorar diferentes variaciones y combinaciones.

El siguiente paso en el desarrollo de la herramienta de asistencia fue describir la relación entre cada parte de la estructura del objeto. Para esto, se utilizaron los principios establecidos en el trabajo de McKay, Stiny y De Pennington (McKay, Stiny, & De Pennington, 2016), donde se identificaron dos tipos de relaciones en una estructura de diseño: composición y conexión.

En primer lugar, se estudiaron las partes que conforman el objeto y se definieron los diferentes tipos de cuerpos presentes en la composición. Según McKay (McKay et al., 2016), se distinguieron tres tipos de cuerpos: el cuerpo principal, los cuerpos secundarios y los cuerpos de conexión (Fig. 2).

El cuerpo principal fue descrito como el elemento dominante de la composición. Es el cuerpo de mayor importancia estructural y ocupa una posición relevante en la forma del objeto. Además, el cuerpo principal ejerce influencia sobre los demás elementos de la composición.

Por otro lado, los cuerpos secundarios son aquellos que generalmente son más pequeños que el cuerpo principal. Estos cuerpos son influenciados por el cuerpo principal y desempeñan un papel complementario en la composición global del objeto.

Finalmente, se identificaron los cuerpos de conexión, los cuales tienen como objetivo principal la unión y conexión entre los diferentes cuerpos que conforman el objeto. Estos cuerpos, como el cuerpo principal y los secundarios, son fundamentales para mantener la integridad estructural y funcional del objeto.

Al describir estas relaciones y definir los diferentes tipos de cuerpos en la composición del objeto, se sentaron las bases para la generación de nuevas formas y estructuras mediante la herramienta de asistencia. Estos conceptos permitieron establecer pautas claras para la organización y conexión de los elementos en el proceso de diseño generativo.



Figura 2. Cuerpo principal (izquierda), secundario (centro) y de conexión (derecha).

Durante el análisis realizado, se consideró fundamental obtener una descripción general de los objetos que reflejara sus características proporcionales sustanciales en lugar de una descripción precisa y elemental. Esto permitiría establecer una especificación morfológica básica. Para lograrlo, se empleó el concepto de proporciones generales propuesto por Akner-Koler (Akner-koler, 1994), el cual describe las características proporcionales esenciales de un elemento, a saber: extensión, superficie y masa (Fig. 3).

La dimensión extensional se refiere a la longitud del objeto y proporciona información sobre su forma y tamaño en relación con el entorno. La dimensión superficial se refiere a la planitud o la superficie del objeto, es decir, cómo se extiende visualmente en el espacio y cómo interactúa con la luz y las sombras. Por último, la dimensión masiva se relaciona con el volumen del objeto, es decir, su capacidad para ocupar un espacio tridimensional.

Estas descripciones proporcionales fueron utilizadas para nombrar las categorías propuestas, combinadas con otros conceptos que brindaban una descripción más detallada de cada una de ellas.

Al incorporar estas descripciones proporcionales en el análisis y desarrollo de la herramienta de asistencia, se buscó capturar la esencia morfológica básica de los objetos. Esto permitió establecer una base sólida para la generación de nuevas formas y estructuras en el diseño conceptual.



Figura 3. Elemento extensional (izquierda), superficial (centro) y masivo (derecha).

En el desarrollo de la herramienta de asistencia, también se procedió a describir los detalles esenciales y no esenciales de los objetos. Se consideraron detalles esenciales aquellos que son indispensables para la identificación morfológica del objeto, es decir, aquellos elementos que son necesarios para comprender su forma y estructura fundamental.

Por otro lado, se identificaron los detalles no esenciales, los cuales no están directamente relacionados con el funcionamiento o la identificación del objeto en sí. Estos elementos pueden incluir adornos, ornamentos o características secundarias que no son esenciales para la comprensión básica del objeto.

La distinción entre detalles esenciales y no esenciales resultó importante para el diseño generativo, ya que permitió enfocar la generación de soluciones en los aspectos fundamentales del objeto, evitando la sobrecarga de elementos superfluos. Al centrarse en los detalles esenciales, se buscó obtener soluciones que capturarán la esencia y la identidad del objeto, optimizando así el proceso de diseño conceptual.

Esta diferenciación entre detalles esenciales y no esenciales (Fig. 4) fue considerada en el desarrollo de la herramienta de asistencia, con el objetivo de ofrecer al diseñador la posibilidad de enfocarse en los aspectos clave del objeto durante la etapa conceptual.



Figura 4. A la izquierda detalles esenciales de una cocina, a la derecha los no esenciales.

Después de establecer todos los elementos anteriores, se planteó una clasificación con el objetivo de identificar diferentes familias de objetos que podrían compartir una estructura principal común. Además, se buscaba determinar qué productos requerían la incorporación de subestructuras y detalles representativos en su construcción.

Esta clasificación permitió agrupar los objetos en categorías o familias en base a su estructura fundamental. Se identificaron elementos estructurales clave que eran

compartidos por varios objetos dentro de una misma familia. Esto proporcionó la oportunidad de utilizar subestructuras similares y detalles representativos característicos en la generación de nuevos diseños.

La clasificación propuesta fue un paso importante en el desarrollo de la herramienta de asistencia, ya que facilitó la generación de formas de objetos dentro de cada familia. Al identificar las subestructuras y detalles indispensables, se logró una mayor coherencia y consistencia en los diseños generados. Además, se fomentó la flexibilidad y la diversidad al permitir la creación de productos únicos y distintivos dentro de cada categoría.

Esta clasificación fue una estrategia clave para identificar las familias de objetos y determinar los elementos necesarios en su construcción. Esto sentó las bases para el desarrollo de la herramienta de asistencia en el diseño conceptual de productos.

Para implementar el sistema generativo se utilizó lenguaje de programación visual (LPV). En este caso, se utilizó Rhino-Grasshopper que es una plataforma de LPV de diseño paramétrico que permite, mediante la conexión de componentes, el control de parámetros y flujo de datos que dirigen la forma del modelo paramétrico.

La elección de Rhinoceros y Grasshopper para el desarrollo e implementación de la herramienta se basa en varias razones fundamentales. En primer lugar, Rhinoceros es un software ampliamente reconocido y utilizado en la industria del diseño, especialmente en arquitectura, ingeniería y diseño industrial. Su versatilidad y capacidad para manejar modelos 3D complejos lo convierten en una opción sólida para nuestras necesidades.

Además, Rhinoceros se complementa perfectamente con Grasshopper, una potente extensión de programación visual. Grasshopper nos permite crear algoritmos generativos y paramétricos, lo que resulta crucial en nuestro enfoque de diseño. Esta capacidad nos permite explorar rápidamente diferentes opciones de diseño, generar formas complejas y adaptar nuestros productos a las necesidades específicas de nuestros clientes.

Otra ventaja de utilizar Grasshopper es su interfaz intuitiva y fácil de usar. Esto facilita el proceso de diseño y desarrollo de la herramienta, permitiéndonos iterar rápidamente y realizar ajustes según sea necesario.

Además, la comunidad de usuarios de Rhinoceros y Grasshopper es activa y cuenta con una gran cantidad de recursos disponibles, como plugins, scripts y tutoriales. Esto nos brinda la oportunidad de aprender y mejorar continuamente nuestras habilidades de diseño y programación, y nos permite estar al tanto de las últimas tendencias y técnicas en el campo del diseño generativo.

En resumen, elegimos Rhinoceros y Grasshopper debido a su versatilidad, capacidad generativa, interfaz intuitiva y el respaldo de una comunidad activa. Estas herramientas nos brindan la flexibilidad y las capacidades necesarias para desarrollar nuestra herramienta de manera efectiva y ofrecer soluciones innovadoras en el campo del diseño.

La organización de trabajo dentro del algoritmo propuesto sigue una secuencia específica para lograr el diseño generativo deseado. Primero, se obtiene la estructura

básica del objeto, que sirve como punto de partida para la generación del diseño. A partir de esta estructura, se procede a crear los volúmenes de las partes que conforman los cuerpos del objeto, definiendo sus formas y dimensiones.

Posteriormente, se incorporan subestructuras al diseño principal, que agregan complejidad y detalle al objeto. Estas subestructuras pueden ser elementos decorativos, funcionales o estructurales que enriquecen tanto la apariencia como el rendimiento del objeto final. Al introducir estas subestructuras, se busca añadir elementos complementarios que mejoren la funcionalidad y la estética del diseño global.

La inclusión de estas subestructuras puede implicar la manipulación de parámetros y variables adicionales, permitiendo la personalización y adaptación del objeto según las necesidades y preferencias del usuario. La variedad de subestructuras disponibles proporciona una amplia gama de posibilidades para la configuración y la optimización del diseño.

Adicionalmente, se lleva a cabo la colocación de detalles esenciales y no esenciales en el diseño. Los detalles esenciales se refieren a elementos necesarios para el correcto funcionamiento o utilidad del objeto, tales como conexiones, uniones o mecanismos específicos. Estos detalles son fundamentales para garantizar que el objeto cumpla con su propósito y desempeñe adecuadamente su función principal.

Por otro lado, se consideran los detalles no esenciales, los cuales son elementos opcionales que añaden valor estético o complementan la funcionalidad del objeto. Estos detalles no son estrictamente necesarios desde un punto de vista funcional, pero contribuyen a mejorar la apariencia, la experiencia de uso o incluso aportan características adicionales al objeto.

La colocación estratégica de estos detalles esenciales y no esenciales en el diseño generativo permite crear objetos que no solo cumplan con sus funciones básicas, sino que también sean estéticamente atractivos y personalizables. De esta manera, se busca lograr un equilibrio entre la funcionalidad, la estética y la experiencia del usuario en el diseño final.

Es relevante resaltar que en el desarrollo del algoritmo se emplearon definiciones clave para su implementación exitosa. La estructura básica del objeto se refiere a la configuración fundamental que sirve como base para la construcción del diseño. Es la estructura primaria sobre la cual se desarrolla todo el objeto.

Por otro lado, se utilizan las subestructuras, que son elementos secundarios que se añaden al diseño principal. Estas subestructuras agregan complejidad y detalle al objeto, aportando características adicionales, funcionales o estéticas, que enriquecen el diseño general.

Además, se incorporan elementos adicionales, los cuales son detalles que se incluyen para enriquecer aún más el objeto. Estos elementos pueden ser tanto funcionales como estéticos, y se utilizan para mejorar la utilidad, el aspecto visual o incluso la experiencia de uso del objeto final.

Estas definiciones clave son fundamentales para la comprensión y la implementación adecuada del algoritmo generativo. Al trabajar con la estructura básica, las

subestructuras y los elementos adicionales, se logra desarrollar diseños completos y sofisticados que cumplen con los objetivos estéticos y funcionales establecidos. Son definiciones cardinales para comprender y contextualizar los resultados obtenidos a lo largo del proceso de diseño generativo.

Antes de continuar con la presentación de los resultados obtenidos, es importante enunciar las definiciones que tanto se han hablado hasta ahora: estructura básica del objeto, subestructuras y elementos adicionales.

#### 4.1 Estructura del producto.

La estructura del producto se puede entender como el armazón o esqueleto que proporciona la configuración general del objeto a representar. En esta estructura, se disponen secciones individuales que sirven como espacios para posicionar los volúmenes geométricos que conforman el objeto. El propósito principal de esta configuración es lograr una representación realista y coherente del objeto.

Cada sección de la estructura mantiene una relación intrínseca con las demás partes del objeto, lo que convierte a la estructura en una compleja red de partes interconectadas. Estas relaciones y conexiones, tanto en términos de organización como de conectividad, son las que finalmente dan forma al objeto y establecen su interrelación. Cada parte individual del objeto cuenta con parámetros definidos que abarcan aspectos como su forma, tamaño y posición. A su vez, estas partes están vinculadas mediante una serie de relaciones que imponen una estructura al conjunto, influyendo en la forma, el movimiento, la funcionalidad y otros aspectos del objeto.

La comprensión y consideración de la estructura del producto, junto con sus partes y relaciones internas, desempeñan un papel crucial en el diseño generativo. Estas consideraciones son fundamentales para sentar las bases de la generación de nuevas formas y la exploración de posibilidades creativas en el diseño. La literatura, incluyendo investigaciones como el trabajo de (Mitra et al., 2014), respalda y enriquece la importancia de esta comprensión de la estructura en el diseño de productos.

El análisis detallado de la estructura de un producto permite a los diseñadores identificar los componentes clave, las conexiones entre ellos y su función en conjunto. Esto es esencial para garantizar que las nuevas formas generadas no solo sean estéticas, sino también funcionales y prácticas. Además, comprender la estructura existente proporciona un valioso punto de partida para la innovación y la mejora del diseño, permitiendo a los diseñadores explorar cómo optimizar o reinventar las relaciones y funciones dentro del producto.

El trabajo de (Mitra et al., 2014) y otras investigaciones en el campo del diseño de productos refuerza la idea de que la estructura es un aspecto crítico que debe considerarse en el diseño generativo. Estos estudios aportan conocimientos y enfoques que pueden aplicarse para lograr diseños más efectivos y creativos, respaldando así el proceso de diseño generativo en diversas industrias y aplicaciones.

En resumen, la comprensión de las estructuras comunes que definen la morfología de objetos, así como la relación estrecha entre los objetos de una misma colección, es esencial para desarrollar herramientas de diseño generativo efectivas. Esto permite la

generación de nuevas formas y diseños creativos, manteniendo la coherencia y funcionalidad dentro de un grupo funcional específico.

#### 4.2 Subestructuras.

Las subestructuras son elementos que se añaden a la estructura principal del objeto con el fin de agregar características de forma, tanto morfológicas como funcionales, que contribuyen a su identificación y distinción. Estas subestructuras se generan ensamblando partes de formas existentes dentro de una colección, manteniendo la estructura original del objeto (Mitra et al., 2014).

La generación de formas novedosas se logra al combinar diferentes partes o elementos de la colección de manera creativa y única. Estas subestructuras pueden adoptar diferentes formas y se pueden representar en formato de líneas o puntos de referencia, dependiendo de la naturaleza y la finalidad del objeto.

La incorporación de subestructuras permite explorar nuevas posibilidades de diseño dentro de los límites establecidos por la estructura principal. Al agregar características adicionales a través de las subestructuras, se enriquece la morfología y la funcionalidad del objeto, proporcionando una mayor variedad y adaptabilidad a las necesidades y preferencias del usuario.

Es importante destacar que las subestructuras no alteran la estructura básica del objeto, sino que complementan y realzan su forma y función. Al preservar la estructura original, se asegura la coherencia y la identificación del objeto dentro de la colección a la que pertenece, al tiempo que se fomenta la innovación y la personalización en el proceso de diseño.

En el caso de las líneas como subestructuras, estas se definen como configuraciones alámbricas que se extienden desde un punto de la estructura principal hacia otro punto en el espacio de modelación. Estas líneas tienen la función de proporcionar una referencia para la colocación de cuerpos volumétricos adicionales que contribuyen a la construcción del producto.

Es importante destacar que estas líneas no se consideran elementos de detalle esencial o no esencial, ya que forman parte de la morfología general del objeto y no están directamente relacionadas con la funcionalidad específica o los detalles decorativos. Su propósito principal es definir la forma y la orientación de los cuerpos volumétricos que se agregan al objeto.

Al utilizar líneas como subestructuras, se pueden crear configuraciones y geometrías complejas alrededor de la estructura principal, lo que permite una mayor flexibilidad en el diseño y la posibilidad de explorar nuevas formas y expresiones estéticas. Estas líneas actúan como guías visuales y espaciales para la colocación precisa de los elementos volumétricos, asegurando la coherencia y la integridad del diseño generativo.

### 4.3 Elementos adicionales.

Los elementos adicionales pueden incluir tanto detalles esenciales como detalles no esenciales. Los detalles esenciales son aquellos elementos necesarios para la identificación y funcionalidad del objeto. Estos detalles son indispensables y están directamente relacionados con el propósito y uso del objeto. Pueden ser componentes estructurales, mecanismos, conexiones o cualquier otro elemento que cumpla una función específica en el objeto.

Por otro lado, los detalles no esenciales son elementos que no son necesarios para la funcionalidad del objeto, pero que contribuyen a su composición gráfica y estética. Estos detalles pueden incluir ornamentaciones, patrones, texturas, o cualquier otro elemento que aporte valor estético al objeto.

La inclusión de elementos adicionales, tanto esenciales como no esenciales, permite personalizar el diseño y brinda la oportunidad de crear objetos más atractivos y visualmente interesantes. Al combinar detalles esenciales y no esenciales, se logra un equilibrio entre la funcionalidad y la estética, creando objetos que no solo cumplen con su propósito principal, sino que también resultan agradables a la vista y generan una conexión emocional con el usuario.

Dicho lo anterior, el objetivo de estos elementos es brindar al usuario una forma fácil y accesible de construir e integrar elementos adicionales en el diseño del producto que se desea representar. Se busca ofrecer herramientas y métodos que simplifiquen el proceso de incorporación de estos elementos, permitiendo al usuario explorar y experimentar con diferentes posibilidades de diseño.

Es importante destacar que estos elementos adicionales deben contar con una amplia generación de soluciones de diseño. Esto significa que deben ofrecer múltiples opciones y variedad en cuanto a formas, estilos y características. Esto es fundamental para cumplir con el objetivo general del algoritmo, que es proporcionar una gama diversa de formas y opciones de diseño para el producto final.

Al permitir una amplia generación de soluciones de diseño, se fomenta la creatividad y la personalización. Los usuarios tienen la libertad de explorar diferentes combinaciones y variaciones, adaptando los elementos adicionales a sus necesidades y preferencias específicas. Esto no solo enriquece la experiencia de diseño, sino que también permite la creación de productos únicos y distintivos.

Por otro lado, se pretende que los elementos adicionales sean capaces de reproducirse directamente en los cuerpos volumétricos que componen las partes del objeto y que son establecidos con anterioridad en la estructura principal o subestructura. En el desarrollo de estos componentes, se antepuso principalmente que estos entrañen propiedades de variabilidad de forma y facilidad de control en su posición geométrica.

### 4.4. La creación de los componentes.

En el ámbito del diseño, el uso de ordenadores ha sido una herramienta omnipresente en las últimas décadas. Sin embargo, su papel se ha limitado en gran medida a funciones de dibujo y almacenamiento de información. En este sentido, es crucial diferenciar entre

el simple acto de plasmar gráficamente una idea preconcebida, es decir, dibujar o modelar, y el verdadero proceso de creación, que implica la generación y materialización de un objeto desde cero utilizando el sistema informático como medio principal.

Durante mucho tiempo, el diseño en el ordenador se ha centrado en representar y manipular objetos existentes de manera digital, lo que ha limitado su potencial creativo y generativo. Sin embargo, la tecnología y las herramientas de diseño han evolucionado significativamente, permitiendo ahora explorar nuevas posibilidades y enfoques en el diseño asistido por ordenador.

En este contexto, se plantea la necesidad de utilizar el poder y la versatilidad de los sistemas informáticos de manera más amplia, aprovechando su capacidad para generar soluciones de diseño innovadoras y explorar ideas completamente nuevas. Esta transición implica un cambio de paradigma, pasando de utilizar el ordenador como una herramienta pasiva a convertirlo en un instrumento activo de creación.

La elección del software adecuado desempeña un papel crucial en esta transformación. Se busca una plataforma que vaya más allá de las funcionalidades tradicionales de dibujo y almacenamiento, y que ofrezca herramientas y capacidades que fomenten la generación y materialización de ideas creativas. Este software debe permitir a los diseñadores trabajar directamente en el sistema informático, desde la concepción inicial hasta la realización final del objeto, ofreciendo un entorno flexible y potente que facilite la experimentación y la exploración de múltiples alternativas de diseño.

En la siguiente sección, exploraremos la elección del software de diseño y la plataforma seleccionada, considerando cómo se alinea con la visión de utilizar el sistema informático como un verdadero instrumento de creación y cómo puede potenciar el proceso de diseño generativo y conceptual.

En el contexto del desarrollo de la herramienta de diseño asistido, es crucial tomar decisiones fundamentales en cuanto al software a utilizar. La elección del software de diseño no solo afecta la viabilidad y funcionalidad de la herramienta, sino que también influye en la eficiencia del proceso de diseño y en la calidad de los resultados obtenidos. En esta sección, exploraremos los motivos detrás de la elección de un software específico y cómo este se alinea con los objetivos y requisitos del proyecto.

Los componentes necesarios para abordar este tema fueron desarrollados principalmente en el lenguaje de programación Python dentro de Grasshopper. La elección de Python se basó en varias razones fundamentales.

En primer lugar, Python es un lenguaje de programación altamente versátil y flexible. Admite múltiples paradigmas de programación, lo que brinda una amplia variedad de enfoques para abordar problemas y desarrollar soluciones. Esto fue esencial para la implementación de los componentes requeridos en el algoritmo, ya que se necesitaba un lenguaje que permitiera la creación de funciones y estructuras de datos adecuadas para el diseño generado.

Además, Python ofrece la posibilidad de generar componentes instalables dentro de Grasshopper. Esto significa que los componentes desarrollados en Python pueden ser fácilmente integrados en el flujo de trabajo de diseño dentro de Grasshopper, proporcionando una interfaz intuitiva y accesible para los usuarios. Esta capacidad de

crear componentes instalables fue especialmente beneficiosa, ya que facilita la interacción con el usuario y le permite utilizarlos de manera eficiente.

Otro factor importante que influyó en la elección de Python fue el tiempo de iteración reducido para ejecutar el algoritmo. Python es conocido por ser un lenguaje eficiente en términos de velocidad de ejecución, lo que resulta en tiempos de cálculo más rápidos. Esto fue crucial para garantizar una experiencia de diseño fluida y ágil, donde los cambios en los parámetros o en la configuración del algoritmo se reflejarán rápidamente en el modelo 3D generado.

En resumen, el lenguaje de programación Python fue elegido para desarrollar los componentes necesarios en el algoritmo debido a su versatilidad, capacidad de generar componentes instalables en Grasshopper y su tiempo de ejecución eficiente. Estas características permitieron la implementación de un sistema generativo robusto y de alto rendimiento dentro del entorno de diseño de Grasshopper.

#### 4.5. Selección del Software CAD

En la actualidad, los avances en el rendimiento de los sistemas informáticos han abierto nuevas posibilidades en el campo del diseño. Estas mejoras en el procesamiento de datos han brindado a los ordenadores un alto potencial como herramienta para la exploración de soluciones en el diseño (Krish, 2011). Ya no se trata simplemente de definir o resolver la geometría de un objeto, sino de trabajar con una amplia gama de datos relacionados con reglas específicas que un ordenador puede procesar e interpretar.

El diseño generativo, en combinación con este potencial informático, nos permite abordar los desafíos del diseño a un nivel que sería difícil de alcanzar únicamente con la capacidad humana. Al utilizar algoritmos y técnicas generativas, podemos establecer un conjunto de parámetros y reglas, permitiendo al ordenador generar múltiples soluciones de diseño que cumplan con criterios específicos. Este enfoque nos brinda la capacidad de explorar un amplio espectro de posibilidades y descubrir soluciones innovadoras y fuera de lo común.

La capacidad de procesamiento de datos de los ordenadores permite realizar cálculos complejos y análisis exhaustivos en un corto período de tiempo. Esto acelera el proceso de diseño y nos brinda la oportunidad de evaluar rápidamente múltiples alternativas, permitiendo una toma de decisiones más informada y fundamentada en datos. Además, la capacidad de iterar y ajustar rápidamente los parámetros de diseño nos permite perfeccionar y refinar continuamente nuestras soluciones.

Al aprovechar el potencial informático en el diseño generativo, podemos superar los límites de nuestra propia creatividad y explorar nuevas fronteras en la concepción y materialización de objetos. Esto nos permite enfrentar desafíos de diseño complejos y encontrar soluciones que de otro modo serían difíciles de imaginar.

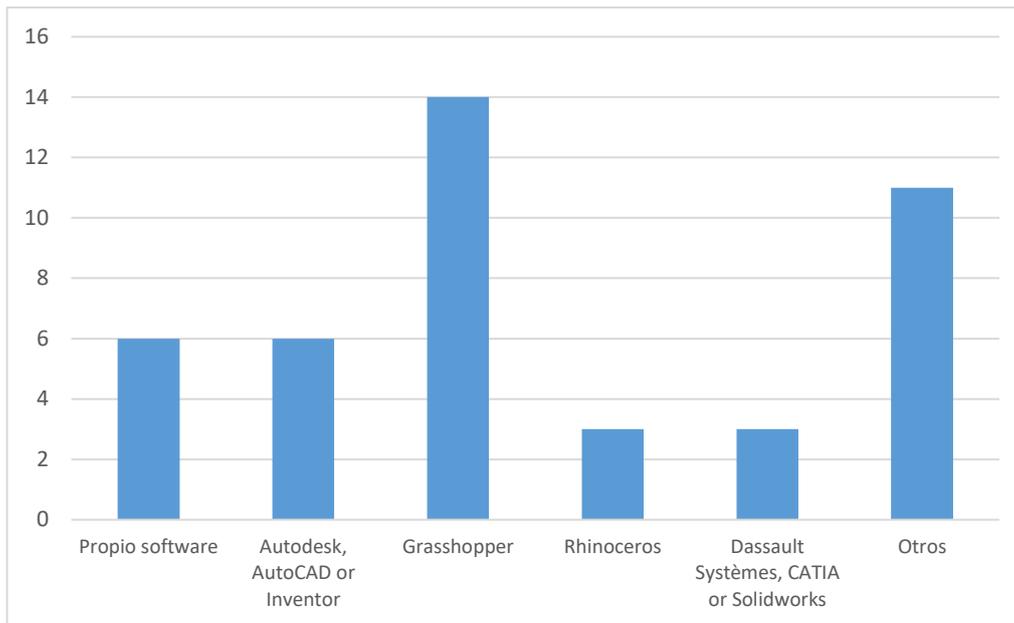
Con el uso del ordenador, el Diseño Generativo puede constituir una forma diferente de romper bloqueos creativos al dejar que el ordenador genere alternativas. El diseñador puede explorar una gama mucho más amplia de posibles diseños de productos en un tiempo comparativamente menor. Es una disciplina relativamente nueva en algunos

campos como el diseño industrial. Está más extendido en el diseño de arquitectura, donde la naturaleza algorítmica de la geometría generada encaja perfectamente en la estética de los edificios modernos.

En cuanto al aspecto tecnológico, es importante señalar que Diseño Generativo es un método que utiliza capacidades computacionales para ayudar a los diseñadores y/o automatizar algunas partes del proceso de diseño (Singh & Gu, 2011). Aquí, los lenguajes de programación visual como Grasshopper, se están volviendo populares debido a su facilidad de uso y su rápido aprendizaje. La investigación de (Leitão, Santos, & Lopes, 2012) es un estudio comparativo entre los lenguajes de programación textual modernos y los lenguajes de programación visual.

El diseño generativo, con su enfoque en el diseño asistido por ordenador, se distingue del sistema CAD tradicional al poner énfasis en la exploración de transformaciones de forma a través del control de parámetros.(Lin & Lee, 2013). Chase (S C Chase, 2005) argumenta que el diseño CAD tradicional carece del potencial exploratorio del diseño generativo, ya que los sistemas CAD actúan como ayudas de diseño donde el usuario ingresa y manipula formas directamente, en contraste con la computación generativa cuyo poder para explorar soluciones es mucho más amplio. Sin embargo, para que un sistema de diseño generativo sea efectivo, debe contar con un mecanismo de fluidez comunicativa entre el diseñador y el sistema (Miquel Prats et al., 2009). Esto significa que el diseñador debe ser capaz de comunicarse de manera efectiva con el ordenador y viceversa, para lograr una colaboración fructífera y una generación de formas que se ajuste a los criterios y necesidades específicas del proyecto. El diseño generativo ofrece la posibilidad de explorar nuevas soluciones y encontrar diseños innovadores, pero para aprovechar al máximo su potencial, es esencial establecer una comunicación fluida y eficiente entre el diseñador y la máquina.

En el campo del diseño generativo, existen numerosas herramientas de software que permiten la interacción del usuario con este enfoque. La gráfica 1 muestra las principales herramientas CAD utilizadas en los trabajos documentados en este estudio. Es notable el uso de Rhinoceros / Grasshopper en esta investigación. Rhinoceros es ampliamente reconocido y utilizado en la industria del diseño debido a su versatilidad y capacidad para manejar modelos 3D complejos. Por su parte, Grasshopper es una poderosa plataforma de diseño paramétrico que permite a los diseñadores controlar parámetros y flujos de datos para dirigir la forma del modelo paramétrico. La combinación de Rhinoceros y Grasshopper ofrece a los diseñadores una herramienta flexible y eficiente para explorar una amplia gama de posibles diseños de productos en un tiempo relativamente corto. Además, estas herramientas permiten una comunicación fluida entre el diseñador y el ordenador, lo que facilita la co-creación de diseños que se adaptan a las necesidades específicas del proyecto.



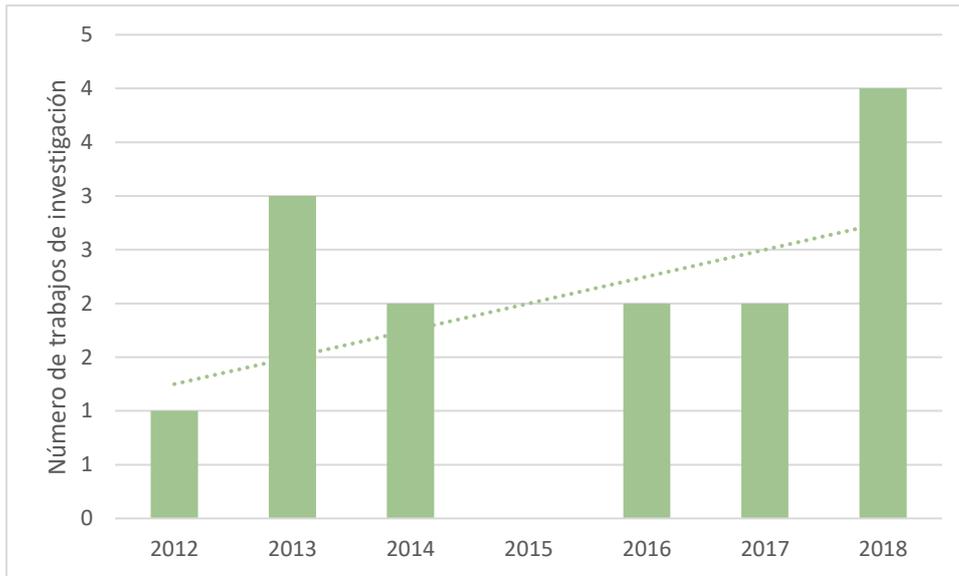
Otros: Solidworks, Unigraphics, Microstation, Blender 3D Software, EduDesign (3D Studio plugin), CATIA, Trimble Sketchup, XTreme ToolkitPro 2008, Maya, CityEngine.

Gráfica 1. Herramientas CAD utilizadas para DG. Fuente: Elaboración propia.

La creciente popularidad de Grasshopper en el campo del diseño generativo se encuentra respaldada por una serie de investigaciones significativas. Lin & Lee enfatizan que el entorno de programación extensible de Grasshopper brinda a los diseñadores la capacidad de concebir una amplia variedad de soluciones de diseño potenciales dentro de un espacio de solución definido, en contraposición a la construcción de un modelo digital final rígido (Lin & Lee, 2013). Esta flexibilidad y libertad para explorar diversas opciones de diseño convierten a Grasshopper en una herramienta atractiva para diseñadores que desean experimentar con nuevas ideas y enfoques creativos.

Además, el trabajo de Leitão reconoce que los lenguajes de programación visual, como Grasshopper, resultan ser más productivos y motivadores, especialmente para los principiantes en el campo del diseño generativo (Leitão et al., 2012). La facilidad de uso y la rápida curva de aprendizaje hacen que Grasshopper sea una elección popular entre los diseñadores que desean sumergirse en el diseño generativo sin necesidad de poseer un conocimiento profundo de programación.

Estos estudios e investigaciones respaldan la creciente tendencia de popularidad de Grasshopper en la investigación contemporánea sobre diseño generativo, como se puede apreciar en la gráfica 2. La versatilidad, la capacidad para fomentar la exploración creativa y la facilidad de uso convierten a Grasshopper en una herramienta poderosa para los diseñadores que aspiran a aprovechar al máximo el potencial del diseño generativo en sus proyectos. Su capacidad para impulsar la innovación y la experimentación en el diseño lo posiciona como una elección sólida y atractiva en la comunidad de diseñadores.



Gráfica 2. Línea de tendencia en el uso de Grasshopper por años. Fuente: Elaboración propia.

## 5. DISEÑO DE LA HERRAMIENTA

Durante el análisis de los productos para la elaboración de la clasificación de las estructuras, se pudo identificar claramente dos familias principales de objetos: aquellos cuya forma general es horizontal y aquellos cuya forma general es vertical.

En la familia de objetos con forma general horizontal, se encontraron elementos como mesas, sillas, escritorios y otros muebles que tienden a tener una disposición más extendida en el plano horizontal. Estos objetos suelen ser utilizados para proporcionar superficies de apoyo y áreas de trabajo.

Por otro lado, en la familia de objetos con forma general vertical, se identificaron elementos como estanterías, armarios, torres y otros objetos que se extienden predominantemente en sentido vertical. Estos objetos suelen utilizarse para almacenar y organizar elementos, aprovechando el espacio en altura.

Es importante destacar que esta clasificación se basó en la forma general y la disposición principal de los objetos, permitiendo agruparlos de manera efectiva según su morfología predominante. Esta clasificación será fundamental para la implementación del algoritmo generativo, ya que permitirá establecer patrones y reglas específicas para cada familia de objetos, optimizando así el proceso de diseño generativo.

En la descripción de cada una de estas familias de objetos, se observa que en aquellos con clasificación vertical, el cuerpo principal del objeto, es decir, el elemento dominante, se encuentra en una disposición vertical. Sin embargo, los cuerpos secundarios y los detalles esenciales y no esenciales pueden estar representados en distintas direcciones, tanto verticales como horizontales. Esto significa que estos objetos pueden tener componentes adicionales que se extiendan tanto en sentido vertical como en sentido horizontal, dependiendo de la función y el diseño específico del objeto.

Por otro lado, en las familias de objetos con clasificación horizontal, el cuerpo principal se encuentra en una posición horizontal. Al igual que en los objetos con clasificación vertical, los cuerpos secundarios y los detalles esenciales y no esenciales pueden estar representados en diferentes direcciones, tanto horizontales como verticales.

En resumen, la disposición principal del cuerpo principal del objeto varía según la familia de clasificación (vertical u horizontal). Sin embargo, los demás elementos que conforman el objeto pueden estar dispuestos en múltiples direcciones, permitiendo una mayor flexibilidad y versatilidad en el diseño generativo. Esta característica es

fundamental para lograr la variedad y diversidad en las soluciones de diseño generadas por el algoritmo.

Dentro de la clasificación general vertical, se proponen tres subcategorías: vertical extensional, vertical superficial y vertical masivo.

Dentro de la clasificación vertical, el primer grupo se refiere a la categoría "vertical extensional" (Fig. 5). En esta categoría, las características morfológicas del objeto muestran que el cuerpo principal está conectado a cuerpos secundarios por una estructura alámbrica que sirve como elemento de conexión entre los cuerpos superiores e inferiores. Esto permite una extensión vertical del objeto, creando una forma que se proyecta hacia arriba o hacia abajo desde el cuerpo principal. Esta configuración proporciona una apariencia estilizada y alargada al objeto, que puede ser utilizada en diversos contextos de diseño.



Figura 5. Vertical extensional.

Dentro de la clasificación vertical, encontramos el grupo denominado "vertical superficial" (Fig. 6). En este grupo, el elemento superior del objeto es una superficie plana de soporte. En la mayoría de los casos, se observa la repetitividad de cuerpos secundarios de forma radial y/u horizontal alrededor del cuerpo principal. Estos cuerpos secundarios pueden tener una configuración alámbrica, similar a la configuración extensional del grupo anterior, pero en este caso, no funcionan como cuerpos de conexión entre los elementos. La presencia de una superficie plana en la parte superior del objeto le confiere una apariencia más estable y equilibrada, lo que puede ser útil en productos que requieren un soporte sólido o una base firme. La repetitividad de los cuerpos secundarios puede agregar un patrón estético interesante al diseño, y la combinación de diferentes configuraciones radiales y horizontales proporciona una rica variedad de formas posibles dentro de esta categoría.



Figura 6. Vertical superficial.

Dentro de la clasificación vertical, el tercer grupo se denomina "vertical masivo" (Fig. 7). En este grupo, el cuerpo principal del objeto tiene proporciones masivas, lo que significa

que representa un volumen significativo. En la mayoría de los casos, estos cuerpos principales tienen una geometría rectangular, lo que les confiere una apariencia sólida y robusta. En este grupo, no se observa la presencia de cuerpos de conexión, lo que significa que el cuerpo principal tiene una relación directa con los cuerpos secundarios y los detalles esenciales o no esenciales del diseño. Esta falta de cuerpos de conexión permite una mayor cohesión y continuidad en la forma del objeto, lo que puede ser útil en productos que requieren una apariencia monolítica o escultórica. La combinación de proporciones masivas y geometría rectangular da lugar a diseños imponentes y poderosos que destacan por su presencia visual.



Figura 7. Vertical masivo

Dentro del grupo vertical extensional, se encuentra una única subcategoría conocida como "extensional añadido" (Fig. 8). En esta clasificación, se observa la presencia de una subestructura que se separa de la dirección principal de la morfología del objeto y puede surgir de cualquier parte del mismo. Esta subestructura añadida se conecta de manera independiente al cuerpo principal y se extiende hacia una dirección diferente, lo que genera una forma más compleja y dinámica. Esta característica agrega un nivel adicional de interés visual al diseño, ya que crea una conexión inesperada entre diferentes partes del objeto. La extensión añadida puede adoptar diversas formas y direcciones, lo que brinda al diseñador una mayor libertad para experimentar y explorar diferentes configuraciones. Además, esta subcategoría puede agregar funcionalidad adicional al objeto, ya que puede albergar elementos complementarios o cumplir con propósitos específicos dentro del diseño. En resumen, la subcategoría de "extensional añadido" dentro del grupo vertical extensional permite una mayor complejidad y versatilidad en la forma y función del objeto, haciendo que el diseño sea más interesante y atractivo para el usuario.



Figura 8. Extensional añadido.

En el grupo vertical superficial, se encuentra la subcategoría "superficial añadido" (Fig. 9). Al igual que en la categoría inmediata superior, el cuerpo principal es una superficie plana de soporte. Sin embargo, en esta subcategoría es necesaria la incorporación de una subestructura adicional para representar cuerpos secundarios y/o detalles esenciales o no esenciales. La subestructura se conecta a la superficie del cuerpo principal y proporciona soporte y estabilidad a los elementos agregados a lo largo de la superficie. Esta subcategoría ofrece una mayor flexibilidad en la creación de formas y configuraciones, permitiendo diseños más complejos y detallados que se ajustan a requisitos específicos del proyecto. Así, la subcategoría "superficial añadido" proporciona una amplia gama de posibilidades para la creación de objetos estéticamente atractivos y funcionalmente eficientes.



Figura 9. Superficial añadido.

Dentro del grupo vertical masivo, se pueden identificar dos subcategorías: "masivo añadido" y "masivos compactos" (Fig. 10). En ambas subcategorías, el concepto de "añadido" se refiere a la necesidad de incorporar subestructuras adicionales para representar los detalles esenciales del objeto. En la subcategoría de "masivo añadido", se requiere agregar subestructuras a la estructura principal para la inclusión de detalles esenciales que enriquezcan la morfología del objeto. Por otro lado, en la subcategoría de "masivos compactos", no existen cuerpos de conexión, pero es necesario añadir una subestructura a la estructura principal para representar los detalles esenciales del objeto de manera más compacta y funcional. Esta subcategoría ofrece opciones para la creación de objetos masivos con detalles cuidadosamente incorporados, sin la necesidad de cuerpos de conexión adicionales, lo que permite un diseño más eficiente y estéticamente agradable. El uso de subestructuras en estas categorías proporciona mayor flexibilidad en el diseño y la posibilidad de crear formas más complejas y detalladas en la representación de objetos verticales masivos.



a)



b)



Figura 10. a) Masivo añadido. b) Masivo compacto

Dentro de la clasificación general horizontal, se pueden distinguir dos subcategorías: "horizontal masivo" y "horizontal cóncavo" (o tipo "C") (Fig. 11). En el primer grupo, al igual que su categoría análoga en la clasificación vertical, el cuerpo principal del objeto se concentra en una sola masa central. Esto permite una estructura sólida y masiva que proporciona una apariencia robusta y fuerte al objeto. Por otro lado, en el segundo grupo, la forma general de la estructura se asemeja a la letra "C", lo que implica que dos o más secciones de la estructura se encuentran en esta dirección. En la mayoría de los casos, este grupo no cuenta con cuerpos de conexión adicionales, lo que enfatiza la continuidad de la forma en "C" y el espacio intermedio que se crea, el cual puede ser utilizado para la posible colocación de detalles esenciales o no esenciales. La clasificación general horizontal ofrece opciones diversas para la creación de objetos con morfologías distintas y adaptadas a diferentes necesidades y estilos de diseño.



Figura 11. a) Horizontal masivo. b) Horizontal cóncavo.

Dentro del grupo horizontal masivo, se clasifican dos subcategorías: el masivo añadido y el masivo compacto (Fig. 12). En el caso del masivo compacto, al igual que su categoría superior en la clasificación vertical, cuenta con las mismas características. El cuerpo principal del objeto tiene una forma masiva y sólida, y se le añade un detalle esencial, como una agarradera, en la parte superior. Los demás detalles esenciales son representados con subestructuras adicionales que complementan la morfología general del objeto.

Por otro lado, en el grupo masivo añadido, los objetos cuentan con cuerpos secundarios que están conectados por elementos de conexión, principalmente alámbricos (configuración extensional) o de superficie, dependiendo de su función. Estos elementos de conexión permiten la integración de los cuerpos secundarios con el cuerpo principal del objeto. Además, en este grupo también existen detalles esenciales que son representados con subestructuras adicionales, lo que agrega complejidad y detalle a la morfología del objeto.

En ambos grupos, la clasificación horizontal masiva ofrece una variedad de opciones para la creación de objetos con una apariencia sólida y robusta, con detalles esenciales que contribuyen a la funcionalidad y estética del producto. La combinación de la estructura principal y las subestructuras permite generar objetos con formas

innovadoras y diferentes usos, lo que amplía las posibilidades de diseño generativo en esta categoría.



Figura 12. a) Masivo añadido. b) Masivo compacto.

Dentro de la subcategoría horizontal masivo añadido, se encuentra una clasificación inferior denominada masivo complejo o tipo “Z” (Fig. 13). Estos objetos presentan una estructura con secciones en diagonal, compuesta principalmente por un cuerpo principal, varios cuerpos secundarios y elementos de conexión que los unen. Los detalles esenciales y no esenciales están presentes en varias partes del objeto, lo que añade complejidad y riqueza a su morfología.

La forma de la estructura se asemeja a una “Z”, lo que le confiere un aspecto único y distintivo. Esta configuración permite que el objeto tenga una apariencia interesante y atractiva, al mismo tiempo que cumple con su función específica. La presencia de detalles esenciales y no esenciales en diferentes partes del objeto garantiza una experiencia de diseño generativo más versátil y creativa.

El masivo complejo o tipo “Z” representa una oportunidad para explorar nuevas soluciones de diseño y expandir las posibilidades del diseño generativo en el contexto de objetos con estructuras masivas y complejas. La combinación de elementos de conexión y detalles en diferentes direcciones crea una morfología única que puede adaptarse a diversas aplicaciones y requerimientos funcionales.



Figura 13. Masivo complejo.

Finalmente, en el grupo horizontal cóncavo, se encuentra el grupo denominado cóncavo parcial (Fig. 14), en el cual básicamente, no es necesaria toda la estructura origen (de la clasificación inmediata superior) para formar el objeto. El cuerpo principal se encuentra en forma horizontal y el cuerpo secundario se distribuye de forma vertical.



Figura 14. Cóncavo parcial.

La figura siguiente (Fig. 15) muestra una recopilación de todos los grupos previamente explicados, lo que permite visualizar la jerarquía de clasificación. Además, se identifican las categorías análogas mediante líneas punteadas, que representan grupos que comparten gran parte de sus características, pero en diferentes direcciones.

Esta representación gráfica es de gran utilidad para comprender la relación entre las diferentes categorías y subcategorías, así como para identificar similitudes y diferencias entre ellas. La jerarquía permite una organización clara y ordenada de los distintos grupos, lo que facilita el proceso de diseño generativo y la exploración de diferentes opciones.

Al identificar las categorías análogas, se pueden encontrar patrones y tendencias en la morfología de los objetos, lo que puede ser útil para inspirar nuevos diseños y crear soluciones innovadoras. Además, esta visualización ayuda a los diseñadores a comprender mejor las características y restricciones de cada categoría, lo que les permite tomar decisiones más informadas durante el proceso de diseño.

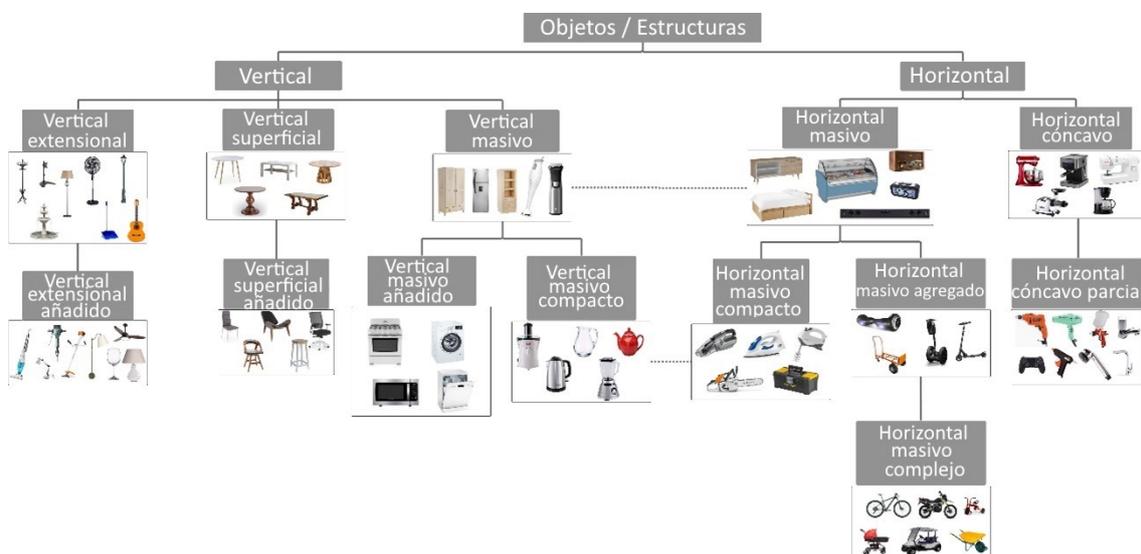


Figura 15. Diagrama de la clasificación. Fuente: Elaboración propia.

### 5.1 El algoritmo

Durante el proceso de construcción del algoritmo, se llevó a cabo la creación de objetos que compartían la misma estructura principal, lo que significa que pertenecían al mismo grupo según la clasificación previamente establecida. La finalidad de esta estrategia fue comprobar la versatilidad para generar diversos productos utilizando los mismos

parámetros estructurales. Además, se buscó verificar la capacidad de adaptación de los componentes insertados como subestructuras y/o detalles en cada uno de los objetos generados. De esta manera, se buscaba demostrar la eficacia del algoritmo en la generación de soluciones de diseño variadas y la capacidad de los elementos adicionales para enriquecer la morfología de los objetos.

Para organizar el flujo de trabajo dentro del algoritmo, se establecieron 4 secciones principales (Fig. 16), el de la estructura básica de los productos, el de creación de volúmenes, el de creación de subestructuras y finalmente, el de incrustación de elementos adicionales.

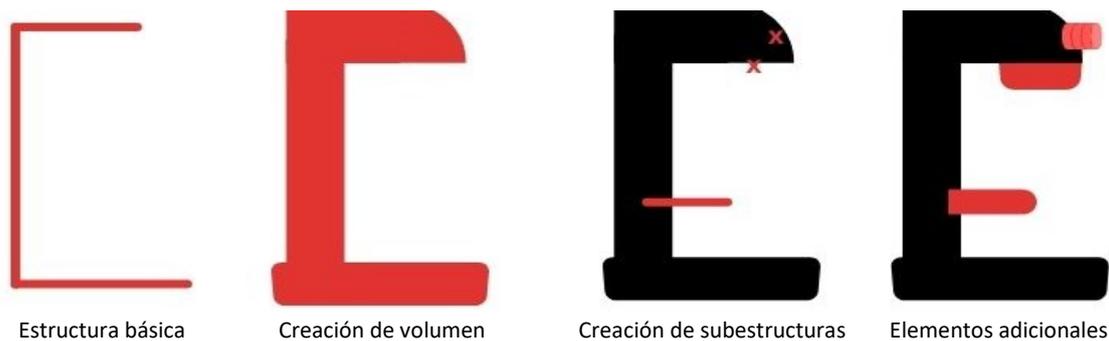


Figura 16. Flujo de trabajo. Fuente: Elaboración propia.

En la sección de la estructura básica, se parte de un conjunto de líneas dibujadas en el entorno de Rhino que representa la estructura clasificada en la clasificación propuesta. Estas líneas son importadas a Grasshopper para el funcionamiento del algoritmo. Se utilizaron las versiones de Rhinoceros 6 SR13 y Grasshopper Wednesday, 27 February 2019 00:37 Build 1.0.0007, y la versión de Python 6.13.190058.371.

El primer procesamiento que se realiza es mediante un componente modificador de curvas, con la finalidad de generar variaciones de postura en la forma. Este componente descompone la curva original para obtener los vértices de inicio y fin de cada sección de línea que la componen. Luego, se aleatorizan los parámetros de estos vértices para generar nuevos puntos de inicio y fin para la nueva sección de línea, definiendo así la nueva posición de la estructura. En esta misma sección, se obtienen los segmentos individuales que conforman la estructura para posteriormente colocar la forma volumétrica que representará cada parte del producto en cada uno de estos segmentos.

Continuando con el funcionamiento del algoritmo, la sección destinada para la creación de los volúmenes que conforman las partes básicas del producto es fundamental para dar forma a cada elemento del diseño. Como se mencionó anteriormente, estos volúmenes se colocan en cada sección individual que conforma la estructura del producto.

Para esta parte del algoritmo, se utiliza un componente que toma como entrada la sección de la estructura, una línea de referencia y parámetros numéricos que servirán para la creación del volumen. El componente utiliza estos datos para generar la forma tridimensional que representa cada parte del producto.

Es importante destacar que el componente de creación de volúmenes es altamente paramétrico, lo que permite al diseñador ajustar fácilmente las dimensiones y formas de los elementos básicos del producto. Esto proporciona una gran flexibilidad en la generación de distintas formas y diseños, lo que permite explorar una amplia variedad de posibilidades antes de llegar a la solución final.

El componente descrito para la creación de volúmenes utiliza una serie de perfiles dibujados previamente en el entorno de Rhino, ya sea mediante curvas o líneas. Estos perfiles son trasladados a Grasshopper para su procesamiento. En esta sección del algoritmo, se utilizan otros componentes preparados que permiten generar los volúmenes por medio de técnicas como extrusión, revolución, barrido, entre otras. La elección del componente específico depende de la configuración de la parte del producto que se desea representar.

Para lograr una mayor flexibilidad en la generación de formas, los perfiles de los elementos básicos también se someten a modificaciones similares a la descomposición de las líneas que conforman la estructura principal. Esto permite explorar formas variables y adaptar los volúmenes a las necesidades específicas de cada parte del diseño. La capacidad de modificar los perfiles y ajustar los parámetros numéricos en el componente de creación de volúmenes brinda un alto grado de control y creatividad en el proceso de diseño.

La combinación de perfiles, técnicas de generación de volúmenes y ajustes paramétricos proporciona una amplia gama de posibilidades para la creación de elementos básicos que formarán parte del producto final. Esta diversidad de opciones facilita la exploración de distintas alternativas de diseño y la generación de formas innovadoras y variadas.

Una vez que se han generado los volúmenes principales que conforman el cuerpo del producto, se inicia el proceso de modificación y refinamiento. Para esto, se utilizan los mismos componentes previamente descritos, pero en esta ocasión, se aplican operaciones booleanas básicas para generar formas más detalladas y complejas.

Las operaciones booleanas permiten combinar, cortar, unir y modificar los volúmenes de manera precisa. Por ejemplo, se pueden realizar uniones para fusionar elementos, cortes para crear formas específicas o intersecciones para eliminar partes innecesarias. Estas operaciones se aplican de forma iterativa para obtener el nivel de detalle deseado en el diseño.

Además de las operaciones booleanas, también se utilizan componentes adicionales para ajustar y refinar los volúmenes. Esto incluye componentes para suavizar bordes, redondear esquinas, ajustar tamaños y proporciones, entre otros. La combinación de estas operaciones y ajustes permite crear formas más complejas y detalladas que enriquecen la apariencia del producto final.

El proceso de modificación y refinamiento es altamente iterativo, lo que significa que se realizan múltiples ajustes y cambios hasta alcanzar la forma deseada. El diseño generativo facilita este proceso, ya que permite explorar rápidamente diversas opciones y evaluar cómo afectan los cambios a la apariencia y funcionalidad del producto.

Siguiendo con la definición volumétrica del producto, el siguiente paso es la incorporación de las subestructuras que permiten agregar detalles y características

específicas al diseño. Estas subestructuras pueden ser líneas o puntos, según las necesidades y requerimientos del objeto en proceso de diseño.

Para facilitar esta tarea, se han desarrollado una serie de componentes instalables en el algoritmo que permiten agregar las subestructuras de manera rápida y eficiente. Estos componentes ofrecen diversas opciones para la colocación y configuración de las subestructuras, lo que brinda al diseñador un alto grado de control y flexibilidad en el proceso creativo.

Por ejemplo, mediante estos componentes se pueden añadir líneas de soporte para mejorar la estabilidad del producto, puntos de referencia para guiar la colocación de elementos adicionales o líneas decorativas que realcen la estética del diseño. La incorporación de estas subestructuras enriquece la morfología del objeto y permite crear soluciones más detalladas y personalizadas.

Además, al ser componentes instalables, el algoritmo se vuelve altamente versátil y escalable, ya que se pueden agregar nuevos componentes según las necesidades específicas de cada proyecto. Esto facilita la adaptación del algoritmo a diferentes tipos de productos y diseños, ampliando así su potencial creativo y aplicativo.

En este punto del proceso de diseño, se tiene la opción de generar volúmenes utilizando los mismos componentes preparados para la sección de creación de volúmenes, o bien, pasar a la incorporación de detalles esenciales mediante el uso de los componentes instalables desarrollados para este propósito.

Si se elige generar volúmenes, se pueden utilizar los componentes previamente diseñados para producir formas más detalladas mediante operaciones booleanas básicas. Estos componentes ofrecen una variedad de opciones para crear volúmenes complejos y personalizados, lo que permite explorar diferentes alternativas y soluciones creativas.

Por otro lado, si se opta por incorporar detalles esenciales al diseño, se pueden utilizar los componentes instalables presentados en este trabajo. Estos componentes permiten generar elementos básicos como botones, pantallas, terminados superficiales, mandos, y otros detalles que enriquecen la funcionalidad y estética del producto.

Sin embargo, para llevar a cabo la incorporación de estos detalles esenciales, es necesario generar previamente los elementos de referencia, que son las subestructuras. Estas subestructuras actúan como guías para la colocación y orientación de los detalles esenciales, asegurando que estos se integren de manera adecuada en el diseño final.

En otras palabras, en esta etapa del algoritmo, se tiene la opción de generar volúmenes o incorporar detalles esenciales. En ambos casos, los componentes instalables desarrollados para el algoritmo brindan un alto grado de flexibilidad y control, lo que permite explorar una amplia gama de posibilidades creativas y generar diseños únicos y personalizados. Además, la generación previa de subestructuras garantiza la correcta integración de los detalles esenciales en el producto final.

## 5.2 El grupo de componentes

Una vez definida la forma general del objeto mediante la estructura principal, se procedió a incorporar elementos de subestructuras y detalles esenciales o no esenciales. Para lograr esto, se desarrollaron componentes en Python que permitieran generar una variedad de formas para estos elementos. Los componentes se dividieron en dos tipos: aquellos que generan líneas o puntos, considerados como elementos subestructurales, y aquellos que generan formas volumétricas, principalmente para representar detalles esenciales o no esenciales. Es importante resaltar que, en este último grupo de componentes, es necesario tener puntos o líneas como referencia para su correcto funcionamiento, por lo que los componentes del primer grupo son indispensables para ello.

Para ilustrar el funcionamiento de estos componentes, se presenta un ejemplo de construcción de un electrodoméstico y algunas de sus partes, utilizando ambos grupos de componentes.

### 5.2.1 Componentes subestructurales.

Como se mencionó previamente, los componentes desarrollados tienen como función principal la generación de puntos o líneas que se clasifican como subestructuras. Estas subestructuras tienen la finalidad de servir como elementos auxiliares para la construcción de partes volumétricas del objeto. En otras palabras, son puntos de referencia que permiten establecer la ubicación y forma de los detalles esenciales o no esenciales que se incorporan al diseño. Estos componentes ofrecen flexibilidad y diversidad en la generación de subestructuras, lo que permite explorar diferentes configuraciones y estilos para los detalles del objeto.

#### Structure Line.

Este componente es una herramienta que genera una línea que parte desde un punto ubicado sobre una línea de la estructura principal (Fig. 17). Su objetivo es crear una línea que pueda ser fácilmente manipulada en dirección y longitud, lo que permite crear subestructuras para agregar elementos adicionales al objeto. Esta línea actúa como una guía para la colocación de detalles esenciales o no esenciales en el diseño del producto.

Para ilustrar su funcionamiento, se presenta un ejemplo de una estructura "cóncava horizontal". En esta configuración, la línea generada por el componente se extiende desde un punto específico sobre la estructura principal, permitiendo establecer la dirección y longitud de la subestructura. Esta subestructura servirá como base para agregar elementos adicionales al objeto, como botones, mandos u otros detalles estéticos o funcionales. La flexibilidad y facilidad de manipulación de la línea generada por el componente facilita la exploración de diferentes opciones de diseño y la creación de formas variadas y originales para el producto final.

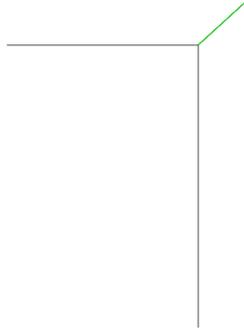


Figura 17. Creación de línea auxiliar a partir de estructura básica.

#### Auxiliary Point.

El componente "Auxiliary Point" (Fig. 18) tiene como objetivo generar un punto auxiliar sobre una superficie ya creada. Este componente utiliza un "MD Slider" que permite proyectar un punto sobre cualquier cara del objeto volumétrico creado previamente. La forma de generar el punto es similar al componente "Structure Line", pero en este caso se obtiene la intersección de una curva con la superficie donde se desea ubicar el punto.

Al utilizar el "MD Slider", se puede ajustar la posición del punto en la superficie del objeto, lo que brinda flexibilidad para colocar detalles esenciales o no esenciales en puntos específicos del diseño. La capacidad de proyectar puntos sobre diferentes superficies permite la creación de elementos precisos y ajustados al objeto volumétrico, lo que enriquece la forma y la estética del producto final. Además, al igual que con el componente "Structure Line", la facilidad de manipulación de este componente en Grasshopper facilita la exploración de diferentes opciones de diseño y la generación de formas variadas para el producto.

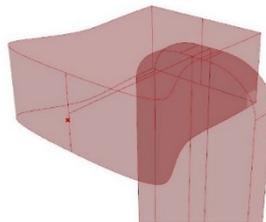


Figura 18. Inserción de punto auxiliar.

#### Array Points.

El componente "Array Points" (Fig. 19) es una herramienta versátil que permite crear una serie de formaciones de puntos en diferentes configuraciones. Estas formaciones o matrices pueden ser lineales, circulares o rectangulares, lo que brinda una gran flexibilidad en la generación de diseños.

Para utilizar este componente, se necesita especificar un punto inicial que servirá como referencia para controlar la posición del primer elemento proyectado en la geometría o superficie donde se desea realizar la formación. A partir de este punto de inicio, el

componente permite indicar el número de elementos a formar, la distancia entre ellos y, en el caso de formaciones circulares, el ángulo de estos.

La función de "Array Points" resulta especialmente útil para crear patrones repetitivos y alinear detalles esenciales o elementos decorativos en el diseño. Al ajustar los parámetros de distancia y ángulo, se puede lograr una amplia variedad de formas y disposiciones para los puntos generados, lo que permite agregar complejidad y dinamismo a la geometría del objeto. La posibilidad de crear formaciones lineales, circulares o rectangulares amplía las opciones de diseño y brinda al diseñador una mayor libertad para explorar diferentes estilos y composiciones.

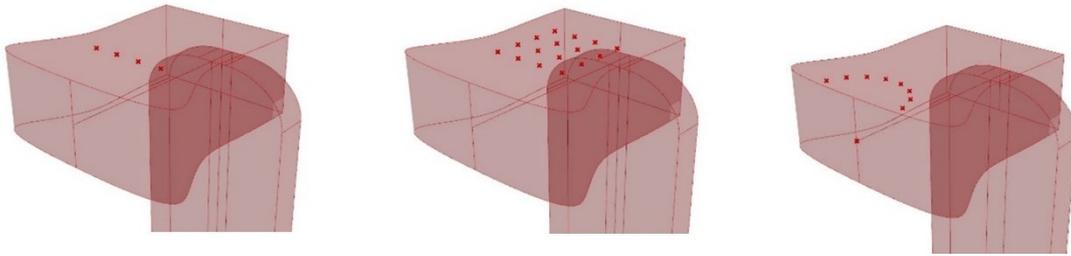


Figura 19. Array points a través del mismo componente.

#### Curves on brep.

El componente "Curves on Brep" (Fig. 20) es una herramienta que permite proyectar una o varias curvas sobre un cuerpo (brep) con el propósito de generar una serie de subestructuras con forma de anillos que sirvan para crear detalles en el objeto. Este proceso se realiza sobre una sección de la estructura principal del objeto, lo que brinda flexibilidad para elegir el rango a abarcar en la línea para la proyección de las curvas sobre las superficies.

Una vez creadas estas curvas sobre el brep, se abren diversas posibilidades para su manipulación. Pueden utilizarse para efectuar barridos, generando así elementos con volumen que sigan la trayectoria de la curva proyectada. También se pueden aplicar componentes como "pipe" para dar grosor a las curvas y crear elementos cilíndricos o tubulares. Esta versatilidad en la manipulación de las curvas proyectadas permite agregar detalles y elementos decorativos al objeto de forma sencilla y precisa.

El componente "Curves on Brep" es una herramienta útil para la creación de detalles esenciales o no esenciales en el diseño del objeto, ya que permite generar formas complejas y detalladas de manera eficiente. Al combinarlo con otros componentes del algoritmo, como los mencionados anteriormente, se pueden lograr diseños más elaborados y personalizados. La capacidad de generar subestructuras con forma de anillos a partir de curvas proyectadas ofrece una amplia gama de posibilidades creativas para el diseñador, permitiéndole explorar distintas opciones de diseño y agregar un toque único y distintivo al producto final.

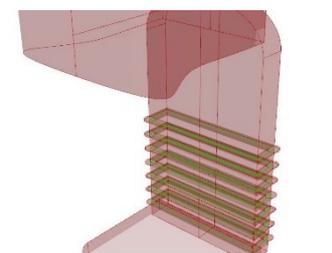


Figura 20. Curvas proyectadas sobre volúmenes.

### 5.2.2 Componentes de generación de formas.

Los componentes de generación de formas desempeñan un papel fundamental en el proceso de diseño generativo, ya que su principal objetivo es crear formas que representen tanto los detalles esenciales como los no esenciales del objeto en desarrollo. Además, estos componentes permiten generar elementos que son representativos de la morfología general del producto final.

Estos componentes ofrecen al diseñador una amplia gama de posibilidades para agregar detalles y elementos decorativos al diseño. Al utilizarlos, el diseñador puede explorar diferentes configuraciones y variaciones de formas, lo que facilita la creación de diseños innovadores y únicos.

Los detalles esenciales son aquellos elementos que son imprescindibles para el funcionamiento y uso del objeto. Por ejemplo, en el diseño de un vehículo, los detalles esenciales podrían incluir el volante, los pedales y los asientos. Por otro lado, los detalles no esenciales son elementos adicionales que agregan valor estético o funcional al objeto, pero que no son fundamentales para su funcionamiento. Estos pueden incluir adornos, patrones decorativos o características estilísticas específicas.

Además de los detalles, los componentes de generación de formas también permiten crear elementos que son representativos de la morfología general del objeto. Estos elementos pueden ser características estructurales o estilísticas que definen la apariencia y forma del objeto en su conjunto.

En definitiva, los componentes de generación de formas son herramientas esenciales para el diseñador generativo, ya que le brindan la capacidad de explorar y crear una amplia variedad de formas y elementos que enriquecen el diseño final del objeto. Estos componentes son fundamentales para lograr diseños únicos, funcionales y estéticamente atractivos que se adapten a las necesidades y requerimientos específicos del proyecto.

#### Reorient Object.

El componente Reorient Object (Fig. 21) desempeña un papel fundamental en el proceso de diseño, ya que su función principal es permitir la reorientación de cualquier geometría sobre la superficie de un objeto sin importar la cara en la que se desee representar la forma. Este componente brinda al diseñador una gran flexibilidad y libertad para colocar y posicionar elementos geométricos de manera precisa y controlada.

La utilización de este componente es muy sencilla. El usuario puede crear la forma geométrica que desee en cualquier parte del espacio de Rhino, sin restricciones de tamaño o forma. Luego, al utilizar el componente Reorient Object, la geometría creada se moverá y ajustará automáticamente para adaptarse a la superficie de la geometría base.

Esto permite al diseñador explorar diferentes configuraciones y posiciones para el objeto geométrico, lo que facilita la creación de diseños complejos y detallados. Además, el componente garantiza que la forma insertada se encuentre siempre

perpendicular a las superficies de la geometría base, lo que asegura una representación precisa y realista.

El componente Reorient Object es especialmente útil para agregar detalles esenciales o no esenciales a la estructura principal del objeto. Por ejemplo, se puede utilizar para colocar botones, pantallas, mandos u otros elementos decorativos en cualquier parte del objeto. También es útil para la incorporación de elementos representativos de la morfología general del diseño.

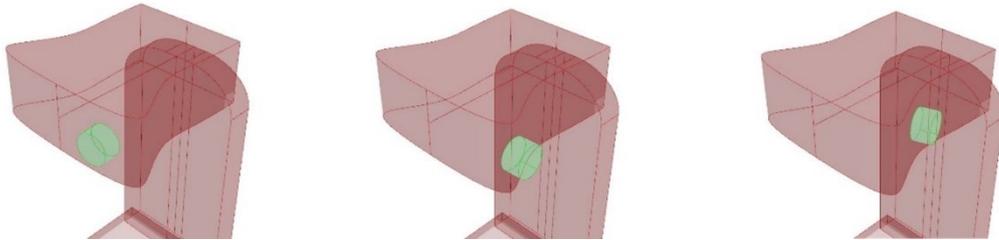


Figura 21. Reorientación de geometrías conforme a las superficies del objeto.

#### Volume on brep.

Su función principal es generar un volumen a partir de un perfil creado previamente. El diseñador puede proporcionar un perfil geométrico previamente creado en cualquier parte del espacio de Rhino, y el componente se encarga de generar una extrusión de este perfil para crear un objeto sólido. Luego, se tapa la extrusión para cerrar el volumen y obtener un objeto completo.

Una de las ventajas de este componente es que permite al diseñador modificar las dimensiones del objeto generando una escala. Esto brinda una gran flexibilidad para ajustar el tamaño del volumen generado según las necesidades del diseño.

Al igual que el componente Reorient Object, el Volume on brep (Fig. 22) también reorienta la geometría para que sea perpendicular a la superficie del objeto base. Esto asegura una representación precisa y realista del volumen generado, y facilita su integración con la estructura principal del objeto.

El control de la posición del volumen generado se realiza mediante un MD\_Slider, lo que permite al diseñador ajustar la ubicación y orientación del objeto en relación con el objeto base.

A diferencia del componente Reorient Object, el Volume on brep no necesita ninguna geometría creada previamente, ya que genera internamente una serie de geometrías. Esto simplifica el proceso de diseño y agiliza la creación de formas volumétricas.

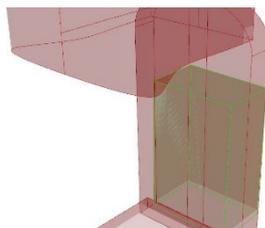


Figura 22. Colocación de una geometría volumétrica sobre un objeto.

### Buttons.

Este componente permite incorporar botones en el cuerpo del objeto de una manera fácil y personalizada. Este componente se utiliza mediante la selección de un punto (subestructura) y un perfil previamente dibujado que define la forma del botón. Esta característica proporciona al diseñador una gran libertad para diseñar botones con formas y estilos únicos que se ajusten a los requerimientos específicos del proyecto.

Una vez seleccionados el punto y el perfil, el componente Buttons (Fig. 23) permite al usuario ajustar la profundidad y el tamaño del botón mediante un factor de escala. Esta funcionalidad es especialmente útil para adaptar el botón al diseño general del objeto y garantizar una apariencia coherente y armoniosa en el producto final.

La capacidad de personalización que ofrece el componente Buttons es clave para el diseño generativo, ya que permite a los diseñadores explorar diferentes formas y estilos de botones de manera rápida y eficiente. Esto facilita la generación de ideas innovadoras y la creación de diseños únicos y atractivos.

Además, el componente Buttons se integra perfectamente con otras subestructuras generadas previamente, lo que garantiza una representación precisa y coherente de los botones en el objeto final. Esta integración fluida entre diferentes elementos del diseño es fundamental para lograr un diseño generativo exitoso y bien estructurado.

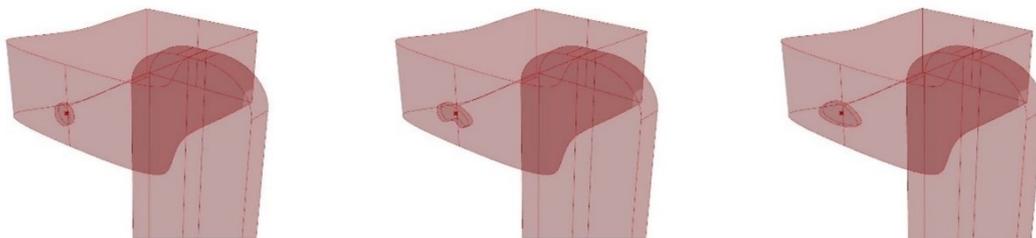


Figura 23. Creación de botonees mediante un perfil dibujado por el usuario.

### Random button generator.

El componente "Random Button Generator" es una herramienta altamente versátil y poderosa que ha sido diseñada específicamente para proporcionar una amplia diversidad de botones diferentes que pueden ser utilizados como detalles esenciales en la morfología del objeto.

Este componente es capaz de generar una variedad de formas de botones, ofreciendo al usuario 5 perfiles de botones establecidos, cada uno de los cuales ha sido cuidadosamente diseñado para ofrecer una apariencia única y atractiva. Además, estos perfiles se pueden modificar de manera aleatoria, lo que agrega un nivel adicional de variabilidad y creatividad en el diseño. El diseñador tiene un control completo sobre el nivel de variabilidad mediante un factor numérico, lo que permite ajustar la cantidad de cambios aleatorios aplicados a los perfiles de botones.

La capacidad de generar formas de botones únicas y variadas es fundamental para el diseño generativo, ya que permite explorar una amplia gama de opciones de diseño y generar soluciones originales y creativas. Con el componente "Random Button Generator" (Fig. 24), el diseñador tiene la libertad de experimentar con diferentes

formas, estilos y combinaciones de botones, lo que facilita la creación de diseños innovadores y atractivos.

La integración de este componente con otros elementos del diseño, como las subestructuras y los detalles adicionales, garantiza una representación coherente y precisa en el objeto final. La versatilidad del componente también lo hace adecuado para una variedad de proyectos y sectores, desde el diseño de productos industriales hasta el diseño de objetos de consumo.

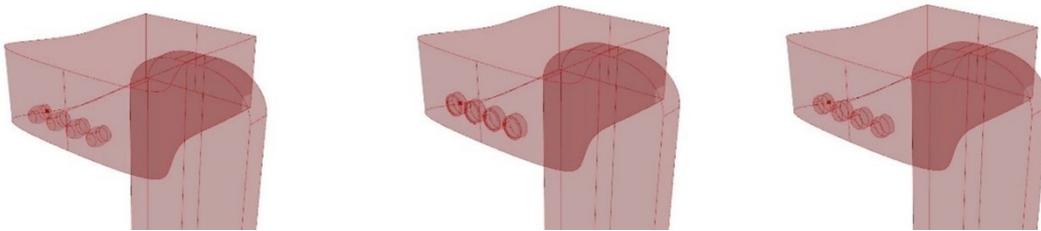


Figura 24. Generación de botones

#### Random pommel generator.

El componente "Random Pommel Generator" (Fig. 25) es una herramienta que complementa al "Random Button Generator" y tiene la finalidad de crear componentes más parecidos a pomos o perillas que se ajusten perfectamente al objeto bajo diseño.

Al igual que su predecesor, este componente ofrece 5 perfiles establecidos de pomos, cada uno de los cuales ha sido diseñado cuidadosamente para ofrecer una variedad de estilos y formas. Estos perfiles se pueden modificar de manera aleatoria, lo que permite al diseñador obtener una diversidad de pomos únicos y personalizados. El control sobre el nivel de variabilidad se realiza mediante un factor numérico, lo que brinda la flexibilidad necesaria para ajustar el grado de cambios aleatorios aplicados a los perfiles de los pomos.

La funcionalidad del "Random Pommel Generator" es esencial para el diseño generativo, ya que agrega detalles esenciales como pomos o perillas que aportan funcionalidad y estética al objeto final. Los pomos son elementos importantes en el diseño de productos y objetos, ya que permiten una interacción cómoda y eficiente con el usuario.

La incorporación de estos pomos generados aleatoriamente aporta un toque de creatividad y originalidad al objeto, lo que puede marcar la diferencia en términos de diseño y estética. Los pomos creados con este componente se adaptan perfectamente a la morfología del objeto, lo que garantiza una apariencia armoniosa y coherente.

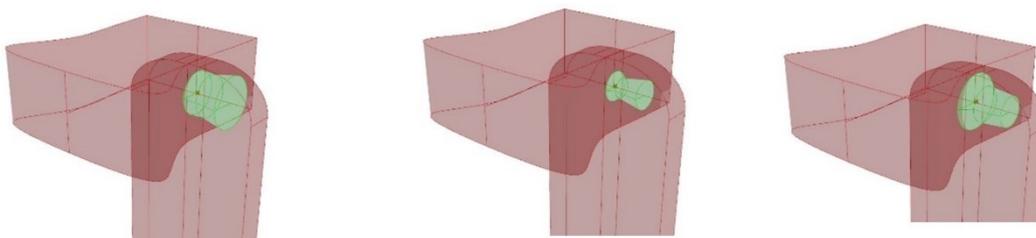


Figura 25. Pomos generados.

### Screen or void.

El componente "Screen or Void" (Fig. 26) es una herramienta versátil que ofrece una doble funcionalidad en el diseño. Su principal función es la creación de cubiertas o pantallas que pueden ser utilizadas para recubrir diferentes partes del objeto. Estas cubiertas pueden tener distintas formas y dimensiones, y permiten agregar detalles estéticos y funcionales al diseño.

Además de su función como cubierta, este componente también puede ser utilizado para crear rebajes o vaciados en el cuerpo del objeto. Esto se logra mediante el uso de un componente botón como input, que permite cambiar la dirección de la colocación de la geometría. Al insertar una diferencia de sólidos, es posible realizar la resta del material y obtener un rebaje en la forma deseada.

El componente "Screen or Void" ofrece al usuario diversas opciones de personalización. Se pueden modificar las dimensiones de largo y ancho de la cubierta o del rebaje, lo que permite adaptarla a las necesidades específicas del diseño. También es posible agregar redondeos en las esquinas para darle un acabado más suave y estilizado. Además, se puede ajustar la profundidad de la cubierta o el rebaje para obtener el efecto deseado.

La versatilidad y facilidad de uso del componente "Screen or Void" lo convierten en una herramienta de gran utilidad para el diseño conceptual. Permite al diseñador agregar detalles funcionales y estéticos de manera rápida y sencilla, lo que enriquece la morfología del objeto.

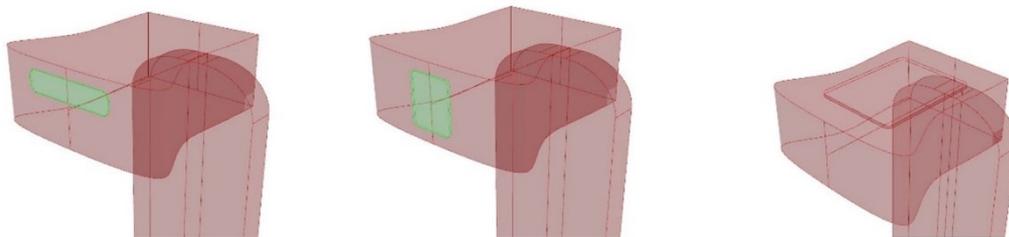


Figura 26. Pantallas o vaciados.

### Edges in contours.

El componente "Edges in contours" (Fig. 27) es una herramienta muy útil que brinda al diseñador la posibilidad de crear bordes en los contornos de algún cuerpo geométrico del objeto. Esto se logra de manera sencilla, identificando el límite superficial del objeto base y creando un sólido alrededor de este límite para representar los marcos o bordes deseados.

Este componente permite realzar y destacar ciertas áreas del objeto, dándole un aspecto más definido y estilizado. Los bordes creados pueden tener diferentes formas y tamaños, lo que permite al diseñador experimentar con distintos estilos y estéticas.

Además, el componente "Edges in contours" ofrece la posibilidad de ajustar la profundidad de los bordes, lo que permite controlar la intensidad del efecto deseado. También se pueden modificar otros parámetros, como el radio de los bordes o la distancia entre ellos, para obtener el resultado exacto que se busca.

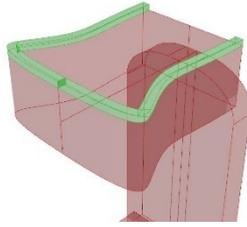


Figura 27. Creación de bordes en contornos.

### Surface pattern.

El componente "Surface pattern" (Fig. 28) es una herramienta que cumple una función muy específica en el diseño generativo. Su objetivo principal es crear un patrón homogéneo en una superficie seleccionada del objeto. Este patrón puede consistir en cuadrados, cuadrados con esquinas redondeadas o círculos, lo que le brinda al diseñador la libertad de elegir el estilo que mejor se adapte al diseño.

Una de las ventajas de este componente es que permite al usuario modificar el tamaño de las geometrías del patrón, así como su profundidad en el cuerpo del objeto. Esto brinda la posibilidad de personalizar y ajustar el patrón según las necesidades y preferencias del diseñador.

Es importante mencionar que, de acuerdo con la clasificación de detalles en este estudio, el componente "Surface pattern" se clasificaría como un detalle no esencial y únicamente estético. Esto significa que su función principal es mejorar la estética y apariencia del objeto, sin tener un impacto directo en su funcionalidad.

Sin embargo, es relevante mencionar que este componente puede requerir una cantidad considerable de recursos del ordenador debido a la complejidad del patrón generado. Por esta razón, se ha dejado su optimización para trabajos futuros, buscando mejorar su rendimiento y eficiencia en el uso de recursos.



Figura 28. Patrón sobre superficies

## 6. APLICACIÓN PRÁCTICA Y RESULTADOS.

El caso de estudio que presentamos a continuación es un ejemplo concreto de cómo se aplica la metodología desarrollada en este proyecto de diseño. En este caso, se ha utilizado el workflow descrito (Fig. 29) y los componentes creados para generar dos objetos diferentes: una cafetera y una cinta de gimnasio. Estos dos objetos comparten la misma estructura principal, lo que ilustra la versatilidad y flexibilidad de la metodología en la creación de diseños variados y personalizados.

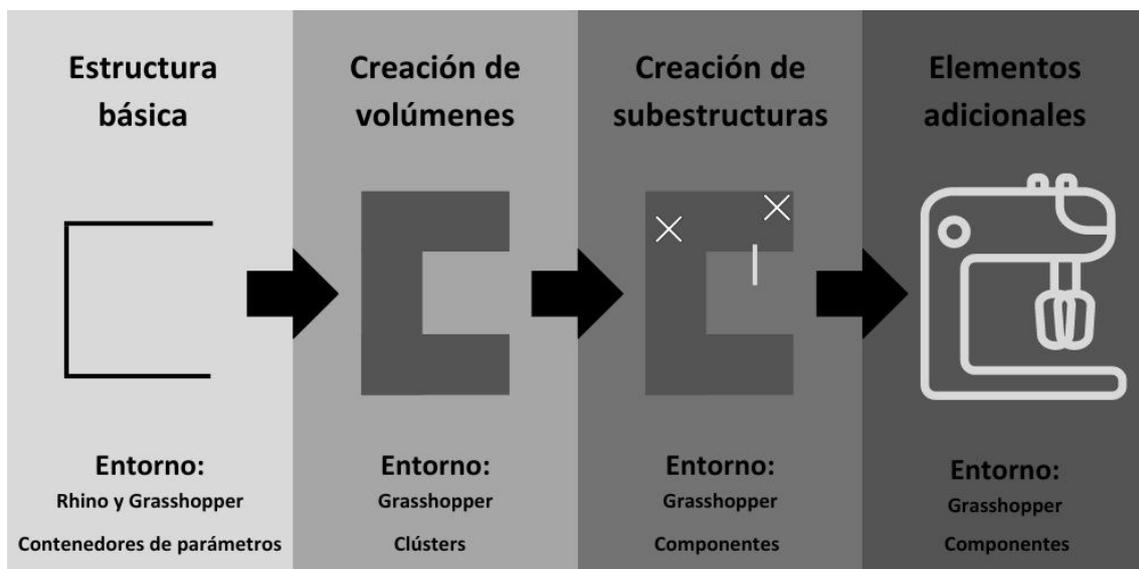


Figura 29. Flujo de trabajo en el algoritmo. Fuente: Elaboración propia.

Es importante destacar que tanto la cafetera como la cinta de gimnasio comparten una característica común en su diseño: ambas se basan en la estructura clasificada como "cóncavo horizontal". Esta estructura básica se caracteriza por una disposición principal de la forma del objeto en la que su morfología general presenta una concavidad en forma de "C" en su orientación horizontal.

Para comenzar, se parte de la estructura principal del objeto, que en este caso es el contorno general tanto de la cafetera como de la cinta de gimnasio. Esta estructura se crea inicialmente en el entorno de Rhino (Fig. 30) y se traslada a Grasshopper para su

procesamiento. La capacidad de Grasshopper para adaptar la forma y posición de la estructura principal permite explorar diferentes configuraciones de diseño para ambos objetos.

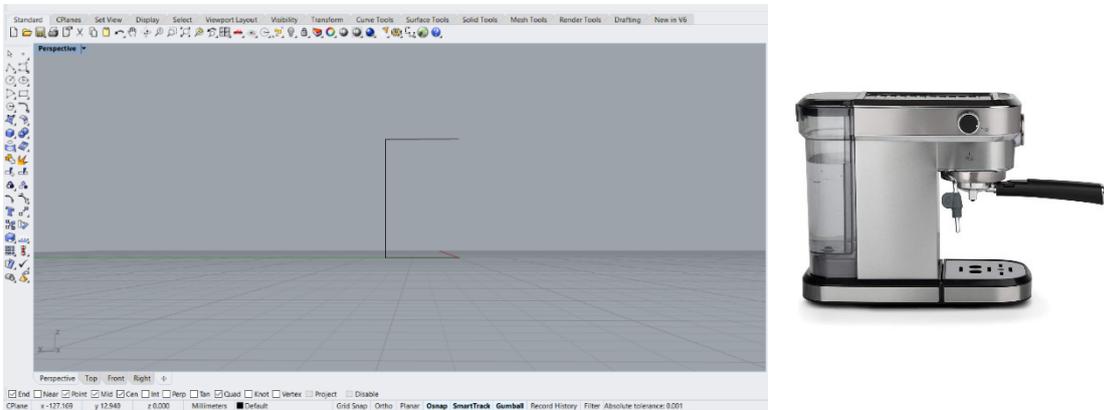


Figura 30. Estructura básica de un modelo de cafetera.

Es relevante destacar que, aunque ambos productos comparten la misma estructura principal, las proporciones de los cuerpos principales y/o secundarios pueden variar entre ellos. Para lograr esta adaptación, es necesario ajustar la longitud de las líneas que conforman la estructura principal. Este proceso se realiza de manera sencilla en el entorno de Rhino (Fig. 31) mediante la manipulación de los puntos de inicio de cada una de las líneas de la estructura.

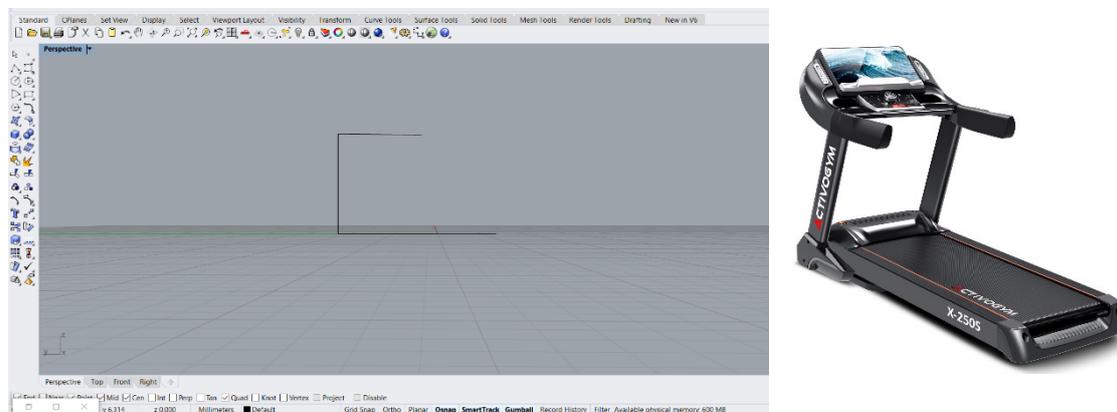


Figura 31. Estructura básica de una cinta de correr.

Una vez que hemos establecido la estructura principal del objeto, avanzamos hacia el entorno de Grasshopper para dar continuidad al diseño de los elementos que compondrán el producto. En esta fase, manipulamos la estructura básica y creamos subestructuras que desempeñan un papel crucial en la generación y posicionamiento de los volúmenes. Para llevar a cabo esta tarea de manera eficiente, hacemos uso de clústeres diseñados específicamente con este propósito de creación de volúmenes.

Antes de adentrarnos en la siguiente etapa, es esencial comprender cómo se generan los volúmenes de los cuerpos principales y secundarios. Estos volúmenes se crean mediante clústeres (Fig. 32) que requieren información de entrada específica. Dicha

información comprende la estructura principal, el selector del número de línea estructural, la curva a transformar, la curva de referencia, la curva dimensional y los valores numéricos para las dimensiones X e Y. A continuación, se detalla cada una de estas entradas.



Figura 32. Clúster para generar volúmenes.

La estructura básica es un conjunto de líneas originalmente diseñadas en el entorno de Rhino y posteriormente importadas a Grasshopper mediante un contenedor de parámetros. Esta estructura desempeña un papel fundamental, ya que sirve como la base sobre la cual se llevarán a cabo las transformaciones y generaciones de volúmenes. Está compuesta por una serie de líneas que definen la disposición general del objeto, así como la ubicación de sus cuerpos principales y secundarios.

El selector de número de línea estructural determina cuál de las secciones independientes de la estructura básica principal se utilizará como ubicación para la generación del volumen correspondiente. Este selector permite flexibilidad al momento de adaptar la forma y disposición de los volúmenes en diferentes partes del objeto.

La curva a transformar, por su parte, representa el perfil base de referencia que guiará la creación del volumen. Estas curvas pueden ser de diversos tipos, como líneas simples o perfiles cerrados, y son indispensables para realizar operaciones tridimensionales como barridos, extrusiones, revoluciones o utilizar la herramienta Loft. En esencia, estas curvas establecen la forma que se desea lograr en la generación del volumen.

Además, se utilizan dos curvas adicionales: la curva de referencia y la curva dimensional. La curva de referencia tiene un propósito específico dentro del clúster de manipulación de curvas, mientras que la curva dimensional cumple su función de acuerdo con su nombre, proporcionando información sobre las dimensiones que se desean lograr en la generación del volumen.

Por último, los valores numéricos representan los controles que permiten ajustar las dimensiones y características específicas de los volúmenes generados. Estos valores numéricos proporcionan una manera precisa de personalizar y adaptar los volúmenes a las necesidades específicas del diseño. En conjunto, estas entradas y controles forman parte integral del proceso de generación y transformación de volúmenes en la metodología de diseño utilizada.

En cuanto a la sección de salida de este clúster, se generan varios elementos que desempeñan un papel fundamental en los clústeres de operaciones tridimensionales previamente mencionados, como los de barridos, extrusiones, revoluciones y Loft (Fig. 33). Estos elementos de salida incluyen lo siguiente: "Dest curve," que proporciona una línea de destino; "segmento," que representa un segmento; "plano," que se refiere a un

plano; "loft X" y "loft Y," que corresponden a los parámetros utilizados en la operación de Loft; y "ext/rev X" y "ext/rev Y," que son utilizados en las operaciones de extrusión y revolución, respectivamente.

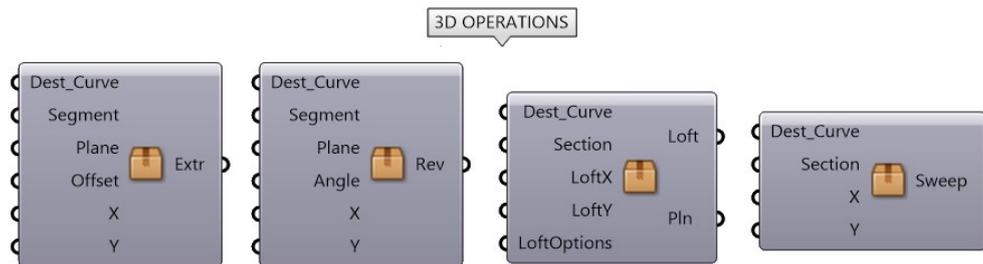


Figura 33. Clúster de generadores tridimensionales.

#### Clúster de Extrusión:

El clúster de extrusión se utiliza para generar volúmenes tridimensionales a partir de las líneas de la estructura básica y los perfiles de referencia. Este componente permite extender o extruir los perfiles a lo largo de las líneas de la estructura, creando cuerpos sólidos con secciones transversales determinadas por los perfiles de referencia. El usuario puede controlar la longitud y dirección de la extrusión, lo que permite crear diferentes volúmenes y formas a lo largo de la estructura principal.

#### Clúster de Revolución:

El clúster de revolución se utiliza para crear volúmenes mediante la rotación de perfiles de referencia alrededor de un eje. Este componente permite generar objetos sólidos con simetría rotacional, como columnas, postes o elementos cilíndricos. El usuario puede definir el ángulo de rotación y el perfil de referencia, lo que le brinda control sobre la forma y dimensiones de los volúmenes generados.

#### Clúster de Loft:

El clúster de loft se utiliza para crear volúmenes complejos interpolando entre perfiles de referencia. Este componente permite generar objetos sólidos con transiciones suaves entre diferentes secciones transversales. El usuario puede definir múltiples perfiles de referencia y controlar la forma de interpolación entre ellos, lo que facilita la creación de formas orgánicas y fluidas.

#### Clúster de Barrido:

El clúster de barrido se utiliza para crear volúmenes al extruir perfiles de referencia a lo largo de una trayectoria definida por una línea de la estructura principal. Este componente permite generar objetos sólidos con formas que siguen la dirección de la trayectoria, lo que es útil para crear elementos como mangos, pasamanos o detalles curvos en el objeto. El usuario puede controlar la distancia de extrusión y la forma de los perfiles, lo que brinda flexibilidad en el diseño de los volúmenes.

Para el caso de estudio de la cafetera se han determinado tres cuerpos principales dentro de su estructura básica que constituyen la morfología primordial. El primero de ellos es la base de la cafetera (Fig. 34), que se sitúa en la parte inferior y sirve como soporte para todo el objeto. Esta base proporciona estabilidad y eleva el cuerpo de la cafetera del plano de trabajo.

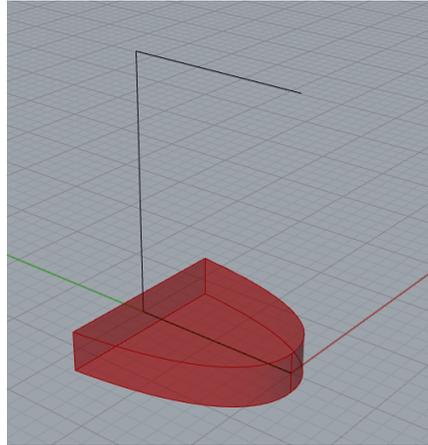


Figura 34. Base de cafetera.

El segundo cuerpo corresponde a la parte superior de la cafetera (Fig. 35), donde se encuentran ubicados los controles y otros elementos de manipulación. En esta sección, se incorporarán detalles esenciales como botones, perillas y pantallas para permitir al usuario interactuar con la cafetera de manera cómoda y funcional.

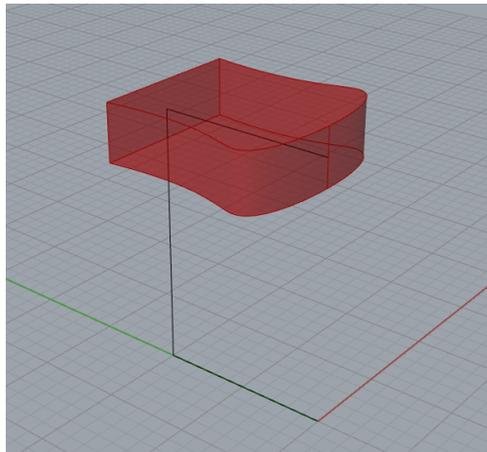


Figura 35. Cuerpo superior de la cafetera.

Finalmente, el tercer cuerpo es el que conecta tanto el cuerpo superior como el inferior de la cafetera y da cabida al área en donde se depositará el café (Fig. 36). Este cuerpo actúa como un elemento de unión entre las dos partes principales de la cafetera y define la funcionalidad principal del objeto, que es preparar y servir café.

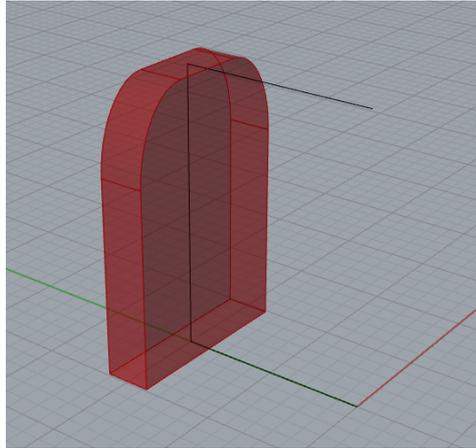


Figura 36. Cuerpo de conexión.

Cabe señalar que, para la creación de los tres cuerpos principales de la cafetera, se ha utilizado exclusivamente el clúster de extrusión dentro del proceso de diseño. Esto se debe a que el clúster de extrusión proporciona la forma adecuada que se busca obtener para estos cuerpos, permitiendo una generación eficiente y precisa de los volúmenes necesarios.

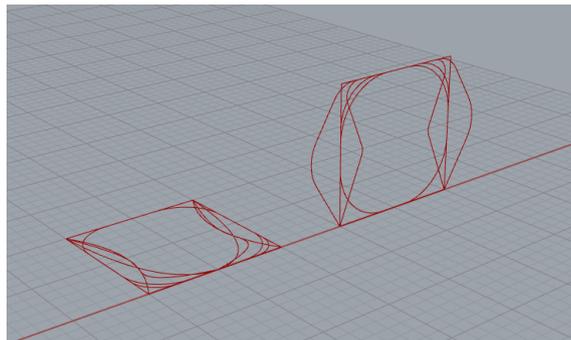


Figura 37. Perfiles para la creación del volumen.

Para el diseño de cada uno de los cuerpos de la cafetera, se han empleado perfiles específicos que capturan la forma deseada de cada elemento (Fig. 37). Estos perfiles fueron inicialmente creados en el entorno de Rhino y posteriormente importados a Grasshopper para su posterior manipulación y modificación aleatoria. Una vez en Grasshopper, se utilizó el clúster de extrusión para transformar estos perfiles en volúmenes tridimensionales que representan los cuerpos principales de la cafetera.

La extrusión se ha ajustado de acuerdo con las dimensiones y proporciones requeridas para cada cuerpo, permitiendo una flexibilidad en el diseño y una adaptación precisa a la estructura principal. Gracias a la versatilidad de este clúster, se ha logrado obtener las formas necesarias para la base, la parte superior y el cuerpo central de la cafetera de manera eficiente y coherente con el diseño conceptual.

La selección de los distintos perfiles creados en Rhino para representar los cuerpos de la cafetera se puede realizar de dos maneras. En primer lugar, se puede utilizar un selector numérico que permite saltar de un perfil a otro de manera controlada y

secuencial. Alternativamente, se puede aplicar un selector aleatorio (Fig. 38) para introducir una mayor variabilidad en las formas generadas.

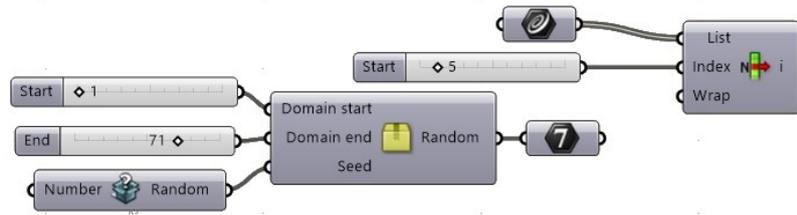


Figura 38. Selector de perfiles.

Además, estos perfiles pueden ser modificados mediante un clúster de modificación de curvas. Este clúster actúa ajustando los puntos de las curvas en los distintos ejes, lo que da lugar a formas que no siempre son idénticas entre sí. La cantidad de aleatoriedad en estas modificaciones se puede ajustar a elección del usuario, lo que permite controlar el nivel de variabilidad en las formas generadas.

Una vez que los volúmenes de los cuerpos principales del objeto han sido generados, se procede a crear las subestructuras necesarias que servirán de base para la incorporación de elementos adicionales. Estos elementos adicionales pueden ser tanto detalles esenciales como detalles no esenciales en el diseño del objeto. En el caso específico de la cafetera, se han aplicado subestructuras en forma de puntos de referencia que serán fundamentales para la generación de botones, pantallas y otras partes características de la cafetera.

Estos puntos de referencia (Fig. 39) pueden tomar diversas formas y configuraciones. Pueden ser puntos individuales o estar dispuestos en alguna formación específica, según las necesidades de diseño del objeto. Estos puntos servirán como base para la incorporación de los detalles esenciales y no esenciales que darán forma y funcionalidad a la cafetera, permitiendo su manipulación y uso esperado.

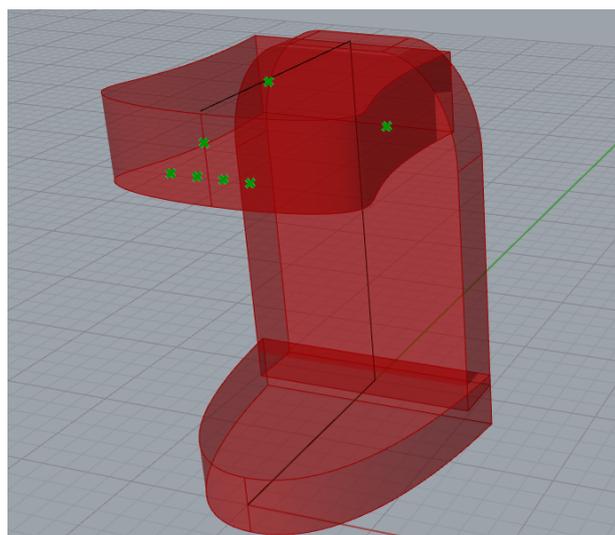


Figura 39. Puntos de referencia (subestructuras).

La ubicación y el control de estas subestructuras se llevaron a cabo mediante la implementación de los componentes creados específicamente para este propósito. Dos de los componentes clave utilizados fueron el "Auxiliary Point" y el "Array Points" (Fig. 40).

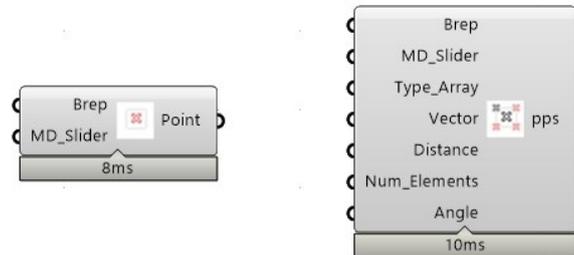


Figura 40. Componentes de puntos auxiliares.

El componente "Auxiliary Point" permitió proyectar puntos auxiliares sobre las superficies ya creadas, utilizando un MD Slider para ajustar la posición de estos puntos. Esto resultó fundamental para la colocación precisa de las subestructuras en la cafetera, asegurando que estuvieran ubicadas en las áreas necesarias para la generación de los elementos adicionales.

Por otro lado, el componente "Array Points" facilitó la creación de formaciones o matrices de puntos en diversas configuraciones, como lineales, circulares o rectangulares. Esto permitió una distribución uniforme de las subestructuras en el objeto y brindó flexibilidad en la disposición de los elementos adicionales, como los botones frontales de la cafetera. Los parámetros ajustables, como el número de elementos, la distancia entre ellos y el ángulo de disposición, permitieron adaptar las subestructuras a las necesidades de diseño de cada detalle.

En la siguiente etapa del proceso de diseño, se procedió a la creación de los elementos adicionales, los cuales se clasificaron en dos categorías: detalles esenciales y detalles no esenciales. Esta distinción se basó en la función y relevancia de cada elemento en la cafetera diseñada.

Los detalles esenciales comprenden aquellos elementos que son fundamentales para la usabilidad y funcionamiento adecuado de la cafetera. Estos incluyen elementos de manipulación esenciales, como botones y perillas para controlar las funciones de la cafetera (Fig. 41). Además, se consideraron partes específicas de una cafetera, como el depósito de líquidos, ya que son componentes necesarios para su operación principal.

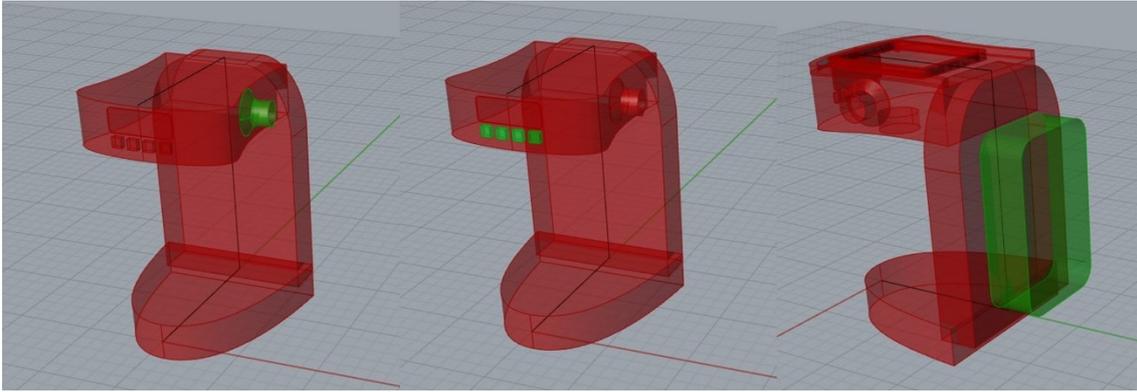


Figura 41. Detalles esenciales.

Por otro lado, los detalles no esenciales (Fig. 42) engloban elementos que, aunque pueden mejorar la estética o la experiencia visual del producto, no son esenciales para su funcionamiento principal. Estos detalles pueden incluir pantallas decorativas, botones adicionales que no son necesarios para la operación principal, bordes en contornos u otros elementos que no tienen un impacto crítico en la función principal de la cafetera.

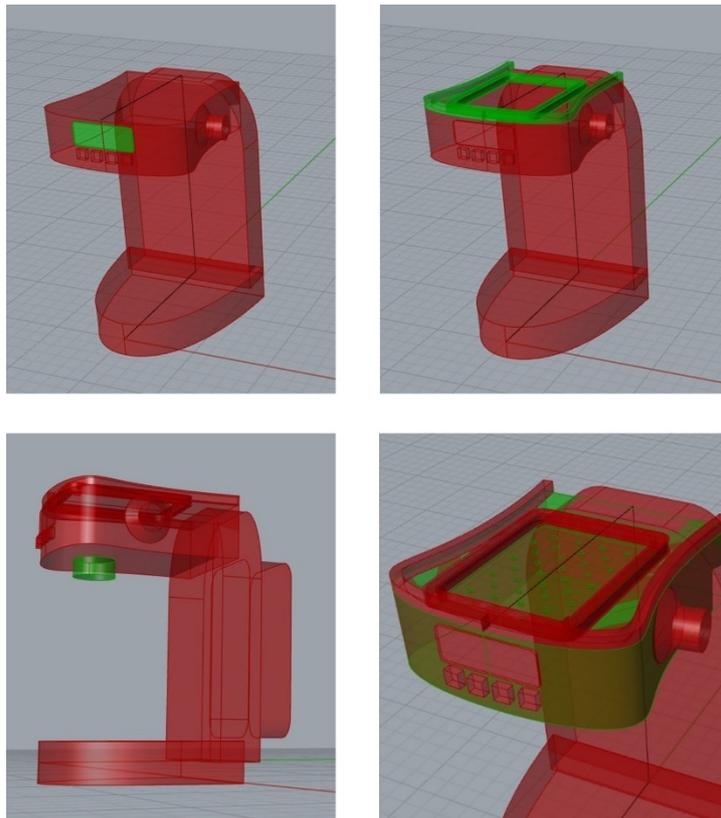


Figura 42. Ejemplos de detalles no esenciales.

La distinción entre detalles esenciales y no esenciales permitió centrar los esfuerzos de diseño en los elementos críticos para la funcionalidad de la cafetera, mientras que se mantenía la flexibilidad para agregar elementos decorativos o estéticos según fuera necesario. Este enfoque garantizó que la cafetera cumpliera con sus funciones principales de manera efectiva, al tiempo que permitía la inclusión de detalles adicionales para mejorar su apariencia visual.

Cabe señalar que, al igual que con las subestructuras, la creación de los elementos adicionales se llevó a cabo utilizando los componentes específicamente diseñados para esta tarea. Se hicieron uso de los componentes de generación de botones, perillas, pantallas y otros detalles que fueron desarrollados como parte de la metodología de diseño.

Los componentes de generación de botones (Fig. 43) fueron fundamentales para la creación de una variedad de botones con formas diversas que cumplen con la función de elementos de manipulación esenciales en la cafetera. Estos botones se generaron de manera automática a través del componente que permite crear nuevas formas en cada iteración, lo que resultó en una variedad de diseños de controles para la cafetera. Además, se contó con la opción de utilizar dos componentes diferentes para la generación de botones: uno que genera botones de manera totalmente automática y otro que permite generar botones a partir de un perfil específico, brindando al usuario la flexibilidad de integrar formas personalizadas si así lo deseaba.

Esta capacidad de generar botones de forma automática o personalizada permitió adaptar los controles de la cafetera a los requisitos de diseño y funcionalidad específicos, lo que resultó en una mayor versatilidad y creatividad en la creación de los elementos esenciales de la cafetera.

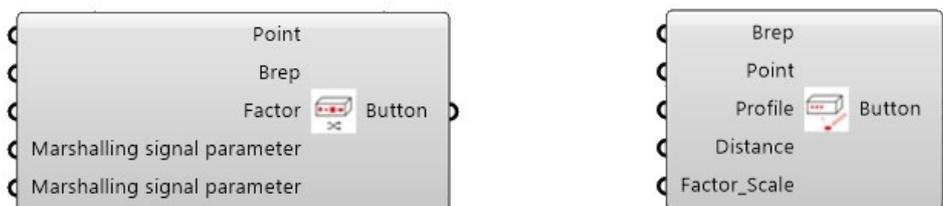


Figura 43. Componentes generadores de botones.

Del mismo modo, los componentes diseñados para la generación de perillas (Fig. 44) permitieron crear perillas con formas adecuadas para su función específica en la cafetera. Estas perillas se generaron de manera similar a los botones automáticos.

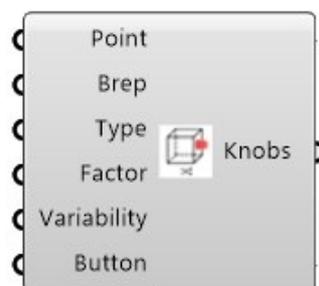


Figura 44. Componente para generar las perillas.

Además de los componentes de generación de botones y perillas, se emplearon componentes específicos para la creación de pantallas y otros detalles decorativos o no esenciales en la cafetera. Estos componentes ofrecieron la posibilidad de incorporar

elementos visuales que mejoraron la apariencia estética del producto sin comprometer su funcionalidad principal.

En particular, se utilizaron dos componentes clave: "Screen or Void" y "Edges in Contours" (Fig. 45). El componente "Screen or Void" permitió la creación de pantallas que podían ser colocadas en diversas partes del volumen seleccionado, lo que agregó un elemento de diseño visualmente atractivo a la cafetera. Además, este componente también permitió la creación de vaciados en el volumen, lo que brindó opciones adicionales de diseño.

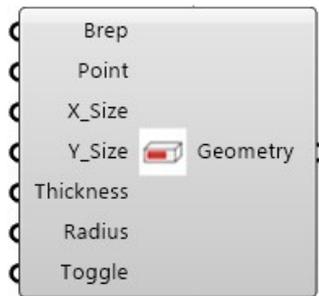


Figura 45. Componente generador de pantallas.

Por otro lado, el componente "Edges in Contours" (Fig. 46) facilitó la generación de contornos en los bordes de la cafetera, mejorando significativamente su estética. Estos contornos agregaron detalles decorativos que realzaron la apariencia general del objeto, contribuyendo así a su atractivo visual. En conjunto, estos componentes proporcionaron las herramientas necesarias para lograr un diseño estilizado y atractivo para la cafetera.

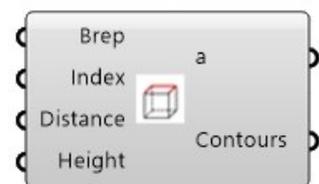


Figura 46. Componente para bordes.

En el proceso de diseño de la cafetera utilizando el algoritmo y los componentes previamente explicados, se generaron varios modelos que exploraron diversas opciones estilísticas y visualmente funcionales (Fig. 47).

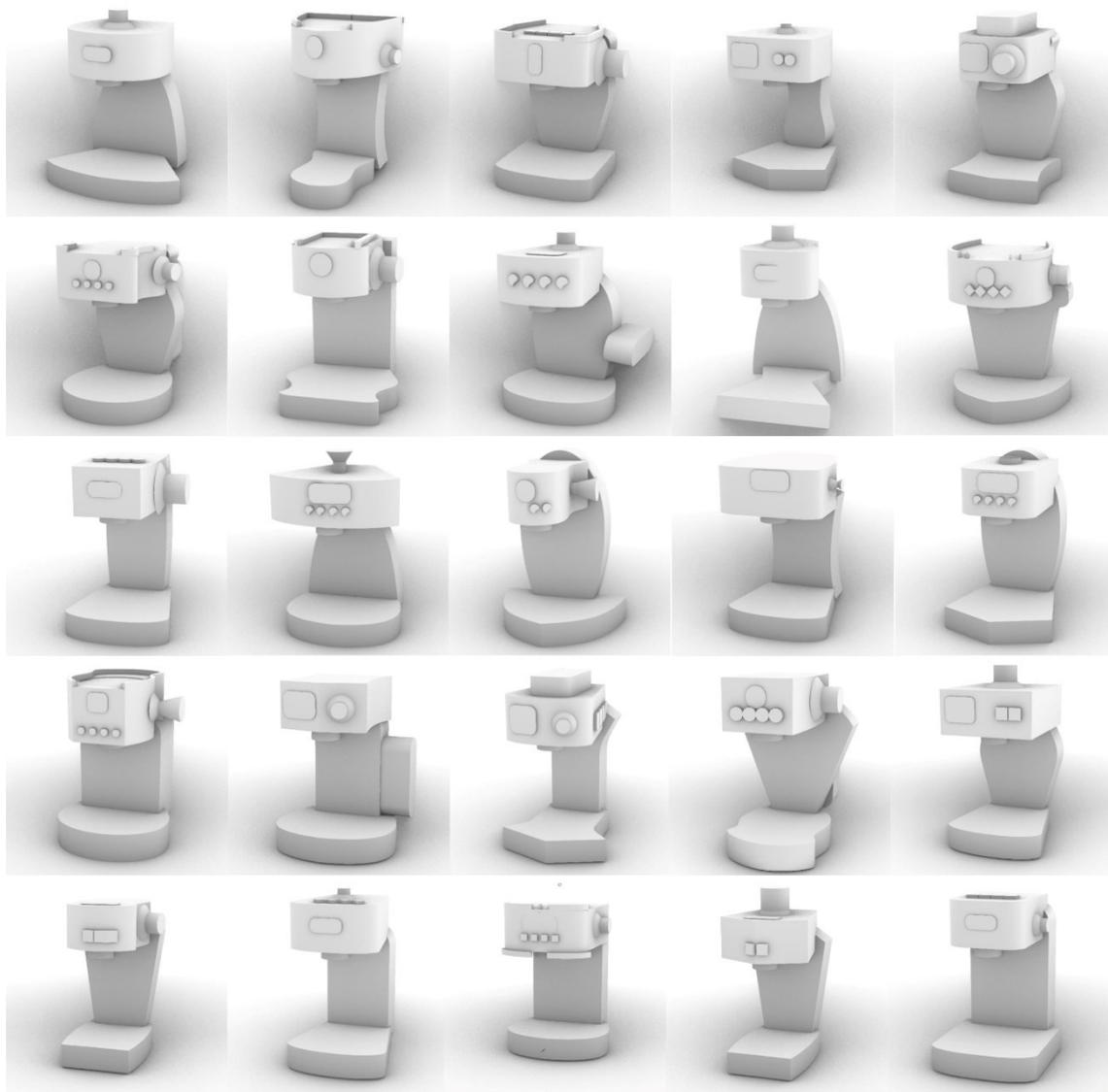


Figura 47. Ejemplos de soluciones de diseño generadas.

Cada uno de estos modelos se creó utilizando el mismo algoritmo y los componentes descritos anteriormente, lo que demuestra la versatilidad y flexibilidad del enfoque de diseño. La capacidad de adaptar la estructura principal, generar subestructuras y agregar elementos esenciales y no esenciales permitió explorar una amplia variedad de estilos y funciones en la creación de diferentes modelos de cafetera. Estos modelos sirvieron como punto de partida para la toma de decisiones en el proceso de diseño final.

Ahora bien, en el proceso de diseño de la cinta de correr utilizando la misma estructura básica, el mismo flujo de trabajo y los componentes previamente establecidos ofrece un ejemplo convincente de la versatilidad de este enfoque de diseño. A continuación, profundizaremos en el desarrollo de diseño de este segundo producto, destacando cómo se aplicaron los mismos principios que se utilizaron para la cafetera.

Como se mencionó anteriormente, la estructura principal compartida entre la cafetera y la cinta de correr es una disposición "cóncava horizontal". Esta forma de base presenta

una curva que se asemeja a la letra "C" en su orientación horizontal y proporciona una plataforma sólida para ambos productos.

A pesar de ser la misma estructura básica, es crucial adaptarla a las necesidades específicas de una cinta de correr. En este caso, se llevaron a cabo modificaciones para garantizar que la estructura proporcionara estabilidad, soporte y espacio suficiente para el área de carrera. Esto incluyó ajustes en la longitud y la disposición de algunas líneas de la estructura para adaptarla a la representación de la cinta de correr (Fig. 48).

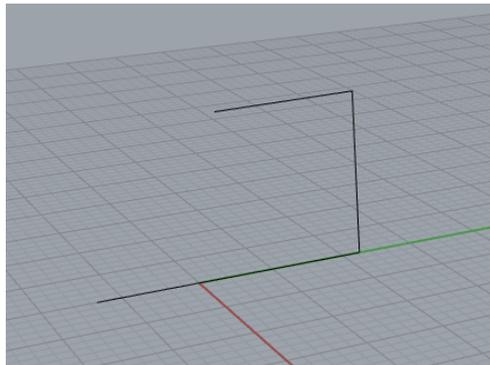


Figura 48. Estructura básica de la cinta de correr.

Los perfiles empleados en el diseño de la cinta de correr fueron elegidos cuidadosamente y transferidos desde el entorno de modelado de Rhino. Estos perfiles fueron objeto de ajustes y personalizaciones específicas para satisfacer las necesidades del diseño de la cinta de correr, aprovechando para ello el clúster de modificación de curvas. Esto permitió una adaptación precisa de la forma de la superficie de carrera y otros componentes cruciales. Es importante señalar que estos perfiles son completamente editables por parte del usuario, lo que brinda flexibilidad para ajustar y perfeccionar el diseño. Además, existe la posibilidad de incorporar nuevos perfiles para enriquecer aún más las opciones disponibles y diversificar los resultados obtenidos con la herramienta de diseño.

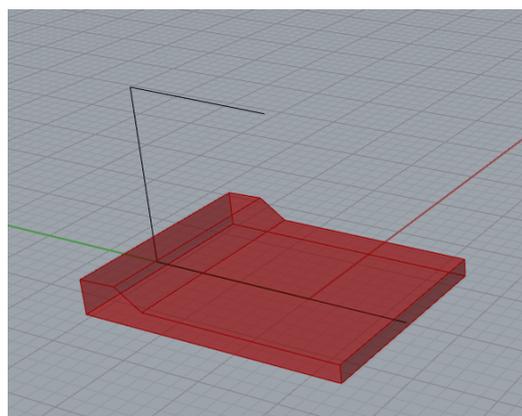


Figura 49. Volumen creado a partir de un perfil seleccionado.

Es importante destacar que, en este caso, se hicieron uso tanto del clúster de extrusión como del clúster de loft, ya que este último ofreció resultados más adecuados para la creación de volúmenes extensionales. Se empleó el clúster de loft para generar volúmenes como la parte que conecta los apoyabrazos y la base (Fig. 49), así como para

diseñar los propios apoyabrazos del producto (Fig. 50). Esto permitió obtener formas más suaves y extensas de manera eficaz.

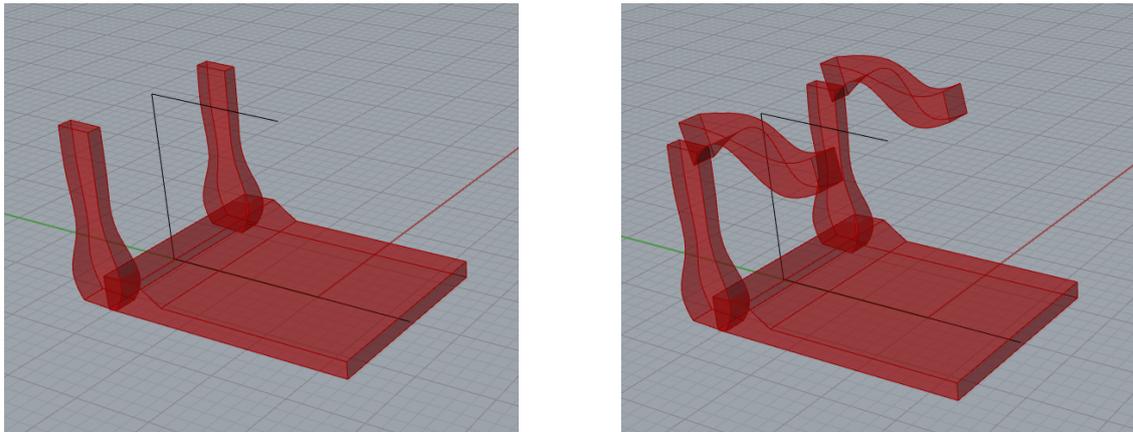


Figura 50. Volúmenes extensionales.

De igual manera, se recurrió al clúster de loft para diseñar el panel de control que albergaría botones y pantallas. Este enfoque se aplicó con éxito para obtener una forma continua y armoniosa en el panel de control, lo que resultó en una estética coherente con el diseño general de la cinta de correr.

Para la cinta de correr, se emplearon los mismos componentes "Auxiliary Point" y "Array Points" que se utilizaron en el diseño de la cafetera para crear subestructuras estratégicas. Estas subestructuras se ubicaron de manera estratégica para servir como puntos de referencia destinados a la colocación de elementos adicionales, como botones de control y detalles decorativos. Antes de abordar la generación de los volúmenes principales de la cinta de correr, se creó una línea de subestructura especial que partía desde la parte superior de la estructura básica. Esta línea fue diseñada específicamente para acomodar la construcción del panel de control, botones y pantallas del producto, garantizando así su funcionalidad y usabilidad.

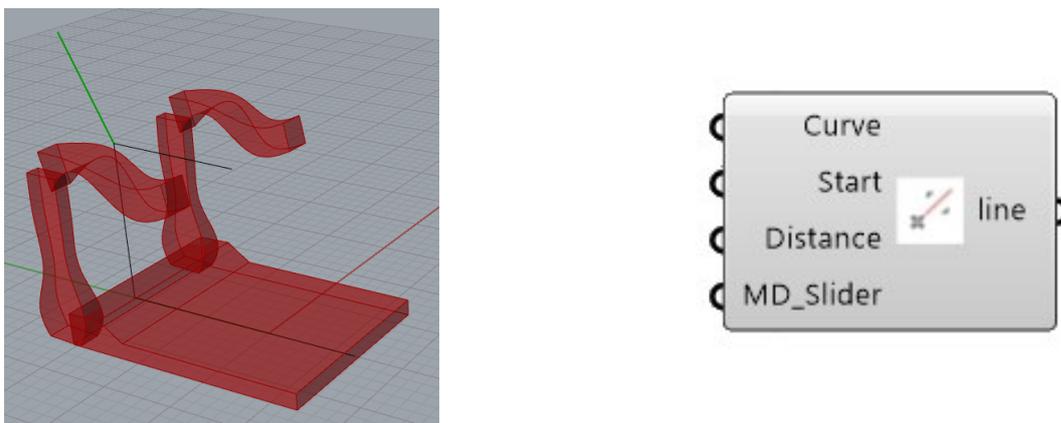


Figura 51. Creación de subestructura para panel de control.

Una vez que se dispone de la línea de subestructura (Fig. 51), se procede a la colocación de un volumen, siguiendo el mismo proceso empleado para el resto de la estructura

básica. En este caso particular, se utilizó el clúster de loft para generar dicho volumen de manera eficiente y más precisa, asegurando que se integre armoniosamente con la estructura principal de la cinta de correr (Fig. 52).

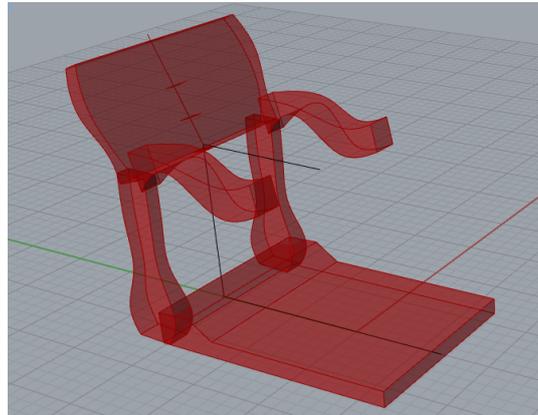


Figura 52. Creación de volumen en la subestructura.

Para diseñar los elementos que formarán parte del panel de control (Fig. 53), se siguió un proceso similar al utilizado para los elementos adicionales de la cafetera. Se comenzó creando puntos de referencia utilizando los componentes específicos desarrollados para este propósito, que servirían como base para la colocación de pantallas y botones. En primer lugar, se abordaron los elementos esenciales, como la pantalla de visualización, asegurando que estos componentes cumplan con las funciones principales requeridas en el producto.

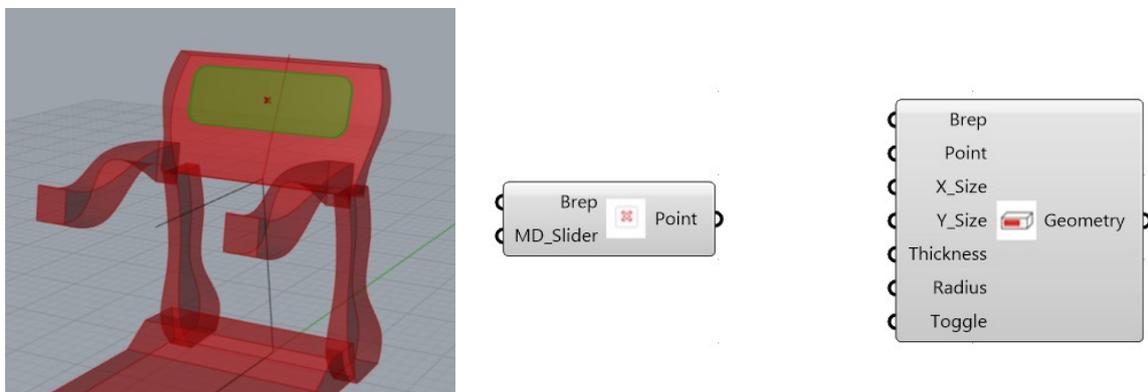


Figura 53. Implementación de elementos adicionales.

De manera similar, se abordó la creación de botones y otros controles esenciales para el panel de control (Fig. 54). En este caso, se emplearon los componentes de puntos en matriz y componentes automáticos para la generación de botones. Estos componentes facilitaron la creación de los controles necesarios para el funcionamiento y la interfaz de usuario de la cinta de correr.

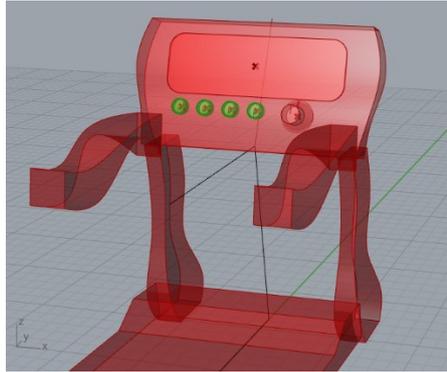


Figura 54. Implementación de controles.

A continuación (Fig. 55), se presentan los diferentes diseños generados para el producto de la cinta de correr utilizando el algoritmo y la metodología descrita anteriormente. Estos diseños muestran cómo la misma estructura básica y el conjunto de componentes pueden adaptarse de manera efectiva para crear una variedad de productos con distintos estilos y características. Cada diseño se ha personalizado para cumplir con los requisitos específicos de la cinta de correr, incluyendo la disposición de la superficie de carrera, los controles de usuario y otros elementos clave. Esta flexibilidad demuestra la eficacia y la versatilidad del enfoque de diseño paramétrico en la creación de productos innovadores y funcionales.

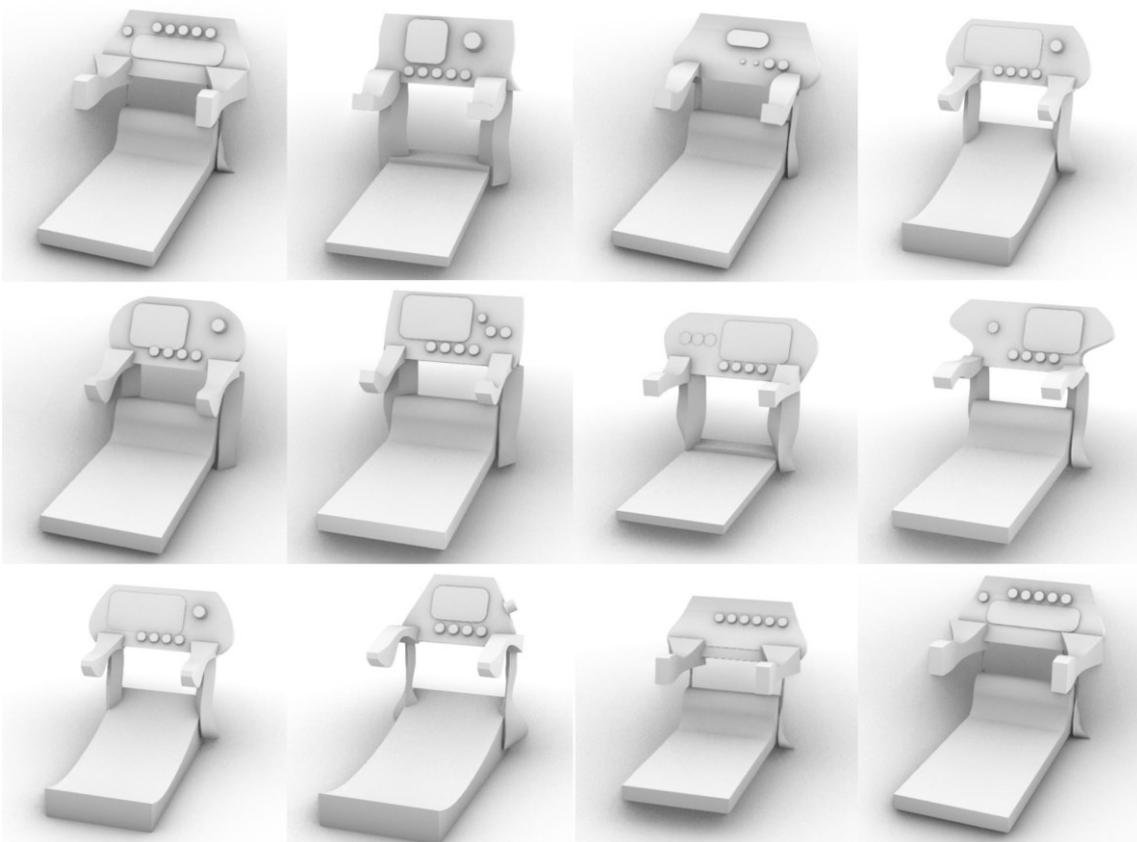


Figura 55. Ejemplos de soluciones de diseño para una cinta de correr.

El uso del algoritmo y la misma estructura básica en el diseño tanto de la cafetera como de la cinta de correr ha brindado una serie de beneficios significativos en el proceso de diseño y desarrollo de productos. En primer lugar, la reutilización de la estructura básica ha permitido un ahorro considerable de tiempo y recursos, ya que no fue necesario diseñar una estructura desde cero para cada producto. Esto agilizó el proceso de diseño y facilitó la experimentación con diferentes conceptos.

Además, la flexibilidad del algoritmo y de los componentes desarrollados ha permitido adaptar la misma estructura básica a las necesidades específicas de cada producto. Esto ha demostrado la capacidad del enfoque paramétrico para generar productos variados y personalizados a partir de una base común.

Otro beneficio clave ha sido la posibilidad de realizar cambios y ajustes de manera rápida y sencilla. Al modificar los parámetros en el algoritmo o los componentes, se pueden explorar diferentes diseños y variaciones sin tener que rediseñar todo el producto desde cero. Esto ha facilitado la iteración y la optimización del diseño.

Adicionalmente, el uso de subestructuras y elementos adicionales ha permitido la creación de detalles esenciales y no esenciales de manera eficiente. Los componentes diseñados para generar botones, pantallas y otros elementos han mejorado la estética y la usabilidad de los productos.

En resumen, el enfoque de diseño paramétrico y la reutilización de la misma estructura básica han resultado en un proceso de diseño más eficiente, flexible y versátil. Esto ha llevado a la creación de productos innovadores y personalizados que cumplen con los requisitos específicos de cada producto, al tiempo que se ahorran recursos y tiempo en el proceso de desarrollo.

El enfoque de diseño generativo utilizado en el desarrollo de la cafetera y la cinta de correr presenta una serie de ventajas significativas en comparación con el método tradicional de diseño asistido por ordenador (CAD). A continuación, se detallan las razones por las cuales el diseño generativo es una opción superior en este contexto:

**Eficiencia en la Creación de Formas:** El diseño generativo se destaca por su eficacia en la generación de formas complejas y personalizadas mediante algoritmos y reglas paramétricas. En el contexto de la cafetera y la cinta de correr, esta eficiencia se traduce en la capacidad de diseñar con rapidez las estructuras principales y secundarias, incluida la estructura "cóncavo horizontal". Esto supone un ahorro considerable de tiempo en comparación con el laborioso proceso de modelado manual en CAD. En CAD, la creación de formas complejas puede ser ardua, ya que a menudo carece de una estructura básica optimizada para la generación de volúmenes y se centra en la edición individual de elementos no paramétricos.

**Exploración Iterativa:** El diseño generativo fomenta la iteración y la exploración de diversas opciones de diseño de manera eficiente. Los diseñadores tienen la flexibilidad de ajustar los parámetros y evaluar múltiples variaciones de forma rápida y sencilla. En contraposición, el CAD tradicional a menudo requiere modificaciones manuales en cada iteración, lo que puede consumir una cantidad significativa de tiempo y recursos. Además, en el CAD tradicional, las modificaciones en elementos individuales pueden desencadenar desafíos en la gestión de las relaciones entre dimensiones y referencias

geométricas, lo que a su vez puede llevar a una revisión más exhaustiva del modelo tridimensional.

**Flexibilidad y Adaptabilidad:** El uso de componentes paramétricos y algoritmos en el diseño generativo proporciona una alta flexibilidad y adaptabilidad. Los cambios en el diseño se pueden realizar con facilidad al ajustar los parámetros, lo que es particularmente útil cuando se trabaja en productos con múltiples componentes y detalles, como los controles y elementos estéticos de un producto.

**Reutilización de Diseños:** El diseño generativo permite una eficiente reutilización de diseños paramétricos. La misma estructura principal puede servir como base para la creación de diversos productos, lo que resulta en un ahorro considerable de tiempo y esfuerzo en comparación con el enfoque CAD tradicional, donde generalmente se debe comenzar cada diseño desde cero. Los elementos considerados tanto detalles esenciales como no esenciales pueden reutilizarse y modificarse fácilmente en distintos productos, permitiendo cambios en su tamaño, forma y posición en tiempo real, algo que suele ser difícil de lograr en un diseño CAD tradicional. Para este trabajo en concreto, los componentes desarrollados, como los generadores de botones, pantallas y otros detalles, automatizan tareas repetitivas y detalladas. Esto garantiza la coherencia y la precisión en la colocación de elementos adicionales esenciales y no esenciales en el producto final, lo que sería una tarea manual en el diseño CAD tradicional.

## 7. CONCLUSIONES.

Esta tesis doctoral ha explorado y desarrollado una herramienta asistencial destinada a mejorar el proceso de diseño generativo en la etapa conceptual. A lo largo de esta investigación, se han abordado diversos aspectos fundamentales, y se han alcanzado hallazgos significativos que contribuyen de manera destacada al campo del diseño asistido por ordenador. Además, se ha propuesto una clasificación de estructuras básicas que engloba una amplia variedad de productos. La investigación comprendió una búsqueda exhaustiva de la historia del diseño generativo, con la recopilación de los trabajos más representativos y la identificación de la evolución de este enfoque a lo largo del tiempo, junto con sus diversas aplicaciones en distintos sectores.

Se destacó la trascendencia de la estética en el diseño, y su relevancia fue resaltada no solo en la conformación visual del objeto, sino también como un componente en la identificación conceptual del mismo. Se determinó que la estética desempeña un papel importante desde las etapas conceptuales del diseño, y se incorporaron elementos específicos destinados a enriquecerla. Siguiendo las contribuciones de investigadores como (Reich, 1993) y otros, se evidenció que la apreciación estética no debe pasar desapercibida en los sistemas de soporte de diseño computacional. La herramienta desarrollada integra este factor como un elemento destacado en la generación de soluciones estéticamente representativas.

Asimismo, las investigaciones de (Stephen N. Smyth, 2000) y (Yanagisawa & Fukuda, 2005) han resaltado la necesidad de lograr una integración efectiva entre forma y función desde las etapas iniciales del diseño. La herramienta asistencial se ha diseñado estratégicamente para aprovechar el potencial de los sistemas CAD, facilitando así la creación de diseños conceptuales que cumplen con este requisito. Esta conjunción de factores contribuye significativamente a la eficiencia y efectividad del proceso de diseño generativo, permitiendo a los diseñadores explorar, conceptualizar y desarrollar productos de manera más rápida.

Los estudios de (Chang et al., 2016), (Krish, 2011) y (Rosenman, 1997) han enfatizado consistentemente el impacto positivo del uso del ordenador y las herramientas CAD en la creatividad del diseñador, particularmente en las fases iniciales del proceso de diseño. La herramienta desarrollada reconoce y capitaliza este potencial, proporcionando un entorno propicio para la generación creativa de soluciones.

En relación con la viabilidad de las soluciones generadas, un aspecto subrayado por (Krish, 2011) y (Rosenman, 1997), se consideró esencial en el diseño de la herramienta

asistencial. Se buscó garantizar que las soluciones propuestas no solo fueran estéticamente distintivas, sino también viables desde el punto de vista práctico.

Al considerar la eficiencia del diseño generativo, concluimos que esta metodología utilizada en el desarrollo de productos ofrece ventajas significativas en términos de eficiencia, flexibilidad, adaptabilidad y automatización en comparación con el diseño CAD tradicional. Facilita a los diseñadores la creación de productos de manera más rápida y efectiva, les permite explorar múltiples opciones de diseño y mantiene una mayor coherencia en todo el proceso de diseño y desarrollo de productos. Estos hallazgos respaldan la adopción y aplicación continuada del diseño generativo en el ámbito del diseño de productos.

A pesar de su logrado desarrollo inicial, la herramienta presenta desafíos que sugieren áreas de mejora para investigaciones futuras. Un aspecto crítico por considerar en trabajos posteriores es la continua optimización de la herramienta, especialmente en términos de rendimiento y eficiencia, principalmente en la manipulación de datos complejos y geometrías avanzadas. Además, se reconoce la importancia de realizar pruebas de usuario para validar la herramienta, verificar su eficacia y usabilidad, y permitir a los usuarios familiarizarse con el uso de los componentes y la interacción fluida con el flujo de trabajo.

Asimismo, se vislumbran áreas prometedoras para investigaciones futuras, como la integración de inteligencia artificial y aprendizaje automático para personalizar aún más las soluciones generadas. También, se plantea la expansión de la herramienta para abordar aplicaciones específicas en diversos sectores, lo que aumentaría su utilidad y aplicabilidad en diferentes contextos.

En resumen, esta tesis doctoral culmina con el desarrollo de una herramienta asistencial sólida y eficaz que aborda de manera integral los desafíos comunes en el diseño generativo. La combinación de la estética, el uso efectivo del ordenador y la viabilidad de las soluciones generadas ofrece una alternativa para las herramientas de diseño generativo en la etapa conceptual. Estos hallazgos no solo enriquecen el conocimiento académico en el campo del diseño asistido por ordenador, sino que también tienen aplicaciones prácticas significativas en la industria del diseño de productos.

## 8. CONTRIBUCIONES CIENTÍFICAS.

### Revistas:

Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio; Acosta-Zazueta, Gonzalo. (2020) A 3D shape generative method for aesthetic product design. Design Studies (66)144 - 176. 10.1016/j.destud.2019.11.003

### Participación en Congresos:

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2022). Evolution And Future Trends In Generative Product Design. EN 2nd International Conference on Architecture and Design (DARCH 2022). (74 - 85). Online: OCERINT.

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2022). Algorithmic Design Of Products Based On Their Basic Structure. EN 2nd International Conference on Architecture and Design (DARCH 2022). (68 – 73). Online: OCERINT.

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2021). Generative Design Software And Different Approaches. EN 1st International Conference on Architecture and Design (DARCH 2021). (101 - 108). Online: International Organization Center of Academic Research.

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2021). El diseño generativo en la etapa del diseño conceptual. EN Coloquio Internacional de Investigación en Diseño 2021. Diseño, Sociedad y Pandemia.

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2021). Generación de estructuras adaptables en producto en el diseño generativo. EN Coloquio Internacional de Investigación en Diseño 2021. Diseño, Sociedad y Pandemia.

Acosta-Zazueta, Gonzalo Rafael; Alcaide-Marzal, Jorge; Diego-Mas, Jose Antonio (2021). The Structure Of Objects And Its Use In Design With Generative Algorithms. EN 1st International Conference on Architecture and Design (DARCH 2021). Online: International Organization Center of Academic Research.



## 9. BIBLIOGRAFÍA

- Abdullah, H. K., & Kamara, J. M. (2013). Parametric Design Procedures: A New Approach to Generative-Form in the Conceptual Design Phase. *Aei 2013*, (April), 334–343. <https://doi.org/10.1061/9780784412909.032>
- Agarwal, M., & Cagan, J. (1998). A blend of different tastes: The language of coffeemakers. *Environment and Planning B: Planning and Design*, 25(2), 205–226. <https://doi.org/10.1068/b250205>
- Agarwal, Manish;, Cagan, J., & Constantine, K. G. (1999). Influencing generative design through continuous evaluation: Associating costs with the coffeemaker shape grammar. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 13(4), 253–275. <https://doi.org/10.1017/S0890060499134024>
- Akner-koler, C. (1994). *Three-Dimensional Visual Analysis*. Stockholm, Sweden: REPROPRINT.
- Albers, A., Rovira, N. L., Aguayo, H., & Maier, T. (2008). Optimization with genetic algorithms and splines as a way for computer aided innovation: Follow up of an example with crankshafts. *IFIP International Federation for Information Processing*, 277, 7–18. [https://doi.org/10.1007/978-0-387-09697-1\\_1](https://doi.org/10.1007/978-0-387-09697-1_1)
- Alcaide-Marzal, J., Diego-Mas, J. A., & Acosta-Zazueta, G. (2020). A 3D shape generative method for aesthetic product design. *Design Studies*, 66, 144–176. <https://doi.org/10.1016/j.destud.2019.11.003>
- Apineda, L. (1993). On computational models of drafting and design. *Design Studies*, 14(2), 124–156. [https://doi.org/10.1016/0142-694X\(93\)80044-D](https://doi.org/10.1016/0142-694X(93)80044-D)
- Barros, M., Duarte, J. P., & Chaparro, B. M. (2011). Thonet chair design grammar: a step towards the mass customization of furniture. *Proceedings of the 14th International CAAD Futures Conference*, 181–200.
- Barros, M., Duarte, J. P., & Chaparro, B. M. (2015). A Grammar-Based Model for the Mass Customisation of Chairs: Modelling the Optimisation Part. *Nexus Network Journal*, 17(3), 875–898. <https://doi.org/10.1007/s00004-015-0265-5>
- Bidgoli, A., & Veloso, P. (2018). DeepCloud: the application of a data-driven generative model in design. *Recalibration: On Imprecision and Infidelity Paper Proceedings Book for the 2018 Association of Computer Aided Design in Architecture*

- Conference, (December), 176–185. Retrieved from [http://papers.cumincad.org/data/works/att/acadia18\\_176.pdf](http://papers.cumincad.org/data/works/att/acadia18_176.pdf)
- Brown, K. N., McMahon, C. A., & Sims Williams, J. H. (1996). Describing process plans as the formal semantics of a language of shape. *Artificial Intelligence in Engineering*, 10(2), 153–169. [https://doi.org/10.1016/0954-1810\(95\)00025-9](https://doi.org/10.1016/0954-1810(95)00025-9)
- Buelinckx, H. (1993). Wren's language of City church designs: a formal generative classification. *Environment and Planning B: Planning and Design*, Vol. 20, pp. 645 – 676. <https://doi.org/10.1068/b200645>
- Çağdaş, G. (1996). A shape grammar model for designing row-houses. *Design Studies*, 17(1), 35–51. [https://doi.org/10.1016/0142-694X\(95\)00005-C](https://doi.org/10.1016/0142-694X(95)00005-C)
- Caldas, L. (2008). Generation of energy-efficient architecture solutions applying GENE\_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics*, 22(1), 59–70. <https://doi.org/10.1016/j.aei.2007.08.012>
- Castro E Costa, E., & Duarte, J. P. (2013). Mass customization of ceramic tableware through digital technology. *Green Design, Materials and Manufacturing Processes - Proceedings of the 2nd International Conference on Sustainable Intelligent Manufacturing, SIM 2013*, 467–471. <https://doi.org/10.1201/b15002-91>
- Chang, Y. S., Chien, Y. H., Lin, H. C., Chen, M. Y., & Hsieh, H. H. (2016). Effects of 3D CAD applications on the design creativity of students with different representational abilities. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2016.08.024>
- Chase, S C. (2005). Generative design tools for novice designers: Issues for selection. *Automation in Construction*, 14(6), 689–698.
- Chase, Scott C. (2002). A model for user interaction in grammar-based design systems. *Automation in Construction*, 11(2), 161–172. [https://doi.org/10.1016/S0926-5805\(00\)00101-1](https://doi.org/10.1016/S0926-5805(00)00101-1)
- Chau, H. H. (2002). Preserving brand identity in engineering design using a grammatical approach Preserving brand identity in engineering design.
- Chau, H. H., Chen, X., McKay, A., & de Pennington, A. (2004). Evaluation of a 3D Shape Grammar Implementation. In *Design Computing and Cognition '04* (pp. 357–376). [https://doi.org/10.1007/978-1-4020-2393-4\\_19](https://doi.org/10.1007/978-1-4020-2393-4_19)
- Chen, & Owen, C. L. (1998). A study of computer-supported formal design. *Design Studies*, 19(3), 331–359. [https://doi.org/10.1016/S0142-694X\(98\)00002-7](https://doi.org/10.1016/S0142-694X(98)00002-7)
- Chen, X., & McKay, A. (2004). Package shape design principles to support brand identity. *14th IAPRI World ...*, 1–14. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:PACKAGE+SHAPE+DESIGN+PRINCIPLES+TO+SUPPORT+BRAND+IDENTITY#0>
- Chiou, S.-C., & Krishnamurti, R. (1995). The grammar of Taiwanese traditional vernacular dwellings. *Environment and Planning B: Planning and Design*, Vol. 22, pp. 689 – 720. <https://doi.org/10.1068/b220689>

- Cluzel, F., Yannou, B., & Dihlmann, M. (2012). Using evolutionary design to interactively sketch car silhouettes and stimulate designers creativity. *Engineering Applications of Artificial Intelligence*, 25(7), 1413–1424. <https://doi.org/10.1016/j.engappai.2012.02.011>
- Cui, J., & Tang, M. X. (2013). Integrating shape grammars into a generative system for Zhuang ethnic embroidery design exploration. *CAD Computer Aided Design*, 45(3), 591–604. <https://doi.org/10.1016/j.cad.2012.08.002>
- Cunningham, J. D., Shu, D., Simpson, T. W., & Tucker, C. S. (2020). A sparsity preserving genetic algorithm for extracting diverse functional 3D designs from deep generative neural networks. *Design Science*, 6(May), 1–33. <https://doi.org/10.1017/dsj.2020.9>
- Deaton, J. D., & Grandhi, R. V. (2014). A survey of structural and multidisciplinary continuum topology optimization: Post 2000. *Structural and Multidisciplinary Optimization*, Vol. 49, pp. 1–38. <https://doi.org/10.1007/s00158-013-0956-z>
- Dincer, A. E., Tong, H., & Cagdas, G. (2012). An Interaction form Generation Tool: EduDesign. *Procedia - Social and Behavioral Sciences*, 51, 74–78. <https://doi.org/10.1016/j.sbspro.2012.08.121>
- Do, E. Y.-L., Gross, M. D., Neiman, B., & Zimring, C. (2000). Intentions in and relations among design drawings. *Design Studies*, 21(5), 483–503.
- Dorsey, J., Xu, S., Smedresman, G., Rushmeier, H., & McMillan, L. (2007). The mental canvas: A tool for conceptual architectural design and analysis. *Proceedings - Pacific Conference on Computer Graphics and Applications*, 201–210. <https://doi.org/10.1109/PG.2007.62>
- Downing, F., & Flemming, U. (1981). The bungalows of Buffalo. *Environment and Planning B: Planning and Design*, 8(3), 269 – 293. <https://doi.org/10.1068/b080269>
- Duarte, J.P., & Simondetti, A. (1997). Basic grammars and rapid prototyping. *EG-SEA-AI Workshop '97, Applications of Artificial Intelligence in Structural Engineering*. Tampere, Finland.
- Duarte, José P. (2005). A discursive grammar for customizing mass housing: The case of Siza's houses at Malagueira. *Automation in Construction*, 14(2 SPEC. ISS.), 265–275. <https://doi.org/10.1016/j.autcon.2004.07.013>
- Flasiński, M. (1995). Use of graph grammars for the description of mechanical parts. *Computer-Aided Design*, 27(6), 403–433. [https://doi.org/10.1016/0010-4485\(94\)00015-6](https://doi.org/10.1016/0010-4485(94)00015-6)
- Flemming, U. (1981). The secret of the Casa Giuliani Frigerio. *Environment and Planning B: Planning and Design*, 8(1), 87–96. <https://doi.org/10.1068/b080087>
- Flemming, Ulrich. (1987). More than the sum of parts: the grammar of Queen Anne houses. *Environment and Planning B: Planning and Design*, Vol. 14, pp. 323–350. <https://doi.org/10.1068/b140323>

- Francalanza, E., Fenech, A., & Cutajar, P. (2018). Generative design in the development of a robotic manipulator. *Procedia CIRP*, 67, 244–249. <https://doi.org/10.1016/j.procir.2017.12.207>
- Garcia, S. (2017). Classifications of Shape Grammars. In *Design Computing and Cognition '16* (pp. 229–248). [https://doi.org/10.1007/978-3-319-44989-0\\_13](https://doi.org/10.1007/978-3-319-44989-0_13)
- Garcia, S., & Menezes Leitão, A. (2018). Shape grammars as design tools: An implementation of a multipurpose chair grammar. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 32(2), 240–255. <https://doi.org/10.1017/S0890060417000610>
- Gay, A., & Samar, L. (2007). *El Diseño Industrial en la Historia*. Argentina: Ediciones TeC.
- Gero, J. S. (1996). Creativity, emergence and evolution in Design. *Knowledge-Based Systems*, 9(1996), 435–448. [https://doi.org/10.1016/S0950-7051\(96\)01054-4](https://doi.org/10.1016/S0950-7051(96)01054-4)
- Gips, J. (1999). Computer Implementation of Shape Grammars. *NSF Workshop on Shape Computation*, 1–11. Retrieved from <http://www.shapegrammar.org/implement.pdf>
- Goel, V. (1995). *Sketches of Thought*. MIT Press.
- Granadeiro, V., Pina, L., Duarte, J. P. J. P., Correia, J. R. J. R., & Leal, V. M. S. V. M. S. (2013). A general indirect representation for optimization of generative design systems by genetic algorithms: Application to a shape grammar-based design system. *Automation in Construction*, 35, 374–382. <https://doi.org/10.1016/j.autcon.2013.05.012>
- Grasl, T., & Economou, A. (2011). GRAPE: Using graph grammars to implement shape grammars. *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*.
- Gunpinar, E., & Gunpinar, S. (2018). A shape sampling technique via particle tracing for CAD models. *Graphical Models*, 96, 11–29. <https://doi.org/10.1016/j.gmod.2018.01.003>
- Hemberg, M., & O'Reilly, U.-M. (2004). Extending Grammatical Evolution to Evolve Digital Surfaces with Genr8. *Genetic Programming 7th European Conference, EuroGP 2004, Proceedings*, 3003, 299–308. [https://doi.org/10.1007/978-3-540-24650-3\\_28](https://doi.org/10.1007/978-3-540-24650-3_28)
- Herbert, T., Sanders, I., & Mills, G. (1994). African shape grammar: a language of linear Ndebele homesteads. *Environment and Planning B: Planning and Design*, 21(4), 453 – 476. <https://doi.org/10.1068/b210453>
- Heusler, W., & Kadija, K. (2018). Advanced design of complex façades. *Intelligent Buildings International*, 10(4), 220–233. <https://doi.org/10.1080/17508975.2018.1493979>
- Hsiao, S.-W., & Chen, C.-H. (1997). A semantic and shape grammar based approach for product design. *Design Studies*, 18(3), 275–296. [https://doi.org/10.1016/S0142-694X\(97\)00037-9](https://doi.org/10.1016/S0142-694X(97)00037-9)

- Hsiao, S.-W. S. W., & Huang, H. H. . (2002). A neural network based approach for product form design. *Design Studies*, 23(1), 67–84.  
[https://doi.org/http://dx.doi.org/10.1016/S0142-694X\(01\)00015-1](https://doi.org/http://dx.doi.org/10.1016/S0142-694X(01)00015-1)
- Hsiao, S.-W., & Wang, H.-P. (1998). Applying the semantic transformation method to product form design. *Design Studies*, 19(3), 309–330.  
[https://doi.org/10.1016/S0142-694X\(98\)00009-X](https://doi.org/10.1016/S0142-694X(98)00009-X)
- Huang, Y., & Li, J. Y. (2014). Generative product design inspired by natural information. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8521 LNCS(PART 1), 583–593.  
[https://doi.org/10.1007/978-3-319-07731-4\\_57](https://doi.org/10.1007/978-3-319-07731-4_57)
- Hybs, I., & Gero, J. S. (1992). An evolutionary process model of design. *Design Studies*, 13(3), 273–290. [https://doi.org/10.1016/0142-694X\(92\)90216-W](https://doi.org/10.1016/0142-694X(92)90216-W)
- Ikonen, T. J., & Söbester, A. (2018). Statistical analysis of control parameters in evolutionary map L-systems-based topology optimization. *Structural and Multidisciplinary Optimization*, 58(3), 997–1013. <https://doi.org/10.1007/s00158-018-1943-1>
- Ji, G. (2016). *Digital generation of chinese ice-ray lat- tice designs*. (2013), 85–94.
- Kallio, K. (2005). 3D6B Editor : Projective 3D Sketching with Line-Based Rendering. *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 1–7.  
<https://doi.org/10.2312/SBM/SBM05/073-079>
- Kazi, R. H., Grossman, T., Cheong, H., Hashemi, A., & Fitzmaurice, G. (2017). DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17*, 401–414.  
<https://doi.org/10.1145/3126594.3126662>
- Khan, S., & Gunpinar, E. (2018). Sampling CAD models via an extended teaching–learning-based optimization technique. *CAD Computer Aided Design*.  
<https://doi.org/10.1016/j.cad.2018.03.003>
- Khemlani, L. (1995). GENWIN: A generative computer tool for window design in energy-conscious architecture. *Building and Environment*, 30(1), 73–81.  
[https://doi.org/10.1016/0360-1323\(94\)E0027-O](https://doi.org/10.1016/0360-1323(94)E0027-O)
- Kicinger, R., Arciszewski, T., & De Jong, K. (2005). Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers and Structures*, 83(23–24), 1943–1978. <https://doi.org/10.1016/j.compstruc.2005.03.002>
- Kielarova, S. W., Pradujphonphet, P., & Bohez, I. L. J. (2014). An approach of generative design system: Jewelry design application. *IEEE International Conference on Industrial Engineering and Engineering Management*, 1329–1333.  
<https://doi.org/10.1109/IEEM.2013.6962626>
- Kielarova, Somlak Wannarumon, Pradujphonphet, P., & Bohez, E. L. J. (2015). New interactive-generative design system: Hybrid of shape grammar and evolutionary design - An application of jewelry design. *Lecture Notes in Computer Science*

- (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). [https://doi.org/10.1007/978-3-319-20466-6\\_33](https://doi.org/10.1007/978-3-319-20466-6_33)
- Kielarova, Somlak Wannarumon, & Sansri, S. (2016). *Shape Optimization in Product Design Using Interactive Genetic Algorithm Integrated with Multi-objective Optimization*. *10607*, 76–86. <https://doi.org/10.1007/978-3-319-69456-6>
- Knight, T. W. (1980). The generation of Hepplewhite-style chair-back designs. *Environment and Planning B: Planning and Design*, *7*(2), 227–238. <https://doi.org/10.1068/b070227>
- Koenig, R., Miao, Y., Knecht, K., Buš, P., & Mei-Chih, C. (2017). Interactive urban synthesis: Computational methods for fast prototyping of urban design proposals. *Communications in Computer and Information Science*, *724*, 23–41. [https://doi.org/10.1007/978-981-10-5197-5\\_2](https://doi.org/10.1007/978-981-10-5197-5_2)
- Koning, H., & Eizenberg, J. (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B: Planning and Design*, *8*(3), 295–323. <https://doi.org/10.1068/b080295>
- Krish, S. (2011). A practical generative design method. *Computer-Aided Design*, *43*(1), 88–100. <https://doi.org/10.1016/j.cad.2010.09.009>
- Kwon, D. Y., Gross, M. D., & Yi-Luen Do, E. (2009). ArchiDNA: An interactive system for creating 2D and 3D conceptual drawings in architectural design. *CAD Computer Aided Design*, *41*(3), 159–172. <https://doi.org/10.1016/j.cad.2008.07.007>
- Lee, H. C., Herawan, T., & Noraziah, A. (2012). Evolutionary grammars based design framework for product innovation. *Procedia Technology*, *1*, 132–136. <https://doi.org/10.1016/j.protcy.2012.02.026>
- Lee, H. C., & Tang, M. X. (2009). Evolving product form designs using parametric shape grammars integrated with genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *23*(02), 131–158. <https://doi.org/10.1017/S0890060409000031>
- Lee, L. C. J., & Lin, M. H. (2014). Experimental research in applying generative design and 3D printers in user participating design. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8517 LNCS*(PART 1), 296–307. [https://doi.org/10.1007/978-3-319-07668-3\\_29](https://doi.org/10.1007/978-3-319-07668-3_29)
- Leitão, A., Santos, L., & Lopes, J. (2012). Programming Languages for Generative Design: A Comparative Study. *International Journal of Architectural Computing*, *10*(1), 139–162. <https://doi.org/10.1260/1478-0771.10.1.139>
- Liebich, T. (1994). A design grammar for architectural languages. *Automation in Construction*, *2*(4), 261–273. [https://doi.org/10.1016/0926-5805\(94\)90002-7](https://doi.org/10.1016/0926-5805(94)90002-7)
- Lin, M.-H., & Lee, L.-C. (2013). An Experimental Study for Applying Generative Design to Electronic Consumer Products. *Part IV LNCS*, *8015*, 392–401. [https://doi.org/10.1007/978-3-642-39253-5\\_43](https://doi.org/10.1007/978-3-642-39253-5_43)

- Liu, Y., Xu, C., Pan, Z., & Pan, Y. (2006). Semantic modeling for ancient architecture of digital heritage. *Computers and Graphics (Pergamon)*, 30(5), 800–814. <https://doi.org/10.1016/j.cag.2006.07.008>
- Lubart, T. (2005). How can computers be partners in the creative process: Classification and commentary on the Special Issue. *International Journal of Human Computer Studies*, 63(4-5 SPEC. ISS.), 365–369. <https://doi.org/10.1016/j.ijhcs.2005.04.002>
- Mandić, M., & Tepavčević, B. (2015). Analysis of shape grammar application as a tool for urban design. *Environment and Planning B: Planning and Design*, 42(4), 675–687. <https://doi.org/10.1068/b130084p>
- McCormack, J. P., Cagan, J., & Vogel, C. M. (2004). Speaking the Buick language: Capturing, understanding, and exploring brand identity with shape grammars. *Design Studies*, 25(1), 1–29. [https://doi.org/10.1016/S0142-694X\(03\)00023-1](https://doi.org/10.1016/S0142-694X(03)00023-1)
- McDermott, J. (2012). Graph grammars as a representation for interactive evolutionary 3D design. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7247 LNCS, 199–210. [https://doi.org/10.1007/978-3-642-29142-5\\_18](https://doi.org/10.1007/978-3-642-29142-5_18)
- McKay, A., Ang, M. C., Chau, H. H., & Pennington, A. de. (2006). Combining Evolutionary Algorithms And Shape Grammars To Generate Branded Product Design. In *Design Computing and Cognition '06* (pp. 521–539). [https://doi.org/10.1007/978-1-4020-5131-9\\_27](https://doi.org/10.1007/978-1-4020-5131-9_27)
- McKay, A., Stiny, G. N., & De Pennington, A. (2016). Principles for the definition of design structures. *International Journal of Computer Integrated Manufacturing*, 29(3), 237–250. <https://doi.org/10.1080/0951192X.2014.1003412>
- Medjdoub, B., & Bi, G. (2018). Parametric-based distribution duct routing generation using constraint-based design approach. *Automation in Construction*. <https://doi.org/10.1016/j.autcon.2018.02.006>
- Miao, Y., Koenig, R., Knecht, K., Konieva, K., Buš, P., & Chang, M. C. (2018). Computational urban design prototyping: Interactive planning synthesis methods—a case study in Cape Town. *International Journal of Architectural Computing*, 16(3), 212–226. <https://doi.org/10.1177/1478077118798395>
- Mitra, N. J., Wand, M., Zhang, H., Cohen-Or, D., Kim, V., & Huang, Q. X. (2014). Structure-aware shape processing. *ACM SIGGRAPH 2014 Courses, SIGGRAPH 2014*. <https://doi.org/10.1145/2614028.2615401>
- Neill, M. O., McDermott, J., Swafford, J. M., Byrne, J., Hemberg, E., Brabazon, A., & Engineering, C. (2010). Evolutionary design using grammatical evolution and shape grammars : designing a shelter. *Int. J. Design Engineering*, 3(1), 4–24. <https://doi.org/doi:10.1504/IJDE.2010.032820>
- Obrenovic, Z. (2009). Sketching and Creativity. *Creativity*.
- Orsborn, S., Cagan, J., Pawlicki, R., & Smith, R. C. (2006). Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 20(3),

217–246. <https://doi.org/10.1017/S0890060406060185>

- Ozdemir, S., & Ozdemir, Y. (2018). Prioritizing store plan alternatives produced with shape grammar using multi-criteria decision-making techniques. *Environment and Planning B: Urban Analytics and City Science*, 45(4), 751–771. <https://doi.org/10.1177/0265813516686566>
- Paczkowski, P., Kim, M., & Morvan, Y. (2011). Insitu: sketching architectural designs in context. *ACM Transactions on Graphics*, 30(6), 182:1-182:10. <https://doi.org/2024156.2024216>
- Pantazis, E., & Gerber, D. (2018). A framework for generating and evaluating façade designs using a multi-agent system approach. *International Journal of Architectural Computing*, 16(4), 248–270. <https://doi.org/10.1177/1478077118805874>
- Pollalis, S. N. (1994). Towards The Automation of Conceptual Design. In *Automation Based Creative Design—Research and Perspectives* (pp. 429–445). <https://doi.org/10.1016/B978-0-444-89870-8.50029-7>
- Prats, M, Earl, C., Garner, S., & Jowers, I. (2006). Shape exploration of designs in a style: Toward generation of product designs. *Artificial Intelligence for Engineering Design and Manufacturing*, 20, 201–215. <https://doi.org/10.1017/S0890060406060173>
- Prats, Miquel, Lim, S., Jowers, I., Garner, S. W., & Chase, S. (2009). Transforming shape in design: observations from studies of sketching. *Design Studies*, 30(5), 503–520. <https://doi.org/10.1016/j.destud.2009.04.002>
- Pugliese, M. J., & Cagan, J. (2002). Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar. *Research in Engineering Design*, 13(3), 139–156. <https://doi.org/10.1007/s00163-002-0013-1>
- Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *CAD Computer Aided Design*, 43(3), 303–315. <https://doi.org/10.1016/j.cad.2010.12.015>
- Regenwetter, L., Nobari, A. H., & Ahmed, F. (2022). Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*, 144(7). <https://doi.org/10.1115/1.4053859>
- Reich, Y. (1993). A model of aesthetic judgment in design. *Artificial Intelligence in Engineering*, 8(2), 141–153. [https://doi.org/10.1016/0954-1810\(93\)90023-9](https://doi.org/10.1016/0954-1810(93)90023-9)
- Renner, G., & Ekárt, A. (2003). Genetic algorithms in computer aided design. *Computer-Aided Design*, 35(8), 709–726. [https://doi.org/10.1016/S0010-4485\(03\)00003-4](https://doi.org/10.1016/S0010-4485(03)00003-4)
- Robertson, B. F., & Radcliffe, D. F. (2009). Impact of CAD tools on creative problem solving in engineering design. *CAD Computer Aided Design*. <https://doi.org/10.1016/j.cad.2008.06.007>

- Rosenman, M. A. (1997). An exploration into evolutionary models for non-routine design. *Artificial Intelligence in Engineering*, 11(3), 287–293.  
[https://doi.org/10.1016/S0954-1810\(96\)00046-5](https://doi.org/10.1016/S0954-1810(96)00046-5)
- Ruiz-Montiel, M., Boned, J., Gavilanes, J., Jiménez, E., Mandow, L., & Pérez-De-La-Cruz, J. L. (2013). Design with shape grammars and reinforcement learning. *Advanced Engineering Informatics*, 27(2), 230–245.  
<https://doi.org/10.1016/j.aei.2012.12.004>
- Sansri, S., & Kielarova, S. W. (2018). *Multi-Objective Shape Optimization in Generative Design: Art Deco Double Clip Brooch Jewelry Design*. 450.  
<https://doi.org/10.1007/978-981-10-6454-8>
- Sass, L. (2008). A physical design grammar: A production system for layered manufacturing machines. *Automation in Construction*, 17(6), 691–704.  
<https://doi.org/10.1016/j.autcon.2007.12.003>
- Schon, D. A., & Wiggins, G. (1992). Kinds of seeing and their functions in designing. *Design Studies*, 13(2), 135–156.
- Seeböhm, T., & Wallace, W. (1998). Rule-based representation of design in architectural practice. *Automation in Construction*, 8(1), 73–85.  
[https://doi.org/10.1016/S0926-5805\(98\)00066-1](https://doi.org/10.1016/S0926-5805(98)00066-1)
- Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards integrated performance-driven generative design tools. *Automation in Construction*, 14(2 SPEC. ISS.), 253–264.
- Shea, K., & Cagan, J. (1999). The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. *Design Studies*, 20(1), 3–23. [https://doi.org/10.1016/S0142-694X\(98\)00019-2](https://doi.org/10.1016/S0142-694X(98)00019-2)
- Shieh, M. D., Li, Y., & Yang, C. C. (2018). Comparison of multi-objective evolutionary algorithms in hybrid Kansei engineering system for product form design. *Advanced Engineering Informatics*, 36, 31–42.  
<https://doi.org/10.1016/j.aei.2018.02.002>
- Singh, V., & Gu, N. (2011). Towards an integrated generative design framework. *Design Studies*, *In Press*,. <https://doi.org/10.1016/j.destud.2011.06.001>
- Smyth, M., & Edmonds, E. (2000). Supporting design through the strategic use of shape grammars. *Knowledge-Based Systems*, 13(6), 385–393.  
[https://doi.org/10.1016/S0950-7051\(00\)00079-4](https://doi.org/10.1016/S0950-7051(00)00079-4)
- Stephen N. Smyth, D. R. W. (2000). Towards the Synthesis of Aesthetic Product Form. *ASME 2000 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, 1–8. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.5001&rep=rep1&type=pdf>
- Stiny, G., & Mitchell, W. J. (1978). The Palladian grammar. *Environment and Planning B: Planning and Design*, 5(1), 5–18. <https://doi.org/10.1068/b050005>

- Stiny, George. (1977). Ice-ray: a note on the generation of Chinese lattice designs. *Environment and Planning B: Planning and Design*.  
<https://doi.org/10.1068/b040089>
- Stiny, George. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Urban Analytics and City Science*, 7(3), 343–351.  
<https://doi.org/10.1068/b070343>
- Stiny, George, & Gips, J. (1972). Shape grammars and the generative specification of painting and sculpture. *Information Processing 71 Proceedings of the IFIP Congress 1971. Volume 2, 71*, 1460–1465. <https://doi.org/d:1526281>
- Stouffs, R. (2018). Description grammars: A general notation. *Environment and Planning B: Urban Analytics and City Science*, 45(1), 106–123.  
<https://doi.org/10.1177/0265813516667300>
- Strug, B., Grabska, E., & Ślusarczyk, G. (2014). Supporting the design process with hypergraph genetic operators. *Advanced Engineering Informatics*, 28(1), 11–27.  
<https://doi.org/10.1016/j.aei.2013.10.002>
- Sun, Y., & Taplin, J. (2018). Adapting principles of developmental biology and agent-based modelling for automated urban residential layout design. *Environment and Planning B: Urban Analytics and City Science*, 45(5), 973–993.  
<https://doi.org/10.1177/2399808317690156>
- Suwa, M., & Tversky, B. (1997). What do architects and students perceive in their design sketches? A protocol analysis. *Design Studies*, 18(4), 385–403.
- Talbott, K. (2007). Context-sensitivity and visual engagement in generative systems. *Automation in Construction*, 16(1), 54–60.  
<https://doi.org/10.1016/j.autcon.2005.10.009>
- Tsai, H. C., Chen, T. L., Tsai, H. J., & Hung, F. K. (2010). Computer-Aided Form Generation for Product Design. *Advanced Materials Research*, 97–101, 3785–3788. <https://doi.org/10.4028/www.scientific.net/AMR.97-101.3785>
- Turrin, M., Von Buelow, P., & Stouffs, R. (2011). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. *Advanced Engineering Informatics*, 25(4), 656–675.  
<https://doi.org/10.1016/j.aei.2011.07.009>
- Tyflopoulos, E., Flem, D. T., Steinert, M., & Olsen, A. (2018). *State of the art of generative design and topology optimization and potential research needs*. 1–15.
- Veloso, P., Celani, G., & Scheeren, R. (2018). From the generation of layouts to the production of construction documents: An application in the customization of apartment plans. *Automation in Construction*, 96, 224–235.  
<https://doi.org/10.1016/j.autcon.2018.09.013>
- Wang, L., Chen, K. W., Janssen, P., & Ji, G. (2020). Enabling optimisation-based exploration for building massing design: A coding-free evolutionary building massing design toolkit in rhino-grasshopper. *RE: Anthropocene, Design in the Age of Humans - Proceedings of the 25th International Conference on Computer-Aided*

*Architectural Design Research in Asia, CAADRIA 2020, 1.*

- Wang, Yedong, Jing, S., Liu, Y., Song, G., Qie, L., & Xing, H. (2018). Generative design method for lattice structure with hollow struts of variable wall thickness. *Advances in Mechanical Engineering, 10*(3).  
<https://doi.org/10.1177/1687814017752482>
- Wang, Yufei, & Duarte, J. P. (2001). Automatic generation and fabrication of designs. *Automation in Construction, 11*(3), 291–302. [https://doi.org/10.1016/S0926-5805\(00\)00112-6](https://doi.org/10.1016/S0926-5805(00)00112-6)
- Wu, Q. (2005). Bracket teaching program: A shape grammar interpreter. *Automation in Construction, 14*(6), 716–723. <https://doi.org/10.1016/j.autcon.2004.12.006>
- Yan, M. (1992). A knowledge-based design system for Yunnan-style house floor plans. *Environment and Planning B: Planning and Design, 19*(5), 559–571.  
<https://doi.org/10.1068/b190559>
- Yanagisawa, H., & Fukuda, S. (2005). Interactive Reduct Evolutional Computation for Aesthetic Design. *Journal of Computing and Information Science in Engineering, 5*(1), 1. <https://doi.org/10.1115/1.1846055>

# ANEXO A: CÓDIGO PYTHON DE COMPONENTES

## A.1 Componente "Structure Line":

```
from ghpythonlib.componentbase import executingcomponent as component
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs
```

```
class MyComponent(component):
```

```
    def RunScript(self, Curve, Parameter, Distance, MD_Slider):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs
        import Rhino as rn

        if Curve == None:
            Curve =
gh.Line(gh.ConstructPoint(0,0,0),gh.ConstructPoint(0,0,0))
            print "Falta Curve"

        if Distance == None:
            Distance = 0.00001

        if Parameter == None:
            Parameter = 0

        if MD_Slider == None:
            MD_Slider = gh.ConstructPoint(0,0,0)

        point1 = gh.EvaluateCurve(Curve, Parameter)[0]

        print MD_Slider

        center = point1
        radius = Distance

        mysphere = rs.AddSphere(center, radius)
        mysphere = rs.coercesurface(mysphere)

        u = gh.DeconstructVector(MD_Slider)[0]
        v = gh.DeconstructVector(MD_Slider)[1]
        dom = rn.Geometry.Interval(0.00,1.00)
        mysphere.SetDomain(0,dom)
        mysphere.SetDomain(1,dom)
        print mysphere.Domain(0)
        print mysphere.Domain(1)
        pp = rs.EvaluateSurface(mysphere, u, v)
        line = gh.Line(point1, pp)
```

```
# return outputs if you have them; here I try it for you:  
return line
```

## A.2 Componente "Auxiliary Point"

```
from ghpythonlib.componentbase import executingcomponent as component
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs

class MyComponent(component):

    def RunScript(self, Curve, Loft, Parameter, MD_Slider):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs
        import Rhino as rn

        if Curve == None:
            Curve =
gh.Line(gh.ConstructPoint(0,0,0),gh.ConstructPoint(0,0,0))
            print "Falta Curve"

        if Parameter == None:
            Parameter = 0

        if MD_Slider == None:
            MD_Slider = gh.ConstructPoint(0,0,0)

        point1 = gh.EvaluateCurve(Curve, Parameter)[0]

        print MD_Slider

        center = point1
        radius = 500

        mysphere = rs.AddSphere(center, radius)
        mysphere = rs.coercesurface(mysphere)

        u = gh.DeconstructVector(MD_Slider)[0]
        v = gh.DeconstructVector(MD_Slider)[1]
        dom = rn.Geometry.Interval(0.00,1.00)
        mysphere.SetDomain(0,dom)
        mysphere.SetDomain(1,dom)
        print mysphere.Domain(0)
        print mysphere.Domain(1)

        p = rs.EvaluateSurface(mysphere, u, v)
        line = gh.Line(point1, p)
        pp = gh.BrepXCurve(Loft, line)[1]

        # return outputs if you have them; here I try it for you:
        return pp
```

### A.3 Componente "Array Points":

```
from ghpythonlib.componentbase import executingcomponent as component
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs

class MyComponent(component):

    def RunScript(self, Curve, Loft, Handler, Parameter, Type_Array, Vector,
Distance, Num_Elements, Angle):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs

        point = gh.EvaluateCurve(Curve, Parameter)[0]
        line = gh.Line(point, Handler)
        pp = gh.BrepXCurve(Loft, line)[1]
        pt = gh.EndPoints(Curve)[1]
        if Vector == "x" or Vector == "X":
            dir= gh.UnitX(Distance)
        else:
            if Vector == "y" or Vector == "Y":
                dir = gh.Unity(Distance)
            else:
                print "verify 'x' or 'y' vector"

        if Type_Array == 1:
            #temp = list[0]#solo para seleccionar el index 0 de la lista
            list1 = gh.LinearArray(pp, dir,
gh.IntegerDivision(Num_Elements,2)+1)#para distribuir los puntos por igual a
ambos lados
            list2 = gh.LinearArray(pp, gh.Negative(dir),
round(Num_Elements/2))
            z = gh.Merge(list1[0],list2[0]) #juntar las listas
            pps = list(set(z)) #remover elementos repetidos de la lista
        else:
            if Type_Array == 2:
                pps = gh.PolarArray(pp, gh.XYPlane(pt), Num_Elements,
gh.Radians(Angle))[0]
            else:
                if Type_Array == 3:
                    pps = gh.RectangularArray(pp,
gh.Rectangle(gh.XYPlane(pt), Distance, Distance,0)[0], Num_Elements,
Num_Elements)[0]
                else:
                    print "error"
                    pps = gh.ConstructPoint(0,0,0)

        # return outputs if you have them; here I try it for you:
        return (pp, pps)
```

#### A.4 Componente "Curves on brep":

```
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs

class MyComponent(component):

    def RunScript(self, Curve, Brep, Count, From, To):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs
        import Rhino as rn
        import Rhino.Geometry as rg

        if Curve == None:
            Curve =
gh.JoinCurves(gh.Line(gh.ConstructPoint(0,0,0),gh.ConstructPoint(1,1,1)),gh.L
ine(gh.ConstructPoint(0,0,0),gh.ConstructPoint(1,1,2)))
            print "Falta Curve"
        if From == None:
            Parameter = 0
        if To == None:
            Parameter = 0
        if Count == None:
            Count = 0
        if Brep == None:
            Brep = gh.Sphere(gh.ConstructPoint(0,0,0), 0.01)

        Curve.Domain = rg.Interval(0,1)
        from_p = gh.EvaluateCurve(Curve, From)[0]
        to_p = gh.EvaluateCurve(Curve, To)[0]
        line = gh.Line(from_p, to_p)
        direction = gh.RotateAxis(Curve, gh.Radians(90), gh.LineSDL(from_p,
gh.UnitX(1), 1))[0]

        curve_array = gh.CurveArray(from_p, line, Count)[0]
        project_p = gh.ProjectPoint(curve_array, direction, Brep)[0]
        distance = gh.Distance(from_p, project_p)
        radius = gh.Addition(distance, gh.Multiplication(distance, 100))
        circles = gh.Circle(project_p, radius)
        project_curve = gh.Project(circles, Brep, direction)

        # return outputs if you have them; here I try it for you:
        return project_curve
```

#### A.5 Componente "Reorient Object":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs

origin = gh.BrepClosestPoint(point, loft)[0]
axis = gh.BrepClosestPoint(point, loft)[1]
plane = gh.PlaneNormal(origin, axis)
geo_orient = gh.Orient(geo, gh.XYPlane(gh.ConstructPoint(0,0,0)), plane)[0]
geo_scale = gh.Scale(geo_orient, gh.Area(geo_orient)[1], factor)[0]
obj = gh.CapHoles(gh.Extrude(geo_scale, gh.Amplitude(plane, dist)))
```

## A.6 Componente "Volume on brep":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs
import Rhino as rn

if Curve == None:
    Curve = gh.Line(gh.ConstructPoint(0,0,0),gh.ConstructPoint(0,0,0))
    print "Falta Curve"

if Parameter == None:
    Parameter = 0

if MD_Slider == None:
    MD_Slider = gh.ConstructPoint(0,0,0)

point1 = gh.EvaluateCurve(Curve, Parameter)[0]

print MD_Slider

center = point1
radius = 500

mysphere = rs.AddSphere(center, radius)
mysphere = rs.coercesurface(mysphere)

u = gh.DeconstructVector(MD_Slider)[0]
v = gh.DeconstructVector(MD_Slider)[1]
dom = rn.Geometry.Interval(0.00,1.00)
mysphere.SetDomain(0,dom)
mysphere.SetDomain(1,dom)
print mysphere.Domain(0)
print mysphere.Domain(1)

p = rs.EvaluateSurface(mysphere, u, v)
line = gh.Line(point1, p)
pp = gh.BrepXCurve(Loft, line)[1]

zero = gh.ConstructPoint(0,0,0)
centroid = gh.Area(geo)[1]
vector2p = gh.Vector2Pt(centroid, zero, False)[0]
geo0 = gh.Move(geo, vector2p)[0]

origin = gh.BrepClosestPoint(pp, Loft)[0]
axis = gh.BrepClosestPoint(pp, Loft)[1]
plane = gh.PlaneNormal(origin, axis)
geo_orient = gh.Orient(geo0, gh.XYPlane(gh.ConstructPoint(0,0,0)), plane)[0]
geo_scale = gh.Scale(geo_orient, gh.Area(geo_orient)[1],factor)[0]
obj = gh.CapHoles(gh.Extrude(geo_scale, gh.Amplitude(plane, dist)))
```

### A.7 Componente "Buttons":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs

zero = gh.ConstructPoint(0,0,0)
centroid = gh.Area(geo)[1]
vector2p = gh.Vector2Pt(centroid, zero, False)[0]
geo0 = gh.Move(geo, vector2p)[0]

origin = gh.BrepClosestPoint(point, loft)[0]
axis = gh.BrepClosestPoint(point, loft)[1]
plane = gh.PlaneNormal(origin, axis)
geo_orient = gh.Orient(geo0, gh.XYPlane(gh.ConstructPoint(0,0,0)), plane)[0]
geo_scale = gh.Scale(geo_orient, gh.Area(geo_orient)[1], factor)[0]
obj = gh.CapHoles(gh.Extrude(geo_scale, gh.Amplitude(plane, dist)))
```

#### A.8 Componente "Random button generator":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs
A = 0
B = 0
C = 0
D = 0
E = 0
Type = 1

if Type == 1:
    A = 0.5
    B = 0.5
    C = 0.08
    D = 0.1
    E = 0.07

#Variability
ratio = Variability
a = gh.Heteroptera.RandomizeNumbers(A, ratio,
gh.Heteroptera.SeedGenerator(Button))
b = gh.Heteroptera.RandomizeNumbers(B, ratio,
gh.Heteroptera.SeedGenerator(Button))
c = gh.Heteroptera.RandomizeNumbers(C, ratio,
gh.Heteroptera.SeedGenerator(Button))
d = gh.Heteroptera.RandomizeNumbers(D, ratio,
gh.Heteroptera.SeedGenerator(Button))
e = gh.Heteroptera.RandomizeNumbers(E, ratio,
gh.Heteroptera.SeedGenerator(Button))

#points
p0 = gh.ConstructPoint(0, 0, 0)
p1 = gh.ConstructPoint(0, 0, a)
p2 = gh.ConstructPoint(b, 0, a)
p3 = gh.ConstructPoint(gh.Addition(b,c), 0, gh.Subtraction(b, d))
p4 = gh.ConstructPoint(gh.Addition(b, c), 0, gh.Subtraction(a, e))
p5 = gh.ConstructPoint(gh.Addition(b, c), 0, 0)

#lines
line = gh.Line(p0, p1)
line1 = gh.Line(p1, p2)
arc = gh.Arc3Pt(p2, p4, p3)[0]
line2 = gh.Line(p3, p5)
polyline = gh.JoinCurves([ line1, arc, line2], False)
surface = gh.Revolution(polyline, line, gh.Multiplication(2,gh.Pi(1)))

vector = gh.Vector2Pt(p0, Point, False)[0]
geo = gh.Move(surface, vector)[0]
plane = gh.XYPlane(Point)

origin = gh.BrepClosestPoint(Point, Brep)[0]
zAxis= gh.BrepClosestPoint(Point, Brep)[1]
target = gh.PlaneNormal(origin, zAxis)

geo1 = gh.Orient(geo, plane, target)[0]
Geometry = gh.Scale(geo1, Point, Factor)[0]
```

### A.9 Componente "Random pommel generator":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs
A = 0
B = 0
C = 0
D = 0
E = 0

if Type == 1:
    A = 0.7
    B = 0.4
    C = 0.25
    D = 0.4
    E = 0.5

if Type == 2:
    A = 0.45
    B = 0.4
    C = 0.25
    D = 0.4
    E = 0.5

if Type == 3:
    A = 0.45
    B = 0.2
    C = 0.2
    D = 0.4
    E = 0.7

if Type == 4:
    A = 0.2
    B = 0.22
    C = 0.22
    D = 0.4
    E = 0.7

if Type == 5:
    A = 0.249
    B = 0.26
    C = 0.515
    D = 0.545
    E = 0.362

#Variability
ratio = Variability
a = gh.Heteroptera.RandomizeNumbers(A, ratio,
gh.Heteroptera.SeedGenerator(Button))
b = gh.Heteroptera.RandomizeNumbers(B, ratio,
gh.Heteroptera.SeedGenerator(Button))
c = gh.Heteroptera.RandomizeNumbers(C, ratio,
gh.Heteroptera.SeedGenerator(Button))
d = gh.Heteroptera.RandomizeNumbers(D, ratio,
gh.Heteroptera.SeedGenerator(Button))
e = gh.Heteroptera.RandomizeNumbers(E, ratio,
gh.Heteroptera.SeedGenerator(Button))

#points
p0 = gh.ConstructPoint(0, 0, 0)
```

```

p1 = gh.ConstructPoint(gh.Negative(a), 0, 0)
p2 = gh.ConstructPoint(gh.Negative(b), 0, C)
p3 = gh.ConstructPoint(gh.Negative(d), 0, gh.Addition(c,e))
p4 = gh.ConstructPoint(0, 0, gh.Addition(c,e))

#lines
polyline = gh.PolyLine([p0,p1,p2,p3,p4], False)
line = gh.Line(p0, p4)
surface = gh.Revolution(polyline, line, gh.Multiplication(2,gh.Pi(1)))

vector = gh.Vector2Pt(p0, Point, False)[0]
geo = gh.Move(surface, vector)[0]
plane = gh.XYPlane(Point)

origin = gh.BrepClosestPoint(Point, Brep)[0]
zAxis= gh.BrepClosestPoint(Point, Brep)[1]
target = gh.PlaneNormal(origin, zAxis)

geo1 = gh.Orient(geo, plane, target)[0]
Geometry = gh.Scale(geo1, Point, Factor)[0]

```

#### A.10 Componente "Screen or void":

```
import ghpythonlib.components as gh
import rhinoscriptsyntax as rs

origin = gh.ConstructPoint(0,0,0)
rectangle = gh.Rectangle(gh.ConstructPoint(0,0,0), X_Size, Y_Size, Radius)[0]
vector = gh.Vector2Pt(gh.Area(rectangle)[1], Point, False)[0]
geo = gh.Move(rectangle, vector)[0]
plane = gh.XYPlane(Point)

origin = gh.BrepClosestPoint(Point, Brep)[0]
zAxis= gh.BrepClosestPoint(Point, Brep)[1]
target = gh.PlaneNormal(origin, zAxis)

geo1 = gh.Orient(geo, plane, target)[0]

if Toggle == False:
    amplitude = Thickness
else:
    amplitude = gh.Negative(Thickness)

Geometry = gh.CapHoles(gh.Extrude(geo1, gh.Amplitude(target, amplitude)))
```

### A.11 Componente "Edges in contours":

```
from ghpythonlib.componentbase import executingcomponent as component
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs

class MyComponent(component):

    def RunScript(self, Brep, Index, Distance, Height):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs
        import Rhino as rn

        if Brep == None:
            Brep =
gh.BoxRectangle(gh.Rectangle(rn.Geometry.Plane.WorldXY,0.1,0.1,0)[0],
gh.UnitZ(0.1))
        if Index == None:
            Index = 0
        if Distance == None:
            Distance = 0

        list = gh.DeconstructBrep(Brep)[1]
        curve = gh.ListItem(list, Index, True)
        point = gh.EndPoints(curve)[1]
        Normal = gh.BrepClosestPoint(point, Brep)[1]
        offset = gh.OffsetCurve(curve, Distance, Normal, 1)
        point2 = gh.EndPoints(offset)[1]
        Normal2 = gh.BrepClosestPoint(point2, Brep)[1]
        curves = gh.Merge(curve, offset)
        #curves.Flatten(0)

        pt = rn.Geometry.Point3d.Unset
        Loft =
rn.Geometry.Brep.CreateFromLoft(curves,pt,pt,rn.Geometry.LoftType.Normal,False)

        Contour = gh.Extrude(Loft, gh.Amplitude(Normal2, Height))
        plane2 = rn.Geometry.Plane.WorldYZ
        Mirror_Contour= gh.Mirror(Contour, plane2)[0]

        # return outputs if you have them; here I try it for you:
        return (Contour, Mirror_Contour)
```

## A.12 Component "Surface pattern":

```
from ghpythonlib.componentbase import executingcomponent as component
import Grasshopper, GhPython
import System
import Rhino
import rhinoscriptsyntax as rs

class MyComponent(component):

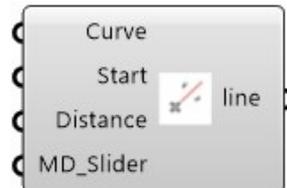
    def RunScript(self, Brep, index, Size, Length, Radius):
        import ghpythonlib.components as gh
        import rhinoscriptsyntax as rs

        face = gh.ListItem(gh.DeconstructBrep(Brep)[0], index, True)
        panel = gh.LunchBox.DiamondPanels(face, 10, 10)[0]
        vertices = gh.DeconstructBrep(panel)[2]
        #flatten = gh.FlattenTree(vertices, {0})[0]
        incurve = gh.PointInCurve(vertices, face)[0]
        dispatch = gh.Dispatch(vertices, incurve)[0]
        closestp = gh.BrepClosestPoint(dispatch, face)[0]
        closestn = gh.BrepClosestPoint(dispatch, face)[1]
        plane = gh.PlaneNormal(closestp, closestn)
        rectangle = gh.Rectangle(plane,
gh.ConstructDomain(gh.Negative(Size), Size),
gh.ConstructDomain(gh.Negative(Size), Size), Radius)[0]
        extrude = gh.Extrude(rectangle, gh.Amplitude(closestn,
gh.Negative(Length)))
        cap = gh.CapHoles(extrude)
        Geo = gh.SolidDifference(Brep, cap)

        # return outputs if you have them; here I try it for you:
        return Geo
```

# ANEXO B: MANUAL DE USO DE COMPONENTES

## B.1 Componente "Structure Line" :



**Descripción:** El componente "Structure Line" es una herramienta de diseño generativo que genera una línea a partir de un punto específico ubicado sobre una línea de la estructura principal del objeto. Su propósito principal es proporcionar una línea que se pueda ajustar fácilmente en términos de dirección y longitud. Esta línea actúa como una guía para la creación de subestructuras que permiten la incorporación de elementos adicionales en el diseño del objeto. Estos elementos pueden ser tanto detalles esenciales como no esenciales para el producto final.

### **Instrucciones de Uso:**

1. **Inicio del Componente:** Insertar el componente en el entorno de Grasshopper.

2. **Configuración de Parámetros de Entrada:**

Define los siguientes parámetros de entrada para el componente "Structure Line":

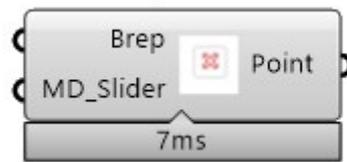
- **Curve (Curva):** Proporciona la línea base o curva sobre la cual se generará la línea.
- **Start (Inicio):** Define el punto de inicio desde el cual se generará la línea.
- **Distance (Distancia):** Establece la distancia que determinará la longitud de la línea.
- **MD\_Slider (Control MD):** Ajusta este control según sea necesario para influir en la generación de la línea.

3. **Generación de la línea:** La línea generada partirá desde el punto específico en la curva base, siguiendo la dirección y longitud definidas.

4. **Manipulación de la línea:** En cualquier momento, puedes ajustar la dirección y longitud de la línea generada según tus necesidades y preferencias.

5. **Parámetros de Salida:** El componente proporcionará una salida llamada "line", que representa la línea generada. Puedes utilizar esta salida en tu proyecto de diseño generativo.

## B.2 Componente "Auxiliary Point"



**Descripción:** El componente "Auxiliary Point" es una herramienta diseñada para generar un punto auxiliar en una superficie ya creada. Este componente utiliza un control llamado "MD Slider," que permite proyectar un punto sobre cualquier cara del objeto volumétrico que hayas definido previamente. La forma de generación del punto es similar a la del componente "Structure Line," pero en este caso, se obtiene al intersectar una curva con la superficie donde deseas ubicar el punto.

### Instrucciones de Uso:

#### 1. Configuración Inicial:

- Asegurarse de haber definido la superficie sobre la cual deseas colocar un punto auxiliar.

#### 2. Parámetros de Entrada:

- Define los siguientes parámetros de entrada para el componente "Auxiliary Point":
  - **Brep (Brep):** Ingresar el objeto volumétrico (Brep) que contiene la superficie sobre la cual deseas proyectar el punto auxiliar.
  - **MD Slider (Control MD):** Este control permite ajustar la posición del punto auxiliar en la superficie.

#### 3. Generación del Punto Auxiliar:

- Una vez que hayas configurado los parámetros de entrada, activa el componente "Auxiliary Point" para generar el punto auxiliar.
- El punto auxiliar se ubicará en la superficie del objeto en el punto de intersección entre la curva definida y la superficie.

#### 4. Ajuste de la Posición del Punto:

- Utiliza el "MD Slider" para ajustar la posición del punto auxiliar en la superficie del objeto volumétrico.
- Este ajuste te brinda flexibilidad para colocar detalles esenciales o no esenciales en puntos específicos de tu diseño.

#### 5. Creación de Elementos en la Superficie:

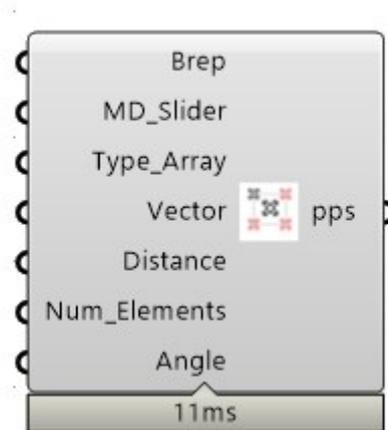
- La capacidad de proyectar puntos sobre diferentes superficies te permite crear elementos precisos y ajustados al objeto volumétrico.

- Estos elementos enriquecen la forma y la estética del producto final.

**6. Parámetros de Salida:**

- El componente proporcionará una salida llamada "Point" (Punto) que representa el punto auxiliar generado.

### B.3 Componente "Array Points"



**Descripción:** El componente "Array Points" es una herramienta versátil que te permite crear formaciones de puntos en diversas configuraciones. Estas formaciones, o matrices, pueden ser lineales, circulares o rectangulares, lo que brinda una gran flexibilidad en la generación de diseños.

#### **Instrucciones de Uso:**

##### **1. Configuración Inicial:**

- Asegúrate de haber definido la geometría o superficie donde deseas crear la formación de puntos.

##### **2. Parámetros de Entrada:**

- Define los siguientes parámetros de entrada para el componente "Array Points":
  - **Brep (Brep):** Ingresa la geometría o superficie en la que deseas crear la formación de puntos.
  - **MD Slider (Control MD):** Este control servirá como punto inicial de referencia para controlar la posición del primer elemento proyectado.
  - **Type\_Array (Tipo de Formación):** Elige el tipo de formación que deseas crear, que puede ser lineal, circular o rectangular.
  - **Vector (Vector):** Define el vector que determinará la dirección de la formación.
  - **Distance (Distancia):** Indica la distancia entre los elementos de la formación.
  - **Num\_Elements (Número de Elementos):** Especifica la cantidad de elementos a formar.

- **Angle (Ángulo)** (solo para formaciones circulares): Establece el ángulo de separación entre los elementos.

### 3. **Generación de la Formación de Puntos:**

- La formación de puntos se creará a partir del punto de inicio definido por el "MD Slider" y se extenderá en la dirección y con la distancia, número de elementos y ángulo (si es aplicable) que hayas especificado.

### 4. **Ajuste y Exploración:**

- Experimenta ajustando los parámetros de distancia, número de elementos, ángulo y tipo de formación para obtener la configuración deseada.
- La formación de puntos resultante puede utilizarse como puntos de referencia para crear y colocar patrones repetitivos, alinear detalles esenciales o elementos decorativos en tu diseño.

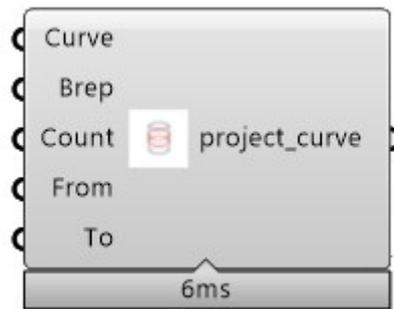
### 5. **Amplía tus Opciones de Diseño:**

- La versatilidad de "Array Points" te permite crear una amplia variedad de formas y disposiciones para los puntos generados.
- Explora diferentes estilos y composiciones en tus diseños gracias a la capacidad de crear formaciones lineales, circulares o rectangulares.

### 6. **Parámetros de Salida:**

- El componente proporcionará una salida llamada "pps" que representa la formación de puntos generada.

## B.4 Componente "Curves on Brep"



**Descripción:** El componente "Curves on Brep" es una herramienta versátil que te permite proyectar una o varias curvas sobre un cuerpo (brep) con el propósito de generar una serie de subestructuras con forma de anillos. Estos anillos pueden utilizarse para crear detalles y elementos decorativos en tu objeto de diseño.

### Instrucciones de Uso:

#### 1. Configuración Inicial:

- Asegúrate de haber definido la geometría o superficie (brep) donde deseas realizar la proyección de las curvas.

#### 2. Parámetros de Entrada:

- Define los siguientes parámetros de entrada para el componente "Curves on Brep":
  - **Curve (Curva):** Ingresa la(s) curva(s) que deseas proyectar sobre el brep.
  - **Brep (Brep):** Indica el brep donde se realizará la proyección de las curvas.
  - **Count (Cantidad):** Especifica la cantidad de anillos o subestructuras que se generarán a lo largo de la proyección de las curvas.
  - **From (Desde) y To (Hasta):** Establece el rango dentro del brep donde se llevará a cabo la proyección de las curvas. Puedes controlar el inicio y final de la proyección.

#### 3. Proyección de Curvas:

- Se generarán subestructuras con forma de anillos a lo largo de la proyección de las curvas.

#### 4. Manipulación de Curvas Proyectadas:

- Las curvas proyectadas pueden utilizarse para diversas aplicaciones:

- **Barridos (Sweeps):** Puedes efectuar barridos para crear elementos con volumen que sigan la trayectoria de las curvas proyectadas.
- **"Pipe" u otros componentes:** Aplica componentes como "pipe" para dar grosor a las curvas y crear elementos cilíndricos o tubulares.

#### 5. **Creación de Detalles:**

- Utiliza las subestructuras con forma de anillos generadas para agregar detalles y elementos decorativos al objeto de forma precisa y eficiente.

#### 6. **Amplía tus Opciones de Diseño:**

- "Curves on Brep" es útil para la creación de detalles esenciales o no esenciales en el diseño de objetos, permitiendo la generación de formas complejas y detalladas.
- Combina este componente con otros del algoritmo para lograr diseños más elaborados y personalizados.

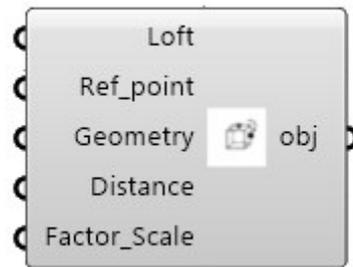
#### 7. **Posibilidades Creativas:**

- La capacidad de generar subestructuras con forma de anillos a partir de curvas proyectadas ofrece una amplia gama de posibilidades creativas. Se puede experimentar con diferentes opciones de diseño y agregar elementos distintivos al producto final.

#### 8. **Parámetros de Salida:**

- El componente proporcionará una salida llamada "project\_curve" que representa las subestructuras con forma de anillos generadas a partir de las curvas proyectadas.

## B.5 Componente "Reorient Object"



El componente "Reorient Object" es una herramienta potente en el proceso de diseño de objetos, diseñada para permitir la reorientación de cualquier geometría sobre la superficie de un objeto sin importar la cara en la que se desee representar la forma. Este componente brinda a los diseñadores una gran flexibilidad y libertad para colocar y posicionar elementos geométricos de manera precisa y controlada.

### **Instrucciones de Uso:**

#### **1. Configuración Inicial:**

- Asegúrate de haber definido la geometría base (normalmente un objeto con distintas superficies) sobre la que deseas colocar la nueva geometría.

#### **2. Creación de la Geometría a Reorientar:**

- Diseña la forma geométrica que deseas reorientar en cualquier parte del espacio de tu programa de diseño (por ejemplo, en Rhino). No hay restricciones en cuanto al tamaño o forma de esta geometría.

#### **3. Configuración de Parámetros de Entrada:**

- Define los siguientes parámetros de entrada para el componente "Reorient Object":
  - **Loft (Brep):** Ingresa la geometría base (generalmente un brep) sobre la que reorientarás la nueva geometría.
  - **Ref\_point (Punto de Referencia):** Indica el punto de referencia en la geometría base donde se alineará la nueva forma.
  - **Geometry (Geometría):** Selecciona la geometría que deseas reorientar.
  - **Distance (Distancia):** Especifica la distancia entre el punto de referencia y la nueva geometría.
  - **Factor\_Scale (Factor de Escala):** Ajusta el factor de escala si deseas cambiar el tamaño de la geometría reorientada.

#### **4. Proceso de Reorientación:**

- La nueva geometría se alineará de manera que sea perpendicular a las superficies de la geometría base.

#### **5. Exploración de Posiciones y Configuraciones:**

- Utiliza este componente para explorar diferentes configuraciones y posiciones para la geometría que estás reorientando. Esto facilita la creación de diseños complejos y detallados.

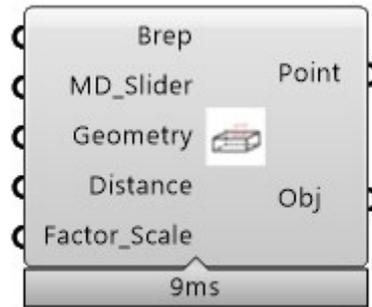
#### **6. Añadir Detalles al Diseño:**

- El componente "Reorient Object" es especialmente útil para agregar detalles esenciales o no esenciales a la estructura principal del objeto. Puedes utilizarlo para colocar botones, pantallas, mandos u otros elementos decorativos en cualquier parte del objeto.

#### **Parámetros de Salida:**

- El componente proporcionará una salida llamada "obj" que representa la nueva geometría reorientada y ajustada a la geometría base.

## B.6 Componente "Volume on brep"



**Descripción:** El componente "Volume on Brep" es una herramienta para el proceso de diseño. Su función principal es generar un volumen a partir de un perfil geométrico previamente creado. El diseñador puede proporcionar un perfil geométrico en cualquier parte del espacio de Rhino, y el componente se encargará de generar una extrusión de este perfil para crear un objeto sólido. Luego, se tapaná la extrusión para cerrar el volumen y obtener un objeto completo.

### **Instrucciones de Uso:**

#### **1. Configuración Inicial:**

- Asegúrate haber definido la geometría base sobre la que deseas colocar el nuevo volumen.

#### **2. Creación de Perfil Geométrico:**

- Diseña el perfil geométrico que deseas utilizar como base para el volumen. Esto puede ser cualquier forma geométrica que se adapte a tus necesidades de diseño.

#### **3. Configuración de Parámetros de Entrada:**

- Define los siguientes parámetros de entrada para el componente "Volume on Brep":
  - **Brep (Brep):** Ingresa la geometría base (generalmente un brep) sobre la que extruirás el perfil geométrico.
  - **MD\_Slider (Punto de Control):** Indica el punto de control que determinará la ubicación y orientación del objeto volumétrico.
  - **Geometry (Geometría):** Selecciona el perfil geométrico que has diseñado.
  - **Distance (Distancia):** Especifica la distancia de extrusión para generar el volumen.

- **Factor\_Scale (Factor de Escala):** Ajusta el factor de escala si deseas modificar las dimensiones del volumen generado.

#### 4. **Generación del Volumen:**

- El componente generará una extrusión a partir del perfil geométrico y cerrará el volumen resultante.

#### 5. **Exploración de Tamaño y Posición:**

- Utiliza este componente para explorar diferentes tamaños y posiciones para el volumen generado. Ajusta los parámetros de distancia y factor de escala según tus necesidades de diseño.

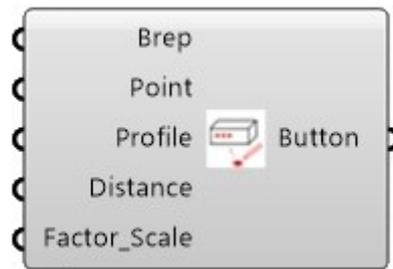
#### 6. **Control de la Posición:**

- Utiliza el MD\_Slider para controlar la ubicación y orientación del objeto volumétrico en relación con la geometría base. Esto te permitirá ajustar la posición del volumen de manera precisa.

#### **Parámetros de Salida:**

- El componente proporcionará dos salidas:
  - "Point (Punto)": Representa el punto de control que determina la ubicación y orientación del objeto volumétrico.
  - "Obj (Objeto)": Es el objeto sólido generado como resultado del proceso.

## B.7 Componente "Buttons"



**Descripción:** El componente "Buttons" es una herramienta interesante en el proceso de diseño que permite incorporar botones personalizados en el cuerpo de un objeto de manera sencilla. Este componente es especialmente útil para agregar detalles esenciales o no esenciales al diseño del producto, como botones, controles u otros elementos que se puedan representar de la misma forma.

### **Instrucciones de Uso:**

#### **1. Configuración Inicial:**

- Asegúrate de haber definido la geometría base del objeto en el que deseas incorporar los botones.

#### **2. Configuración de Parámetros de Entrada:**

- Define los siguientes parámetros de entrada para el componente "Buttons":
  - **Brep (Brep):** Ingresa la geometría base (generalmente un brep) del objeto en el que deseas incorporar los botones.
  - **Point (Punto):** Indica el punto (subestructura) donde deseas colocar el botón.
  - **Profile (Perfil):** Selecciona el perfil previamente dibujado que define la forma y el estilo del botón. Este perfil puede ser cualquier forma geométrica que desees.
  - **Distance (Distancia):** Especifica la profundidad del botón en el objeto.
  - **Factor\_Scale (Factor de Escala):** Ajusta el factor de escala si deseas modificar el tamaño del botón.

#### **3. Ajuste de Profundidad y Tamaño:**

- Utiliza el componente "Buttons" para ajustar la profundidad y el tamaño del botón en función del diseño. El factor de escala te permite personalizar el tamaño del botón de manera eficiente.

#### **4. Diseño Personalizado:**

- Aprovecha la libertad creativa que ofrece el componente "Buttons" para diseñar botones con formas y estilos únicos que se adapten a los requerimientos específicos del diseño.

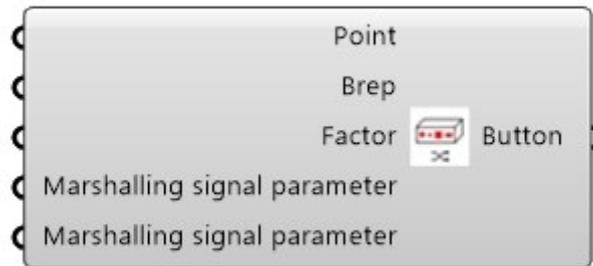
#### **5. Integración con Otros Componentes:**

- El componente "Buttons" se integra perfectamente con otras subestructuras generadas previamente, lo que garantiza una representación precisa y coherente de los botones en el objeto final. Asegúrate de que los botones se alineen y ajusten de manera coherente con el diseño general del objeto.

#### **Parámetros de Salida:**

- El componente proporcionará una salida:
  - "Button (Botón)": Representa el botón personalizado que has diseñado e incorporado en el cuerpo del objeto.

## B.8 Componente "Random button generator"



**Descripción:** El componente "Random Button Generator" es una herramienta versátil y poderosa diseñada para proporcionar una amplia diversidad de botones que pueden utilizarse como detalles esenciales en la morfología del objeto. Este componente es especialmente útil para agregar variedad y originalidad a los botones en tus diseños generativos.

### Instrucciones de Uso:

#### 1. Configuración Inicial:

- Asegúrate de haber definido la geometría base del objeto en el que deseas incorporar botones.

#### 2. Parámetros de Configuración:

- Define los siguientes parámetros de entrada para el componente:
  - **Point (Punto):** Indica el punto (subestructura) donde deseas colocar los botones.
  - **Brep (Brep):** Ingresa la geometría base (generalmente un brep) del objeto en el que deseas incorporar los botones.
  - **Factor (Factor):** Ajusta el nivel de variabilidad en la forma de los botones. Un valor mayor generará botones más diferentes y únicos, mientras que un valor menor mantendrá la forma más cercana a los perfiles establecidos.
  - **Variability (Variabilidad):** Selecciona la señal de variabilidad para controlar el nivel de aleatoriedad en la generación de botones.
  - **Button (Botón):** Selecciona la señal de botón para establecer los perfiles de botones de referencia.

#### 3. Generación de Botones:

- Utiliza el componente "Random Button Generator" para generar botones con formas básicas y sencillas. El componente ofrece cinco perfiles de

botones establecidos, cada uno con una apariencia distinta y atractiva. Con la opción de variabilidad, estas formas predefinidas brindan un abanico amplio de formas.

#### 4. **Ajuste de Variabilidad:**

- Ajusta el parámetro "Factor" para controlar el nivel de variabilidad en la forma de los botones. Un valor alto genera botones más diferentes, mientras que un valor bajo mantiene la forma cercana a los perfiles establecidos.

#### 5. **Diseño Creativo:**

- Experimenta con diferentes valores de "Factor" para lograr botones únicos y atractivos. La variabilidad en la forma de los botones es fundamental para el diseño generativo, ya que permite explorar una amplia gama de opciones de diseño y generar distintas soluciones.

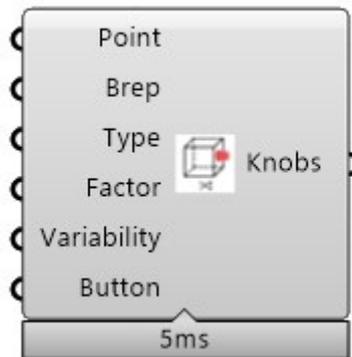
#### 6. **Integración con Otros Componentes:**

- Integra los botones generados por el componente "Random Button Generator" con otras subestructuras y elementos del diseño para obtener una representación coherente en el objeto final.

#### **Parámetros de Salida:**

- El componente proporcionará una salida:
  - "Button (Botón)": Representa los botones generados con formas distintivas y variadas.

## B.9 Componente "Random pommel generator"



**Descripción:** El componente "Random Pommel Generator" es una herramienta que complementa al "Random Button Generator" y está diseñada para crear perillas o pomos que se ajusten perfectamente al objeto bajo diseño. Este componente es interesante para agregar detalles funcionales y estéticos al diseño.

### **Instrucciones de Uso:**

#### **1. Configuración Inicial:**

- Asegúrate de haber definido la geometría base del objeto en el que deseas incorporar pomos.

#### **2. Parámetros de Configuración:**

- Define los siguientes parámetros de entrada para el componente:
  - **Point (Punto):** Indica el punto (subestructura) donde deseas colocar los pomos.
  - **Brep (Brep):** Ingresa la geometría base (generalmente un brep) del objeto en el que deseas incorporar los pomos.
  - **Type (Tipo):** Selecciona el tipo de pomo que deseas generar. El componente ofrece cinco perfiles de pomos establecidos, cada uno con una apariencia distintiva.
  - **Factor (Factor):** Ajusta el nivel de variabilidad en la forma de los pomos. Un valor mayor generará pomos más diferentes, mientras que un valor menor mantendrá la forma más cercana a los perfiles establecidos.
  - **Variability (Variabilidad):** Selecciona la señal de variabilidad para controlar el nivel de aleatoriedad en la generación de pomos.
  - **Button (Botón):** Marca esta opción para generar una nueva opción de solución, es decir una nueva perilla.

#### **3. Generación de Pomos:**

- Utiliza el componente "Random Pommel Generator" para generar pomos con formas básicas y sencillas. Puedes seleccionar el tipo de pomo que se adapte a las necesidades del diseño.

#### **4. Ajuste de Variabilidad:**

- Ajusta el parámetro "Factor" según las necesidades de diseño para lograr pomos con la cantidad deseada de variabilidad.

#### **5. Diseño Creativo:**

- Experimenta con diferentes tipos de pomos, valores de "Factor" y configuraciones para obtener una amplia variedad de opciones de diseño.

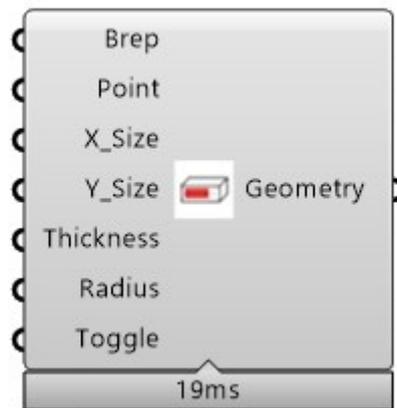
#### **6. Integración con Otros Componentes:**

- Integra los pomos generados por el componente "Random Pommel Generator" con otras subestructuras y elementos del diseño para obtener una representación coherente en el objeto final.

#### **Parámetros de Salida:**

- El componente proporcionará una salida:
  - "Knobs (Perillas o Pomos)": Representa los pomos generados con formas distintivas y variadas.

## B.10 Componente "Screen or void"



**Descripción:** El componente "Screen or Void" es una herramienta versátil que ofrece una doble funcionalidad en el diseño. Puede utilizarse para crear cubiertas o pantallas que recubran partes del objeto, así como para generar rebajes o vaciados en el cuerpo del objeto. Esta versatilidad permite al diseñador agregar detalles funcionales y estéticos de manera rápida y sencilla, enriqueciendo la morfología del objeto.

### Instrucciones de Uso:

#### 1. Configuración Inicial:

- Asegúrate de haber definido la geometría base del objeto en el que deseas agregar cubiertas o rebajes.

#### 2. Parámetros de Configuración:

- Define los siguientes parámetros de entrada para el componente:
  - **Brep (Brep):** Ingresa la geometría base (generalmente un brep) del objeto en el que deseas agregar cubiertas o rebajes.
  - **Point (Punto):** Indica el punto donde deseas colocar la cubierta o el rebaje.
  - **X\_Size (Tamaño en X):** Ajusta el ancho de la cubierta o rebaje en la dirección X.
  - **Y\_Size (Tamaño en Y):** Ajusta el largo de la cubierta o rebaje en la dirección Y.
  - **Thickness (Espesor):** Establece el grosor de la cubierta o rebaje.
  - **Radius (Radio):** Agrega redondeos en las esquinas para un acabado suave y estilizado.
  - **Toggle (Alternar):** Utiliza este componente botón para alternar entre la creación de cubiertas y rebajes.

#### 3. Generación de Cubiertas o Rebajes:

- Utiliza el componente "Screen or Void" para generar cubiertas o rebajes con las dimensiones, espesor y redondeos deseados.

#### 4. **Personalización:**

- Ajusta las dimensiones, espesor y redondeos según las necesidades de tu diseño.

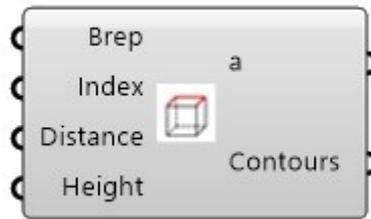
#### 5. **Diseño Creativo:**

- Experimenta con diferentes configuraciones para obtener cubiertas o rebajes con la apariencia y efecto deseados.

#### **Parámetros de Salida:**

- El componente proporcionará una salida:
  - "Geometry (Geometría)": Representa la cubierta o el rebaje generados según la configuración.

## B.11 Componente "Edges in contours"



**Descripción:** El componente "Edges in Contours" es una herramienta útil que permite al diseñador crear bordes en los contornos de un cuerpo geométrico de un objeto. Esta función realza y destaca áreas específicas del objeto, mejorando su definición y estilización. Puedes ajustar la forma, tamaño y profundidad de los bordes, lo que te brinda la flexibilidad necesaria para experimentar con diferentes estilos y estéticas.

### **Instrucciones de Uso:**

#### **1. Configuración Inicial:**

- Asegúrate de haber definido la geometría base del objeto en el que deseas crear bordes en los contornos.

#### **2. Parámetros de Configuración:**

- Define los siguientes parámetros de entrada para el componente:
  - **Brep (Brep):** Ingresa el objeto o geometría en la que deseas crear bordes en los contornos.
  - **Index (Índice):** Indica el índice del contorno dentro del objeto base donde deseas crear los bordes.
  - **Distance (Distancia):** Ajusta la distancia entre los bordes en el contorno. Esto controla la separación entre ellos.
  - **Height (Altura):** Establece la altura o profundidad de los bordes en el contorno. Esto controla la intensidad del efecto.

#### **3. Generación de Bordes en Contornos:**

- Utiliza el componente "Edges in Contours" para crear bordes en el contorno seleccionado. Estos bordes realzarán y destacarán el área específica del objeto.

#### **4. Personalización:**

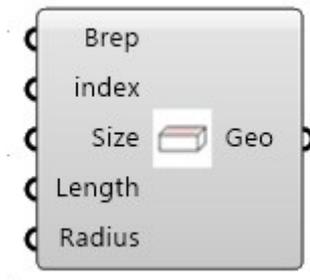
- Experimenta con diferentes configuraciones, como la distancia entre bordes y la altura, para lograr el efecto deseado en tu diseño.

### **Parámetros de Salida:**

- El componente proporcionará dos salidas:

- "a (Contorno)": Representa los bordes en los contornos del objeto base.
- "Contours (Contornos)": Representa los bordes espejo del contorno "a" en los contornos del objeto base.

## B.12 Componente "Surface pattern"



**Descripción:** El componente "Surface Pattern" es una herramienta que se utiliza para crear patrones homogéneos en una superficie seleccionada del objeto en el diseño generativo. Estos patrones pueden consistir en cuadrados, cuadrados con esquinas redondeadas o círculos, lo que brinda al diseñador la flexibilidad de elegir el estilo que mejor se adapte a su diseño.

### Instrucciones de Uso:

#### 1. Configuración Inicial:

- Asegúrate de haber definido la geometría base del objeto en el que deseas aplicar un patrón en la superficie.

#### 2. Personalización del Patrón:

- Configura el patrón a tu gusto utilizando los siguientes parámetros:
  - **Brep (Brep):** Ingresa el objeto o geometría en el que deseas crear un patrón en la superficie.
  - **Index (Índice):** Indica el índice de la superficie donde deseas aplicar el patrón.
  - **Size (Tamaño):** Ajusta el tamaño de las geometrías en el patrón.
  - **Length (Longitud):** Define la profundidad en el cuerpo del objeto.
  - **Radius (Radio):** Si estás utilizando círculos en tu patrón, establece el radio deseado.

#### 3. Generación del Patrón:

- Utiliza el componente "Surface Pattern" para crear el patrón en la superficie seleccionada del objeto. El patrón mejorará la estética y apariencia del objeto.

### Parámetros de Salida:

- El componente proporcionará una salida llamada "Geo (Geometría)" que representará el patrón generado en la superficie seleccionada.

**Nota Importante:** Es importante destacar que el componente "Surface Pattern" puede requerir una cantidad considerable de recursos del ordenador debido a la complejidad de los procesos involucrados en la generación de los patrones. Hay que tener en cuenta que la optimización de este componente se ha dejado para trabajos futuros, con el objetivo de mejorar su rendimiento y eficiencia en el uso de recursos.