



Facultad de Arquitectura e Ingeniería Civil

**Implementación de un algoritmo de optimización multi -  
objeto de una red de barras utilizando un lenguaje de  
programación visual en “Rhinceros” y sus extensiones.**

Autor: David Alejandro Muñoz Avila

Tutor: Ing. Hugo Marcelo Otáñez Gómez.



Quito, 2021

## DECLARACIÓN JURAMENTADA

Yo, DAVID ALEJANDRO MUÑOZ AVILA, con cédula de ciudadanía número 172373462-8, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido previamente presentado para ningún grado a calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.



DAVID ALEJANDRO MUÑOZ AVILA

C.C. 172373462-8

## **DECLARATORIA**

El presente Trabajo de Titulación titulado:

**“ Implementación de un algoritmo de optimización multi - objeto de una red de barras utilizando un lenguaje de programación visual en “Rhinceros” y sus extensiones. ”**

Realizado por:

**DAVID ALEJANDRO MUÑOZ AVILA**

Como requisito para la obtención del Título de:

**INGENIERO CIVIL**

Ha sido dirigido por el profesor

**ING. HUGO MARCELO OTÁÑEZ GÓMEZ, MSC.**

Quien considera que constituye un trabajo original de su autor.

Ing. Hugo Marcelo Otáñez Gómez.

**TUTOR**

## DECLARATORIA DE PROFESORES INFORMANTES

Los profesores informantes:

Ing. Diego Xavier Jara Almeida, Msc, MBA.

Ing. Luis Soria Núñez, MsC.

Después de revisar el trabajo presentado,  
Lo han calificado como apto para su defensa oral ante el tribunal  
examinador

Ing. Luis Soria Núñez, MsC.

Ing. Diego Xavier Jara Almeida, Msc, MBA.

## DEDICATORIA

Dedicado a Dios, a mis padres que con mucho esfuerzo y confianza depositaron en mi persona e invirtieron en mi educacion. A mi hermano por sembrar los cimientos de esta profesion.

## **AGRADECIMIENTO**

En primer lugar agradezco a Dios y a mis padres por motivarme a siempre ser mejor durante el transcurso de mi vida estudiantil y durante mi trayectoria a lo largo de mi vida.

Agradezco a mi tutor el Ing. Marcelo Hugo Otañez Gomez por su apoyo durante el desarrollo de este trabajo de titulación.

A todos los docentes que compartieron sus conocimientos conmigo a lo largo de mi carrera universitaria.

A la Universidad Internacional SEK por brindarme la oportunidad de culminar mi carrera universitaria.

## RESUMEN

Las civilizaciones, a medida que el tiempo transcurre y los avances tecnológicos aumentan a pasos agigantados, la construcción y los diseños arquitectónicos han ido acoplándose a las necesidades requeridas. Optimizar los elementos estructurales se ha ido convirtiendo cada día en un tema de alta relevancia para generar estructuras geométricas no convencionales o de características orgánicas, los lenguajes de programación cada vez son más maleables y didácticos llegando al punto en el que no es necesario ser un experto programador para definir un algoritmo mediante un software de programación visual, dicho esto y mediante las necesidades que exigen las estructuras, en la actualidad se pretende dar una solución estructural a una geometría no convencional que conforme una estructura paramétrica y optimizar el material usado en sus elementos, todo esto será aplicado mediante un algoritmo definido en un software de programación visual.

Existen varios métodos aplicados al análisis estructural mediante softwares definidos los cuales se encargan del diseño estructural, mismos que definen estructuras bajo parámetros en geometrías simples sin mayor grado de complejidad y de una forma rápida, en la actualidad existen nuevas tentativas de análisis estructural por medio del análisis de elementos finitos por medio de nuevos softwares que permiten analizar estructuras con geometrías no convencionales o paramétricas con mayor facilidad y rapidez, estas herramientas han ido tomado fuerza en las últimas décadas gracias a los avances tecnológicos que existen en la actualidad y que permiten a los ingenieros brindar soluciones estructurales y de optimización a geometrías paramétricas.

Herramientas tales como lo son Rhinoceros y su módulo Grasshopper son de gran utilidad en los diseños paramétricos, en adaptar soluciones estructurales y en optimizar las mismas de ser este el caso.

Rhinoceros es un lenguaje de programación visual que por medio de su módulo Grasshopper se puede llegar a obtener una variedad de diseños arquitectónicos y a su vez por medio de algoritmos que permite el software se logran brindar soluciones estructurales y de optimización en tanto a la estructura se refiera.

Las estructuras arquitectónicas siempre han sido y serán un hito que marcará una época que define el avance de las civilizaciones, a lo largo de la historia se ha identificado el nivel tecnológico de una civilización gracias a su arquitectura.

**PALABRAS CLAVE (3)** Algoritmo, optimización, diseño paramétrico.

## **ABSTRACT**

Civilizations as time passes and technological advances increase by leaps and bounds, construction and architectural designs have been adapting to the required needs. Optimizing structural elements has become a highly relevant topic every day to generate unconventional geometric structures or organic characteristics, programming languages are increasingly malleable and didactic, reaching the point where it is not necessary to be an expert. programmer to define an algorithm by means of visual programming software, having said this and by means of the needs demanded by the structures, at present it is intended to provide a structural solution to an unconventional geometry that conforms to a parametric structure and to optimize the material used in its elements, all this will be applied by means of an algorithm defined in a visual programming software.

There are several methods applied to structural analysis using defined softwares, which are in charge of structural design, which define structures under parameters in simple geometries without a greater degree of complexity and in a fast way, at present there are new attempts at structural analysis by means of the finite element analysis by means of new software that allows to analyse structures with unconventional or parametric geometries with greater ease and speed, these tools have gained strength in recent decades thanks to the technological advances that exist today and that allow engineers provide structural and optimization solutions to parametric geometries.

Tools such as Rhinoceros and its Grasehopper module are very useful in parametric designs, in adapting structural solutions and in optimizing them if this is the case.

Rhinoceros is a visual programming language that through its Grasshopper module can obtain a variety of architectural designs and in turn, through algorithms that the software allows, it is possible to provide structural and optimization solutions as far as the structure is concerned.

Architectural structures have always been and will be a milestone that will mark an era that defines the advancement of civilizations, throughout history the technological level of a civilization has been identified thanks to its architecture.

**KEY WORDS (3)** Algorithms, optimization, parametric design.

## CONTENTS

1.	Capítulo 1.....	7
1.1.	Introducción.....	7
1.2.	Justificación.....	7
1.3.	Hipótesis .....	8
1.4.	Antecedentes .....	8
1.5.	Alcance .....	8
1.6.	Objetivo principal.....	9
1.7.	Objetivos secundarios.....	9
1.8.	Limitaciones .....	9
1.9.	Planteamiento del problema.....	9
1.10.	Variables independientes.....	10
1.11.	Variable dependiente.....	11
1.11.1.	Las geometrías paramétricas .....	11
1.11.2.	Método de análisis de la estructura.....	11
1.12.	Relación entre Variables.....	11
1.13.	Metodología.....	12
2.	CAPITULO II (Fundamento Teórico).....	13
2.1.	Parámetro .....	13
2.2.	Diseño paramétrico.....	13
2.3.	Objetivos diseño paramétrico.....	17
2.3.1.	Diseño .....	17
2.3.2.	Relación entre variables/parámetros.....	17
2.3.3.	Resultados paramétricos y /o responsivos.....	17
2.4.	Parametrización: .....	17
2.4.1.	Ejemplos de parametrización: .....	18
2.5.	Buttom-up.....	23
2.6.	Algoritmo.....	28
2.6.1.	Características de los algoritmos .....	29
2.6.2.	Técnicas de representación .....	29
2.6.3.	Lenguaje de Programación Visual.....	29
2.7.	Análisis Estructural.....	31
2.7.1.	Elementos Finitos.....	31

2.8. Optimización estructural.....	37
2.8.1. Introducción.....	37
2.8.2. Definición.....	38
2.8.3. Tipos de optimización.....	39
2.8.4. Algoritmos Evolutivos genéticos.....	41
2.8.5. Aplicaciones.....	42
3. CAPITULO 3 (Herramientas aplicadas al Diseño Paramétrico y al Análisis Estructural).....	44
3.1. Rhinoceros 3D.....	44
3.2. Grasshopper.....	46
3.3. Kiwi!3D.....	48
3.4. Octopus.....	50
3.4.1. Parámetros de genes.....	52
3.4.2. Mapeo o modelado paramétrico.....	53
3.4.3. Objetivos y genotipos.....	53
3.5. Pufferfish.....	55
4. CAPITULO 4 (Caso de estudio).....	56
5. CALCULOS.....	57
5.1. Parámetros de modelación.....	57
5.2. Definición del algoritmo usando Kiwi3D.....	59
5.3. Definición del algoritmo usando Pufferfish.....	69
5.4. Definición del algoritmo y optimización de la geometría en estudio aplicando Octopus.....	75
6. Conclusiones.....	83
7. References.....	87
Ilustración 1, Metropol Parasol(Sevilla); Foto de Fernando Alda; Diseño de J. Mayer. H. Architects.....	15
Ilustración 2, Citado de (RhinoVAULT).....	16
Ilustración 3, McNeel.....	19
Ilustración 4, McNeel.....	20
Ilustración 5, Tambir Sarkar.....	21

Ilustración 6, Zaha Hadid .....	22
Ilustración 7, (Administración de negocios, 2016).....	23
Ilustración 8. Ejemplo de algoritmo de evolución genética. ....	25
Ilustración 9, Padre + Madre = probabilidad del color de ojos resultante.....	26
Ilustración 10, adaptación de la naturaleza para sobrevivir. ....	27
Ilustración 11, evolución adaptativa de pinzones de Darwin, (hongos, 2010).....	28
Ilustración 12, fuente propia.....	31
Ilustración 13, Giuseppe Miralisenna, simulación del análisis de elementos finitos....	33
Ilustración 14, coordenadas nodales (i,j,k) y desplazamientos de los nodos, Valero (2004).....	34
Ilustración 15, Liu y Tovar, 2014 .....	42
Ilustración 16. Logotipo de Rhinoceros 3D. Citado de (McNeel & Associates, 2019)44	
Ilustración 17. Ejemplo de uso de NURBS. Citado de (McNeel, 2019).....	45
Ilustración 18. Estructura generada en Rhinoceros 3D. Fuente propia.....	46
Ilustración 19. logotipo de Grasshopper. Citado (McNeel & Associates, 2019) .....	46
Ilustración 20. Interfaz de Grasshopper. Citado de (Gil & Parsons, 2014).....	47
Ilustración 21. Demostración del uso de un slider en grasshopper, fuente propia ....	48
Ilustración 22. logotipo de Kiwi!3d. Citado de (one!3d).....	49
Ilustración 23. logotipo de Octopus. Citado de (Vierlinger,2012). ....	50
Ilustración 24. Interfaz de Octopus. Citado de (Vierlinger, 2012).....	51
Ilustración 25 parametros o genes.....	52
Ilustración 26 mapeo o modelado parametrico .....	53
Ilustración 27 fenotipos y objetivos .....	54
Ilustración 28 interfaz octopus .....	54
Ilustración 29 logo pufferfish .....	55
Ilustración 30 modelo arquitectonico (rhinoceros).....	58
Ilustración 31 model.....	59
Ilustración 32 IGAsolver .....	59
Ilustración 33 Toggle.....	60
Ilustración 34 modelo de analsis.....	60
Ilustración 35 tipo de analisis .....	61

Ilustración 36 conexión IGAsolver .....	61
Ilustración 37 tipo de elemento estructural.....	61
Ilustración 38 Parametros del elemento estructural .....	62
Ilustración 39 conecciones de parametros del elemento estructural.....	62
Ilustración 40 control de diametros del perfil tubular .....	63
Ilustración 41 parametros de modelado del elemento estructural.....	63
Ilustración 42 parametros de apoyo .....	64
Ilustración 43 tipo de apoyo .....	64
Ilustración 44 definición de parametros en el apoyo .....	65
Ilustración 45 tipo de carga .....	65
Ilustración 46 Definición de parametros del tipo de carga .....	66
Ilustración 47 Parametros del resultado de IGAsolver .....	66
Ilustración 48 Tipo de esfuerzos que puede analizar IGAsolver .....	67
Ilustración 49 Algoritmo Kiwi3D .....	68
Ilustración 50 Resultados Kiwi3D.....	68
Ilustración 51 Pipe Mesh.....	69
Ilustración 52 Deconstruct Mesh.....	70
Ilustración 53 Construct Mesh.....	70
Ilustración 54 Organización de resultados .....	71
Ilustración 55 PufferFish .....	71
Ilustración 56 Mesh Join .....	72
Ilustración 57 Resultados PufferFish y Kiwi3D.....	72
Ilustración 58 Definición de 450 puntos de forma aleatoria .....	73
Ilustración 59 Selección de puntos que se sobreponen .....	74
Ilustración 60 Algoritmo mediante GrassHopper.....	74
Ilustración 61 Resultados graficos de Octopus .....	76
Ilustración 62 Rusultados Octopus .....	77
Ilustración 63 Resultados Octopus .....	77
Ilustración 64 Resultados Octopus .....	78
Ilustración 65 Interfaz Octopus .....	78
Ilustración 66 Resultados graficos Octopus en Rhinoceros.....	79

Ilustración 67 Resultado grafico Octopus .....	79
Ilustración 68 Resultado grafico Octopus .....	80
Ilustración 69 Resultado optimizado Otopus vista superior .....	81
Ilustración 70 Resultado optimizado Otopus vista lateral.....	81
Ilustración 71 Resultado optimizado Otopus vista superior .....	82
Ilustración 72, esquema grafico del algoritmo.....	83

# 1. CAPITULO 1

## 1.1. INTRODUCCIÓN

Las estructuras paramétricas, la optimización de materiales, y la aplicación de herramientas tecnológicas en el ámbito de diseño estructural, han sido un tema de gran relevancia para la construcción en la actualidad, existen varios métodos para dar soluciones estructurales y de optimización a las estructuras mencionadas, en esta tesis se dará una solución estructural mediante un sistema de barras que serán creadas por medio de un algoritmo, este será aplicado por medio de las herramientas digitales de programación visual, que a su vez se encargara de optimizar el material en sus elementos estructurales.

El método para seguir será por medio de un lenguaje de programación visual, se ira creando el algoritmo que definirá los parámetros del diseño estructural y el alcance de optimización según se lo solicite. El paquete de programas que incluye a Rhinoceros, por medio de su módulo Grasshopper, serán las herramientas primordiales para brindar una solución a nuestra estructura impuesta, del mismo modo se hará uso de extensiones compatibles con Grasshopper para realizar los cálculos de optimización y análisis estructural, obteniendo de esta manera como producto final un sistema de barras optimizadas, que brinden una solución estructural a nuestra estructura paramétrica.

## 1.2. JUSTIFICACIÓN

Debido a que la ingeniería y las ciencias de la construcción se han ido adaptando a las nuevas tecnologías y a las necesidades de optimizar el trabajo en varios aspectos del diseño estructural y temas de diseños digitalizados.

Se puede decir que mediante la aplicación de nuevas herramientas tecnológicas de la ingeniería y de la programación visual, se puede crear un algoritmo, que dé solución estructural, a una geometría parametrizada, que puede servir al profesional de la construcción como una herramienta de diseño estructural.

### **1.3. HIPÓTESIS**

Con la aplicación de este algoritmo se desprenderá, como el uso un lenguaje de programación visual en conjunto con el paquete de programas Rhinoceros, Grasshopper, kiwi3D, Pufferfish y Octopus, puede servir a los profesionales de la construcción y a la ingeniería, a solucionar problemas de diseño estructural, de una forma gráfica.

### **1.4. ANTECEDENTES.**

La implementación de algoritmos para la optimización de elementos estructurales, ha ido tomando fuerza en la última década, llevando de esta forma a la ingeniería a un nivel más profundo en el uso de programas que permiten optimizar elementos estructurales, y a su vez aplicar estos conceptos para formar estructuras con geometrías no convencionales, estructuras con formas orgánicas o simplemente el reducir el peso de la estructura.

A medida que la arquitectura va tomando formas vanguardistas, por demanda de la sociedad, las estructuras paramétricas, estructuras con geometrías no convencionales, u aproximándonos a geometrías orgánicas, los arquitectos se han visto en la necesidad de crear mencionadas estructuras, las cuales llegan con la necesidad, de, soluciones de ingeniería aplicada a programas con la capacidad de optimizar y responder a las necesidades de dichas estructuras, mencionado esto se llegó a plantear como una tesis que genere interés en la aplicación de estos métodos, en la que se pueda demostrar como un algoritmo aplicado a un lenguaje de programación visual logra generar una solución estructural a una geometría parametrizada y optimizar una estructura, de esta forma se pretende llegar a generar interés en nuestra sociedad como ya lo es en países más desarrollados.

### **1.5. ALCANCE.**

Obtener un algoritmo que pueda ser ocupado en el paquete de programas Rhinoceros, Grasshopper, kiwi3D, Pufferfish y Octopus. con el fin de obtener proceso que defina un sistema estructural optimizado, mediante un sistema de barras, que dará una solución estructural a una geometría no convencional o paramétrica, que sirva como herramienta para los profesionales de la construcción.

## **1.6. OBJETIVO PRINCIPAL**

Formular un algoritmo que sea aplicable en los programas Rhinoceros, Grasshopper, kiwi3D, Pufferfish y Octopus. El cual pueda brindar una solución estructural a una geometría paramétrica, y a su vez dar resultados gráficos, en el cual se pueda identificar de forma visual el comportamiento mecánico de las mismas, de tal modo este será optimizado tanto en la cantidad y longitud de barras a ocupar en una estructura de acero.

Demostrar la aplicabilidad de los programas Rhinoceros, Grasshopper, kiwi3D, Pufferfish y Octopus. pueden brindar una solución estructural a una geometría paramétrica, a su vez brindar resultados optimizados en tanto a la cantidad de material.

## **1.7. OBJETIVOS SECUNDARIOS**

- Crear un algoritmo que pueda ser usado en los programas antes mencionados.
- Demostrar cómo funcionan los programas con el algoritmo creado.
- Hacer un ejemplo ilustrativo que permita detallar el proceso, y la forma en el que, el conjunto de estos programas en corroboración del uno con el otro, dan soluciones.

## **1.8. LIMITACIONES**

Los elementos que no se abordaran en nuestro estudio principalmente es el sistema constructivo, conexiones, tipos de cimentación, cargas sísmicas ya que todo está aplicado al euro código, tampoco se abordaran temas de detalles constructivos o similares. El proyecto estará direccionado precisamente a demostrar el uso de herramientas de un software de programación visual aplicado directamente al análisis estructural y a la optimización de sus elementos estructurales hasta donde la estructura así lo permita.

## **1.9. PLANTEAMIENTO DEL PROBLEMA.**

En tanto la arquitectura use formas paramétricas y formas geométricas no convencionales en sus diseños, la ingeniería se ve en la necesidad y obligación de dar soluciones estructurales para satisfacer la necesidad de materialización de estos modelos arquitectónicos.

En la actualidad existen varios paquetes para el desarrollo de programación visual, en los cuales se destacan, los programas antes mencionados, ya que los mismos son maleables con el operador al momento de ingresar algoritmos. Lo que ha permitido la optimización de los trabajos en ingeniería y profesiones afines, llegando de este modo a ser utilizadas en estructuras de grandes y pequeñas dimensiones. Es por eso que es menester para esta investigación, incentivar al uso de nuevas herramientas tecnológicas, siendo esto así, se intentara contestar la siguiente pregunta, con el fin de enseñar un proceso que puede servir como herramienta para un profesional de la construcción.

¿Cómo la aplicación de un algoritmo de diseño estructural, creado a partir de un lenguaje de programación visual, puede ser una herramienta de trabajo para el diseño de estructuras?

### **1.10. VARIABLES INDEPENDIENTES.**

- para el caso en estudio la variable independiente es la geometría paramétrica y el método de análisis de la estructura.

- Las geometrías paramétricas.

usualmente son impuestas por los arquitectos diseñadores, mismas que deberán ser construidas bajo un sistema que brinde confort, seguridad y sea sustentable, es entonces donde entra el criterio profesional del diseñador estructural brindando una solución estructural a dicha geometría, por medio de programas de programación visual existentes, en la actualidad se ha facilitado el proceso para brindar varias soluciones estructurales a dichas geometrías con la capacidad de optimizar el material a utilizar.

- Método de análisis de la estructura

Para el método de análisis de la estructura se utilizará un paquete de programas que aplica el concepto de análisis de elementos finitos, de este modo nos recomendará como resultado la mejor opción según la geometría del modelo arquitectónico, dejando finalmente a criterio del ingeniero estructural la elección final para el diseño.

## **1.11. VARIABLE DEPENDIENTE.**

- Algoritmo a programar

Los algoritmos son una herramienta básica, pero de gran ayuda para sistematizar una solución, de este modo por medio de ellos podemos definir parámetros en los cuales sean de gran aporte para la solución de los problemas que se presentan.

Se pretende crear un algoritmo de optimización multi - objeto aplicado a una geometría paramétrica que permita brindar una solución estructural, por medio de un sistema de vigas creada por el algoritmo el software realizara un análisis de las posibles soluciones que existen para dicha geometría.

### **1.11.1. LAS GEOMETRÍAS PARAMÉTRICAS.**

Usualmente son impuestas por los arquitectos diseñadores, mismas que deberán ser construidas bajo un sistema que brinde confort, seguridad y sea sustentable, es entonces donde entra el criterio profesional del diseñador estructural brindando una solución estructural a dicha geometría, por medio de softwares de programación visual existentes en la actualidad se a facilitado el proceso de brindar varias soluciones estructurales a dichas geometrías con la capacidad de optimizar el material a utilizar.

### **1.11.2. MÉTODO DE ANÁLISIS DE LA ESTRUCTURA**

Para el método de análisis de la estructura se utilizará un programa que aplica el concepto de análisis de elementos finitos, de este modo se recomendará como resultado la mejor opción según la geometría del modelo arquitectónico, dejando finalmente a criterio del ingeniero estructural la elección final para el diseño.

## **1.12. RELACIÓN ENTRE VARIABLES.**

El desarrollo de programas aplicados a la construcción y su aplicación ha ido creciendo de forma paulatina en el Ecuador y en el mundo, ya sea en la parte de diseños estructurales o diseños arquitectónicos. En virtud a esto se pretende hacer un algoritmo de optimización multi - objeto aplicado a una geometría paramétrica que permita brindar una solución estructural, por medio de un sistema de vigas creada por el algoritmo antes

mencionado, en uso de los programas ya dichos con anterioridad, en donde se realizará un análisis de las posibles soluciones que existen para dicha geometría.

Es por esto que al ocupar estas variables se busca describir como un algoritmo puede crear y optimizar los elementos que conformaran una estructura definida por una geometría paramétrica.

### **1.13. METODOLOGÍA.**

Para el siguiente trabajo, se ocupará una metodología mixta, en la cual, con un método descriptivo y exploratorio, se demostrará que la aplicación de un algoritmo por medio de un paquete de programas y módulos aplicables a un lenguaje de programación visual, se pueden realizar procesos que lleven a soluciones, mismas que se pueden llegar a través de cálculo manual y tradicional. Pero en cambio la propuesta del trabajo es hacer énfasis en la aplicación de nuevas tecnologías, como herramientas que faciliten el trabajo a los profesionales de la construcción.

Es por esto que, mediante el uso de un paquete de programas de programación visual se pretende llegar a crear un algoritmo que cumpla con la función de implementar un sistema de vigas y genere soluciones estructurales aplicadas a una geometría no convencional, mediante el mismo algoritmo se pretende llegar a optimizar el sistema de vigas aplicadas sobre la geometría impuesta por el diseñador o arquitecto, de este modo se analizaran los resultados estructurales gráficos que nos darán los programas y de esta manera se elegirá el más óptimo según criterios de ingeniería. Hay que recalcar que es un estudio demostrativo, descriptivo, mas no un estudio comparativo, que busque lo correcto o lo incorrecto, más bien el trabajo busca describir otro modo de llegar a soluciones, que pueden ser demorosas, al ser realizadas de forma manual.

## **2. CAPITULO II (FUNDAMENTO TEÓRICO).**

### **2.1. PARÁMETRO.**

Con la capacidad de aplicar límites, rangos de valores y configuraciones específicas. Dentro de un mismo modelo se puede encontrar varios resultados según los parámetros que los delimiten. “Un parámetro es un dato preciso definido para analizar o valorar una situación” (Navarrete, 2014).

El término parámetro es medible y tiene como particularidad describir límites o restricciones. Por lo tanto, es tangible que delimita un sistema (Hudson, 2010).

### **2.2. DISEÑO PARAMÉTRICO.**

El término “paramétrico” se refiere a las relaciones entre todos los elementos del modelo que permiten la coordinación y la gestión de cambios, esta relación la crea el usuario con su trabajo. (Autodesk, 2019)

El diseño paramétrico, es definido como un método de modelación gráfica en forma digitalizada mediante el cual se puede generar diversas formas geométricas, partiendo desde un grupo de elementos tales como: parámetros, operaciones, funciones, relaciones geométricas, entre otros. El principal objetivo del diseño paramétrico es la generación de modelos que se adapten a variaciones numéricas de cada parámetro previamente definido, de este modo, cuando el valor numérico varía, cambian también los resultados, obteniendo de este modo varios en el mismo proceso de diseño.

Un diseño paramétrico es generado a partir de la definición de parámetros, mediante algoritmos se ingresan condiciones de diseño, las condiciones varían en forma numérica permitiendo que el modelo pueda ser modificado cuantas veces el usuario crea conveniente hasta obtener el resultado esperado.

Lo que caracteriza a un diseño paramétrico es la cualidad de ser responsivo, lo cual quiere decir que está en la capacidad de generar varias respuestas a partir del diseño gráfico original.

Según la RAE, diseño se define como: traza o delineación de un edificio o una figura.

El termino paramétrico, el cual se origina en las matemáticas y hace referencia a un valor o medida que es directamente dependiente del mismo, generalmente es representado como una variable manipulable (Gronda, 2011).

Se puede considerar como diseño paramétrico a las utilidades operacionales que se aplican en el desarrollo creativo aplicados en la geometría (WOODBURRY, 2010). Se han sugerido varias estrategias aplicadas al diseño integrado entre ingeniería y la arquitectura para el diseño y solución estructural de geometrías complejas (RAPPAPORT, 2010).

El diseño paramétrico principalmente es aplicado en los diseños arquitectónicos, se aplican recursos computacionales como son los lenguajes de programación visual medio al cual están direccionadas a geometrías y/o al análisis técnico, en proyectos de grandes y pequeñas escalas o experimentales (MEREDITH, 2008).

El diseño paramétrico abarca desde la implementación de curvas hasta la estructuración de edificaciones, en la actualidad se utiliza lenguajes de programación visual o grafica y/o programas de análisis según sea el caso. De tal modo, el diseño paramétrico estaría definido por la vinculación que existe entre los aspectos formales del proyecto, que se pueden definir durante el desarrollo (WOODBURRY, 2010), por lo que se implican nuevos recursos y actividades de diseño arquitectónico.



ZILUSTRACIÓN 1, METROPOL PARASOL(SEVILLA); FOTO DE FERNANDO ALDA;  
DISEÑO DE J. MAYER. H. ARCHITECTS

Los diseños paramétricos requieren una inversión de tiempo superior comparado con un no parametrizado (Muttio, E., Botello, s., & Tapia, M., 2017).

Una de las ventajas que tiene el diseño paramétrico, es la que está en busca de cierta forma la optimización de las obras civiles.

La optimización de la forma aplicado a la fachada con el fin de mejorar es uno de los ejemplos ilustrativos existentes tales como: La iluminación, la ventilación natural o distribución interna de una edificación.

Una de las principales ventajas del modelado paramétrico es la maleabilidad al momento de generar estructuras complejas, aplicando funciones matemáticas como los parámetros, la ilustración 2 muestra un claro ejemplo del diseño paramétrica, este modelo permite mostrar como una estructura está definida en su etapa final, pero en esta etapa de diseño bien se podrían cambiar las formas con tan solo modificar los parámetros ya definidos previamente mediante un algoritmo de diseño gráfico.

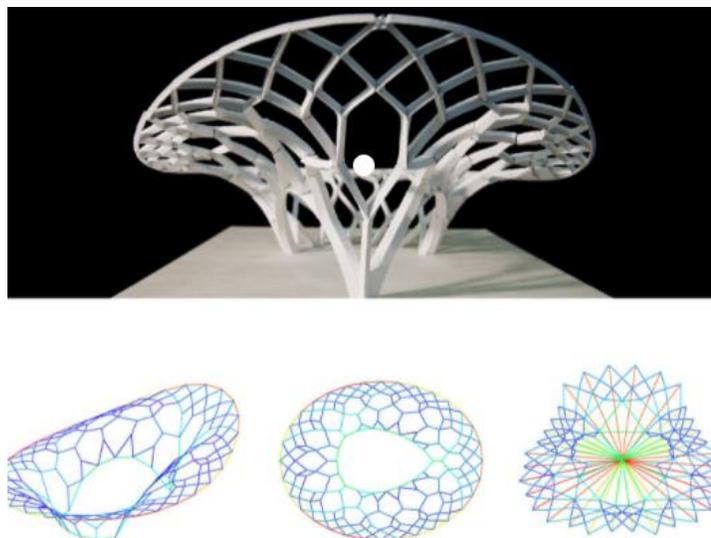


ILUSTRACIÓN 2, CITADO DE (RHINOVAULT).

Los modelos paramétricos usualmente son sometidos de forma rápida y eficiente en experimentos virtuales, con el fin de analizar rendimientos de acuerdo con indicadores de actuaciones previamente definidas.

Una de las principales razones por las que el diseño no parametrizado, en geometrías complejas, tendrán variables limitantes al instante de explorar nuevas posibilidades de modificaciones geométricas, es la falta de parámetros que puedan modificar estas características en forma cuantitativa. El diseño paramétrico pretende resolver las limitaciones mediante algoritmos definidos.

También se puede notar que a través de algoritmos se obtiene alternativas al instante de realizar cambios en el diseño, es capaz de definir la ruta más eficiente, es decir que, crea soluciones que definen las condiciones de forma específica al contexto de diseño como: (eficiencia del uso del material, capacidad portante del material, costos).

## **2.3. OBJETIVOS DISEÑO PARAMÉTRICO.**

### **2.3.1. DISEÑO**

Es un proceso más no un resultado único: al momento de diseñar, existe un desarrollo matemático y geométrico, creando un sistema (algoritmo), el cual permite explorar más de un resultado, manteniendo las premisas de diseño establecidas inicialmente.

### **2.3.2. RELACIÓN ENTRE VARIABLES/PARÁMETROS**

Una vez que se tiene definido un proceso de diseño, mas no, una forma preestablecida, pueden manipularse sus variables y propiedades, mismas que pueden ser modificadas en tiempo real brindando de este modo los resultados, para ser comparados según lo necesite el diseñador, con la finalidad de obtener un producto eficiente.

### **2.3.3. RESULTADOS PARAMÉTRICOS Y /O RESPONSIVOS**

A partir de un diseño paramétrico, existe la posibilidad de generar un diseño inteligente y/o responsivo, estableciendo de este modo un diseño con criterio y que permita explorar formas, adaptándolo de este modo a cualquier situación o contexto técnico. Es decir, se puede adaptar a cualquier parámetro o variable que haya sido integrada en el proceso de diseño, obteniendo de este modo un resultado inteligente y responsivo que pretenderá satisfacer un problema específico.

## **2.4. PARAMETRIZACIÓN:**

Según la Oxford: parametrización es, la acción de dividir o estudiar una forma mediante parámetros.

Se entiende como parametrización, al proceso de relacionar y estudiar formas geométricamente mediante herramientas digitales. Esto da lugar al concepto de arquitectura paramétrica que se basa en:

- El uso de herramientas digitales.
- Creación de modelos innovadores de orden complejo.

- Creación de modelos distintos, basado en la correlación de todos los parámetros

El modelado paramétrico está directamente relacionado con la parametrización, mediante el cual, el diseño arquitectónico en conjunto con el diseño digital produce como resultado geometrías más complejas para las propuestas de edificaciones (MEREDITH, 2008).

#### **2.4.1. EJEMPLOS DE PARAMETRIZACIÓN:**

##### ***2.4.1.1. El folding:***

Se define como la técnica del folding, el diseño para plegarse en un tamaño más pequeño para el almacenamiento o transporte. (press, 2021)

Por medio de este ejemplo se puede apreciar como la técnica del folding de forma responsiva logra reaccionar a factores externos de temperatura, los cual produce la apertura de las placas a medida que la temperatura aumenta en el interior de la estructura. El sistema responde a impulsos térmicos.

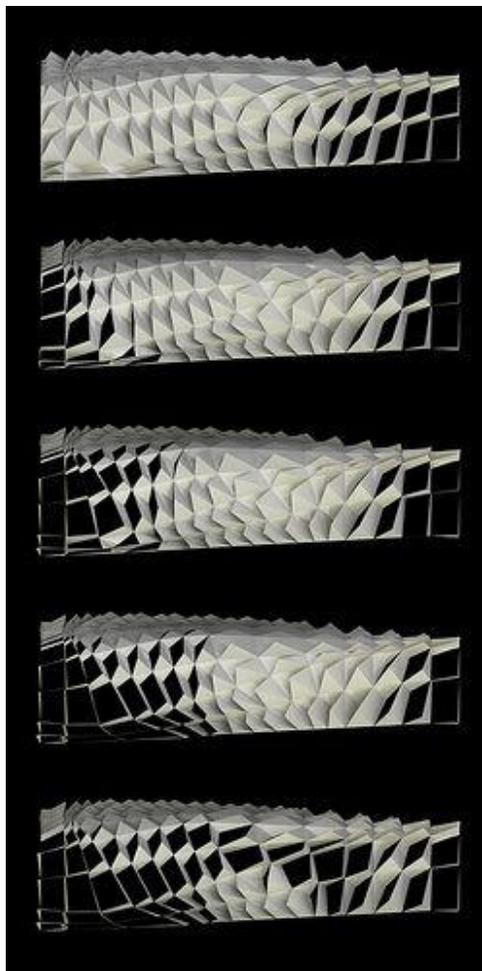


ILUSTRACIÓN 3, MCNEEL.

#### **2.4.1.2. *Teselado:***

El teselado y teselaciones, hacen referencia a una regularidad o patrón de figuras que cubren o pavimentan por completo a una superficie que cumple con dos requisitos, (Muñoz, 2015):

1. Que no queden espacios.
2. Que no se superpongan las figuras.

En la ilustración 4, se puede observar un teselado triangular sobre una malla de superficie compleja, geometría definida a partir del concepto de teselado con división poligonal.

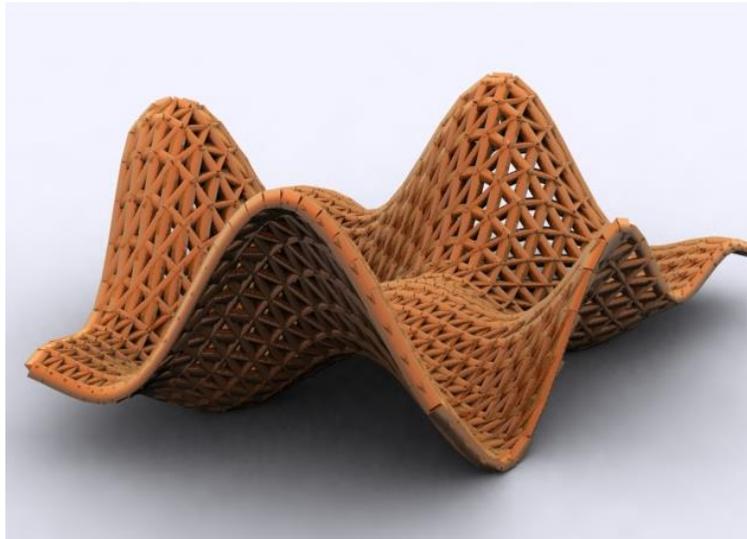


ILUSTRACIÓN 4, McNEEL

#### **2.4.1.3. Sistema Waffle:**

Un sistema de construcción tipo waffle, se caracteriza por tener un refuerzo bidireccional en el exterior del material, de este modo dándole la forma de recipientes cúbicos así generando un waffle. Este tipo de diseños los podemos encontrar en estructuras de hormigón, acero y madera. Una losa que es construida con una geometría tipo waffle le da una estabilidad estructural significativamente mayor y a su vez optimiza la cantidad de material a usar (Arellano-Mendez, 2017).

Como se puede observar en la ilustración 5, este sistema, se puede aplicar en losas y columnas, de este modo los esfuerzos se distribuyen de forma bidireccional cumpliendo con las licitaciones de carga requeridas.

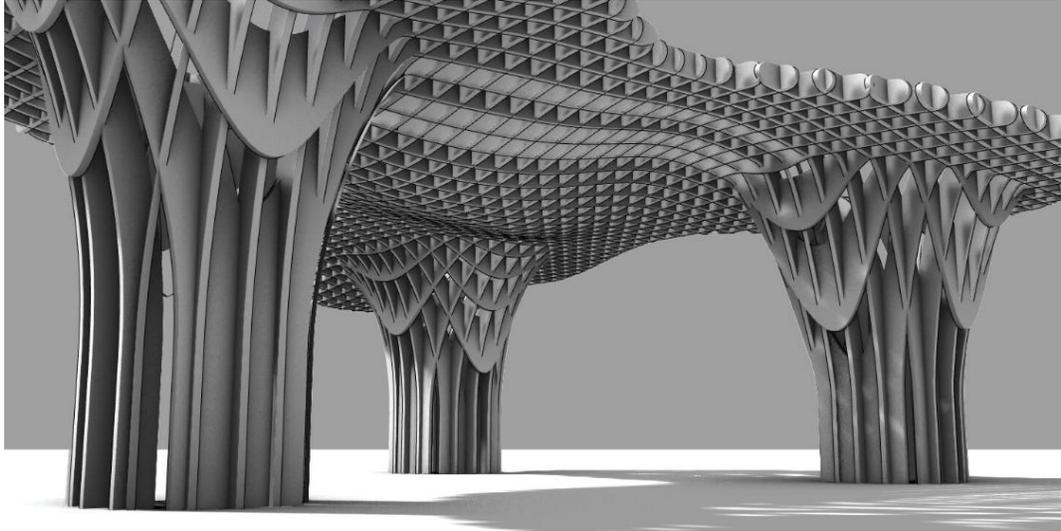


ILUSTRACIÓN 5, TAMBIR SARKAR

#### **2.4.1.4. Parametricismo**

Se define como un estilo de diseño que contiene la arquitectura vanguardista contemporánea, promovido en consecuencia de la arquitectura posmoderna y la arquitectura moderna. Dicho termino fue acuñado por Patrick Shumacher, socio arquitectónico de Zaha Hadid (1950 – 2016).

El término parametricismo, tiene su origen en base al diseño paramétrico, la cual está basada en las restricciones de una ecuación paramétrica. Este a su vez está basado en programas computacionales, algoritmos y computadoras con el fin de manipular ecuaciones con fines de diseño.

El parametricismo rechaza la homogeneización (repetición en serie) como uno de sus principales valores claves al momento de diseñar. Uno de sus objetivos, es diseñar con un mayor grado de complejidad espacial, manteniendo siempre la legibilidad (university, 2020).

La arquitectura atraviesa un ciclo continuo de adaptación innovadora, el entorno arquitectónico que está delimitada por una época socioeconómica del posfordismo. (este término hace referencia al sistema de producción en cadena implementado por Henry Ford a partir del año 1908).

Una de las claves que marcan la arquitectura de vanguardia y el urbanismo que deben abordar, se resumen en el siguiente lema: organizar y articular la creciente complejidad de la sociedad posfordista.

En la actualidad muchos de los estudios arquitectónicos y de ingeniería alrededor del mundo tienen o están implantando estrategias de trabajo sustentados en algoritmos, ya sean estos creados por empresas de programas, estudios especializados o por sus propias empresas.

De este modo una línea que es creada por un algoritmo no es solo una línea, si no que ahora está definida por una conexión inmensa a diversas bases de datos que estos a su vez contienen diversos tipos de información, de su valor monetario, dimensiones, tipo de material, resistencia, por mencionar algunas de las características.

Estos procesos son realizados de forma digitalizada a través de computadores de alta potencia gráfica y velocidad en sus procesadores, de esto dependerá el alcance de creación de los algoritmos dirigidos a la construcción.

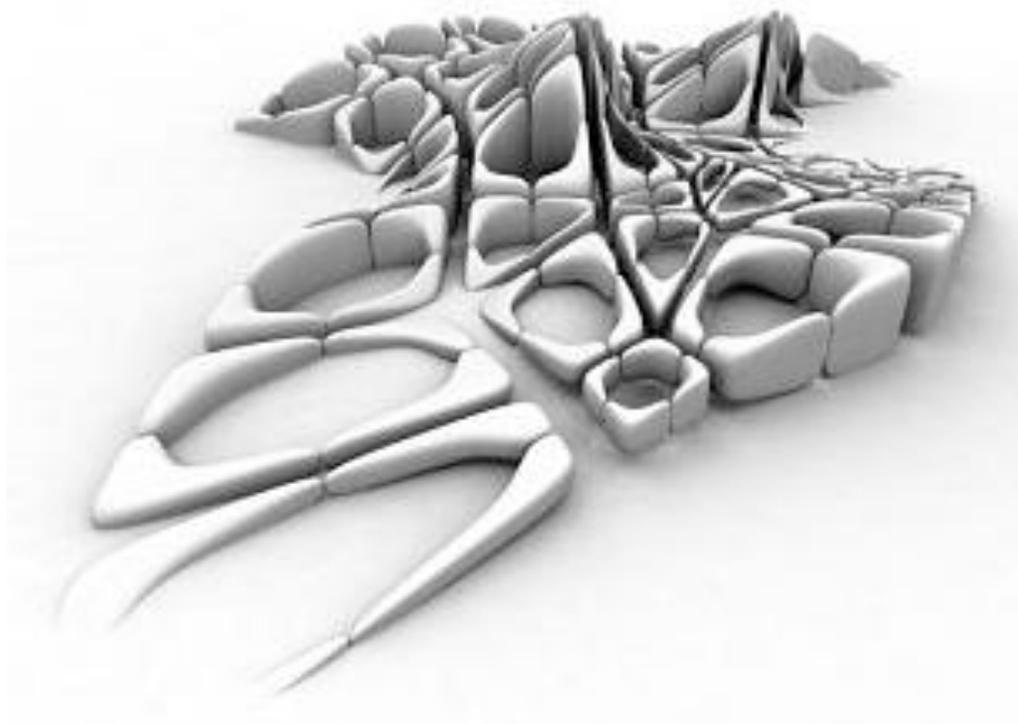


ILUSTRACIÓN 6, ZAHA HADID

## 2.5. BUTTOM-UP

Es una estrategia de procesamiento de información, un análisis de abajo hacia arriba.

En las últimas décadas del siglo XX, la arquitectura se ha actualizado con el concepto de una estrategia Buttom-Up de procesos de información, es decir, empieza a diseñar individualmente pequeñas partes, para, poder ensamblarlas y a su vez forma componentes más grandes, que de la misma forma se enlazan para formar el sistema completo (Palmer S.E., Rosch E., & Chase P., 1981).

Diseñar mediante este proceso nos permite entender que las partes individuales son un conjunto de información que se conectan unos con otros conforme van incrementando el nivel de sus entrelaces, el sistema va creciendo de una forma natural e inteligente, mismo que es adaptable o responsivo a los parámetros que lo definen.

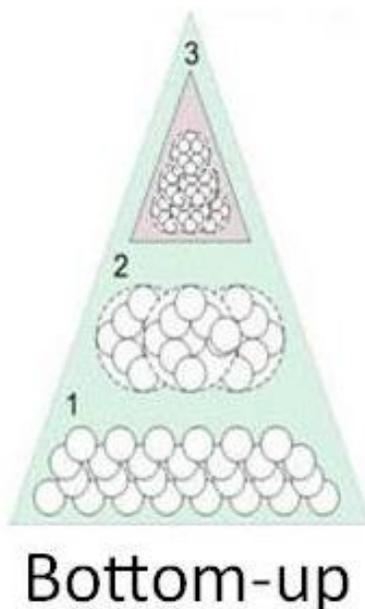


Ilustración 7, (Administración de negocios, 2016)

El proceso de Buttom-Up permitió a la arquitectura definir una estrategia de diseño, como un mapa genético donde se parte de datos, valores o dimensiones primitivas que irán conectándose unos con otros aumentando de nivel en lo que a su jerarquía estructural respecta, es decir, la información primaria ira evolucionando conforme se entrelace con conjuntos de información y de dicho proceso nacerá un nuevo conjunto que trasladara la información heredada de sus congénitos.

Por dar un ejemplo sencillo, si se ingresa como dato, dos puntos, y los unimos; el resultante de este entrelace determina una recta, que es un nivel superior si matemáticamente se habla, a su vez si a esta recta se entrelaza con dos rectas más se obtendrá un triángulo, de este modo esté será nuestro nuevo nivel. Y a su vez si a dicho nivel lo entrelazamos con nuevos datos tendremos un nuevo nivel de información, el proceso se repetirá constantemente siempre y cuando los datos de ingreso sean compatibles para seguir con el proceso evolutivo. Se está generando un algoritmo que pretende llegar a un resultado definitivo de diseño.

Sin embargo, el proceso evolutivo es una ley que no permite retroceder y es físicamente imposible llegar a la involución para recuperar datos desechados en el proceso, por lo tanto, es aquí cuando entra en acción la modelación virtual en la que podemos simular distintos escenarios. Citando a Manuel de Landa, estos conjuntos se vuelven auto organizables y siguen dinámicas no lineales (Palmer S.E., Rosch E., & Chase P., 1981).

Por lo tanto, nuestro ADN virtual, diseñado, podrá responder a los parámetros que nosotros imponamos y de esta forma los modificará en su propio código genético según el resultado que arroje sus datos frente a estos factores.

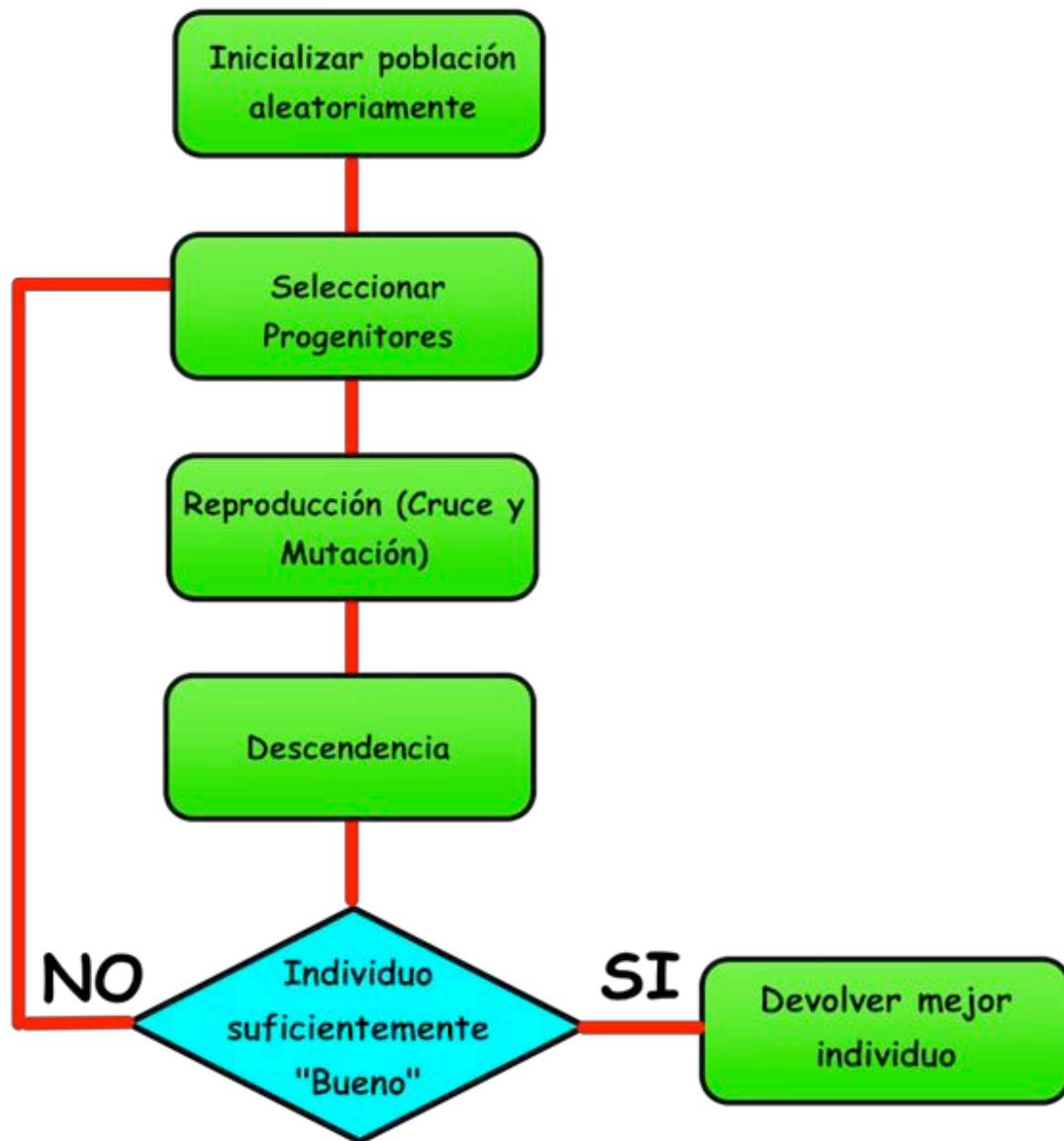


Ilustración 8. Ejemplo de algoritmo de evolución genética.

El aporte que genera esta metodología frente a las anteriores es que brinda la posibilidad de lograr una gran optimización de procesos de diseño. Por medio del proceso de selección de los individuos más óptimos que es realizado por los algoritmos evolutivos genéticos.

El diseño tiende a obedecer parámetros de pensamiento sistemático bajo el concepto darwinista según Guilles Deleuze:

- pensamiento poblacional.

- pensamiento intensivo.

El pensamiento poblacional. - dirigir la atención en términos de Adán y Eva es una forma incorrecta de interpretar este concepto, por lo contrario, se deberá realizarlo en términos de grandes comunidades.

La población está definida por una matriz que generara el contexto mas no al individuo, por lo que, al llevar a cabo un proceso de forma aislada al momento del diseño, este, no permitirá que su evolución progrese ya que esta está definida por un sistema de programación genética con los parámetros previamente definidos en función de su comunidad.

Parent A	Parent B	Odds: Eye Color of the Baby			
	+ 	=	 75%	 18.75%	 6.25%
	+ 	=	 50%	 37.5%	 12.5%
	+ 	=	 50%	 0%	 50%
	+ 	=	 <1%	 75%	 25%
	+ 	=	 0%	 50%	 50%
	+ 	=	 0%	 1%	 99%

Ilustración 9, Padre + Madre = probabilidad del color de ojos resultante

El pensamiento intensivo hace referencia a un proceso de diseño que interactúa en función a las condiciones que permitan evolucionar sus componentes.

Por lo tanto, según esta definición un organismo no podrá evolucionar si su estructura no es capaz de responder a las variables que influyen sobre la misma y terminara por pasar a la historia de la selección natural.

Si a la ingeniería se hace referencia, una estructura que no sea capaz de responder a las variables de carga, esfuerzos o reacciones este no será capaz de sobrevivir a las variables y terminara siendo consumido por las mismas.

Por lo tanto, si se hace referencia a una evolución virtual aplicada a la ingeniería estructural, tiene que ser de mucha importancia el prediseño para que el diseño logre evolucionar.

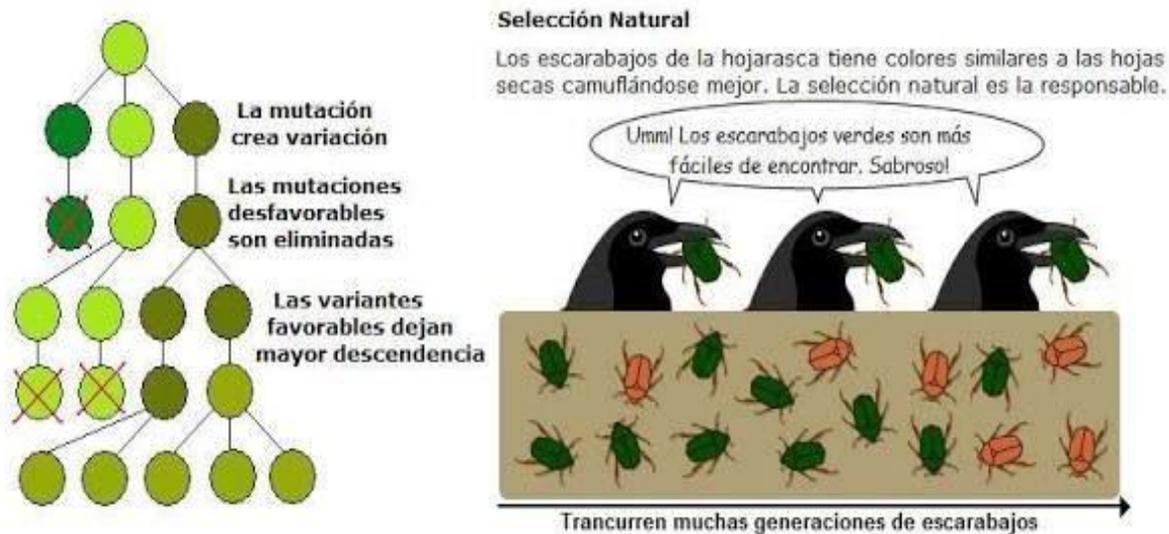


Ilustración 10, adaptación de la naturaleza para sobrevivir.

En conclusión, se puede definir que la arquitectura es siempre paramétrica mas no es una regla general que esta sea desarrollada por algoritmos generativos, debido a que los procesos de diseño se van diferenciando en cuanto a la conceptualización de información, por lo tanto, siempre estará en función de parámetros que deberán ser responsivos.

De este modo, el concepto de algoritmo generativo se entiende como un optimizador de procesos de diseño, mas no como una herramienta, este es el error en el cual el arquitecto cae si abusa de los programas de diseño sin entender lo que conlleva construir todo el diseño.

Como se constata según la información recopilada, esta revolución de algoritmos generados por medio de la modelación virtual no solo puede ser aplicados al campo de

la arquitectura, gradualmente este puede ser ingresado en el campo económico, social, ingenieril, etc. Estos van permitiendo un control más exacto del comportamiento de los parámetros que definan nuestra realidad y mediante la comprensión de estos como dinámicas que no actual de forma lineal, nuestra concepción universal sobre como el todo tendrá la tendencia a evolucionar, como se puede observar en la ilustración 11.

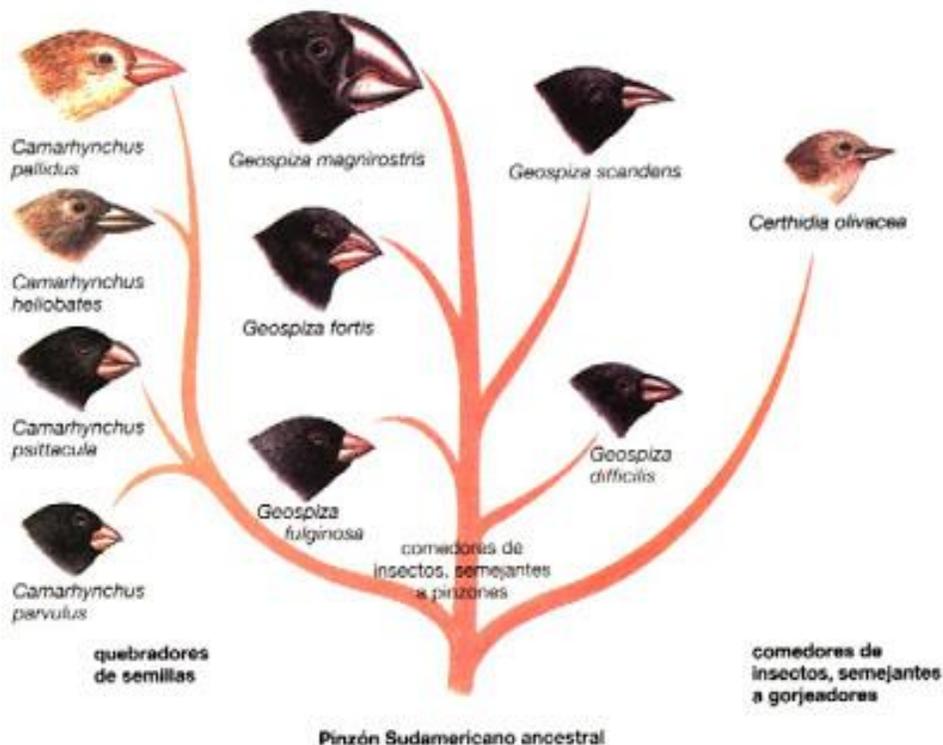


Ilustración 11, evolución adaptativa de pinzones de Darwin, (Hongos, 2010)

## 2.6. ALGORITMO.

Se define como algoritmo a la secuencia de instrucciones o pasos a seguir que representen un modelo de solución para un determinado problema en análisis. O también se puede definir como el conjunto de pasos ordenados que conducen a obtener una solución al problema en análisis (Pinedo, 2002).

Los algoritmos son totalmente independientes de los lenguajes de programación de cualquier tipo. En cualquier caso, el algoritmo puede ser escrito y en consiguiente ser ejecutado en un lenguaje de programación sin importar el tipo. Un algoritmo es una

infraestructura designada a brindar cualquier solución, siendo ejecutada luego en cualquier lenguaje de programación según sea el caso (Michalewicz, 1994).

### **2.6.1. CARACTERÍSTICAS DE LOS ALGORITMOS**

- Preciso. Define de manera rigurosa.
- Definido. Las veces que se siga un algoritmo, se obtendrá el mismo resultado siempre.
- Finito. Tiene un principio y un fin.
- Siempre debe producir un resultado. La información que brinde de salida serán los resultados de seguir las instrucciones.

Dicho esto, se puede concluir que un algoritmo está en la obligación de brindar una solución a un problema, entre dos algoritmos que lleven a la misma solución, siempre será el más corto en ser el seleccionado para cumplir con su función.

### **2.6.2. TÉCNICAS DE REPRESENTACIÓN**

La representación de un algoritmo, usualmente son graficas por medio de figuras geométricas, antes de este ser transformado a un lenguaje de programación cual sea este, se pueden utilizar algunos métodos de representación, escrita, grafica o numérica (Pinedo, 2002). Los métodos más conocidos y ya definidos son:

- Diagramación libre (Diagramas de flujo)
- Diagramas Nassi - Shneiderman
- Seudocódigo
- Lenguaje natural (español, inglés, etc.)
- Fórmulas matemáticas

### **2.6.3. LENGUAJE DE PROGRAMACIÓN VISUAL.**

Un lenguaje de programación visual se caracteriza por emplear bloques con diferentes colores y formas geométricas que puedan acoplarse o enlazarse entre sí (Aguinaga, 2021).

No existe en este lenguaje la sintaxis, ya que sus elementos (bloques) van determinando el orden en los que estos crean conexiones entre sí.

La programación visual. En la aplicación de este lenguaje se realiza la selección de bloques que contienen diferentes funciones que se encuentran en las librerías del mismo modulo, de este modo los bloques seleccionados se van acoplando unos con otros siguiendo un algoritmo predefinido, transformando de esta manera en un programa. Los conjuntos de estos programas a su vez crean una aplicación destinada a brindar soluciones según sea el caso. (Aguinaga, 2021).

#### ***2.6.3.1. Ventajas de la programación visual***

Una de las principales ventajas de la programación visual es la de no requerir el aprendizaje, tratar la sintaxis típica de un lenguaje de programación textual, de este modo el manejo de este tipo de lenguajes viene siendo más sencillo para el usuario.

Otra de las ventajas primordiales en muchos casos, en la programación textual es necesario tener un conocimiento previo de programación compleja, es decir que al momento de escribir el texto se puede cometer errores de variables mal escritas o caracteres de control que se ha olvidado escribir.

#### ***2.6.3.2. Desventajas de la programación visual***

En los proyectos grandes o complejos, los programas visuales tienden a volverse complejos y confusos debido a la cantidad de conexiones que se generaran, tomando en cuenta esto, se puede decir que los lenguajes de programación textual permiten mantener un código ordenado y sofisticado en el cual el mantenimiento será favorable en aplicaciones grandes.

Otra de las desventajas del lenguaje de programación visual es que su desarrollo está dirigido a la creación de cierto tipo de aplicaciones, no permite generar un multipropósito, el lenguaje de programación textual o tradicional permite generalizar el desarrollo de todo tipo de aplicaciones.

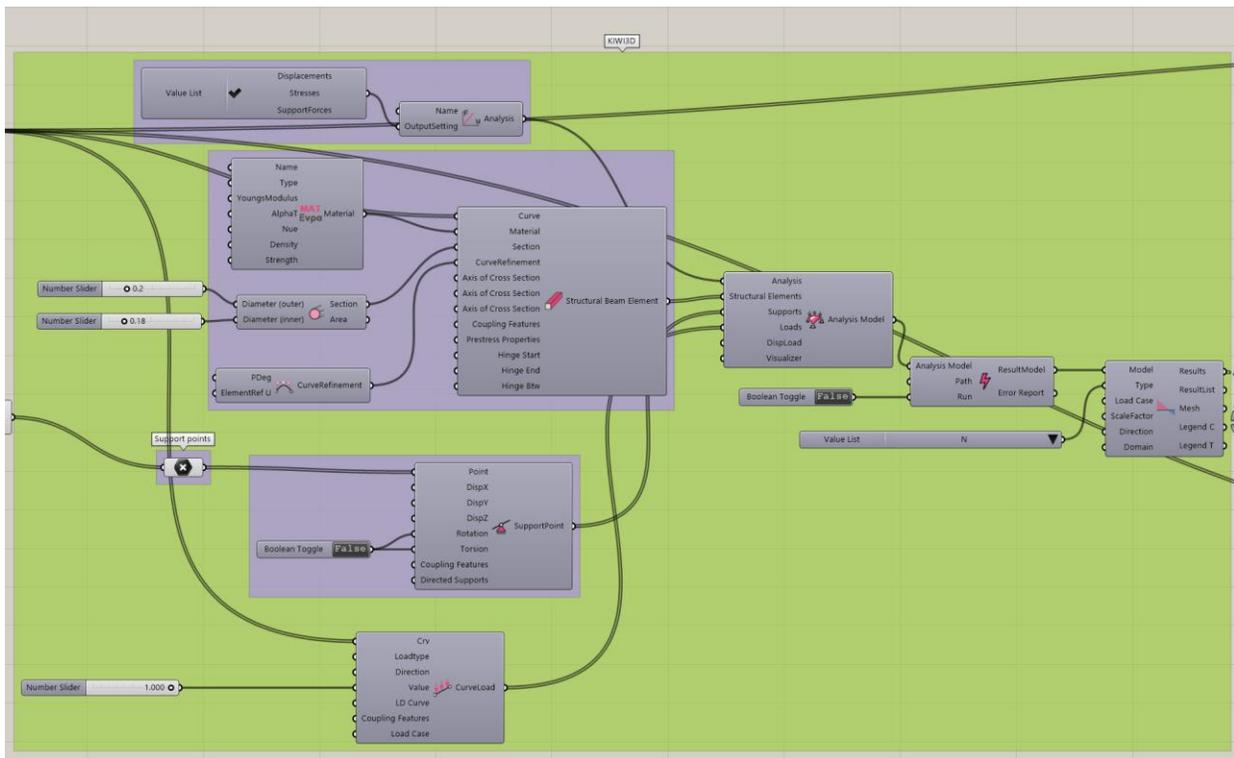


ILUSTRACIÓN 12, FUENTE PROPIA

## 2.7. ANÁLISIS ESTRUCTURAL.

### 2.7.1. ELEMENTOS FINITOS.

La geometría de un elemento estructural, sometida a cargas y a restricciones de movimiento, se subdivide en partes más pequeñas, conocidas como elementos, que representara el dominio continuo del problema. La división de una geometría como tal en elementos de menor dimensión tiende a resolver un problema complejo, al subdividirlos en problemas más simples al momento de analizar, lo que permitirá al ordenador hacer tareas más eficientes al instante de analizar el esfuerzo existente por cada subdivisión existente (Mirlisenna, 2016).

Es decir que el método de los elementos finitos examina la estructura como el conjunto de un numero finito de subdivisiones más pequeñas por medio de un sistema de ecuaciones numéricas.

El método de los elementos finitos propone que; Un número infinito de variables que no se conocen, lleguen a ser sustituidas por un número contable de elementos de comportamiento definido, la forma de las divisiones puede ser, triangulares, cuadradas, cualquier forma geométrica simple, dependiendo del tipo y dimensiones del problema en análisis. Como el número de divisiones del elemento ahora ya es limitado, ahora obtienen el nombre de ELEMENTOS FINITOS (Bathe, 1995).

Definido el termino de elementos finitos, estos están conectados entre sí por puntos, mismos que toman el nombre de Nodos o Puntos Nodales. El conjunto de los elementos y nodos se lo denomina **mall**. La precisión de los cálculos aplicadas a este método dependerá directamente del número de nodos y elementos que conformen la malla en análisis, del tamaño y de los tipos de elementos (Mirlisenna, 2016).

#### **2.7.1.1. El Método General**

Está definido por aproximaciones de problemas continuos, por lo tanto:

- El continuó. - hace referencia a la continuidad del esfuerzo existente a través del elemento, estaría definido por la división del mismo en un número extenso de subdivisiones, mismo que desde ahora se lo llamara como **finto**, cuyo comportamiento está definido mediante un numero finito de parámetros asociados a puntos característicos que se los denomina **nod**. Estos nodos definen las uniones existentes entre cada elemento con sus adyacentes, por medio de un análisis unitario de cada una de las divisiones existentes se llega a los resultados de cada sub elemento por separado para finalmente unir todos estos resultados y tener un producto final de análisis de esfuerzos.
- Un sistema completo es conformado por el ensamble de los elementos, para su solución se sigue las reglas de los problemas discretos.
- Las incógnitas del problema pasan a ser numéricas, para transferir a los nodos los resultados, y estos a su vez conectarse entre ellos para transmitir sus funciones y obtener un producto final.
- El comportamiento en el interior del elemento quedara definido por el producto que adquieren los nodos mediante las adecuadas funciones de interpolación.

El MEF (Método de los Elementos Finitos), por tanto, se transforma un elemento de naturaleza continua a un modelo discreto aproximado, a esto se lo denomina discretización del modelo. El conocimiento del comportamiento interno del modelo del cuerpo aproximado, se obtendrá mediante la interpolación de los valores obtenidos en los nodos. Es por tanto la aproximación de los valores de una función a partir del conocimiento de un número determinado y finito. (Valero, 2004).

### 2.7.1.2. Aplicación del método de los elementos finitos

El método puede ser aplicado en la resolución de problemas de análisis estructural en lo que respecta a la obtención de desplazamientos, deformaciones, tensiones también nos permite representar varios escenarios que evaluara el rendimiento del elemento en lo que concierne a resistencia, rigidez o fatiga mecánico.

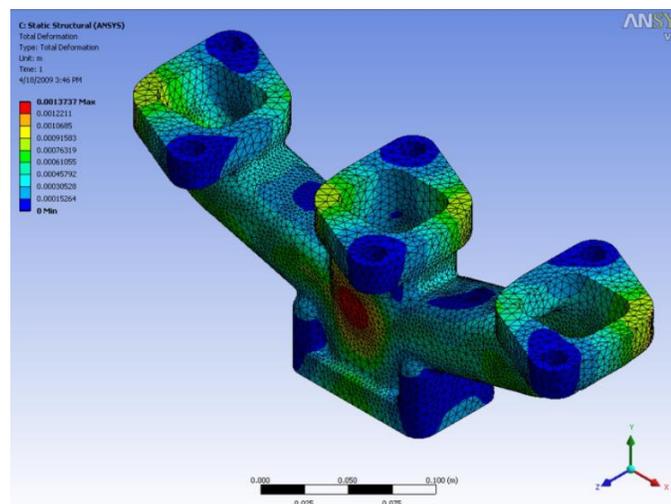


ILUSTRACIÓN 13, GIUSEPPE MIRLISENNA, SIMULACIÓN DEL ANÁLISIS DE ELEMENTOS FINITOS.

La aplicación más intuitiva para comprender el método de los elementos finitos, es el subdividir una placa en sub placas y aplicar un análisis unitario y posterior a esto conectarlos por puntos de control. El método nació gracias a la evolución de aplicaciones a sistemas estructurales.

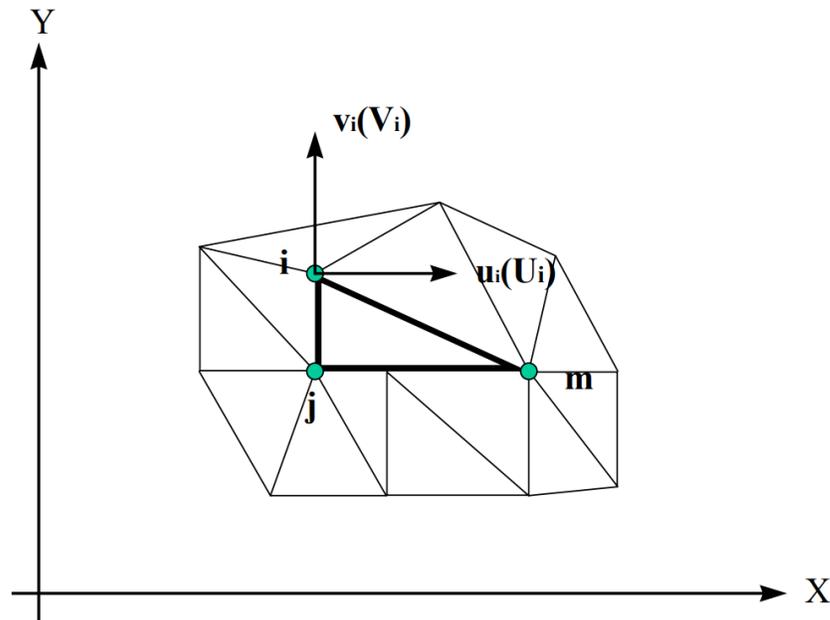


ILUSTRACIÓN 14, COORDENADAS NODALES (I,J,K) Y DESPLAZAMIENTOS DE LOS NODOS, VALERO (2004)

### 2.7.1.3. Ecuaciones generales.

#### 2.7.1.3.1. Campo de deformaciones

Dentro de un campo de deformaciones, un punto cualquiera, sea éste que se encuentre dentro del plano del dominio, está definido por un vector  $\mathbf{u}$ , que consta de tantas componentes como deformaciones existan en el dominio.

Para dicho caso especial es:

$$(\mathbf{u}) = \begin{pmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{pmatrix}$$

Si se tiene un elemento finito que conforma el campo de deformaciones, en el interior se aproximan, por medio de la hipótesis de interpolación, determinando de este modo el promedio de las deformaciones en cada nodo del elemento, dando como resultado los factores de las funciones de interpolación las siguientes:

$$u = \sum N_i U_i \quad v = \sum N_i V_i \quad w = \sum N_i W_i$$

Dicha interpolación se la puede poner en forma matricial:

$$\mathbf{u} = N\delta^e$$

Donde  $\delta^e$  es el vector correspondiente a todas las deformaciones nodales del elemento (figura....):

$$\delta^e = [U_1 \ V_1 \ W_1 \ U_2 \ V_2 \ W_2 \ \dots \ U_n \ V_n \ W_n]^T$$

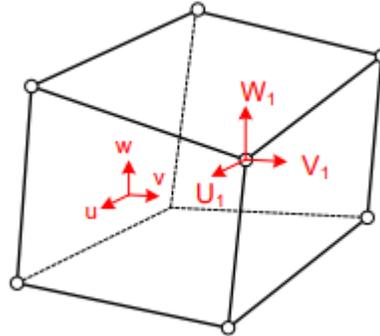


FIGURA 1 DEFORMACIONES EN UN ELEMENTO FINITO

La matriz de funciones de interpolaciones  $N$  tiene tres filas y tantas columnas como grados de libertad exista entre todos los nudos del elemento. La estructuración de esta matriz siempre obedece al tipo:

$$(N) = \begin{bmatrix} N_1 & 0 & 0 & \dots & \dots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & \dots & 0 & 0 & N_n \end{bmatrix}$$

### 2.7.1.3.2. Deformaciones unitarias

Las deformaciones unitarias aplicadas en un punto cualquiera del elemento, suponiendo que tiene pequeñas deformaciones, son:

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zz} \end{Bmatrix} = \begin{Bmatrix} \frac{du}{dx} \\ \frac{dv}{dy} \\ \frac{dw}{dz} \\ \frac{du}{dy} + \frac{dv}{dx} \\ \frac{dv}{dx} + \frac{dw}{dy} \\ \frac{dw}{dx} + \frac{du}{dz} \end{Bmatrix}$$

Y se puede expresar en forma matricial como:

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zz} \end{Bmatrix} = \begin{bmatrix} \frac{d}{dx} & 0 & 0 \\ 0 & \frac{d}{dy} & 0 \\ 0 & 0 & \frac{d}{dz} \\ \frac{d}{dy} & \frac{d}{dx} & 0 \\ 0 & \frac{d}{dz} & \frac{d}{dy} \\ \frac{d}{dz} & 0 & \frac{d}{dx} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = du$$

En esta expresión se puede identificar el operador matricial  $\mathbf{d}$  que permite pasar de las deformaciones de un punto  $\mathbf{u}$  a sus correspondientes deformaciones unitarias  $\varepsilon$ . Este operador produce el mismo número de filas y de deformaciones unitarias que existan en el elemento analizado, de la misma manera este actuará con el número de columnas y los desplazamientos  $\mathbf{u}$ .

Si sustituimos las deformaciones  $\mathbf{u}$  en función de las deformaciones nodales, aplicando la hipótesis de interpolación, se obtiene:

$$\varepsilon = du = dN\delta^e$$

Siendo de esta forma identificada la matriz B:

$$B = dN$$

Tal que cumpla con:

$$\varepsilon = B\delta^e$$

Dada la estructura de la matriz N, la matriz B se puede definir siempre como:

$$B = dN = d = \begin{bmatrix} N_1 & 0 & 0 & \dots & \dots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & \dots & 0 & 0 & N_n \end{bmatrix}$$

$$B = [B_1 \ B_2 \ \dots \ B_n]$$

Tal que las matrices  $B_i$  tiene la siguiente forma:

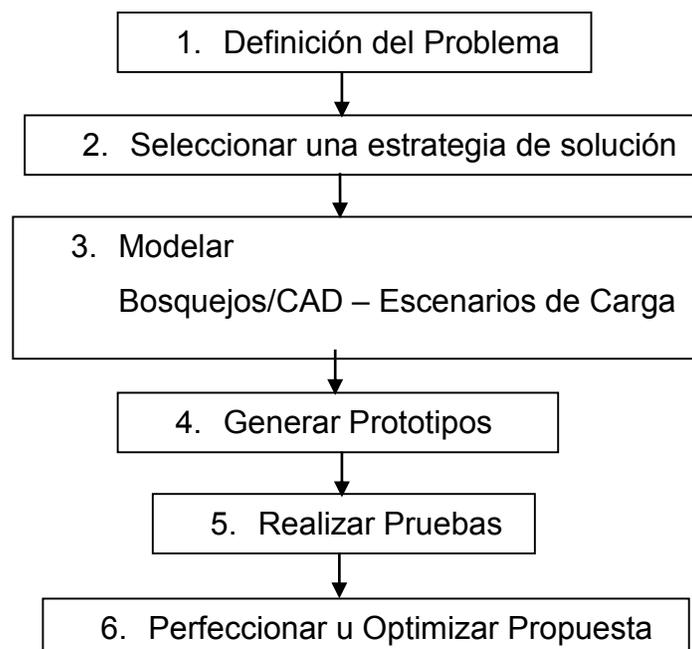
$$B_i = d \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} = \begin{bmatrix} \frac{dN_i}{dx} & 0 & 0 \\ 0 & \frac{dN_i}{dy} & 0 \\ 0 & 0 & \frac{dN_i}{dz} \\ \frac{dN_i}{dy} & \frac{dN_i}{dx} & 0 \\ 0 & \frac{dN_i}{dz} & \frac{dN_i}{dy} \\ \frac{dN_i}{dz} & 0 & \frac{dN_i}{dx} \end{bmatrix}$$

El valor en función de las variables definidas es totalmente general para otros tipos de problemas de elasticidad, como flexión de placas, problemas de revolución, etc. (Lizarza, 2011)

## 2.8. OPTIMIZACIÓN ESTRUCTURAL.

### 2.8.1. INTRODUCCIÓN

Una de las principales labores de un ingeniero es la de diseñar estructuras y analizar el desempeño. Para esto se plantean las actividades que se describen a continuación en el siguiente mapa conceptual:



La finalidad de la optimización es la búsqueda de una solución eficiente, misma brinde las mejores características que brinden una solución a un objetivo planteado por un problema en análisis. Si lo enfocamos a las matemáticas, se dice que es aquel proceso que evalúa en función de la primera derivada el mínimo y el máximo de una función, misma que será seleccionada por el diseñador para evaluar. En el caso muy particular de las estructuras existen varios aspectos que pueden surgir a nuestro interés, como es el caso de la rigidez estructural, el peso de la estructura, la forma y cantidad de material de un perfil dado, etc.

Usualmente la optimización se la realiza a componentes para maximizar su desempeño en la estructura. Gracias a esto la humanidad ha logrado modificar en diversas formas las estructuras que emplea para su resguardo, almacenamiento, transporte, de esta manera buscando encontrar siempre una mejor solución a un problema específico, está a sido la tarea de las últimas décadas de la ciencia y más aún en la ingeniería. (LOPEZ, 2013)

### **2.8.2. DEFINICIÓN.**

La optimización, según la (Oxford Lenguajes) Academia de Lenguas de la Oxford, es el método para determinar valores numéricos de las variables que intervienen en el proceso de ejecución para obtener un el mejor resultado posible.

La optimización estructural aplicado en la ingeniería viene a ser un concepto fundamental.

Un diseño optimo es considerado como tal, cuando se busca adecuar una solución que satisfaga las restricciones de cada parámetro de acuerdo con el criterio entre temas como los son el valor monetario, cantidades de material, facilidad de construcción, etc. (Cerroloza, M., & Annicchiarico, W., 1996).

Cuando a optimización estructural se hace referencia, existen varios métodos que se han ido desarrollando con el transcurso del tiempo en conjunto con los avances tecnológicos y las necesidades que la arquitectura presenta en la actualidad. De acuerdo con el autor Bendsoe, la optimización se clasifica en tres principales categorías que las define por: tamaño (perfiles), geometría y topológica (Bendsoe, 1989).

La optimización de tamaño pretende determinar las medidas ideales de los elementos que conforman la estructura de forma predefinida.

La optimización geométrica pretende determinar la forma óptima de la estructura de topología fija.

La optimización topológica pretende determinar la distribución ideal del material mediante la modificación de la conexión entre distintos nodos (Sanchez, 2012).

### **2.8.3. TIPOS DE OPTIMIZACIÓN.**

Los tipos de optimización son: La heurística y la metaheurística.

#### **2.8.3.1. Optimización Estructural**

Existen varias clasificaciones para los métodos de optimización estructural, siendo Querin quien las distinguió en dos grandes grupos: las Heurísticas y las gradientes, sin embargo, los métodos de optimización estructural son definidos como métodos indirectos, a continuación, se definirán los 3 grandes subgrupos, de forma breve y enunciando algunos métodos ya conocidos que resultan más relevantes entre estos. (LOPEZ, 2013).

#### **2.8.3.2. Métodos de gradiente**

Existen varios métodos capaces de determinar el valor de una variable de diseño aplicado a una función, que resulten en un máximo y un mínimo, en donde se consideraran los parámetros de diseño.

Los principales limitantes de estos métodos están presentes en el cálculo diferencial en la búsqueda de un máximo, aun así, representan una solución más formal desde un punto de vista matemático. En la práctica esto no resulta ser lo más idóneo para obtener una solución a corto plazo en geometrías complejas. (LOPEZ, 2013).

- Algunos ejemplos de estos métodos son:
  - Optimización con restricciones
  - Optimización sin restricciones
  - Método de los multiplicadores de Lagrange
  - Método de Kuhn – Tucker

- Método de programación lineal
- Método de homogenización
- Método de optimización de forma

#### **2.8.3.3. Métodos Heurísticos**

Dichos métodos están generados a través de la observación de la naturaleza y en muchos casos de manera intuitiva y lógica, es por esta razón que no se los puede asociar a una solución con un verdadero óptimo ya que en un estricto sentido al formular la matemática esta no la define como tal. Sin embargo, resulta ser altamente programable y sencillo a la vez, siendo su principal punto débil la falta de un rigor matemático que los defina, aun así, sirven muy bien al instante de resolver distintos tipos de problemas de optimización.

Algunos ejemplos:

- Algoritmos genéticos
- Recocido simulado
- Optimización de forma asistida por computadora

#### **2.8.3.4. Optimización metaheurística.**

El término metaheurística mencionado por primera vez por Fred Glover en 1986 en su artículo denominado la búsqueda del tabú, para llegar a definir que es el proceso de optimización metaheurística primero se debe definir que es la optimización heurística la cual se define como un procedimiento que permite encontrar solución a problemas de gran complejidad de una manera intuitiva, una vez entendido este concepto, la optimización metaheurística es la ejecución de procesos iterativos que permiten guiar y modificar las operaciones heurísticas para brindar una mejor calidad en las soluciones (Sörensen, Sevaux, & Glover, 2018).

Los metaheurísticos son métodos diseñados aproximados que pretenden resolver problemas de optimización combinatoria, en lo cual los métodos heurísticos clásicos no son efectivos, los métodos metaheurísticos proporcionan un modelo general para la creación de nuevos algoritmos híbridos, combinado con la inteligencia artificial, la

evolución biológica y varios mecanismos estadísticos (Mario Cesar Velez & Jose Alejandro Montoya, 2007).

#### **2.8.3.5. Métodos indirectos**

Los métodos indirectos desarrollados en las últimas dos décadas a partir de las investigaciones de Xie, Steven y Querin, con su fundamento matemático que resultan altamente programables y versátiles con los usuarios, sin embargo, estos métodos se basan fundamentalmente en la observación de la naturaleza y su manera de abordar los inconvenientes estructurales. La metodología que este método sigue es simple y no depende de una forma para poder ofrecer un óptimo, sin embargo, los resultados dependerán del tamaño del elemento en estudio. (LOPEZ, 2013)

Los métodos que se usan son:

- Optimización estructural evolutiva (ESO)
- Optimización estructural evolutiva aditiva (AESO)
- Optimización estructural evolutiva bidireccional (BESO)
- Optimización estructural morfológica (MESO)
- Diseño totalmente reforzado
- Creación inteligente de cavidades, entre otros.

#### **2.8.4. ALGORITMOS EVOLUTIVOS GENÉTICOS**

Se define como algoritmos evolutivos genéticos, aquellos que trabajan en función de soluciones y se basan en conceptos Darwinianos de supervivencia de las soluciones más aptas. Los diseños o parámetros que son representados de forma concatenada numéricamente que asumen el nombre de cromosomas en donde los elementos que lo conforman se denominan genes, vienen siendo los parámetros que se pueden manipular para determinar soluciones. Los cromosomas son generados y manipulados de acuerdo con su naturaleza y mecanismo de evolución como por ejemplo la reproducción y la mutación. Tras un proceso a través del tiempo y generar diferentes generaciones los diseños que se obtendrán según la naturaleza deberán ser los óptimos. Si comparamos con la técnica de programación matemática notaremos que los problemas de optimizar

las estructuras a partir de su tamaño y topología no restringirán al método de optimización con algoritmos evolutivos (Sanchez, 2012).

### 2.8.5. APLICACIONES

Los métodos de optimización estructural son aplicados en varios campos de la construcción tales como:

#### 2.8.5.1. Reducción de secciones

En vigas, mediante un software de optimización, como se muestra en la ilustración 5.

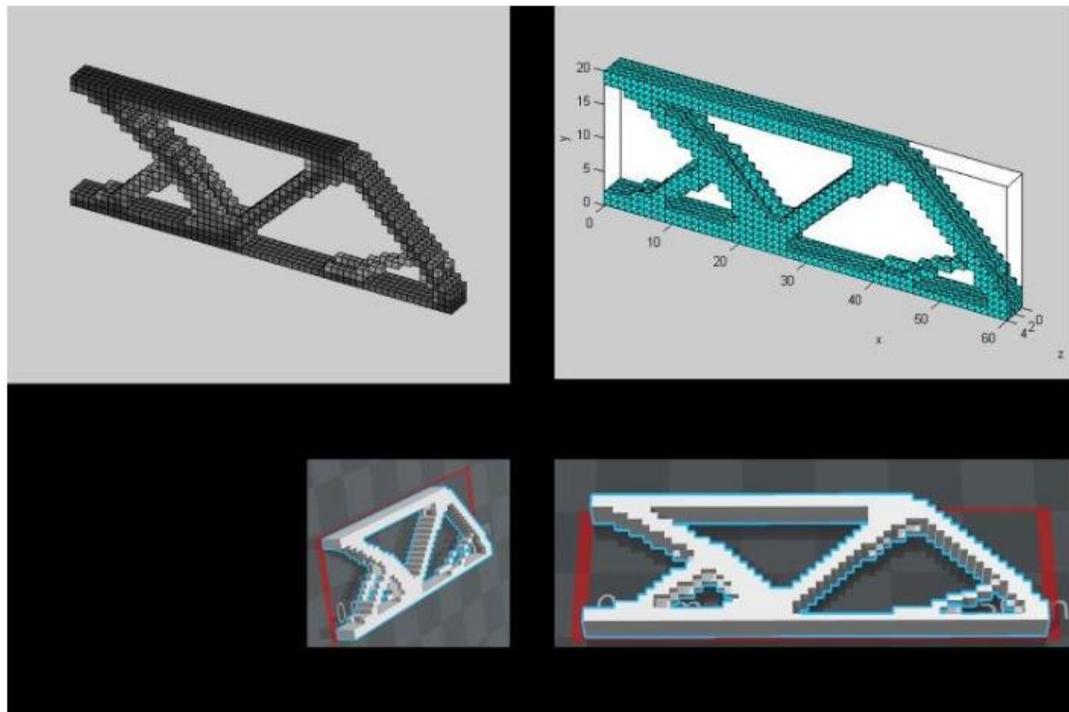


ILUSTRACIÓN 15, LIU Y TOVAR, 2014

La ilustración 5 demuestra los resultados de optimización topológica aplicados a una viga en voladizo obtenido con el código de 169 líneas TOP3D para un 30% de porcentaje de volumen.

### **2.8.5.2. Reducción de costos,**

En cuanto a la reducción del uso de materiales en los elementos estructurales que conforman una estructura, el volumen de material es directamente proporcional al costo de este, es decir que a mayor volumen de material tendremos mayor costo. Al optimizar la cantidad de material se puede reducir considerablemente el costo de este según el porcentaje de optimización que la estructura lo permita.

### **2.8.5.3. Facilidad de construcción.**

se puede definir como facilidad de construcción como el tiempo que tarda en ser ensamblada una estructura y el porcentaje de complejidad de su sistema constructivo.

### 3. CAPITULO 3 (HERRAMIENTAS APLICADAS AL DISEÑO PARAMÉTRICO Y AL ANÁLISIS ESTRUCTURAL).

Existen una variedad extensa de las herramientas que se pueden utilizar para el modelado de una estructura paramétrica, como, para su análisis estructural y optimización, en consecuencia, para nuestro tema de estudio, se hará uso de las herramientas de programación visual y sus módulos correspondientes, de este modo se dará a conocer el uso de las mismas.

#### 3.1. RHINOCEROS 3D.

Rhinoceros 3D, o comúnmente denominado Rhino en el medio del diseño digital, es un programa bajo la licencia (GPL) que permite a los usuarios aspirantes utilizar una versión de prueba de 90 días, esta herramienta permite crear, editar, analizar, animar, renderizar y traducir curvas NURBS, en superficies, sólidos, nubes de puntos y mallas (McNeel & Associates, 2019).



ILUSTRACIÓN 16. LOGOTIPO DE RHINOCEROS 3D. CITADO DE (MCNEEL & ASSOCIATES, 2019)

Las curvas NURBS son representaciones matemáticas de una geometría en 3D que tiene la capacidad de describir cualquier tipo de forma con precisión, abarcando los siguientes: líneas, círculos, arcos o curvas 2D has los sólidos o superficies de forma libre en 3D, ejemplo la ilustración 17, la maleabilidad de los NURBS permite que sea utilizado

en cualquiera de los procesos que van desde ilustrar y animar hasta el proceso de fabricado. (McNeel & Associates, 2019).

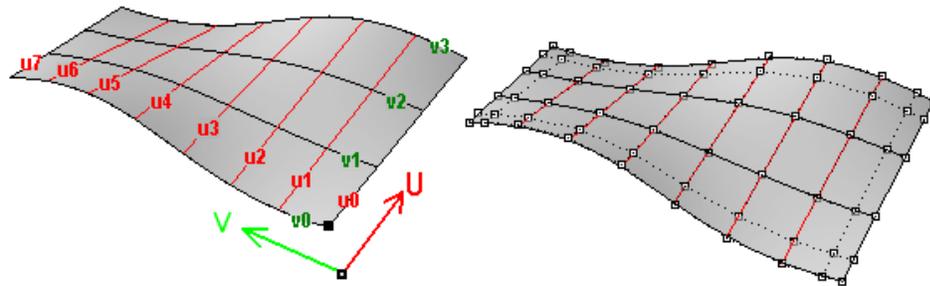


ILUSTRACIÓN 17. EJEMPLO DE USO DE NURBS. CITADO DE (MCNEEL, 2019)

A más de que Rhinoceros 3D posee sus propias herramientas, contiene herramientas que permite a los desarrolladores una mayor flexibilidad con el usuario, como lo es, RhinoCommon (.NET), Grasshopper, rhino.Python, RhinoScript, administrador de tareas para plug-ins del Zoo y Rhino Installer Engine, de este modo Rhino posee un código abierto que permite al usuario interactuar de manera personalizada con el programa (McNeel & Associates, 2019).

RhinoScript y Rhino.Python son potentes lenguajes de scripting de Rhino que, gracias a su manejabilidad y su biblioteca virtual, permite transmitir cualquier programa que se realice en estos entornos. La diferencia que existe entre estos dos módulos, además de su lenguaje de programación es que los programas que se desarrollen con RhinoScript solamente pueden funcionar en Windows, en tanto que los programas desarrollados con Rhino.Python permiten ser utilizados con Windows y Mac (McNeel & Associates, 2019).

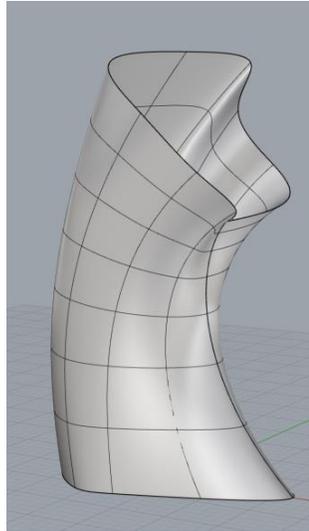


ILUSTRACIÓN 18. ESTRUCTURA GENERADA EN RHINOCEROS 3D. FUENTE PROPIA

### 3.2. GRASSHOPPER

Grasshopper es un editor de programación visual que fue creado por David Rutten en Robert McNeel & Associates. Grasshopper es un componente del programa Rhinoceros 3D que puede ser utilizado por profesionales en diferentes campos de trabajo y estudios aplicados a la arquitectura, ingeniería, diseños industriales y más. Grasshopper en conjunto con Rhino ofrece la oportunidad al usuario de tener un control paramétrico sobre los modelos realizados, además de brindar una plataforma capaz de desarrollar programas de alto nivel sobre una interfase grafica muy intuitiva y de fácil manejo mediante la programación visual (Gil & Parsons, 2014).



ILUSTRACIÓN 19. LOGOTIPO DE GRASSHOPPER. CITADO (MCNEEL & ASSOCIATES, 2019)

Al instante de descargar la versión más actual de Rhino este incluye de forma instantánea el componente Grasshopper, dentro de las opciones de descarga también se lo puede adquirir de Grasshopper3D.com para versiones más antiguas de Rhino. Grasshopper está en la capacidad de diseñar programas visuales denominados definiciones, mismas que están compuestas por nodos y cables (Gil & Parsons, 2014). En la ilustración 20 se muestra la interfaz de Grasshopper.

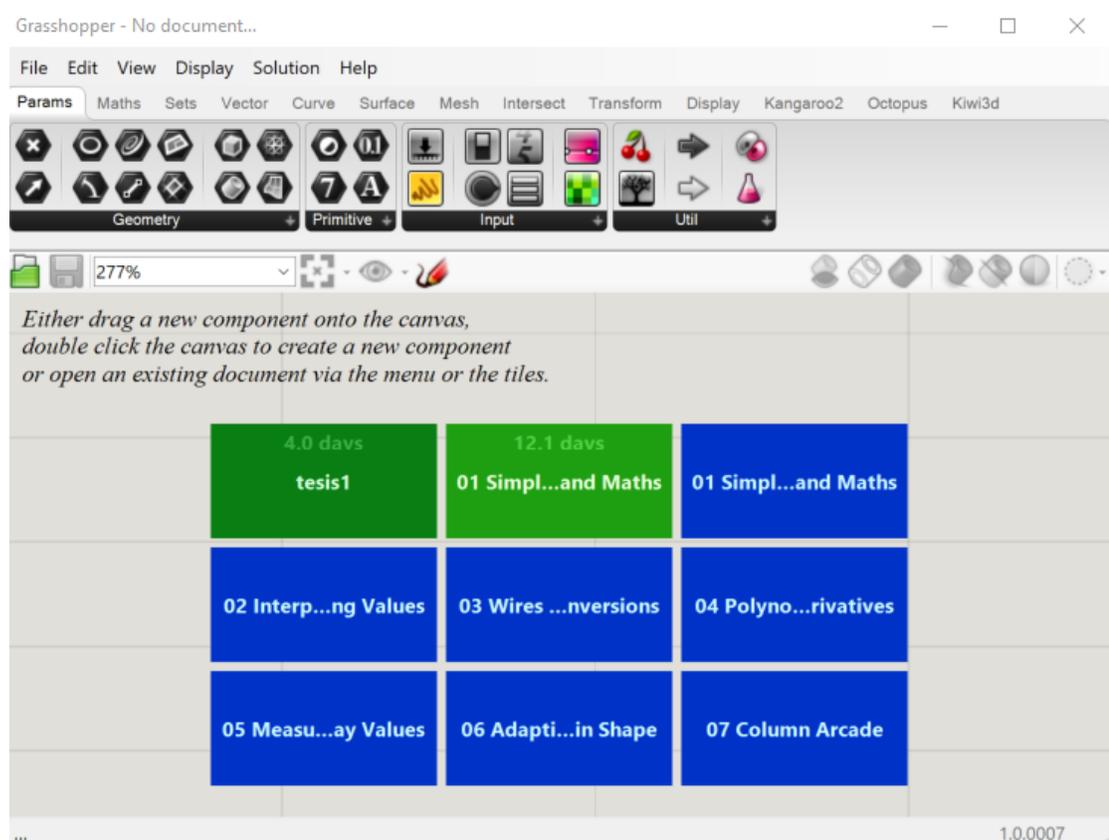


ILUSTRACIÓN 20. INTERFAZ DE GRASSHOPPER. CITADO DE (GIL & PARSONS, 2014).

Mediante componentes denominados como 'slider' el cual es un componente que permite establecer valores numéricos de manera didáctica para el usuario y a partir de los valores generados y con las conexiones adecuadas dentro del programa se puede tener control sobre los elementos programados de forma intuitiva. La conexión que existe entre Grasshopper y Rhino es esencialmente en tiempo real. Al instante de ajustar los parámetros mediante un 'slider' se actualiza el modelo en el Rhino.

Grasshopper representa a través de componentes que se dedican únicamente a desarrollar códigos en distintos lenguajes como los son: Python, Visual Basic y C++ (Gil & Parsons, 2014).

Grasshopper representa una puerta a muchos programas BIM como Tekla (Tekla, s.f.), que permite una comunicación entre todos estos programas de forma sencilla. En la actualidad existen entornos BIM como lo es REVIT (Autodesk,s.f.) que cuenta con Dynamo (Autodesk,s.f.) que funciona de manera muy similar a Grasshopper.

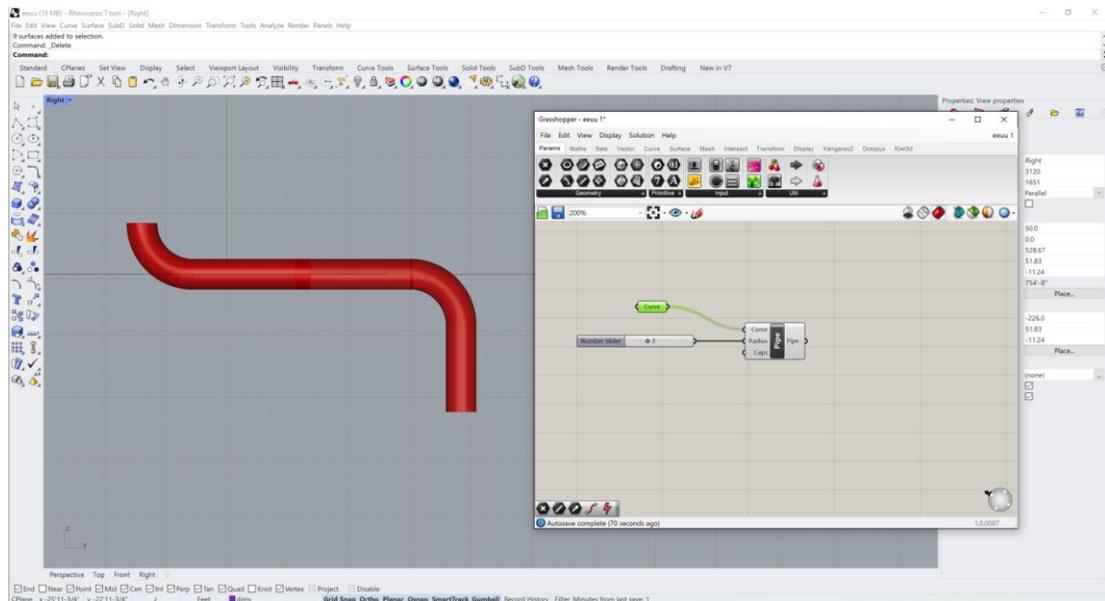


ILUSTRACIÓN 21. DEMOSTRACIÓN DEL USO DE UN SLIDER EN GRASSHOPPER, FUENTE PROPIA

Como se puede observar en la ilustración 21, a una curva definida según parámetros de diseño se le adjuntó un nuevo parámetro por medio de un 'slider', mismo que permite modificar el diámetro de la curva según se le vaya modificando en el mismo.

### 3.3. KIWI!3D.

Kiwi!3D permite la integración del análisis isogeométrico (IGA). La característica especial de IGA radica en el uso de B-splines racionales no uniformes (NURBS) como funciones base para los elementos finitos. Por lo tanto, permite ejecutar simulaciones directamente en NURBS sin necesidad de mallado.

Kiwi! 3D incluye un elemento isogeométrico de carcasa, membrana, viga y cable. Puede manejar tanto superficies recortadas como parches acoplados (modelos de varias superficies). Puede ejecutar problemas de análisis estructural lineales y geoméricamente no lineales. También incluye Búsqueda de formas para estructuras de tracción (Estrategia de referencia actualizada).

Debido a que el estado actual del complemento es un trabajo en continuo progreso, se distribuyen las versiones con una fecha límite de uso. Dentro del mismo tiempo de uso se proporciona una versión actualizada (One!3d, 2020).



ILUSTRACIÓN 22. LOGOTIPO DE KIWII!3D. CITADO DE (ONE!3D)

Kiwi! 3D es un complemento para Rhino y Grasshopper. Kiwi! 3D permite la integración del análisis isogeométrico (IGA) en el entorno CAD y, por lo tanto, crea un intercambio fluido y mutuo de datos de diseño y análisis: un proyecto, un flujo de trabajo, mejores resultados en menos tiempo (One!3d, 2020).

### 3.4. OCTOPUS

Octopus es uno de los componentes adaptables al módulo Grasshopper para dar solución a los problemas con algoritmos evolutivos orientado al diseño paramétrico (Vierlinger, 2012).



ILUSTRACIÓN 23. LOGOTIPO DE OCTOPUS. CITADO DE (VIERLINGER,2012).

Al pertenecer a la familia de los algoritmos evolutivos, octopus trabaja realizando el mismo método de solución lógico, la diferencia se encuentra en que octopus introduce el principio del Pareto para de este modo lograr trabajar a partir de múltiples objetivos (Vierlinger, 2012). El interfaz se puede observar en la ilustración 23, los cuadros que son graficados por el programa representan los resultados que se obtienen tras diferentes iteraciones y para describir el ejemplo cada eje representara un parámetro que dado el caso son distancias, flecha y numero de columnas, a partir de estos parámetros se va situando de manera gráfica los resultados que se obtiene de este proceso.

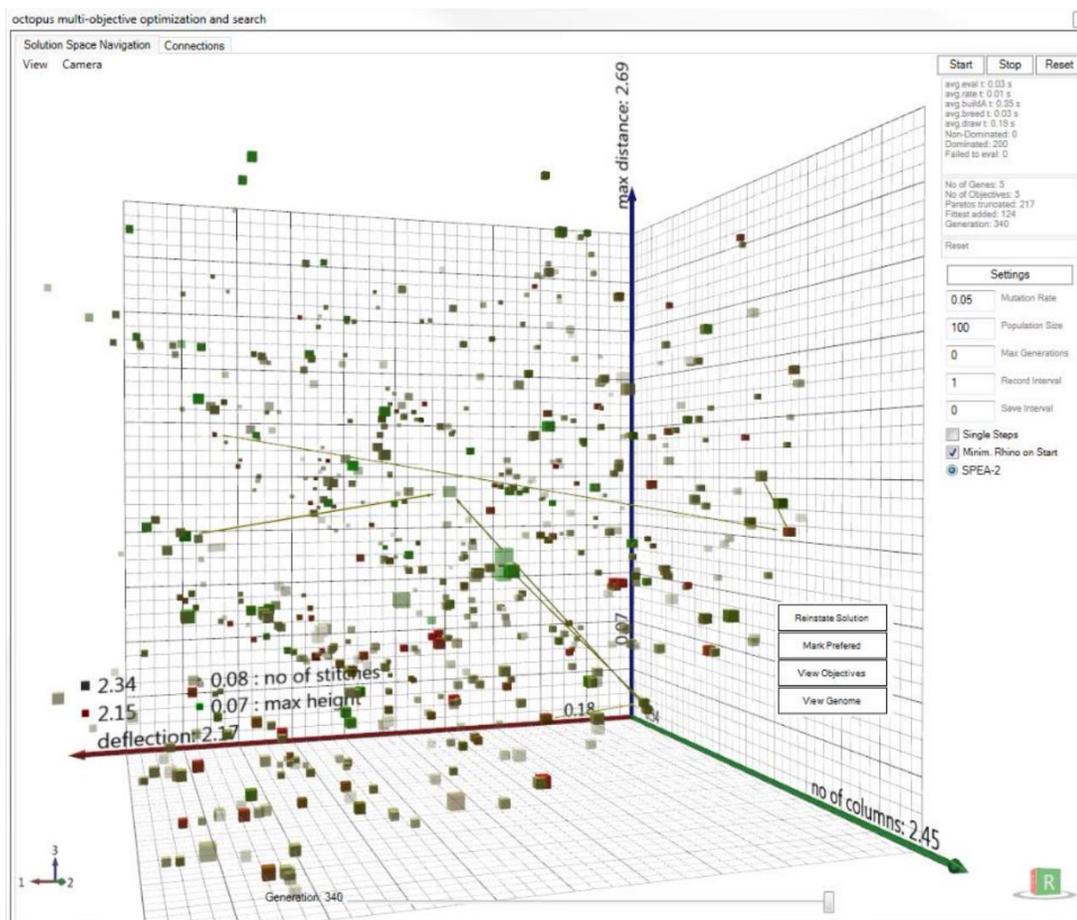


ILUSTRACIÓN 24. INTERFAZ DE OCTOPUS. CITADO DE (VIERLINGER, 2012)

Como parte de las tareas que se pueden realizar en Octopus son las siguientes (Vierlinger, 2012).

- Obtener una diversidad de soluciones para un simple objeto.
- Encontrar la mejor solución entre 2 a X objetivos.
- Mejorar soluciones por objetivos similares.
- Escoger las mejores soluciones durante la búsqueda.
- Cambiar objetivo durante la búsqueda.
- Soluciona modelos 3D para obtener retroalimentación visual.
- Guarda su historial de soluciones.
- Guarda todos los datos en el documento de Grasshopper.
- Exporta resultados a documentos de texto.

Para el uso correcto del módulo Octopus los creadores del programa de optimización multi -objeto proveen de un manual al instante de descargar el mismo, de este modo proveedor pretende su correcto uso.

Los parámetros principales que requiere Octopus es:

- Parámetros de genes, (Parameters of genes).
- Mapeo o modelado paramétrico, (Parametric model or mapping).
- Objetivos y fenotipos, (Objectives and phenotype).

### 3.4.1. PARÁMETROS O GENES

Este parámetro permite al usuario identificar y determinar la cantidad de puntos coordenados que creara el programa, se puede seleccionar de forma aleatoria o de forma puntual y específica según las necesidades del usuario, (Vierlinger, 2012).

Octopus puede conectar a múltiples componentes tales como: (GenePool y NumberSlider), los nombres de esto componentes y otros son citados en Ingles debido a que el programa solo capta este idioma al instante de traducir.

Octopus explorara los posibles valores de los controles deslizantes conectados según su configuración de rango individual, estos valores pueden ser aleatorios. Ilustración 25

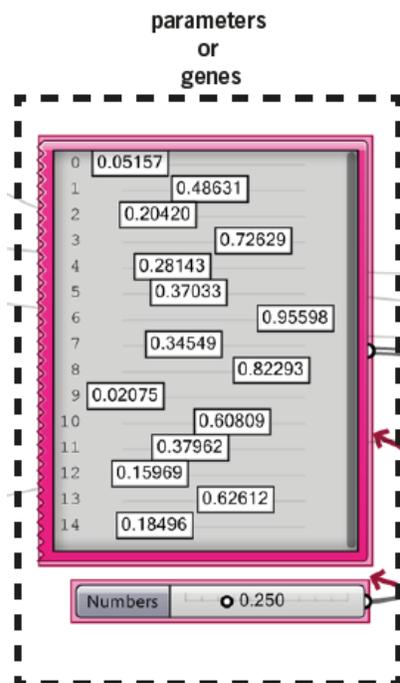


ILUSTRACIÓN 25 PARAMETROS O GENES

### 3.4.2. MAPEO O MODELADO PARAMÉTRICO

Este paso nos indica que debemos tomar algunos parámetros de entrada y realizar varios cálculos, análisis, para la generación del modelado. La duración de estos pasos es importante para un rendimiento óptimo. Se deberán realizar muchas evaluaciones con diferentes valores de parámetros, (Vierlinger, 2012). Ilustración 26.

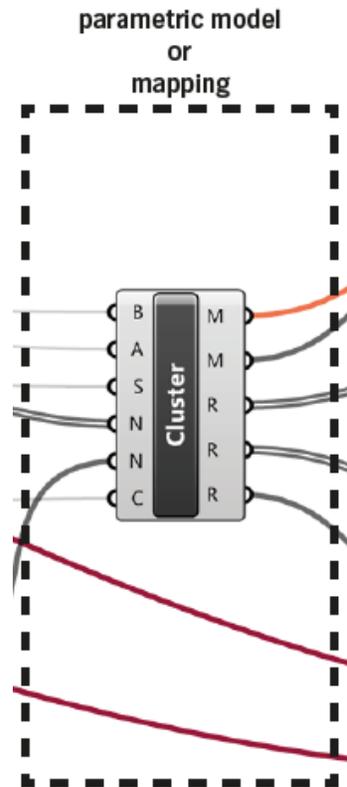


ILUSTRACIÓN 26 MAPEO O MODELADO PARAMETRICO

### 3.4.3. OBJETIVOS Y GENOTIPOS

Para este tipo de parámetro que nos pide el módulo Octopus existen varios tipos, los cuales son:

- Solution Mesh: Esto permite seleccionar una representación de malla 3D de la solución de Octopus para una evaluación visual, cave recalcar que el uso de mallas con un gran número de superficies aumentara el uso de memoria de Octopus y aumenta el de la definición cuando se guarda.
- Numeric objective values [Parámetro numérico]: los valores de aptitud de una solución. Mínimo dos, el máximo es teóricamente ilimitado.

- Textual objectives description [Parámetro de texto]: estos son nombres cortos y simples que describen las dimensiones del objetivo.
- Boolean hard constraints [Parámetro booleano]: Este paso es opcional, se puede conectar con Octopus y esperar un valor VERDADERO para cada solución válida; de lo contrario, se lanza la solución fuera.

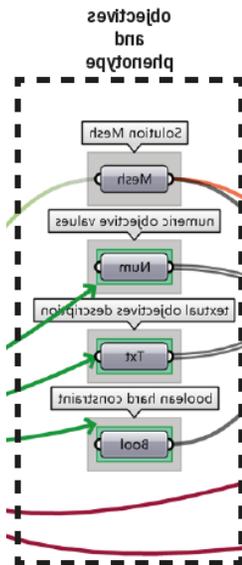


ILUSTRACIÓN 27 FENOTIPOS Y OBJETIVOS

Una vez descrito estos parámetros, Octopus podrá conectarse a estos parámetros para cumplir con su función. Ilustración 28.

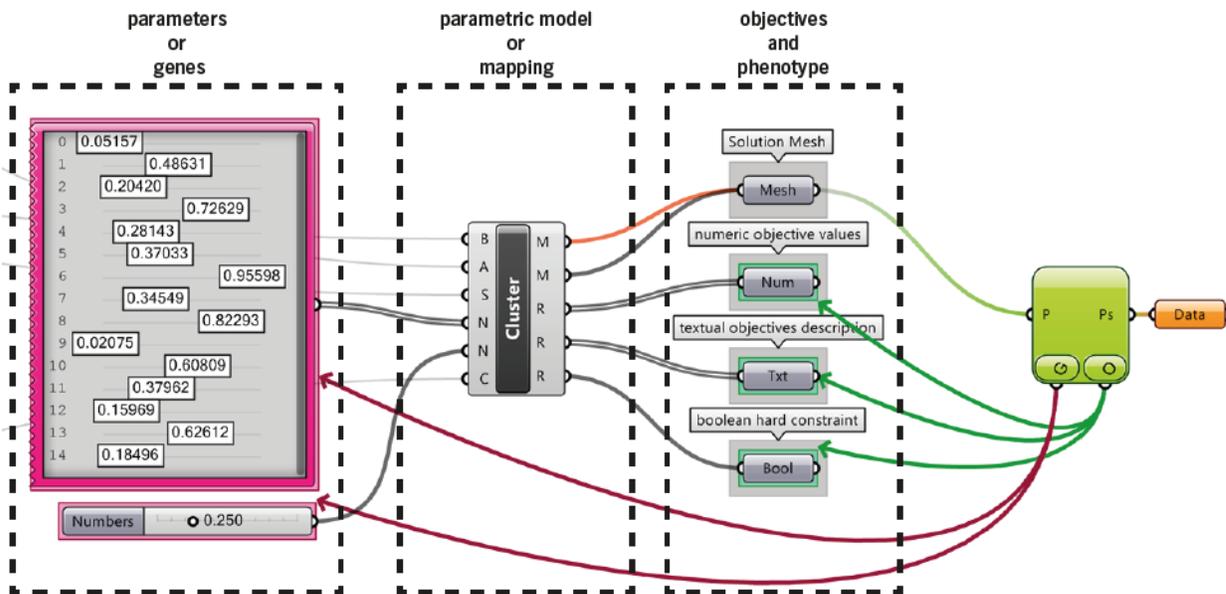


ILUSTRACIÓN 28 INTERFAZ OCTOPUS

### 3.5. PUFFERFISH

Este complemento es un conjunto de 330 componentes que se centra en interpolaciones, mezclas, transformaciones, promedios, transformaciones e interpolaciones, esencialmente cambio de forma. Pufferfish utiliza principalmente parámetros y factores para las entradas para un control más personalizado sobre operaciones como interpolaciones y cuadrículas en lugar de las entradas habituales de recuento de divisiones de Grasshopper. Estos componentes están acompañados de componentes de soporte que son métodos útiles para operaciones de interpolación / mezcla / transformación / celosía, como hacer que las curvas sean compatibles, un mapeador de gráficos de curvas personalizado y una transformación de múltiples subprocesos en una caja trenzada. Además, hay componentes adicionales que simplifican algunas operaciones comunes de Grasshopper, como probar la igualdad dentro de una tolerancia y redondear a los números más cercanos. (Pryor, 2021)



ILUSTRACIÓN 29 LOGO PUFFERFISH

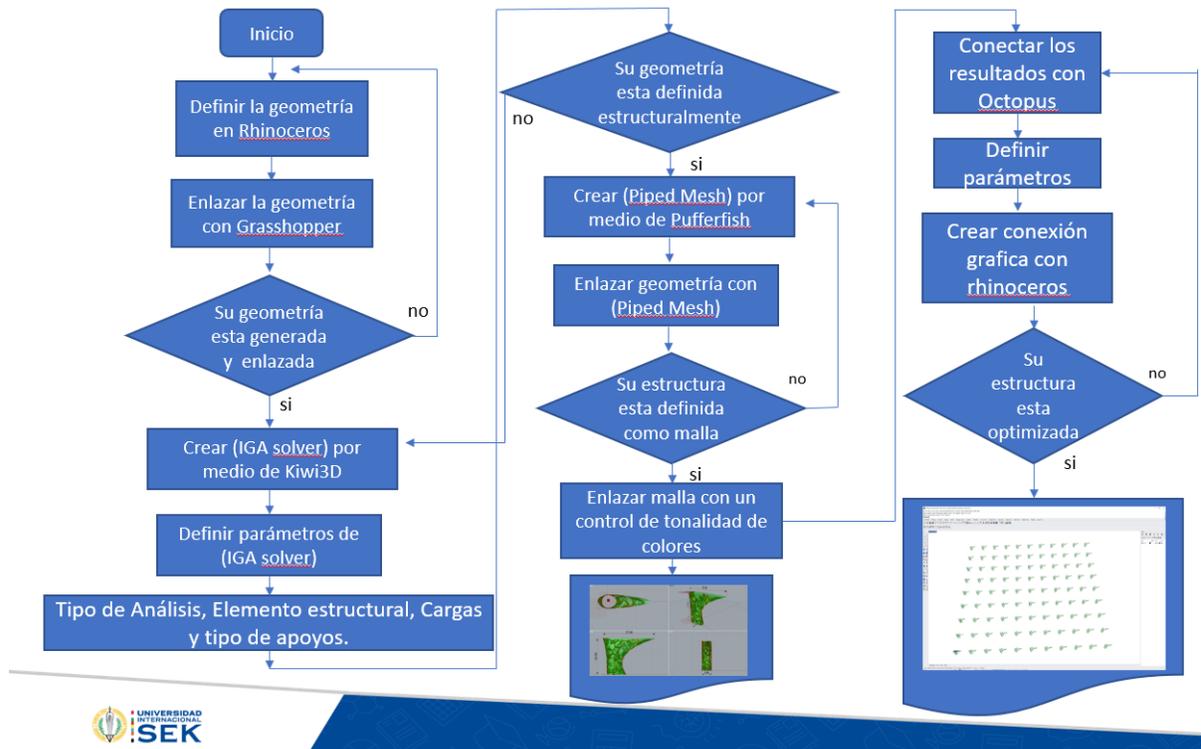
## 4. CAPITULO 4 (CASO DE ESTUDIO)

Una vez que se ha definido que es y cómo funciona el diseño paramétrico por medio del programa Rhino 3D y su módulo Grasshopper, está clara la definición de lo que es un algoritmo genético, ¡los complementos 'Kiwi!3d' para el análisis estructural y 'Octopus' para optimizarlo, a continuación, se define la metodología aplicada y de qué forma se harán uso los programas mencionados anteriormente para optimizar una geometría paramétrica impuesta por un arquitecto a su vez por un diseñador.

Para nuestro caso de estudio se hará uso una geometría parametrizada impuesta por un arquitecto que se muestra en la ilustración 30 está definida por una membrana mismo modelo paramétrico se generó en Rhino 3D.

Existen varios programas con herramientas útiles para la optimización de estructuras, los cuales son de gran ayuda para este tipo de problemas, sin embargo, la tecnología avanza a pasas agigantados brindando herramientas nuevas con un grado de eficiencia que aporta con la solución a este tipo de problemas en un menor tiempo y brindado al usuario una experiencia más versátil al momento de diseñar por medio de las herramientas existentes actualmente en el mercado.

Grasshopper en conjunto con Rhino ofrece la oportunidad al usuario de tener un control paramétrico sobre los modelos previamente realizados o sobre modelos que se puedan programar a través del mismo programa en mención, además de brindar una plataforma capaz de desarrollar programas de alto nivel sobre una interface gráfica muy intuitiva y de fácil manejo mediante la programación visual.



Mediante las herramientas que se menciona en esta tesis se va a demostrar el funcionamiento de las mismas y su aplicación por medio de la creación de un algoritmo, a más de esto, se pretende demostrar como por medio de este programador visual se logra optimizar el número de vigas que conformara nuestro sistema, el programa nos permite realizar estos procesos de una forma gráfica y del mismo modo tendremos resultados gráficos que corroboren con nuestra tesis.

## 5. CALCULOS

A continuación, se presentará en forma de ilustraciones con leyendas el proceso de construcción de nuestro algoritmo.

### 5.1. PARÁMETROS DE MODELACIÓN

La función general del algoritmo en estudio brinda una solución estructural a una geometría no convencional, misma que será impuesta por un diseñador o a su vez por

un arquitecto, la geometría en análisis será dirigida a un espacio público que identifique un sector determinado por el municipio competente.

El algoritmo necesita ser definido por los siguientes datos:

TIPO	DESCRIPCION
Material	Acero
sección	circular
Módulo de young	200000 Mpa
diámetro interno	18 cm
diámetro externo	20 cm

El algoritmo considera que el material es isotrópico. El tipo de modelo para analizar la estructura será Lineal, para este caso en particular solo se evaluarán los esfuerzos.

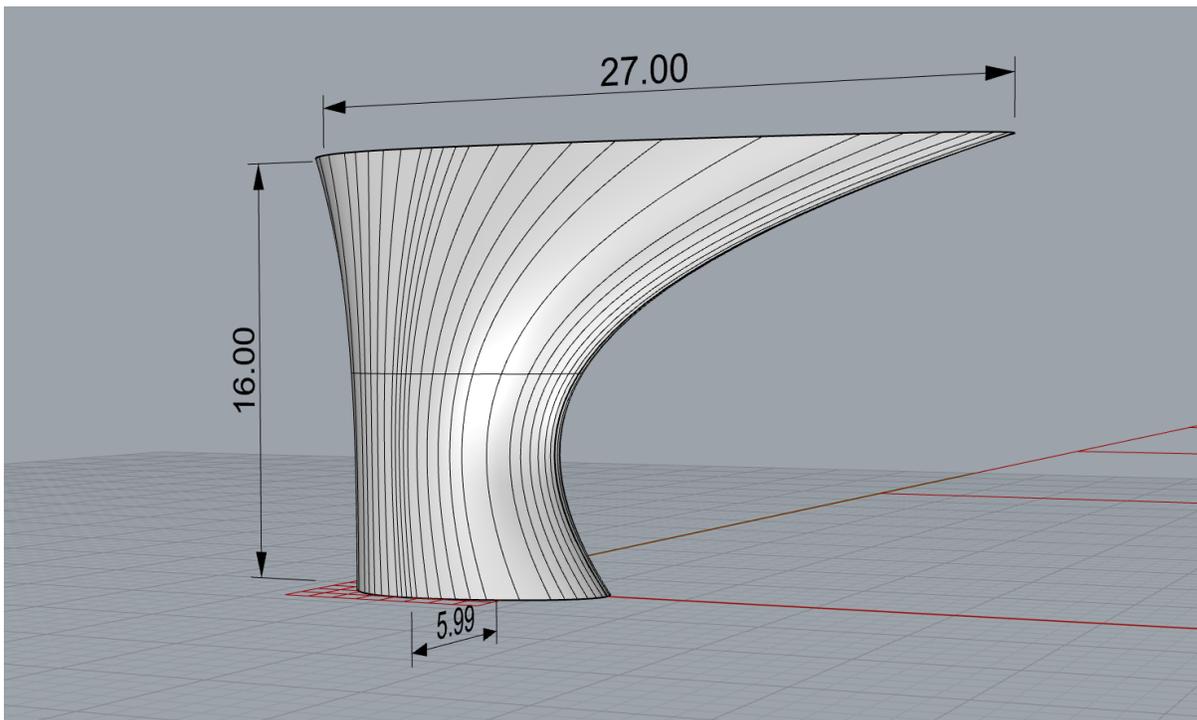


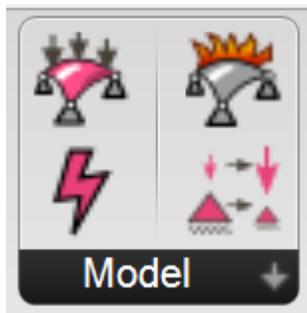
ILUSTRACIÓN 30 MODELO ARQUITECTONICO (RHINOCEROS)

En el siguiente texto se definirán los parámetros de cálculo que necesitará el programa para definir el algoritmo, mismos que serán presentados de forma gráfica y por ende su procedimiento de aplicación en el módulo de Grasshopper.

Cabe recalcar que previo a esto se tendrá que tener definida la geometría en el programa de Rhinoceros 3D, sobre la cual se aplicará el algoritmo que definirá los parámetros de gráficos de la estructura. Ilustración 32.

## 5.2. DEFINICIÓN DEL ALGORITMO USANDO KIW3D

Para definir dicho algoritmo, se procederá hacer uso de los módulos de Grasshopper, por lo tanto y en consecuencia de antes dicho, mediante el módulo Kiwi3D, para nuestro caso de estudio y objetivos se seleccionará un modelo nombrado como: Análisis isogeométrico de IGA, (IGA solver), el cual representará a una simulación. Ilustración 32.



Como se puede observar en la ilustración 31. Se encuentra el icono a seleccionar del IGA solver. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar según sus parámetros.

ILUSTRACIÓN 31

MODEL

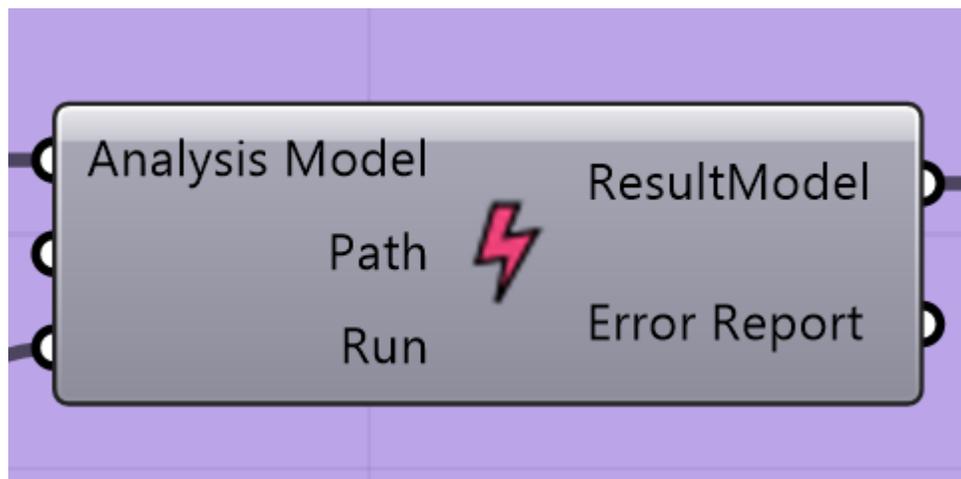


ILUSTRACIÓN 32 IGASOLVER

Este solucionador no funcionara por si solo ya que necesita de parámetros para ir definiendo nuestro algoritmo según nuestras necesidades, dicho esto se procede a definir los parámetros con las opciones que el IGA solicita, como podemos observar en la ilustración 32. Se ha seleccionado un modelo del análisis por medio de una de las opciones de Kiwi3D, mismo que se lo conectara al IGA, para todo esto es necesario poder activar y desactivar la función que está cumpliendo el IGA, para lo cual, se procede a crear un "Toggle" el cual cumple con las funciones de activar o desactivar dicho proceso. Ilustración 33.

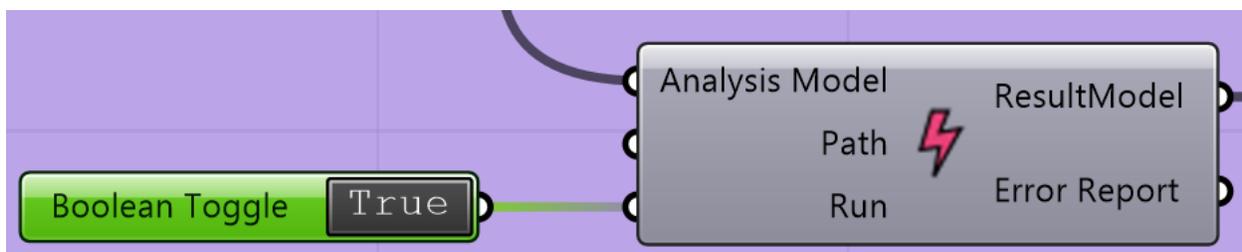
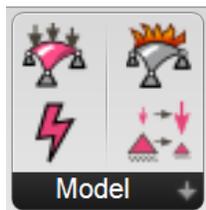


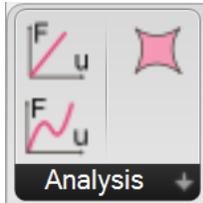
ILUSTRACIÓN 33 TOGGLE

Una vez conectado el modelo del análisis este nos pedirá los parámetros del modelo tales como: tipo de análisis, el elemento estructural que se va a utilizar, el tipo de apoyo, las cargas, entre otros parámetros que se necesiten para nuestro caso de estudio, para nuestro caso de estudio necesitaremos definir como primer paso el tipo de análisis, el tipo de análisis será definido por un análisis lineal, el cual solo se definirá por sus esfuerzos debido que nuestro caso de estudio así lo requiere, los desplazamientos y las fuerzas de apoyo para este caso no serán tomadas en cuenta debido que no es necesario para nuestra demostración. Ilustración 36.



La ilustración número 34. Muestra el icono a seleccionar del modelo del análisis. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar según sus parámetros.

ILUSTRACIÓN 34 MODELO  
DE ANALISIS



La ilustración número 35. muestra el icono a seleccionar del tipo de análisis. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar según sus parámetros.

ILUSTRACIÓN 35 TIPO DE ANALISIS

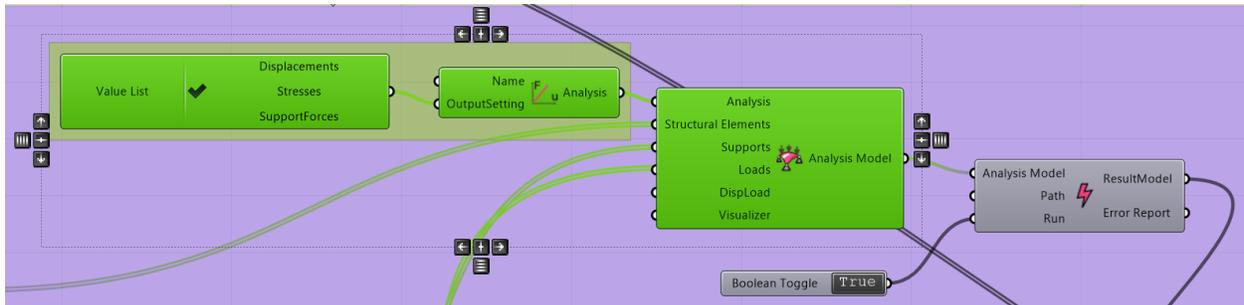
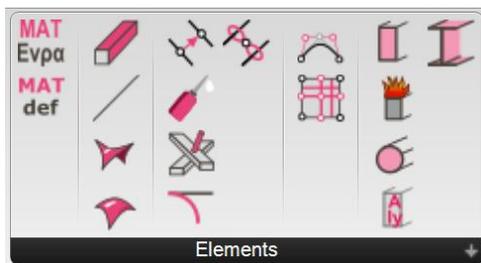


ILUSTRACIÓN 36 CONECCIÓN IGASOLVER

El siguiente parámetro para definir, se refiere al tipo de elemento estructural que se pretende usar, de este modo se definirá como elemento estructural una viga, misma que tiene requerimientos que los defina como tal. Ilustración 38.



La ilustración 37 muestra el icono a seleccionar del tipo de elemento estructural. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar

ILUSTRACIÓN 37 TIPO DE ELEMENTO ESTRUCTURAL

según

sus

parámetros.

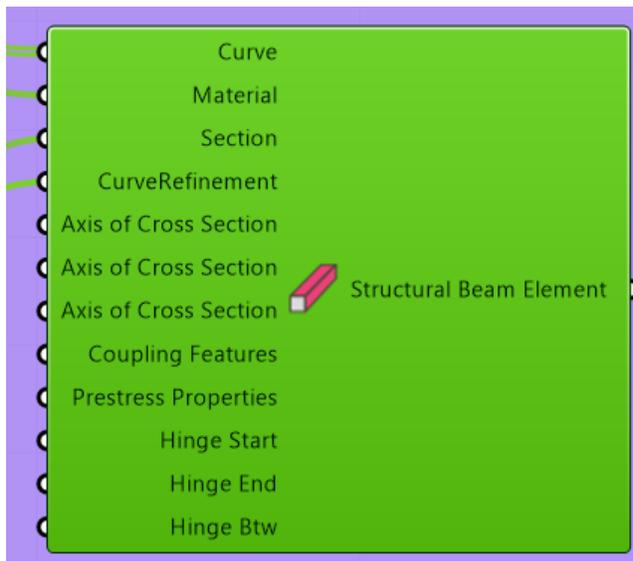


ILUSTRACIÓN 38 PARAMETROS DEL ELEMENTO ESTRUCTURAL

Para que este elemento estructural funcione como tal, este debe cumplir con un mínimo de parámetros requeridos por el programa como lo son: la geometría de las curvas de los elementos de las vigas, el material, la sección transversal y el refinamiento de la curva. Como se puede verificar en la ilustración 39.

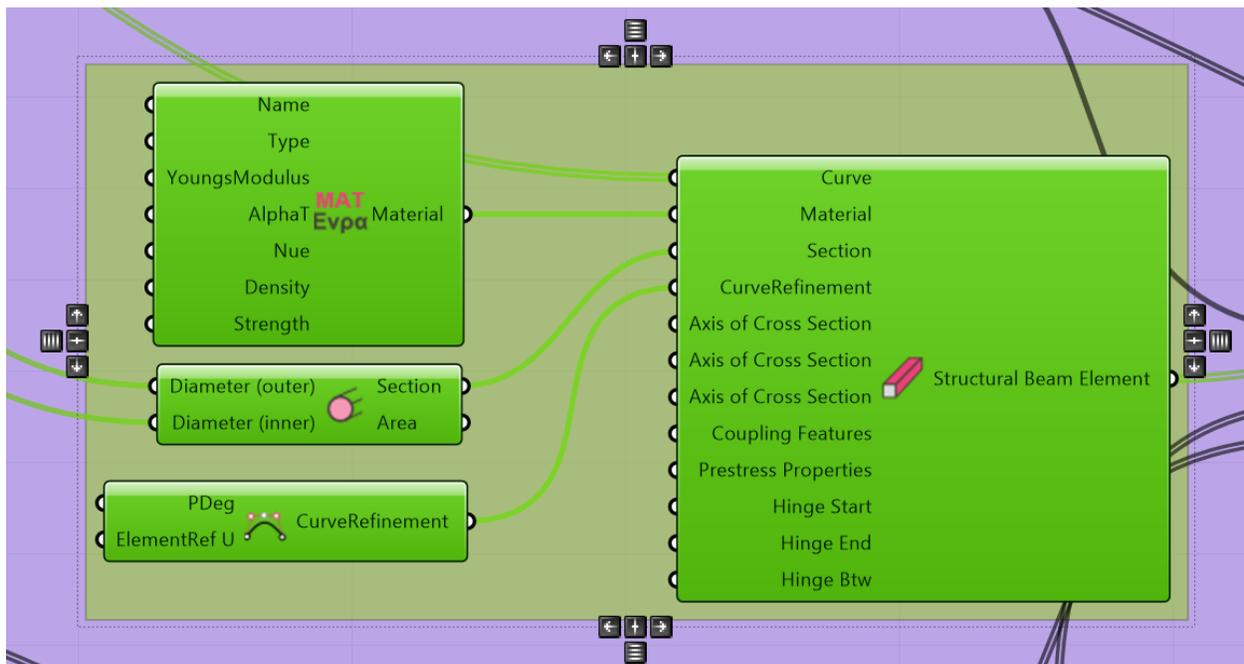


ILUSTRACIÓN 39 CONECCIONES DE PARAMETROS DEL ELEMENTO ESTRUCTURAL

Si a estos módulos los dejamos sin definir sus parámetros el programa los dejara por defecto, y nos dejara bajo los parámetros del Acero estructural con un módulo de Young de 200000 MPa, que para nuestro caso de estudio es correcto, en el caso de querer hacer uso de otro material se deberá definir estos parámetros según las características del material que requiramos.

La sección de nuestro elemento estructural se ha definido como una sección tubular la cual consta de un diámetro interno y un externo, por lo tanto, este es un parámetro que se debe definir tal y como se muestra en la ilustración 40.

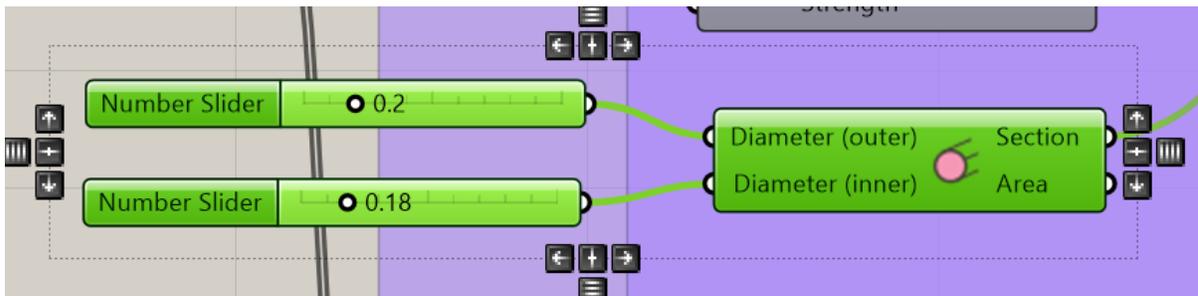


ILUSTRACIÓN 40 CONTROL DE DIAMETROS DEL PERFIL TUBULAR

Una vez definidos dichos parámetros, se obtendrá un producto que se define como el elemento estructural que conformará nuestra estructura. Ilustración 41.

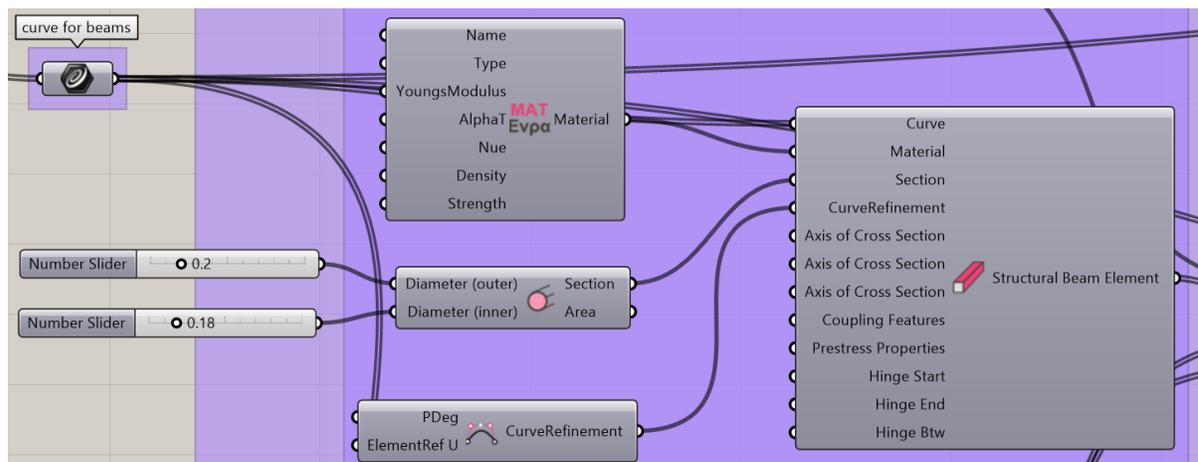


ILUSTRACIÓN 41 PARAMETROS DE MODELADO DEL ELEMENTO ESTRUCTURAL

Uno de los siguientes parámetros a definir, solicitados por el modelo del análisis es el tipo de apoyo que tendrá la estructura o el tipo de soporte. Ilustración 42

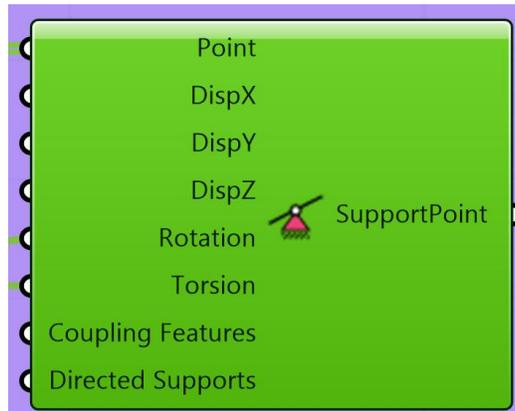


ILUSTRACIÓN 42 PARAMETROS DE APOYO

Mismos que nos solicitara ciertos parámetros de diseño, como en todos los módulos de Grasshopper si no se definen, estos tomaran valores predeterminados por el mismo programa los cuales pueden ser de opción verdadero o falso o a su vez una opción cualquiera que haya sido definida por el fabricante, existen otros parámetros que tienen que ser definidos obligatoriamente para definir el algoritmo, para este caso en singular, si no se define el punto o los puntos sobre los cuales se aplicaran los apoyos el algoritmo presentara un mensaje de error y nos solicitara ingresar los datos faltantes. Ilustración 44.



La ilustración 43, muestra el icono a seleccionar del tipo de apoyo. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar según sus parámetros.

ILUSTRACIÓN 43 TIPO DE APOYO

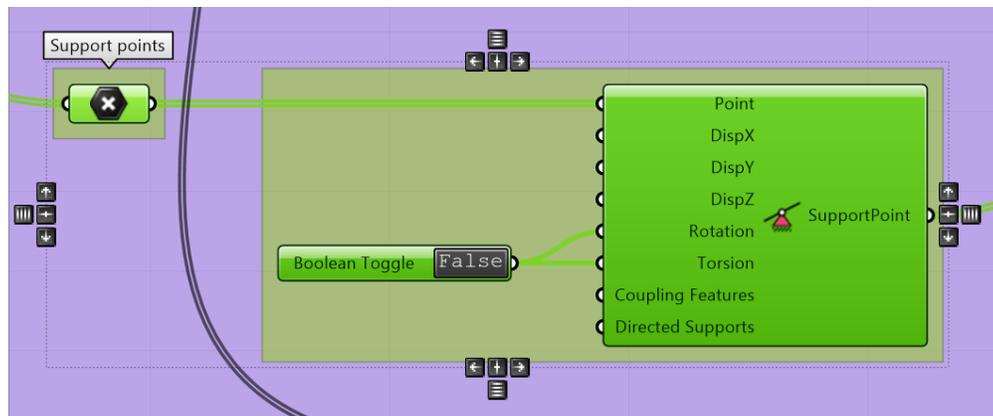
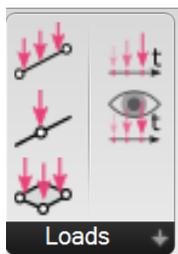


ILUSTRACIÓN 44 DEFINICIÓN DE PARAMETROS EN EL APOYO

Nuestro caso en particular haremos uso de un 'Toggle' el cual lo aplicaremos para activar y desactivar la rotación y la torsión de nuestro apoyo al instante de ejecutar el algoritmo. Ilustración 44.

Para culminar el modelo del análisis es necesario definir las cargas de servicio, para definir estas cargas el programa tiene la opción del tipo de cargas ya sean estas puntuales, uniformes a lo largo de un perfil o sobre una superficie. Ilustración 45.



La ilustración 45, muestra el icono a seleccionar del tipo de carga. Que nos brinda el módulo del Kiwi3D, por medio de este se procede a crear el módulo virtual que nos permitirá programar según sus parámetros.

ILUSTRACIÓN 45 TIPO

DE CARGA

En este caso, como nuestra sección es una barra, el tipo de carga a seleccionar es una carga uniforme a lo largo de la misma.

Los parámetros para definir se describen en la ilustración 46, siendo de manera obligatoria definir la curva sobre la que se va a aplicar nuestra carga, el tipo de cargas a analizar solo es la carga muerta, para nuestro caso de estudio no se ha tomado en cuenta casos de carga como los son las sísmicas o las cargas vivas.

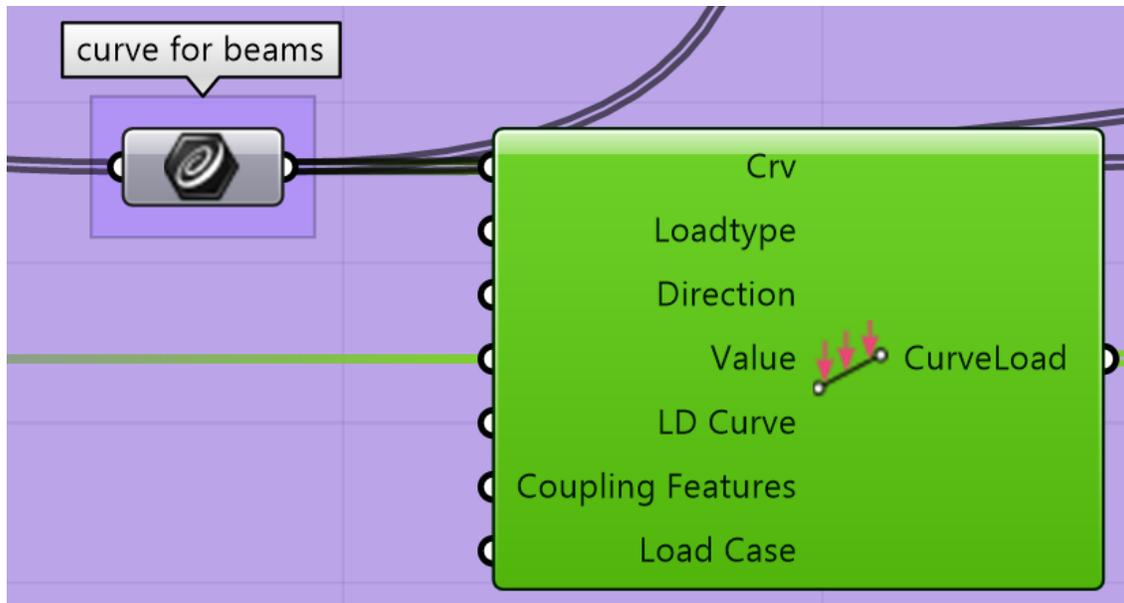


ILUSTRACIÓN 46 DEFINICIÓN DE PARAMETROS DEL TIPO DE CARGA

Para todo este algoritmo es necesario reflejar los resultados, mismos que nos permitirán reflejar de una forma gráfica en el programa Rhinoceros 3D.

Como se puede observar en la Ilustración 47, esto es uno de los parámetros solicitados por IGA solver, los resultados están direccionados en una sola dimensión por lo tanto este será el tipo de modulo que conectaremos a nuestro algoritmo.

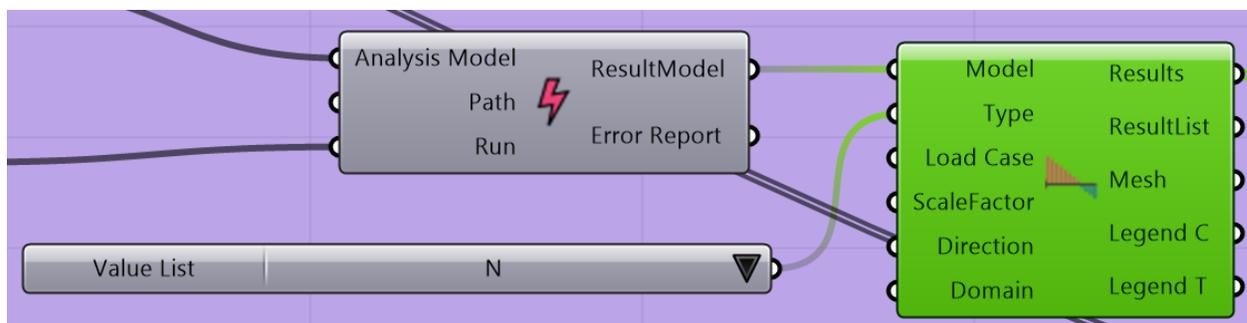


ILUSTRACIÓN 47 PARAMETROS DEL RESULTADO DE IGASOLVER

Existen varios parámetros a definir, en nuestro caso solo se analizará el tipo, para esto necesitamos una lista de valores que se puede generar por el módulo de Grasshopper. Al instante de generar esta lista se presentarán varias opciones como son: la Normal, el

momento 1, el momento 2, el momento total, el cortante 1, el cortante2, entre otros, que de ser el caso deberán ser analizados según su alcance en su proyecto.

Para fines de nuestro alcance solo es necesario analizar la Normal. Ilustración 48.

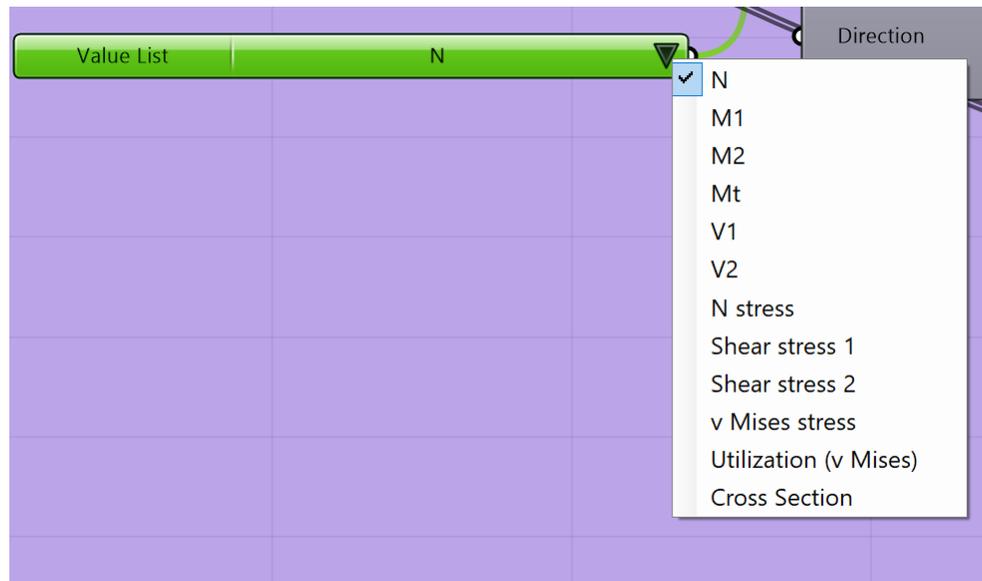


ILUSTRACIÓN 48 TIPO DE ESFUERZOS QUE PUEDE ANALIZAR IGASOLVER

Por tanto, una vez realizado estos pasos y bajo un criterio de ingeniería estructural como los son: el saber definir el tipo de material y sus características físico-mecánicas, el tipo de apoyo que se aplicará a la estructura y el modelo del análisis, se obtendrá un algoritmo, ilustración 49. Esta definirá un sistema de barras sometidas a esfuerzos axiales que brindará una solución de tipo estructural a nuestra geometría. Ilustración 50.

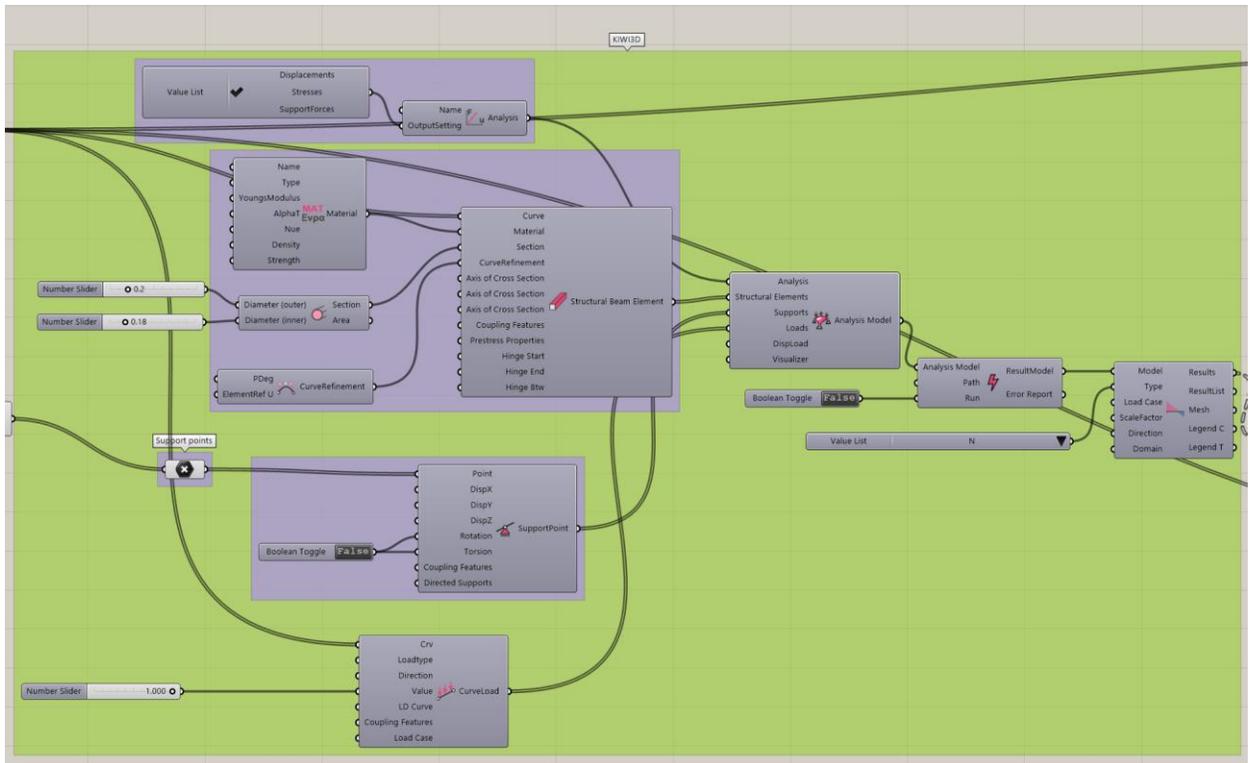


ILUSTRACIÓN 49 ALGORITMO KIWI3D

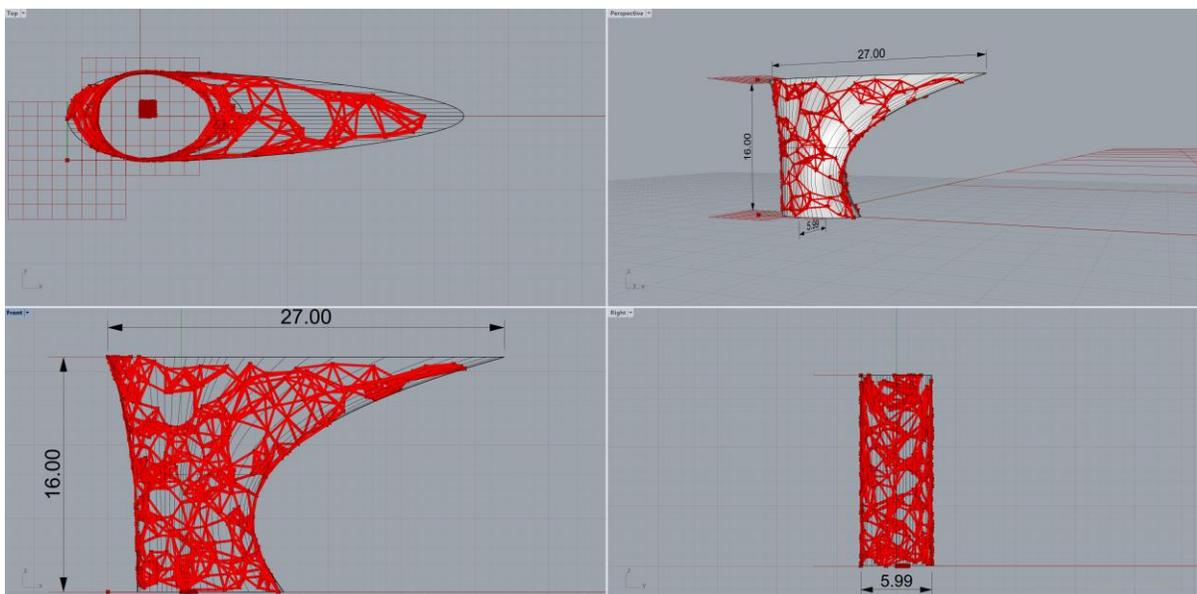


ILUSTRACIÓN 50 RESULTADOS KIWI3D

### 5.3. DEFINICIÓN DEL ALGORITMO USANDO PUFFERFISH

Una vez que se tiene definido un sistema de barras por medio del algoritmo, mismas que están sometidas a esfuerzos axiales para brindar una solución estructural a nuestra geometría, se procede a identificar que barras estas siendo sometidas a mayor esfuerzo.

Para esto es necesario que nuestro algoritmo este identifique cada una de las barras que existe en la geometría y logre identificarlas por unidades, una vez definido esto y por medio del módulo Pufferfish el algoritmo procede a realizar un análisis de elementos finitos por cada barra existente, de este modo y por medio del algoritmo se identificara que barras están sometidas a mayores o menores esfuerzos, de tal manera nuestro algoritmo tiene que brindar una respuesta visual para el operador con la finalidad de que si es necesario se pueda aumentar el número de barras o disminuir si esta lo requiere.

Por lo tanto, para lograr crear nuestro algoritmo es necesario definir ciertos parámetros que los veremos a continuación por medio de este módulo Pufferfish.

Como lo que se quiere analizar es cada una de las barras, se requiere que cada una de estas barras se transforme en una malla para ser analizadas en mínimas partes y conocer cuál es el comportamiento de la barra antes el esfuerzo q está siendo sometida. Para esto se hará uso de una de las herramientas de Pufferfish, 'Parameter Pipe Mesh', misma que exige varios parámetros de diseño como se puede observar en la ilustración 51, para esto tenemos que definir el radio, las curvas sobre las cuales va a actuar nuestro algoritmo y el rango de divisiones que tendrá cada una de las barras que conformen la estructura.

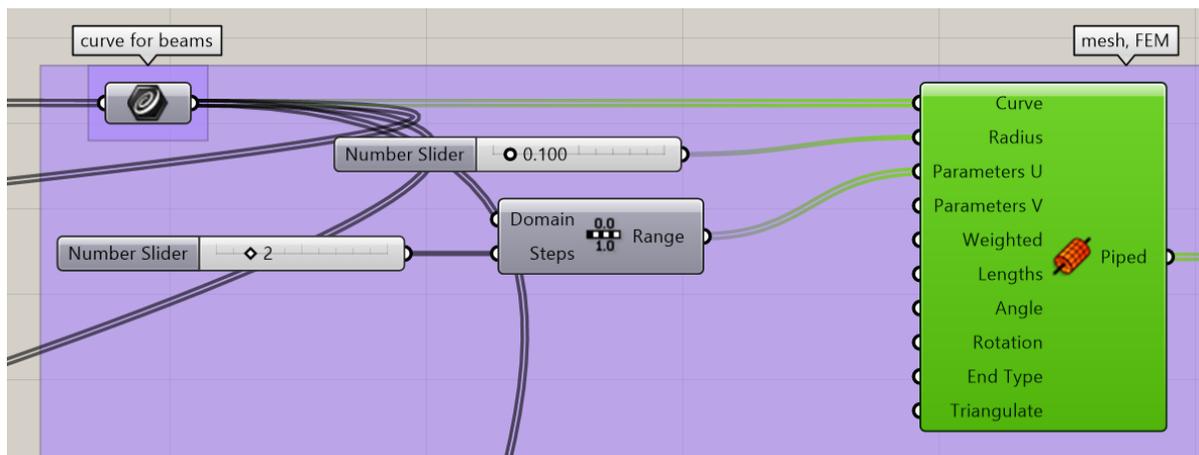


ILUSTRACIÓN 51 PIPE MESH

Con esto se logra obtener un sistema de mallas separadas esto con la finalidad de identificar como está actuando los esfuerzos internos en cada barra y darles tonalidades de colores diferentes según la cantidad de esfuerzo que esté realizando cada barra, para esto se necesita des construir estas mallas por medio de la herramienta 'Deconstruct Mesh' Ilustración 52 y reconstruirla con la herramienta 'Construct Mesh' Ilustración 53, esto se lo hace con la finalidad de lograr colorear las barras al instante que el algoritmo este identificando los esfuerzos por cada barra.

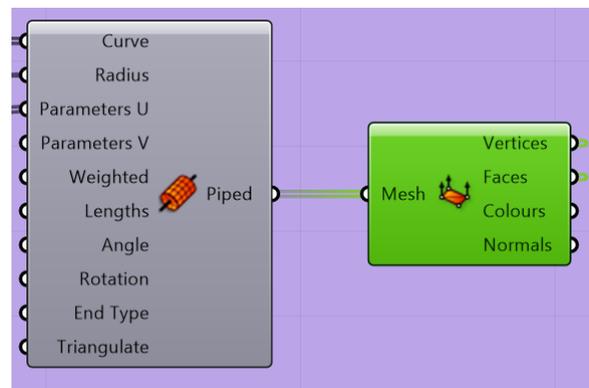


ILUSTRACIÓN 52 DECONSTRUCT MESH

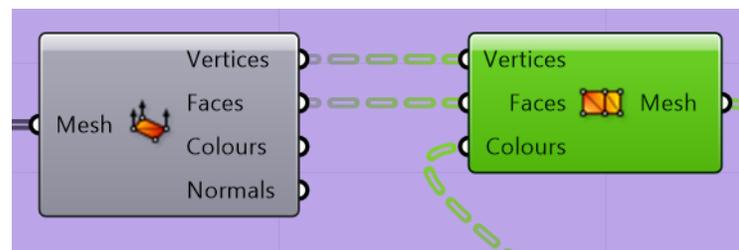


ILUSTRACIÓN 53 CONSTRUCT MESH

A partir de este momento, el algoritmo permite pintar las barras por medio de las herramientas existentes y en conjunto con los resultados que nos brindó el módulo de Kiwi3D.

Los resultados que nos brinda el módulo Kiwi3D deberán ser organizados de tal modo que estos puedan ser interpretados de forma correcta por el algoritmo, a esto nos referimos con que tiene que ser valores absolutos, esfuerzos promedios y seleccionar los valores de mayor a menor dimensión, esta programación se puede apreciar en la ilustración 54.

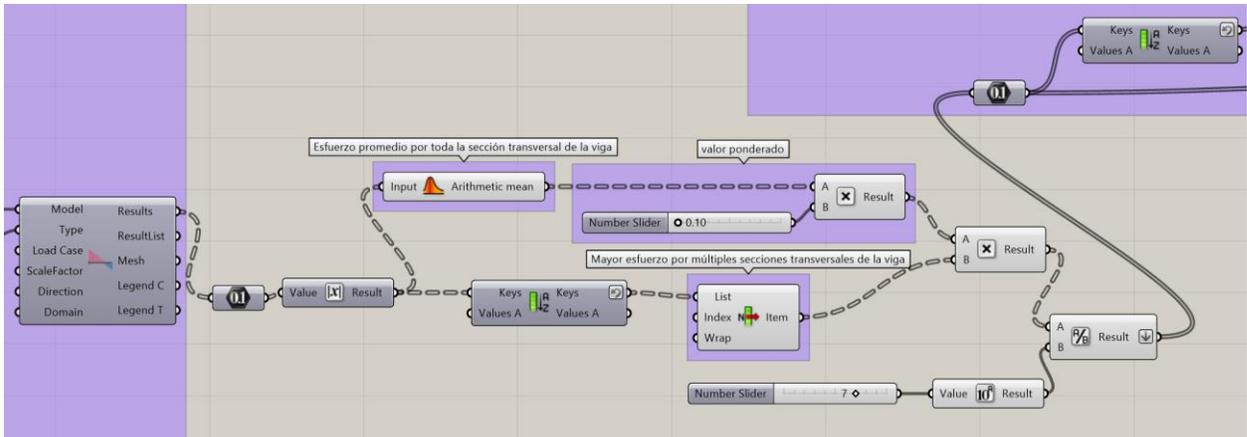


ILUSTRACIÓN 54 ORGANIZACIÓN DE RESULTADOS

Una vez que nuestro algoritmo está definido por los parámetros antes mencionados se procede a decirle al mismo que pinte las barras según su esfuerzo de mayor a menor con tonalidades de colores que varían de los tonos verdes para los valores que estén por debajo del esfuerzo admisible permitido, tonos amarillos para valores que estén en el rango medio de este esfuerzo admisible y finalmente los tonos rojos se aplicaran a los valores que estén por sobre este límite de esfuerzo admisible. Ilustración 55.

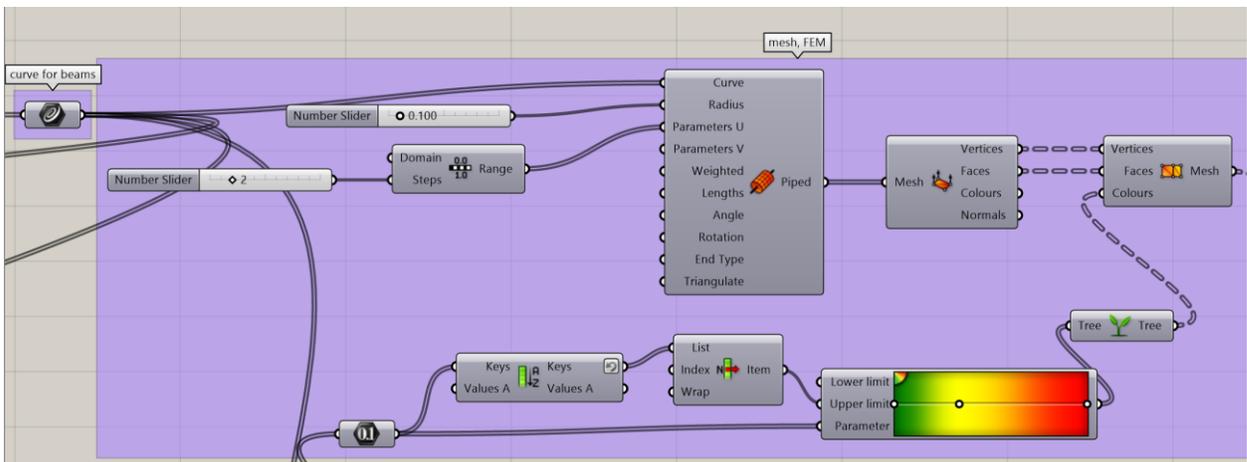


ILUSTRACIÓN 55 PUFFERFISH

Finalmente es necesario unir todas las mallas que se dividieron para lograr determinar estos esfuerzos de una forma gráfica para esto utilizaremos una de las herramientas de Pufferfish 'Mesh Join', ilustración 56.

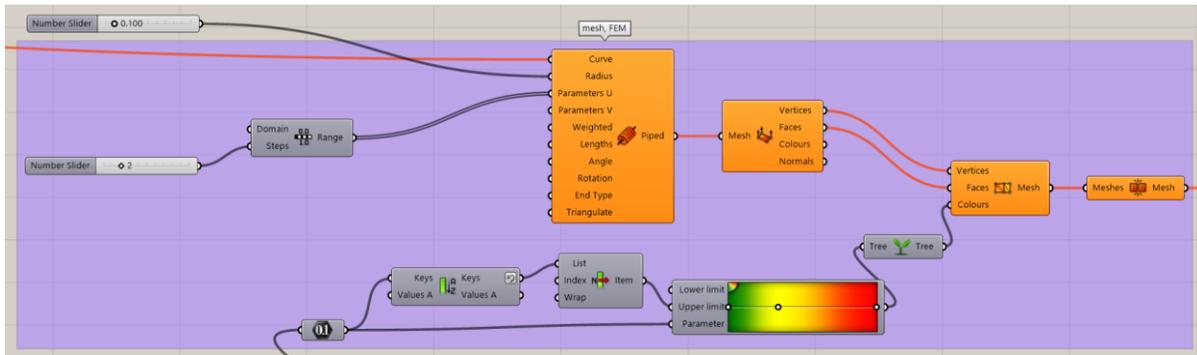


ILUSTRACIÓN 56 MESH JOIN

Con esto se da por finiquitado el análisis y representación gráfica de las barras que conforman la estructura de la geometría. Ilustración 57.

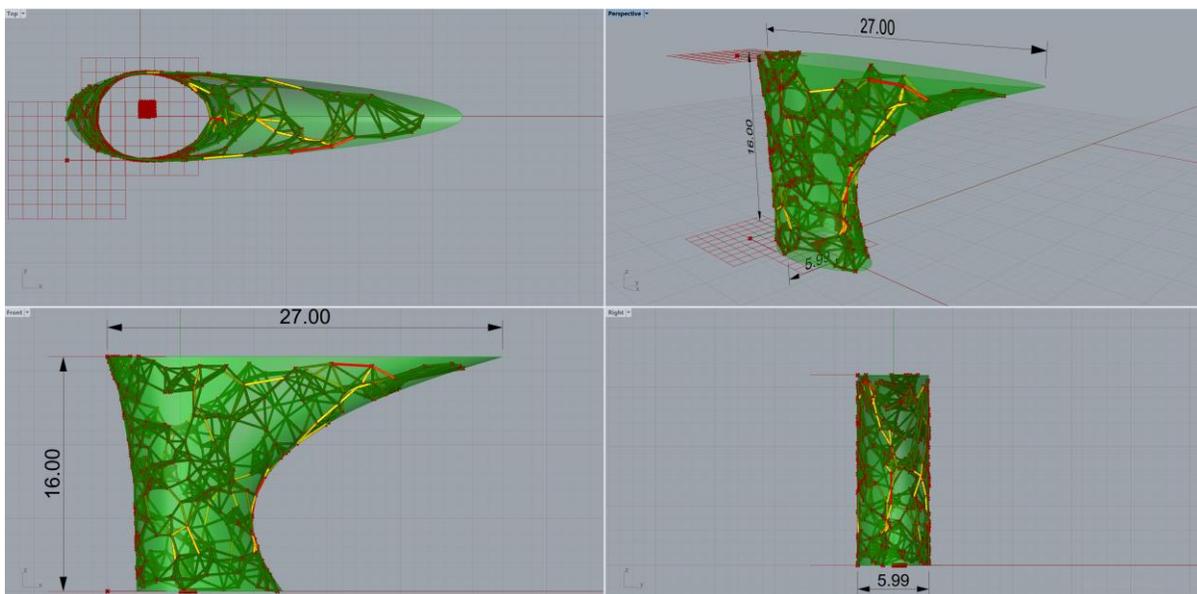


ILUSTRACIÓN 57 RESULTADOS PUFFERFISH Y KIW3D

Como se muestra en la ilustración 57, podemos notar como el algoritmo está actuando de una forma correcta al expresar los resultados de una forma gráfica en la cual el usuario puede distinguir las barras q esta sometidas a mayor o menor esfuerzo, esto permite al usuario poder determinar si es necesario utilizar un mayor número de barras o un menor número.

La cantidad de barras está definida por la cantidad de puntos que se ingresen en el algoritmo esto está definido por una lista aleatoria que es ingresada mediante Grasshopper, en esta lista se ingresarán los parámetros de: cantidad de puntos, el rango de valores y la cantidad de decimales como lo muestra la ilustración 58.

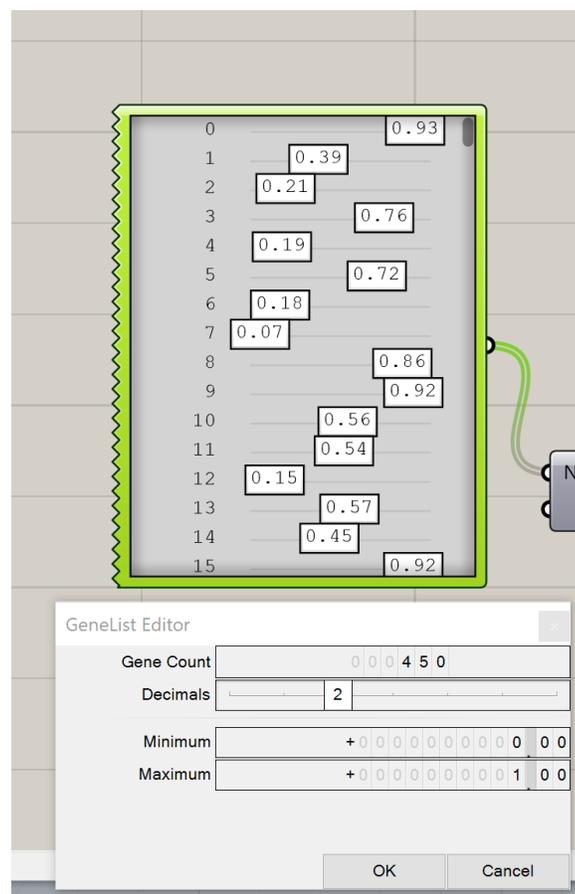


ILUSTRACIÓN 58 DEFINICIÓN DE 450 PUNTOS DE FORMA ALEATORIA

Al ser una selección aleatoria de puntos existe una gran posibilidad de que genere puntos en el mismo lugar geométrico, a mayor cantidad de puntos, mayor posibilidad de que se generen estos puntos repetidos por así llamarlos, para evitar este problema lo

que se hace es indicarle al programa que elimine estos puntos repetidos por medio de uno de los módulos de programación visual de Grasshopper, ilustración 59.

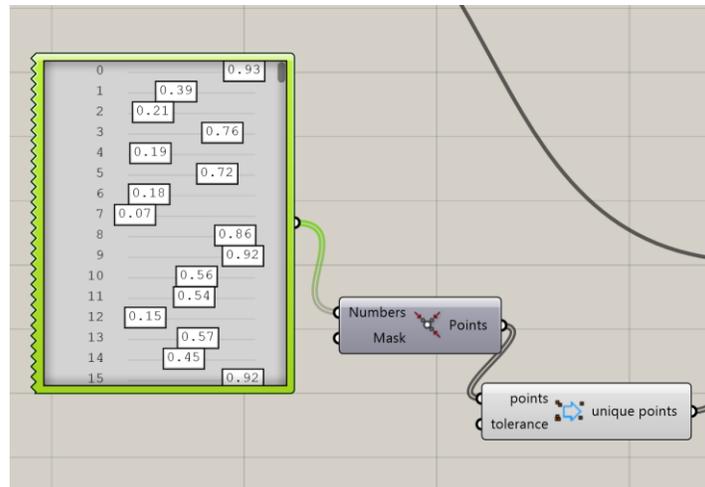


ILUSTRACIÓN 59 SELECCIÓN DE PUNTOS QUE SE SOBREPONEN

Una vez definidos los parámetros de modelación de nuestro algoritmo, obtenemos una solución estructural generada por un programador visual mediante un algoritmo genético. Ilustración 60.

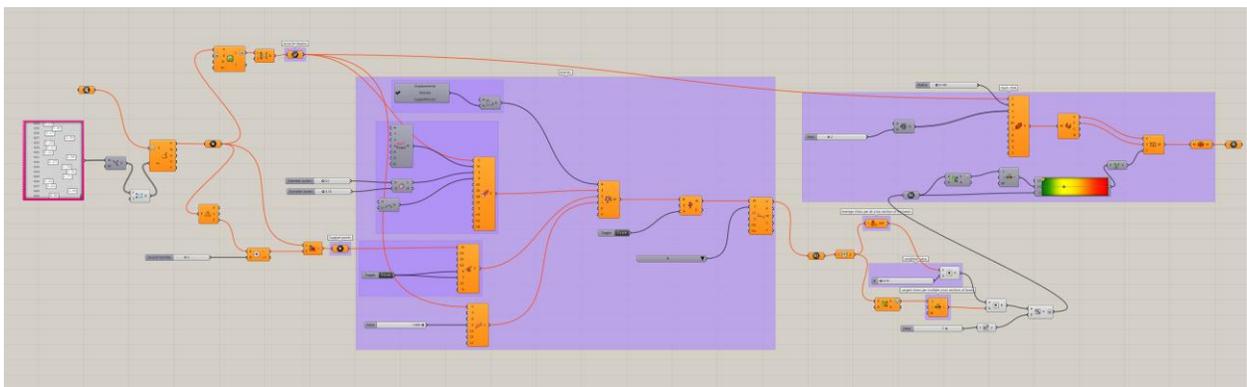


ILUSTRACIÓN 60 ALGORITMO MEDIANTE GRASSHOPPER

A partir de este punto lo que se pretende es optimizar esta estructura tanto en el número de barras como en el largo, esto se lo hará a través de una herramienta de Grasshopper denominada como Octopus.

#### **5.4. DEFINICIÓN DEL ALGORITMO Y OPTIMIZACIÓN DE LA GEOMETRÍA EN ESTUDIO APLICANDO OCTOPUS**

Desde la existencia de las civilizaciones el ser humano ha ido optimizando los recursos que se aplican al instante de ejecutar algún tipo de acción ya sean estos: El tiempo, el material, la fuerza de trabajo, etc.

De tal modo que en este caso de estudio se pretende optimizar una estructura por medio de una de las herramientas que brinda Grasshopper, demostrando que puede generar varios resultados que cumplan con los parámetros definidos en el algoritmo modelado.

En la ilustración 61 se puede observar los resultados del módulo Octopus, estos resultados son el producto de 50/200 iteraciones, Octopus permite 200 iteraciones de un modelo.

En esta representación gráfica que brinda el módulo se puede apreciar los 50 resultados que se encontró, entre estas existen varias respuestas que pueden estar dentro del límite de los esfuerzos admisibles y que cumplen con las licitaciones de carga que se generaron como parámetro en el algoritmo.

Octopus se encarga de encontrar solo las mejores soluciones que cumplan con las solicitudes de carga, de este modo las soluciones generadas son semejantes. Todos los resultados deberán ser analizados uno por uno para, con un criterio profesional seleccionar la que se considere como mejor opción para su diseño final.

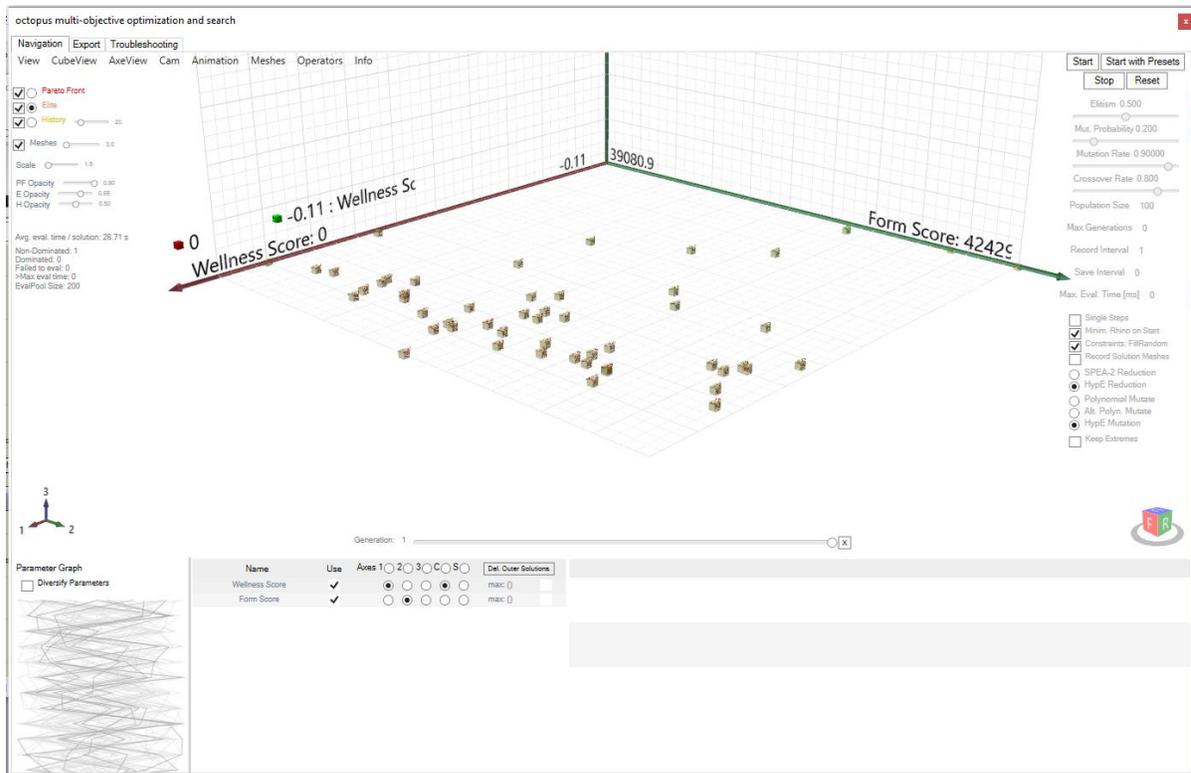


ILUSTRACIÓN 61 RESULTADOS GRAFICOS DE OCTOPUS

Como se puede observar en la ilustración 62, existen modelos de solución que cumplen con la solicitud de esfuerzos mínimos de tal modo que son más el número de barras que conforman la geometría que están siendo sometidas a esfuerzos mayores que en otras soluciones, como ejemplo de esto podemos ver la ilustración 67 y 68, en las cuales se puede comprobar como existen más número de barras con una tonalidad de colores entre rojas y amarillas, mismas que están definiendo la intensidad de esfuerzo que existe en cada barra.

Por lo contrario, si nos dirigimos a la ilustración 70, se puede ver claramente que es una sola la barra que presenta un esfuerzo mayor al resto de barras según nuestro algoritmo.

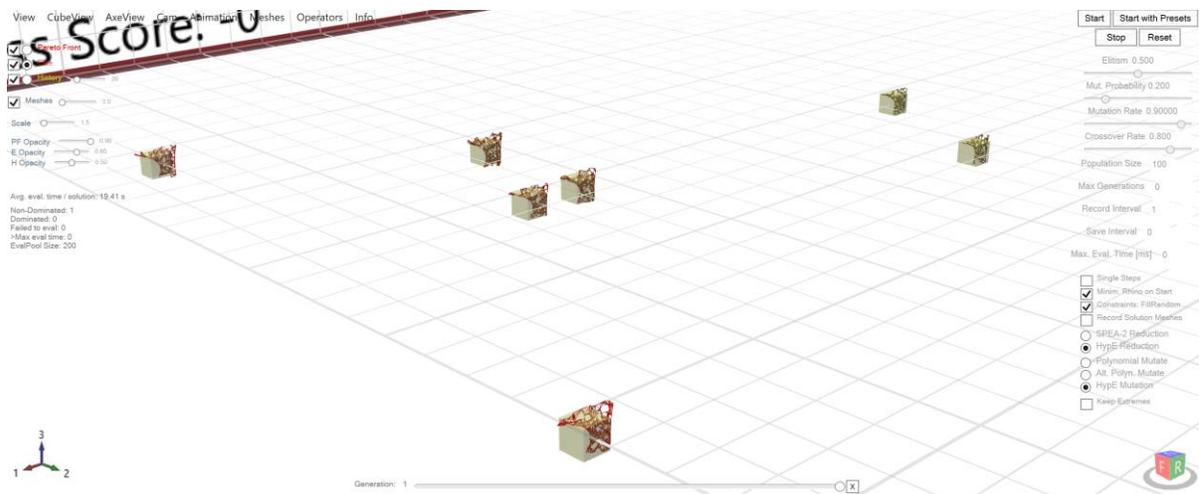


ILUSTRACIÓN 62 RESULTADOS OCTOPUS

De este modo Octopus ira ubicando de forma gráfica y ordenada las soluciones según su desempeño como estructura, en la ilustración 63, es posible definir una de las soluciones más óptimas para el diseño de la estructura, esto será definido de forma visual en el plano generado por Octopus.

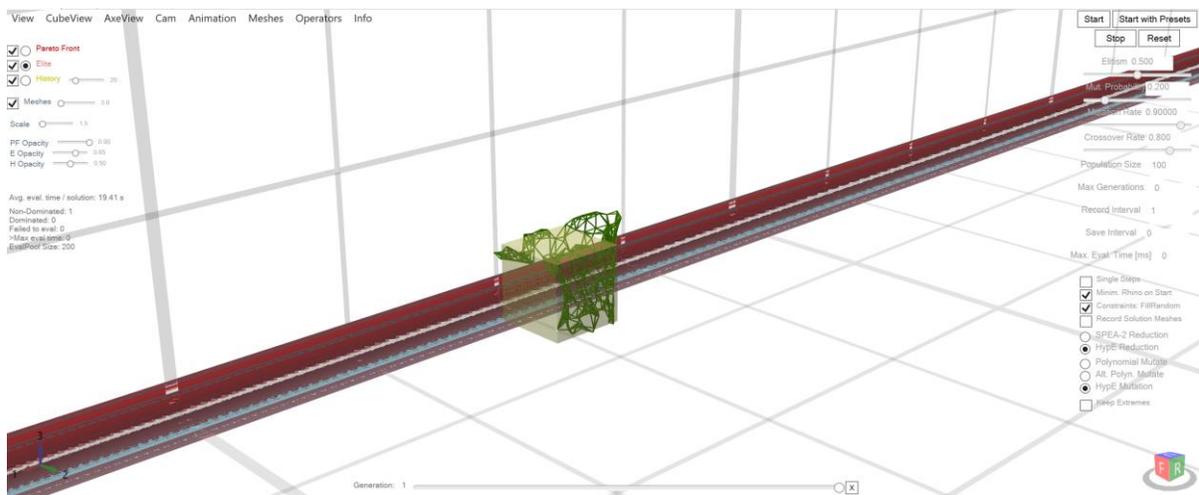


ILUSTRACIÓN 63 RESULTADOS OCTOPUS

Como se mencionó anteriormente no existe una sola solución óptima, ilustración 64.

Dentro de las soluciones que genera Octopus existirán tantas soluciones como el programa logre definir basado en los parámetros de diseño definidos en el algoritmo, estas soluciones sean graficadas por el mismo algoritmo y representadas en Rhinoceros 3D. ilustración 65, ilustración 66.

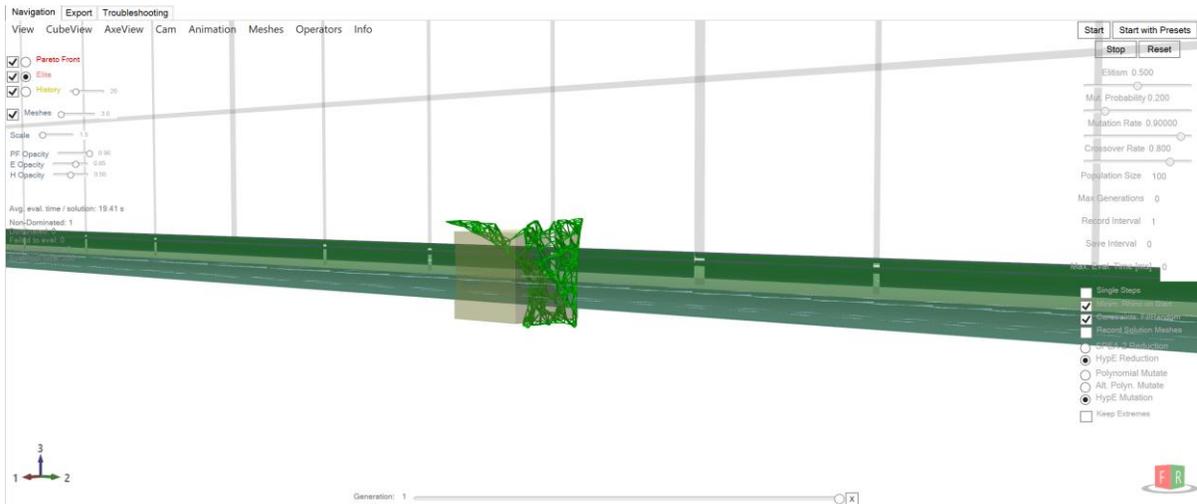


ILUSTRACIÓN 64 RESULTADOS OCTOPUS

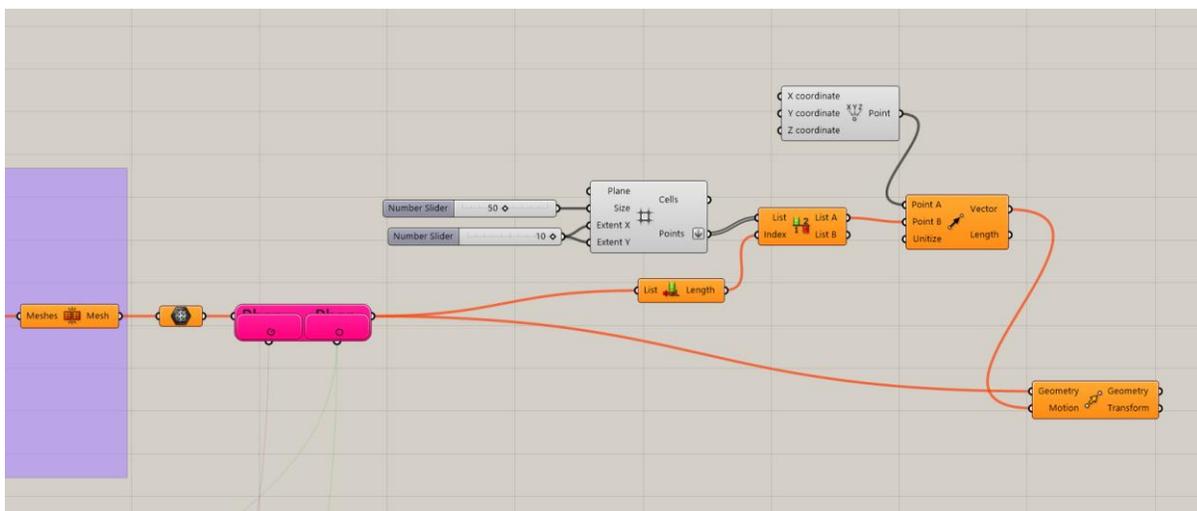


ILUSTRACIÓN 65 INTERFAZ OCTOPUS

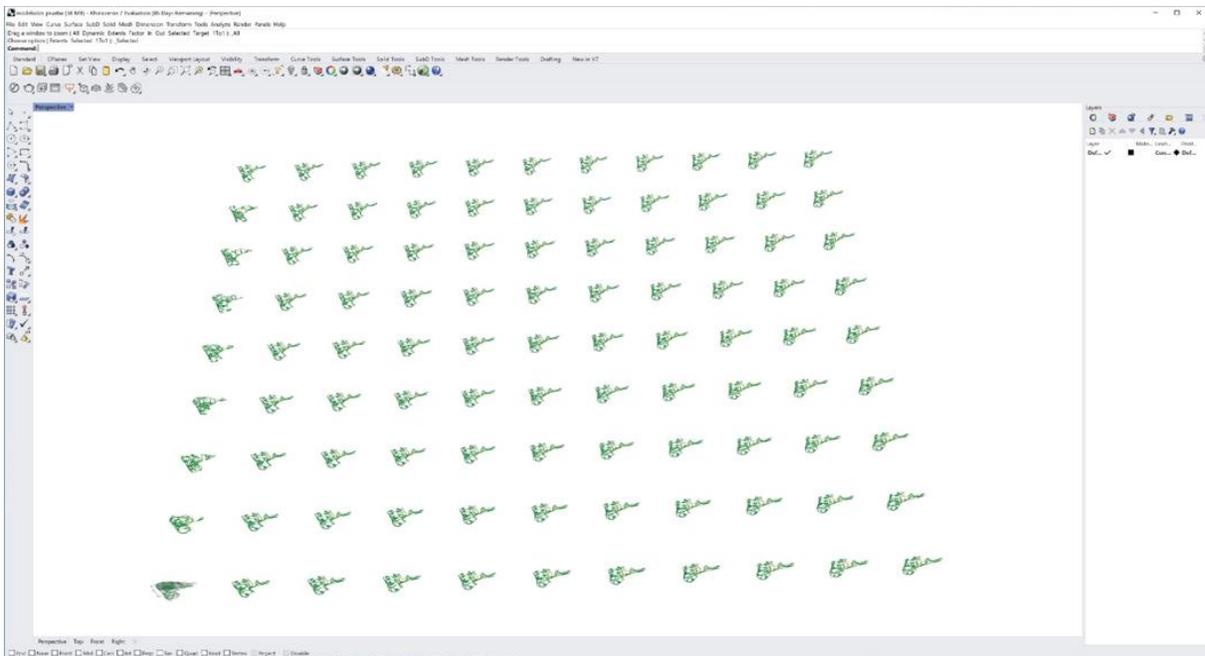


ILUSTRACIÓN 66 RESULTADOS GRAFICOS OCTOPUS EN RHINOCEROS

Una vez que se ha graficado los resultados de Octopus en Rhinoceros 3D, por medio del algoritmo, ilustración 65. Se procederá a implantar esta grafica de forma definitiva mediante el comando Bake (genera de forma gráfica la geometría definida por el algoritmo en Rhino 3D).

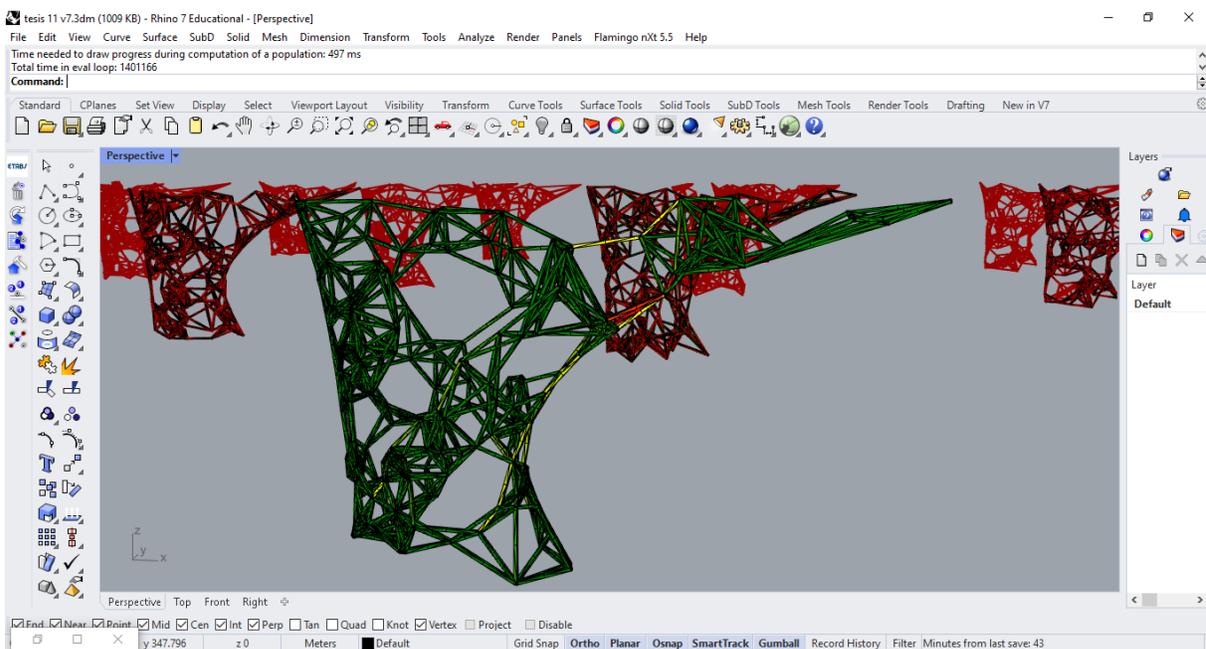


ILUSTRACIÓN 67 RESULTADO GRAFICO OCTOPUS

De esta forma es más fácil identificar para el usuario una solución que se considere como la más óptima.

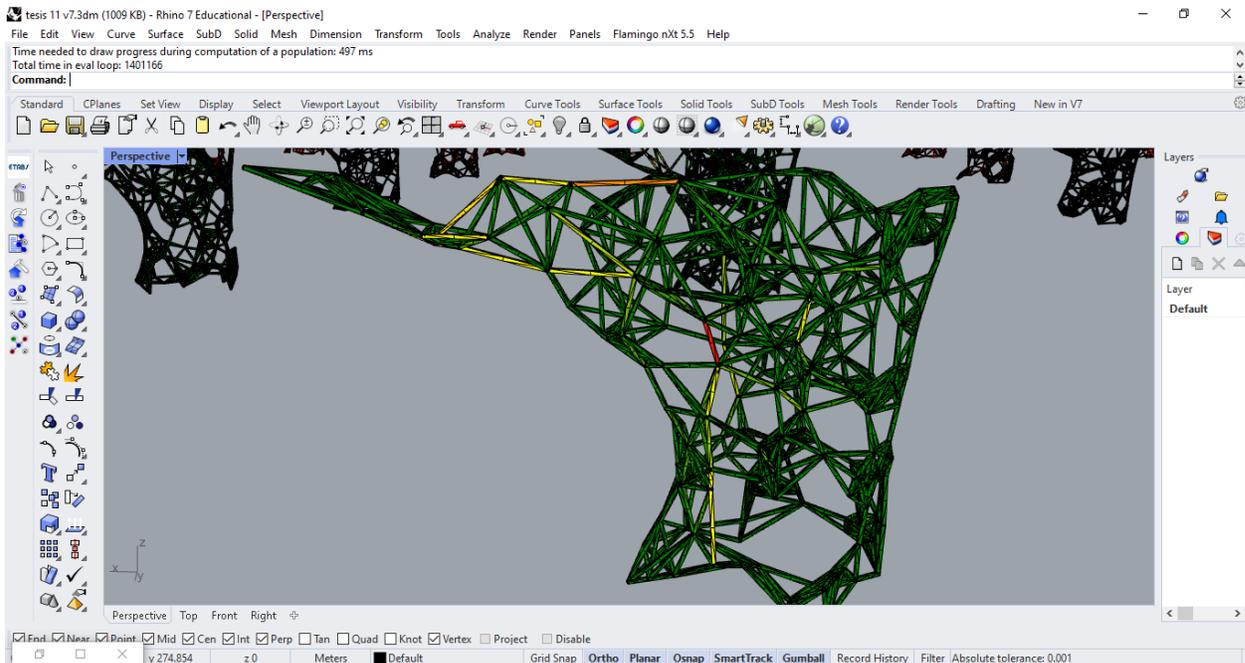


ILUSTRACIÓN 68 RESULTADO GRAFICO OCTOPUS

Una vez que se define una solución que se considerara como la más óptima, se procederá a seleccionarla y separarla del resto de soluciones, esto con la finalidad de poder estudiar de una mejor manera sus elementos estructurales, ya que como es de conocimiento, los parámetros de este diseño son solo el tipo de material, la sección tubular y los radios externo e interno, la ubicación de las barras, el número y sus dimensiones en longitud no están definidas, estas estarán definidas por Octopus, por lo mismo que es el encargado de optimizar el número de barras y su longitud.

La selección del modelo se ha hecho en base al número de barras que está expuesta a mayores esfuerzos, para nuestra selección se ha tomado en consideración la ubicación de estas barras por llamarlas críticas ya que están sometidas a altos esfuerzos axiales, ilustración 70.

A continuación, podemos observar en las ilustraciones 69 – 7, el modelo seleccionado en diferentes perspectivas y se puede apreciar con claridad las barras y su tonalidad de colores que definen la cantidad de esfuerzo que está siendo sometida cada barra.

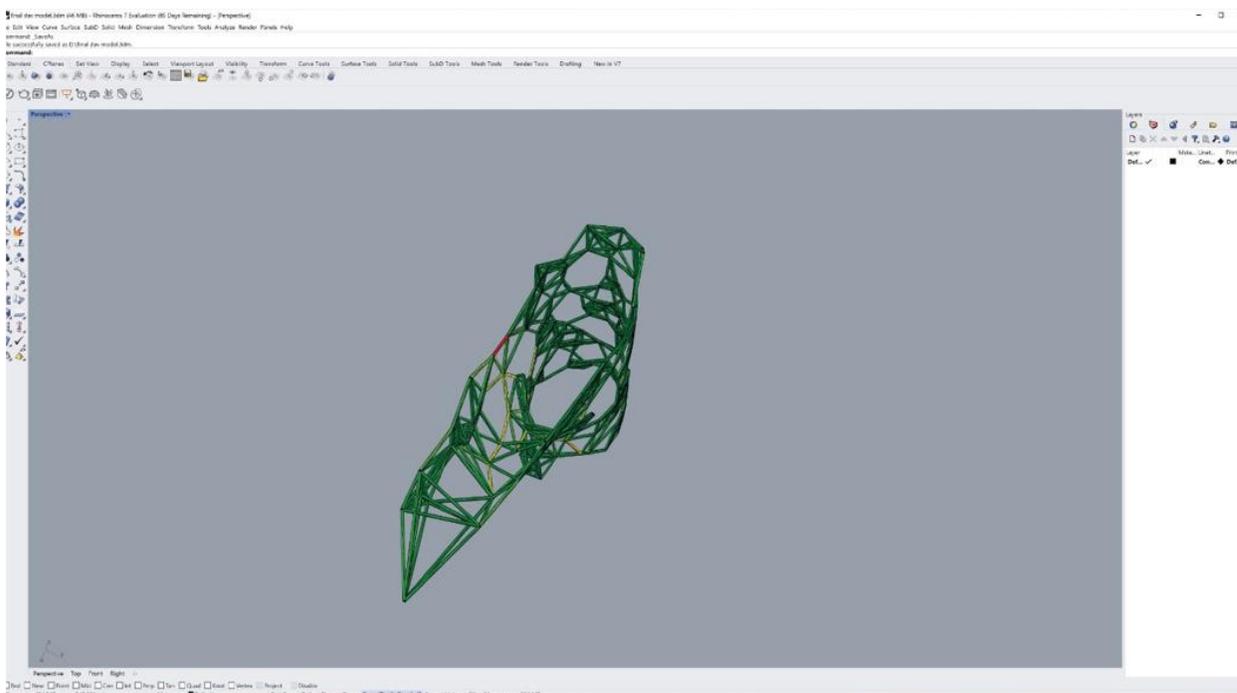


ILUSTRACIÓN 69 RESULTADO OPTIMIZADO OTOPUS VISTA SUPERIOR

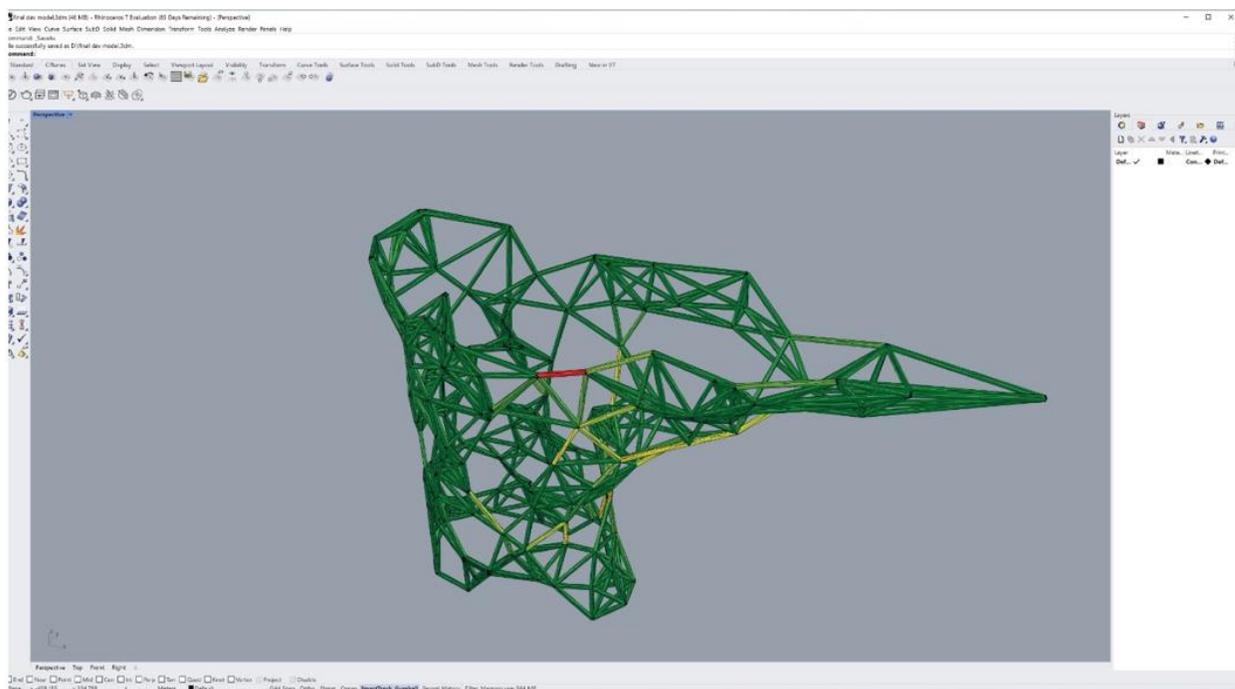


ILUSTRACIÓN 70 RESULTADO OPTIMIZADO OTOPUS VISTA LATERAL

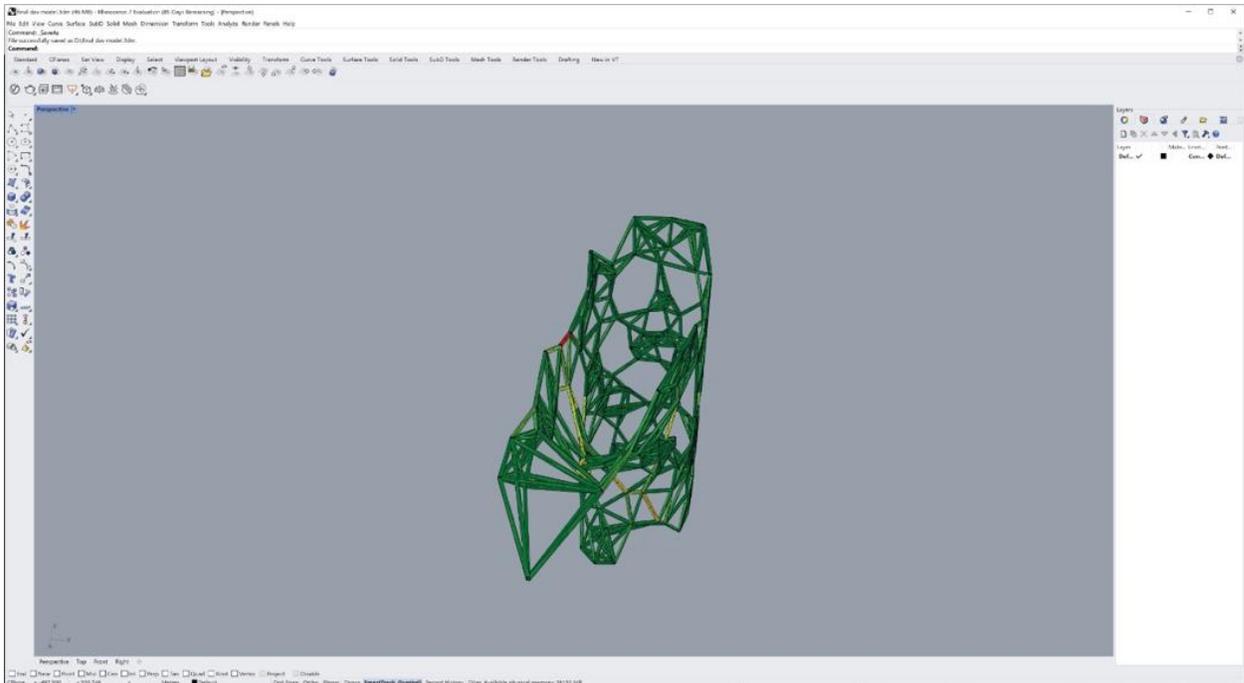


ILUSTRACIÓN 71 RESULTADO OPTIMIZADO OTOPUS VISTA SUPERIOR

Se ha seleccionado esta solución por las siguientes características:

- Ubicación de la barra crítica.
- Número de barras críticas.

Se ha tomado en consideración estos dos parámetros debido a que:

La ubicación de la barra crítica se encuentra en la parte superior de la estructura comprometiendo solo dos nodos y 4 barras, esto permitirá que el cambio de radio en el perfil se mas fácil.

El número de barras críticas al ser mínimo nos ayudara no tener muchos cambios por realizar en la estructura para que esta cumpla con las solicitudes de carga.

Finalmente se presenta un bosquejo de como quedo el algoritmo ilustración 72, mismo que define una solución estructural y las soluciones optimizadas a una geometría parametrizada.

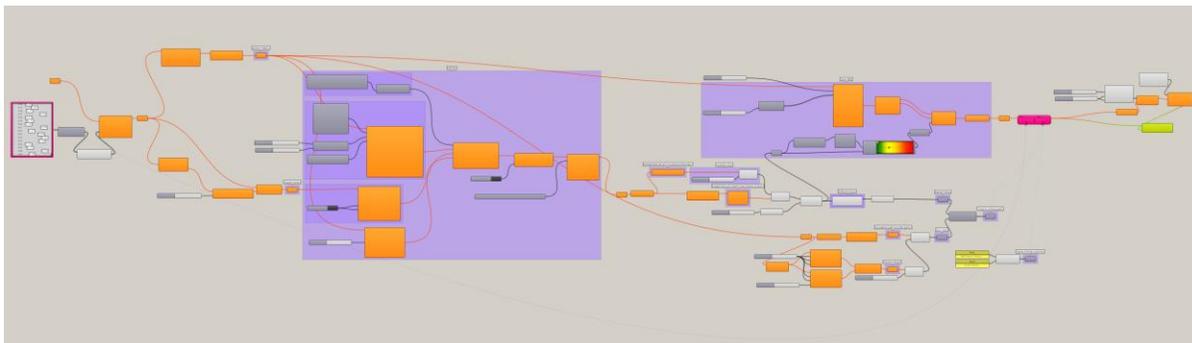


ILUSTRACIÓN 72, ESQUEMA GRAFICO DEL ALGORITMO

## 6. CONCLUSIONES

- Se concluye que es posible crear un algoritmo que genere una solución a estructuras de carácter geométrico no convencional, el cual, a través del uso de un lenguaje de programación visual, hace posible analizar un sistema de barras que permite definir el comportamiento mecánico, individual de cada barra, según los esfuerzos externos a los que este se vea sometido.
- Por medio del algoritmo descrito en esta tesis, se logró generar un resultado gráfico, el cual permite: Identificar y ubicar de forma inmediata las barras que están siendo sometidas a esfuerzos superiores a los admisibles, de esta manera permitiendo al constructor realizar cambios inmediatos en el proceso de diseño, de esta forma permitiendo rediseñar la estructura y logrando optimizar el tiempo que se tardarían analizar esto problemas de la manera tradicional.
- A medida de lo investigado se logró observar que, es necesario un estudio cuantitativo que verse, sobre la cantidad de barras optimizadas, tanto en su longitud por unidad como en la cantidad de barras usadas en la estructura,

debido a que esta investigación tiene como objeto principal la creación de un algoritmo que genere una solución estructural a geometrías no convencionales, demostrar que el uso de nuevas tecnologías aplicadas a la ingeniería pueden optimizar el desempeño de un profesional de la construcción, no se dio énfasis a este estudio cuantitativo que debería ser realizado. Es por esto que se deja sentado la necesidad de dicho estudio cuantitativo, ya que este puede verificar el porcentaje de optimización en la cantidad de material a usar en la construcción de una estructura. Siendo esto así, este estudio serviría de complemento para esta tesis.

- El módulo de Octopus está en la capacidad de brindar soluciones optimizadas, las mismas que trabajan dentro de un rango máximo de 200 iteraciones, estas soluciones tienen que ser seleccionadas y analizadas por un experto entendido en el funcionamiento del programa, siendo esto así, el módulo de Octopus sirve como una herramienta para el profesional, mas no busca reemplazar el criterio y trabajo de un profesional de la construcción
- El desarrollo de esta tesis ha servido para hacer énfasis sobre el proceso de diseño y las herramientas existentes en la actualidad de arquitectura e ingeniería para llevar a cabo diseños de modelos parametrizados, mismos que pueden corroborar con programas existentes décadas antes ya que se puede optimizar en el tiempo de trabajo al hacer uso de estas nuevas herramientas. Desde la creación de la primera civilización, la humanidad ha tenido la necesidad de optimizar, ya sea en espacios, materiales de construcción, tiempos de trabajo, etc. Por este motivo en la actualidad la optimización sigue presente en las mismas formas, con diferentes modos de aplicación debido a la tecnología y avances de la misma se ha logrado obtener proyectos con desarrollo más rápidos, por lo tanto, la optimización es algo q está presente en cada acción que el ser humano quiere ejecutar de una forma correcta, rápida y que sea eficiente. En la actualidad las estructuras están tendiendo a tener formas parametrizadas mismas para las cuales están existiendo herramientas

que permiten diseñarlas, por medio de esta tesis se ha podido corroborar la existencia de las mismas y su aplicación, determinando de este modo los siguientes parámetros:

- Aplicación de la programación visual
  - Determinar una solución estructural por medio de un algoritmo
  - Creación de un algoritmo
  - Identificación de parámetros
  - Selección de modelos optimizados por medio de un programa de optimización multi – objeto
  - Creación de curvas NURBS por medio de un algoritmo
  - Identificación de problemas de diseño por medio del algoritmo
- Desde las investigaciones del método de los elementos finitos, la tecnología a tenido que ir buscando la manera de ir cubriendo estas necesidades de cálculo, a medida que la tecnología avanza este análisis tiende a ser más exacto debido a que podrá descomponer un elemento estructural en lo que se puede definir como infinitas partes, a mayor número de elementos creados por este análisis sobre el elemento base mayor será la capacidad de estudio y análisis que obtendremos, esto nos quiere decir que, podremos tener un análisis más exacto del comportamiento de este elemento siempre y cuando este análisis sea más profundizado y esto dependerá directamente de la potencia de nuestros computadores.

La linealidad o no linealidad de una estructura depende directamente del análisis realizado, todas nuestras estructuras son no lineales, es decir no cumple con la ley de Hooke (esfuerzo deformaciones) ya que al instante de la aplicación de cargas tiende a crear la degradación de la rigidez de la estructura y al eliminar las cargas el material no recupera completamente su forma original.

Por tanto, los métodos de análisis no lineal son de mayor complejidad ya que no solo se considerará la degradación de la rigidez, sino que también se analizara la forma y donde fallara la estructura, a esto nos referimos con la

ubicación de las rotulas plásticas que es el lugar donde se producen este tipo de fallas.

Los métodos lineales son los usados típicamente en las estructuras, por lo que la precisión de sus resultados en los análisis es aceptable. Otra de las ventajas es que los análisis lineales no necesitan computadores de alto rendimiento, por este motivo y por fines de estudio de nuestra tesis se aplicó en nuestro algoritmo un análisis no lineal, mas no con esto se quiere decir que no se pueda crear un algoritmo que realice un análisis no lineal, en la actualidad el rendimiento de los computadores es alto y los análisis no lineales pueden ser definidos con mayor rapidez.

## 7. REFERENCES

- Aguinaga, N. (2021). *LENGUAJES DE PROGRAMACION TEXTUALES Y VISUALES*. Barcelona: cipsa.net CURSOS & MASTER.
- Bathe, K. (1995). *Finite Element Procedures*. USA: Prentice Hall, 2nd edition.
- Bendsoe, M. (1989). Optimal shape design as a material distribution problem. *Struct Multidiscip Optim.* 1, 193 - 202.
- Cerrolaza, M., & Annicchiarico, W. (1996). *Algoritmo de optimizacion estructural basados en simulacion genetica*. Caracas: Universidad Central de Venezuela. Primera Edicion.
- Flake, G. W. (1998). *the computational Beauty of Nature: Computer exploration of fractals, Chaos, Complex Systems, and Adaptation*. ISBN 978-0-262-56127-3.
- García, A. R. (2017). Epidemiología de accidentes de trabajo en Ecuador basado en la base de datos de la Seguridad Social en los años 2014 - 2016. *Scientifica*, 14-18.
- Gil & Parsons. (2014). *The Grasshopper Primer*. London: Thomas Telford. 1st Edition.
- Grona, M. L. (2011). *Fabricacion Digital Y Diseño Parametrico*. Santa fe: Universidad Nacional del Litoral, Santa fe.
- Mario Cesar Velez & Jose Alejandro Montoya. (24 de Julio de 2007). METAHEURISTICOS: UNA ALTERNATIVA PARA LA SOLUCION DE PROBLEMAS COMBINADOS EN ADMINISTRACION DE OPERACIONES. *Revista EIA*, pág. 1.
- McNeel & Associates. (2019). <https://wiki.mcneel.com/rhino/nurbssurfaces>. Obtenido de [https://www.scipedia.com/public/Muttio\\_et\\_al\\_2017a](https://www.scipedia.com/public/Muttio_et_al_2017a): <https://www.rhino3d.com/es/>
- MEREDITH, M. (2008). *Parametric/Algorithmic Architecture*. Barcelona: From Control to Design.
- Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolutions*. New York: Programs. Second extended edition.
- Mirlisenna, G. (22 de Enero de 2016). Metodo de los elementos finitos. *Blog ESSS*, pág. 1.
- Muttio, E., Botello, s., & Tapia, M. (2017). Modelado parametrico mediante programacion visual en el diseno y analisis estructural de edificios. *Revista Mexicana de Metodos Numericos*, 22.

One!3d. (14 de Abril de 2020). *Food4Rhino.com*. Obtenido de Food4Rhino: <https://www.food4rhino.com/app/kiwi3d>

Palmer S.E., Rosch E., & Chase P. (1981). *Canonical Perspective and the Perception of Objects*. USA: L. Erlbaum Associates. .

Pinedo, M. (2002). *Algorithms and Systems*. USA: Prentice-Hall.

RAPPAPORT, N. (2010). Architectural Design. En A. D. Theory, *Architectural Design* (págs. 122-129). Londres: Wiley.

Sanchez, S. (2012). *Optimizacion estructural y topologica de estructuras morfologicamente no identificadas mediante logaritmos geneticos*. Valencia: Universidad Politecnica de Valencia .

Sörensen, Sevaux, & Glover. (2018). *A History of Metaheuristics*. Handbook of Heuristics.

university, R. I. (2020). el parametricismo como modelo erroneo. *arqpress*, 1-2.

Valero, E. F. (2004). *Aportaciones al estudio de las maquinas electricas de flujo axioal mediante la aplicaion del metodo de los elemetos finitos*. Departamento de Ingenieria Electrica. UPC.

Vierlinger. (December de 2012). *Grasshopper*. Obtenido de Octopus: <https://www.grasshopper3d.com/group/octopus?groupUrl=octopus&id=2985220%3A>

WOODBURRY, R. (2010). *Elements of Parametric Design*. New York: Routledge.