

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE ARQUITECTURA Y DISEÑO
FACULTAD DE INGENIERÍA ARQUITECTURA Y DISEÑO
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA

MAESTRÍA Y DOCTORADO EN ARQUITECTURA, URBANISMO Y DISEÑO



Algoritmos y Arquitectura: una Aportación al Diseño

T E S I S

para obtener el grado de

MAESTRO EN ARQUITECTURA, URBANISMO Y DISEÑO

Presenta

AARÓN TADEO ONCHI RASCÓN

Director de Tesis

DR. ALEJANDRO JOSÉ PEIMBERT DUARTE

MEXICALI, BAJA CALIFORNIA

MAYO DEL 2021

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



FACULTAD DE ARQUITECTURA Y DISEÑO
FACULTAD DE INGENIERÍA ARQUITECTURA Y DISEÑO
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA

**MAESTRÍA Y DOCTORADO
EN ARQUITECTURA, URBANISMO Y DISEÑO**

ALGORITMOS Y ARQUITECTURA: UNA APORTACIÓN AL DISEÑO

TESIS PARA OBTENER EL GRADO
DE MAESTRO EN ARQUITECTURA, URBANISMO Y DISEÑO

PRESENTA
AARÓN TADEO ONCHI RASCÓN

Correo electrónico UABC: onchia@uabc.edu.mx

Correo personal: at.onchi@gmail.com

Identificador ORCID: [0000-0003-2167-6758](https://orcid.org/0000-0003-2167-6758)

Director de tesis: Dr. Alejandro José Peimbert Duarte

Identificador ORCID: [0000-0003-4209-0203](https://orcid.org/0000-0003-4209-0203)

Comité tutorial:

M. Arq. Cuauhtémoc Robles Cairo

Identificador ORCID: [0000-0003-2452-4960](https://orcid.org/0000-0003-2452-4960)

Dra. Aurora García García de León

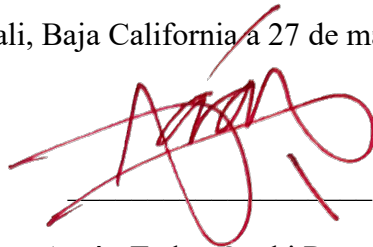
Identificador ORCID: [0000-0003-0583-4998](https://orcid.org/0000-0003-0583-4998)

Código QR vinculado al repositorio virtual del MyDAUD

DECLARACIÓN DE ORIGINALIDAD

Declaro que la tesis que se presenta contiene material original que no ha sido presentado para la obtención de un grado académico o diploma en esta u otra institución de educación superior. Asimismo, declaro que hasta donde yo sé no contiene material previamente publicado o escrito por otra persona excepto donde se reconoce como tal a través de las citas.

Mexicali, Baja California a 27 de mayo de 2021

A handwritten signature in red ink, consisting of several overlapping loops and strokes, positioned above a horizontal line.

Aarón Tadeo Onchi Rascón

AVAL DE TERMINACIÓN DE TESIS

Fecha: 25 de mayo de 2021

DRA. PALOMA RODRÍGUEZ VALENZUELA
Subdirectora de la Facultad de Arquitectura y Diseño
UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Presente.-

Para dar cumplimiento al artículo 106 del ESTATUTO ESCOLAR DE LA UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA, en su TÍTULO CUARTO DE LOS TÍTULOS PROFESIONALES, GRADOS ACADÉMICOS Y DIPLOMAS, CAPÍTULO ÚNICO, sobre los requisitos para la obtención del grado de **Maestría en Arquitectura, Urbanismo y Diseño**, como director de tesis en la Línea de Generación y Aplicación del Conocimiento de Arquitectura y Medio Ambiente de la Facultad de Arquitectura y Diseño. Se emite la presente acta que avala que el estudiante:

AARÓN TADEO ONCHI RASCÓN

ha concluido su tesis de grado titulada:

ALGORITMOS Y ARQUITECTURA: UNA APORTACIÓN AL DISEÑO

Lo anterior en virtud de que se realizaron las correcciones y recomendaciones del comité tutorial, por lo que ha sido aprobada para el proceso de presentación de examen de grado.

DIRECTOR DE TESIS

DR. ALEJANDRO JOSÉ PEIMBERT DUARTE

FIRMA

DEDICATORIA

DEDICO EL PRESENTE TRABAJO DE INVESTIGACIÓN A MIS PADRES Y HERMANOS, SIN ELLOS ESTE TRABAJO NO HUBIERA SIDO POSIBLE. TAMBIÉN A MI PAREJA M.ARQ. GLORIA GABRIELA ALCARAZ ADAME POR SU INCONDICIONAL APOYO.

AGRADECIMIENTOS

AGRADECIMIENTOS AL CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA POR EL APOYO FINANCIERO PARA LA REALIZACIÓN DEL PROGRAMA DE POSGRADO EN INVESTIGACIÓN; AL PROGRAMA DE MAESTRÍA Y DOCTORADO EN ARQUITECTURA, URBANISMO Y DISEÑO POR LA FORMACIÓN Y SOPORTE EN EL DESARROLLO DEL PRESENTE TRABAJO DE INVESTIGACIÓN, EN ESPECIAL AL COORDINADOR DEL MYDAUD DR. GONZALO BOJORQUEZ MORALES.

A MI DIRECTOR DE TESIS DR. ALEJANDRO JOSÉ PEIMBERT DUARTE POR EL SOPORTE Y LA GUÍA EN EL DESARROLLO DEL PRESENTE TRABAJO. A MI COMITÉ TUTORAL CONFORMADO POR EL M. ARQ. CUAUHTÉMOC ROBLES CAIRO Y LA DRA. AURORA GARCÍA GARCÍA DE LEÓN POR EL IMPORTANTE APOYO RECIBIDO CON SUS OBSERVACIONES Y CONSEJOS. AL DR. FERNANDO MENESES-CARLOS POR SUS COMENTARIOS Y EL APOYO COMO REVISOR EXTERNO.

A LOS PROFESORES QUE ME BRINDARON CLASES, EN ESPECIAL AL M. ARQ. ANTONIO LEY GUING Y LA DRA. CRISTINA SOTELO SALAS POR LA INFORMACIÓN, RECURSOS BIBLIOGRÁFICOS Y REFERENCIAS QUE FUERON FUNDAMENTALES EN LA REALIZACIÓN DE LA INVESTIGACIÓN.

RESUMEN

ALGORITMOS Y ARQUITECTURA: UNA APORTACIÓN AL DISEÑO

La presente investigación busca desarrollar una metodología de diseño algorítmico y paramétrico (AP) que auxilie en la resolución del espacio arquitectónico. Introduce y guía al arquitecto en el desarrollo algorítmico con el uso de *Grasshopper* como entorno de programación visual; a fin de verificar la aportación de la tecnología computacional en la optimización de los objetos arquitectónicos.

La metodología propuesta funciona como una herramienta de diseño que incorpora parámetros a considerar bajo dos líneas de análisis: (1) cualitativa, orientada a la solución fenomenológica y semiológica del espacio, (2) cuantitativa, enfocada en la adaptabilidad ambiental y la optimización estructural de un proyecto. Posteriormente se hace un estudio comparativo entre una metodología AP y una tradicional de diseño arquitectónico, con aplicación en el taller de Diseño Arquitectónico 1 y la asignatura Nociones de Arquitectura y Urbanismo Paramétrico de la Facultad de Arquitectura y Diseño de la UABC.

La metodología AP permite al arquitecto explorar complejidad en el diseño arquitectónico y mayor capacidad para optimizar el desempeño de un proyecto a través de los recursos computacionales. Sin embargo, la investigación también muestra que la curva de aprendizaje de la programación digital visual orientada al diseño de objetos es amplia y requiere de práctica.

Palabras clave: Arquitectura computacional, Metodología de diseño, Programación digital visual, Arquitectura Algorítmica, Diseño Paramétrico.

ABSTRACT

ALGORITHMS AND ARCHITECTURE: A CONTRIBUTION TO DESIGN

This research seeks to develop an Algorithmic and Parametric (AP) design methodology that helps in the resolution of architectural space. Introduces and guides the architect in algorithmic development with the use of Grasshopper as a visual scripting environment; to verify the contribution of computational technology in the optimization of architectural objects.

The proposed methodology works as a design tool that incorporates parameters to consider under two lines of analysis: (1) qualitative, oriented to the phenomenological and semiological solution of space, (2) quantitative, focused on environmental adaptability and structural optimization of a project. Subsequently, a comparative study is made between an AP methodology and a traditional one of architectural design, with application in the workshop Architectural Design I and the course Notions of Parametric Architecture and Urbanism of the Faculty of Architecture and Design of the UABC.

The AP methodology allows the architect to explore complexity in architectural design and greater ability to optimize project performance through computational resources. However, research also shows that the learning curve for object-oriented visual scripting is wide and requires practice.

Keywords: *Computational Architecture, Design Methodology, Visual Scripting, Algorithmic Architecture, Parametric Design.*

ÍNDICE DE FIGURAS

<p><i>Figura 1.</i> Proceso de diseño con el uso de la computadora a manera de “producción” o “computarización”; se aprovecha la computadora como una herramienta representacional. Elaboración propia (2019).</p>	3
<p><i>Figura 2.</i> Proceso de diseño con el uso de la computadora a manera de “concepción” o “computación”; se aprovecha la computadora como inteligencia auxiliar en la generación de ideas y objetos de diseño. Elaboración propia (2019).</p>	3
<p><i>Figura 3.</i> Estructura de preguntas y objetivos de investigación. Elaboración propia (2019).. 10</p>	10
<p><i>Figura 4.</i> El algoritmo como traductor o mediador entre el pensamiento humano y el procesador de una computadora. Elaboración propia (2019).</p>	12
<p><i>Figura 5.</i> Ejemplo de algoritmo expresado en diagrama de flujo. Elaboración propia (2019).</p>	13
<p><i>Figura 6.</i> <i>Serpentine Gallery Pavilion</i> de Toyo Ito, proyecto construido (Londres, 2002). Fuente: Ito, T. (2002). <i>Serpentine Gallery Pavilion 2002</i> [Imágen]. Recuperado de http://www.toyo-ito.co.jp/WWW/Project_Descript/2000-/2000-p_08/2000-p_08_en.html. <i>Copyright</i> [2002] © Nacása & Partners Inc.....</p>	14
<p><i>Figura 7.</i> <i>The Morning Line</i> de Aranda & Lasch, proyecto construido (Sevilla, 2008). Fuente: Aranda, B. y Lasch, C. (2008). <i>The Morning Line</i> [Imágen]. Recuperado de http://arandalasch.com/works/the-morning-line/.....</p>	14
<p><i>Figura 8.</i> <i>Mercedes-Benz Museum</i> de UNStudio, proyecto construido (Stuttgart, 2006). Fuente: van Berkel, B. (2006). <i>Mercedes-Benz Museum</i> [Imágen]. Recuperado de http://www.unstudio.com/en/page/12482/mercedes-benz-museum. <i>Copyright</i> [2006] © Eva Bloem.....</p>	16
<p><i>Figura 9.</i> <i>Queen Alia International Airport</i> de Foster + Partners, proyecto construido (Amman, 2012). Fuente: Foster, N. (2012). <i>Queen Alia International Airport</i> [Imágen].</p>	

Recuperado de https://www.fosterandpartners.com/projects/queen-alia-international-airport/ .	
<i>Copyright</i> [2012] © Foster + Partners	16
<i>Figura 10.</i> Las cuatro eras de la relación entre diseño y computación: Diseño Asistido por Computadora (CAD), según Wujec (2017). Elaboración propia (2019).	19
<i>Figura 11.</i> <i>Cardiff Bay Opera House</i> de Greg Lynn, proyecto para concurso (Gales, 1994). Fuente: Lynn, G. (1994). <i>Cardiff Bay Opera House</i> [Imágen render]. Recuperado de http://glform.com/buildings/cardiff-bay-opera-house/ . <i>Copyright</i> [1994] © Greg Lynn FORM	25
<i>Figura 12.</i> <i>Heydar Aliyev Centre</i> de Zaha Hadid Architects, proyecto construido (Azerbaijan, 2012). Fuente: Hadid, Z. (2012). <i>Heydar Aliyev Centre</i> [Imágen]. Recuperado de https://www.zaha-hadid.com/architecture/heydar-aliyev-centre/ . <i>Copyright</i> [2012] © Iwan Baan	26
<i>Figura 13.</i> Diagrama descriptivo de las cuatro eras de la relación entre diseño y computación: Diseño Asistido por Computadora (CAD), contextualizado dentro de las cuatro Revoluciones Industriales según Wujec (2017) y las eras digitales de la Arquitectura según Carpo (2017). Elaboración propia (2019).	29
<i>Figura 14.</i> <i>Research Pavilion 2011</i> de ICD/ITKE <i>University of Stuttgart</i> , proyecto liderado por Achim Menges y construido (Stuttgart, 2011). Fuente: Menges, A. (2011). ICD/ITKE <i>Research Pavilion 2011</i> [Imágen]. Recuperado de http://www.achimmenges.net/?p=5123 . <i>Copyright</i> [2011] © ICD/ITKE <i>University of Stuttgart</i>	34
<i>Figura 15.</i> <i>BUGA Wood Pavilion 2019</i> de ICD/ITKE <i>University of Stuttgart</i> , proyecto liderado por Achim Menges y construido (Heilbronn, 2019). Fuente: Menges, A. (2019). ICD/ITKE <i>Research Pavilion 2019</i> [Imágen]. Recuperado de http://www.achimmenges.net/?p=20987 . <i>Copyright</i> [2019] © ICD/ITKE <i>University of Stuttgart</i>	34

- Figura 16. Tallin Architecture Biennale Pavilion* de Gilles Retsin, proyecto construido (Tallin, 2017). Fuente: Retsin, G. (2017). *Tallin Architecture Biennale Pavilion* [Imágen]. Recuperado de <https://www.retsin.org/Tallinn-Architecture-Biennale-Pavilion>.35
- Figura 17. Bloom, 2012 Olympic Games*, pabellón personalizable de José Sanchez y Alisa Andrasek, proyecto construido (Londres, 2012). Fuente: Sanchez, J. y Andrasek A. (2012). *Bloom* [Imagen]. Recuperado de <https://www.plethora-project.com/bloom>.35
- Figura 18. Gemini* de Neri Oxman, proyecto construido (Paris, 2014). Fuente: Oxman, N. (2014). *Gemini* [Imágen]. Recuperado de <http://neri.media.mit.edu/projects/details/gemini>. Fotografía: Michel Figuet. *Copyright* [2018] © Neri Oxman42
- Figura 19. Mínima | Maxima World Expo Pavilion 2017* de TheVeryMany (Marc Fornes), proyecto construido (Astana, Kazakhstan, 2017). Fuente: Fornes, M (2017). *Mínima | Maxima World Expo Pavilion 2017* [Imagen]. Recuperado de <https://theverymany.com/world-expo-17-astana-kz-1>. Fotografía: NAARO. *Copyright* [2017] © NAARO.43
- Figura 20. Combinational Nest, Tallin Architecture Biennale Pavilion* de José Sánchez, proyecto (Tallin, 2019). Fuente: Sánchez, J. (2019a). *Combinational Nest* [Render]. Recuperado de <https://www.plethora-project.com/combinatorial-nest>.....43
- Figura 21. Nuremberg Concert Hall* de Gilles Retsin, Proyecto (Nuremberg, 2018). Fuente: Retsin, G. (2018). *Nuremberg Concert Hall* [Render]. Recuperado de <https://www.retsin.org/Nuremberg-Concert-Hall>.44
- Figura 22. Estructura del estado del arte de la Arquitectura Algorítmica Paramétrica.* Elaboración propia (2019).46
- Figura 23. Posicionamiento geográfico de la Arquitectura Algorítmica Paramétrica.* Elaboración propia (2019).48

<i>Figura 24.</i> Estructura del marco teórico de la <i>Autopoiesis of Architecture</i> de Schumacher, P. (2011, 2012). Elaboración propia (2019).....	51
<i>Figura 25.</i> Estructura metodológica del Parametricismo de Schumacher, P. (2009, 2012). Elaboración propia (2019).	53
<i>Figura 26.</i> Estructura metodológica de la Arquitectura Algorítmica de Terzidis, K. (2006). Elaboración propia (2019).	56
<i>Figura 27.</i> Aksamija, A. (2016, p. 176). Proceso y método de investigación performativo [diagrama]. Modificación: elaboración propia (2020).....	58
<i>Figura 28.</i> Complemento a proceso y método de investigación performativo de Aksamija (2016, p. 176) [diagrama]. Elaboración propia (2020).	60
<i>Figura 29.</i> Aksamija, A. (2016, p. 175). Modelos de prácticas de investigación performativa - relación con las firmas de diseño arquitectónico- [diagrama]. Modificación: elaboración propia (2020).....	61
<i>Figura 30.</i> Aksamija, A. (2016, p. 81). Proceso de proyecto de Investigación, de Perkins+Will y la Universidad de Cincinnati, 2012 [diagrama].....	62
<i>Figura 31.</i> Wujec, T. (2017, p. 31). Pasos generales de la fabricación digital [diagrama]. Modificación: elaboración propia (2020).	63
<i>Figura 32.</i> Los tres pasos del proceso de fabricación digital en la arquitectura, según el libro <i>Digital Fabrication in Architecture</i> (Dunn, 2012). Elaboración propia (2019).....	64
<i>Figura 33.</i> Stadler, M. M. (2002, p. 66). Diagrama de los siete puentes de Königsberg (izquierda) y Grafo de Euler como demostración topológica a los siete puentes de Königsberg (derecha) [diagrama].....	70
<i>Figura 34.</i> Norberg-Schulz, C. (1971, pp. 21, 22). Esquema topológico del <i>centro</i> y el <i>lugar</i> de las propiedades elementales del espacio existencial [diagrama].....	70

- Figura 35.* Norberg-Schulz, C. (1971, pp. 26, 27). Esquema topológico de la *dirección* y el *camino* de las propiedades elementales del espacio existencial [diagrama]..... 71
- Figura 36.* Norberg-Schulz, C. (1971, p. 28). Esquema topológico del *área* y la *región* de las propiedades elementales del espacio existencia [diagrama]..... 71
- Figura 37.* Aspectos topológicos del espacio cognoscitivo y artístico de Norberg-Schulz (1999; 1971). Elaboración propia (2020)..... 72
- Figura 38.* Estímulos en los sistemas sensoriales de la experiencia espacial de Bloomer y Moore (1983). Elaboración propia (2020). 74
- Figura 39.* *Pabellón Swiss Sound* (Hanover, 2000). Fuente: Zumthor, P. (2000). *Pabellón Swiss Sound* [Imágen]. Recuperado de <https://es.wikiarquitectura.com/edificio/pabellon-swiss-sound/>..... 76
- Figura 40.* *Pabellón CLT Park Harumi* (Tokio, 2019). Fuente: Kuma, K. (2019a). *CLT Park Harumi* [Imágen]. Recuperado de <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-02.jpg>.
Copyright [2019] © Kawasumi • Kobayashi Kenji Photograph Office..... 83
- Figura 41.* *Pabellón CLT Park Harumi* (Tokio, 2019). Fuente: Kuma, K. (2019b). *CLT Park Harumi* [Imágen]. Recuperado de <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-05.jpg>.
Copyright [2019] © Kawasumi • Kobayashi Kenji Photograph Office..... 84
- Figura 42.* *Pabellón Climbable Wooden* (Paris, 2015). Fuente: Kuma, K. (2015a). *Climbable Wooden Pavilion* [Imagen]. Recuperado de https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_11.jpg. 84
- Figura 43.* *Pabellón Climbable Wooden* (Paris, 2015). Fuente: Kuma, K. (2015b). *Climbable Wooden Pavilion* [Imagen]. Recuperado de

https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_31.jpg	85
<i>Figura 44.</i> Parámetros y estrategias para el análisis estructural de la envolvente arquitectónica. Elaboración propia (2020).....	90
<i>Figura 45.</i> Diagrama de parámetros generales para el análisis termo-energético de la envolvente arquitectónica. Elaboración propia (2019).	91
<i>Figura 46.</i> Representación esquemática de los elementos sintácticos de un algoritmo. Recuperado de AAD_ <i>Algorithmis-Aided Design: Parametric Strategies Using Grasshopper</i> , Tedeschi (2014, p. 23). Adaptación y modificación: elaboración propia (2020).	96
<i>Figura 47.</i> Elementos sintácticos de <i>Grasshopper</i> : el parámetro y la función (componente). Elaboración propia (2020).	97
<i>Figura 48.</i> Estructura de componentes de parámetro y de función en <i>Grasshopper</i> . Elaboración propia (2020).	98
<i>Figura 49.</i> Previsualización geométrica de un punto generado en <i>Grasshopper</i> . Elaboración propia (2020).....	99
<i>Figura 50.</i> Previsualización geométrica de una esfera generada en <i>Grasshopper</i> . Elaboración propia (2020).....	99
<i>Figura 51.</i> Elementos de Interfaz de <i>Grasshopper</i> . Elaboración propia (2020).	101
<i>Figura 52.</i> Métodos de conexión de componentes de parámetros y de función en <i>Grasshopper</i> . Elaboración propia (2020).	102
<i>Figura 53.</i> Estado de componente de <i>Grasshopper</i> en color gris. Elaboración propia (2020).	103
<i>Figura 54.</i> Estado de componente de <i>Grasshopper</i> en color gris oscuro. Elaboración propia (2020).....	104

<i>Figura 55.</i> Estado de componente de <i>Grasshopper</i> en color gris opaco. Elaboración propia (2020).....	104
<i>Figura 56.</i> Estado de componente de <i>Grasshopper</i> en color verde. Elaboración propia (2020).	105
<i>Figura 57.</i> Estado de componente de <i>Grasshopper</i> en color amarillo. Elaboración propia (2020).....	105
<i>Figura 58.</i> Estado de componente de <i>Grasshopper</i> en color rojo. Elaboración propia (2020).	105
<i>Figura 59.</i> Ejemplo de lista de información en componente de <i>Grasshopper</i> . Elaboración propia.	106
<i>Figura 60.</i> Relación entre listas de información convergentes en un componente de función. La lista larga prevalece sobre las más cortas. Los últimos elementos de las listas cortas se reproducen para complementar la larga. Elaboración propia (2020).....	107
<i>Figura 61.</i> Ejemplo de uso de componente <i>Shortest List</i> como filtro para hacer que prevalezca la lista corta. Los valores excedentes de la lista larga son omitidos. Elaboración propia (2020).....	108
<i>Figura 62.</i> Ejemplo de uso de componente <i>Cross Reference</i> como filtro para hacer que todos los elementos de las listas interactúen en todas sus posibilidades de relación. Elaboración propia (2020).....	108
<i>Figura 63.</i> Ejemplo de uso de componente <i>List Item</i> como filtro para seleccionar elementos específicos de la lista. Elaboración propia (2020).	109
<i>Figura 64.</i> Ejemplo de uso de componente <i>Partition List</i> como filtro para dividir una lista de información. Elaborar propia (2020).	109
<i>Figura 65.</i> Ejemplo de uso de componente <i>Split List</i> como filtro para seccionar una lista de información. Elaborar propia (2020).	109

<i>Figura 66.</i> Ejemplo de uso de componente <i>Reverse List</i> como filtro para revertir el orden de la lista de información. Elaboración propia (2020).	110
<i>Figura 67.</i> Ejemplo de uso de componente <i>List Length</i> como filtro para determinar la longitud de la lista de información. Elaboración propia (2020).	110
<i>Figura 68.</i> Ejemplo de uso de componente <i>Replace Items</i> para reemplazar algún elemento dentro de la lista de información. Elaboración propia (2020).	110
<i>Figura 69.</i> Ejemplo de uso de componente <i>Shift List</i> para desplazar el orden de los elementos de la lista de información. En el componente se determina la magnitud de desplazamiento y si los elementos desplazados se recuperan en las primeras posiciones vaciadas. Elaboración propia (2020).....	111
<i>Figura 70.</i> Ejemplo de lista de información de <i>Grasshopper</i> , como parámetro output de un componente. En la parte superior derecha se muestra el nombre de la lista: {0;0}. Elaboración propia (2020).....	112
<i>Figura 71.</i> Ejemplo de uso de componente <i>Partition List</i> como filtro para dividir una lista de información. El componente genera dos listas de información que fluyen a través del parámetro <i>output</i> . Elaborar propia (2020).	113
<i>Figura 72.</i> Estructura del árbol de información en <i>Grasshopper</i> . Recuperado de <i>Grasshopper Primer: for version 0.6.0007 - Spanish Edition</i> , Payne e Issa (2009, p. 36).	114
<i>Figura 73.</i> Ejemplo de árbol de información estructurado por dos listas de datos (ramas). Los elementos de cada lista (rama) solo pueden interactuar entre sí. Los elementos de diferentes listas (ramas) no pueden interactuar. Elaboración propia (2020).	115
<i>Figura 74.</i> Ejemplo de uso de componente <i>Flatten Tree</i> como filtro para romper los grupos de listas de datos en una sola que contenga todos los parámetros. Elaborar propia (2020). .	116
<i>Figura 75.</i> Ejemplo de uso de componente <i>Graft Tree</i> como filtro para agrupar los datos en varias listas (ramas) que contengan todos los parámetros. Elaborar propia (2020).....	117

<i>Figura 76.</i> Ejemplo de uso de componente <i>Tree Branch</i> como filtro para seleccionar una lista (rama) de un conjunto de ramas (árbol) que fluyan a través de un componente. Elaborar propia (2020).....	118
<i>Figura 77.</i> Ejemplo de uso de componente <i>Tree Item</i> como filtro para seleccionar uno o varios datos específicos de una lista (rama), en un conjunto de ramas (árbol) que fluyan a través de un componente. Elaborar propia (2020).....	118
<i>Figura 78.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “geometría múltiple en superficie subdividida”. Elaboración propia (2020).	120
<i>Figura 79.</i> Geometría inicial para multiplicar en superficie a través de algoritmo de <i>Grasshopper</i> . Elaboración Propia (2020).	121
<i>Figura 80.</i> Superficie inicial como referencia para aplicación en algoritmo de <i>Grasshopper</i> . Elaboración propia (2020).	122
<i>Figura 81.</i> Resultado de algoritmo de <i>Grasshopper</i> : “geometría múltiple en superficie subdividida”. Elaboración propia (2020).....	122
<i>Figura 82.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “elementos atractores”. El ejemplo muestra el parámetro <i>Point</i> como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).....	123
<i>Figura 83.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “elementos atractores”. El ejemplo muestra el parámetro <i>Curve</i> como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).....	124
<i>Figura 84.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “elementos atractores”. El ejemplo muestra varios parámetros <i>Curve</i> como elementos atractores y rectángulos como objetos a multiplicar. Elaboración propia (2020).....	124

<i>Figura 85.</i> Resultado de algoritmo de <i>Grasshopper</i> : “elementos atractores”. El ejemplo es producto del parámetro <i>Point</i> como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).	126
<i>Figura 86.</i> Resultado de algoritmo de <i>Grasshopper</i> : “elementos atractores”. El ejemplo es producto del parámetro <i>Curve</i> como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).	127
<i>Figura 87.</i> Resultado de algoritmo de <i>Grasshopper</i> : “elementos atractores”. El ejemplo es producto de varios parámetros <i>Curve</i> como elementos atractores y rectángulos como objetos a multiplicar. Elaboración propia (2020).	127
<i>Figura 88.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “elemento atractor en superficie subdividida”. El ejemplo muestra el parámetro <i>Point</i> como elemento atractor con influencia en superficie. Elaboración propia (2020).	128
<i>Figura 89.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “elemento atractor en superficie subdividida”. El ejemplo muestra el parámetro <i>Curve</i> como elemento atractor con influencia en superficie. Elaboración propia (2020).	129
<i>Figura 90.</i> Resultado de algoritmo de <i>Grasshopper</i> : “elemento atractor en superficie subdividida”. El ejemplo es producto del parámetro <i>Point</i> como elemento atractor. Elaboración propia (2020).	131
<i>Figura 91.</i> Resultado de algoritmo de <i>Grasshopper</i> : “elemento atractor en superficie subdividida”. El ejemplo es producto del parámetro <i>Curve</i> como elemento atractor. Elaboración propia (2020).	131
<i>Figura 92.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “Reticula en diagonal (<i>Diagrid</i>) a partir de una superficie”. Estrategia retomada de AAD <i>Algorithmis-Aided Design: Parametric Strategies Using Grasshopper</i> , (Tedeschi, 2014, p. 157). Adaptación y modificación: elaboración propia (2020).	132

<i>Figura 93.</i> Resultado de algoritmo de <i>Grasshopper</i> : “retícula en diagonal (<i>Diagrid</i>) a partir de una superficie”, variante A. Elaboración propia (2020).....	134
<i>Figura 94.</i> Resultado de algoritmo de <i>Grasshopper</i> : “retícula en diagonal (<i>Diagrid</i>) a partir de una superficie”, variante B. Elaboración propia (2020).....	134
<i>Figura 95.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “superficie generada a partir de función matemática”. Estrategia retomada de <i>AAD_Algorithmis-Aided Design: Parametric Strategies Using Grasshopper</i> , (Tedeschi, 2014, p. 105). Adaptación y modificación: elaboración propia (2020).....	135
<i>Figura 96.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir de función matemática”, variante A: superficie NURBS. Elaboración propia (2020).	137
<i>Figura 97.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir de función matemática”, variante B: <i>Mesh</i> . Elaboración propia (2020).....	137
<i>Figura 98.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “superficie generada a partir de puntos aleatorios”. Elaboración propia (2020).	138
<i>Figura 99.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir de puntos aleatorios”, variante: superficie <i>Mesh</i> . Elaboración propia (2020).....	140
<i>Figura 100.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “superficie generada a partir de función matemática gráfica”, variante A: superficie <i>Mesh</i> . Elaboración propia (2020).....	141
<i>Figura 101.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “superficie generada a partir de función matemática gráfica”, variante B: superficie NURBS. Elaboración propia (2020)..	141
<i>Figura 102.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir función matemática gráfica”, variante A: superficie <i>Mesh</i> , función matemática seno. Elaboración propia (2020).....	143

<i>Figura 103.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir función matemática gráfica”, variante B: superficie NURBS, función matemática seno. Elaboración propia (2020).....	144
<i>Figura 104.</i> Resultado de algoritmo de <i>Grasshopper</i> : “superficie generada a partir función matemática gráfica”, variante C: superficie NURBS, función matemática suma de senos. Elaboración propia (2020).	144
<i>Figura 105.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).....	145
<i>Figura 106.</i> Resultado de algoritmo de <i>Grasshopper</i> : “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).....	148
<i>Figura 107.</i> Resultado de algoritmo de <i>Grasshopper</i> : “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).....	148
<i>Figura 108.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “ejemplo de diseño paramétrico de rascacielos”, variante A: envolvente desarrollada con estrategia algorítmica “retícula en diagonal (<i>Diagrid</i>) a partir de una superficie”. Elaboración propia (2020).....	149
<i>Figura 109.</i> Algoritmo de <i>Grasshopper</i> para estrategia básica “ejemplo de diseño paramétrico de rascacielos”, variante B: envolvente desarrollada con estrategia algorítmica “elemento atractor en superficie subdividida”. Elaboración propia (2020).	150
<i>Figura 110.</i> Resultado de algoritmo de <i>Grasshopper</i> : “ejemplo de diseño paramétrico de rascacielos”, variante A: envolvente desarrollada con estrategia algorítmica “retícula en diagonal (<i>Diagrid</i>) a partir de una superficie”. Elaboración propia (2020).....	152
<i>Figura 111.</i> Resultado de algoritmo de <i>Grasshopper</i> : “ejemplo de diseño paramétrico de rascacielos”, variante B: envolvente desarrollada con estrategia algorítmica “elemento atractor en superficie subdividida”. Elaboración propia (2020).	152

<i>Figura 112.</i> Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema metodológico de análisis cualitativo: aspectos fenomenológicos y semiológicos del espacio. Elaboración propia (2020).	155
<i>Figura 113.</i> Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema metodológico de análisis cuantitativo: análisis termo-energético y la estructura. Elaboración propia (2020).	157
<i>Figura 114.</i> Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema global metodológico que incluye las rutas de análisis cualitativas y cuantitativas. Elaboración propia (2020).	160
<i>Figura 115.</i> Algoritmo de <i>Grasshopper</i> “transformación topológica de una superficie con <i>Weaverbird</i> ”. Variante A: manipulación de superficie producto de estrategia “superficie generada a partir de función matemática gráfica”. Elaboración propia (2020).	162
<i>Figura 116.</i> Algoritmo de <i>Grasshopper</i> “transformación topológica de una superficie con <i>Weaverbird</i> ” Variante B: manipulación de superficie producto de estrategia “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).	162
<i>Figura 117.</i> Resultado de algoritmo de <i>Grasshopper</i> “transformación topológica de una superficie con <i>Weaverbird</i> ” Variante A: manipulación de superficie producto de estrategia “superficie generada a partir de función matemática gráfica”. Elaboración propia (2020)...	164
<i>Figura 118.</i> Resultado de algoritmo de <i>Grasshopper</i> “transformación topológica de una superficie con <i>Weaverbird</i> ” Variante B: manipulación de superficie producto de estrategia “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).	164
<i>Figura 119.</i> Algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ”. Variante: estructura extensible. Elaboración propia (2020).	167

<i>Figura 120.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ” Variante: estructura extensible. Elaboración propia (2020). ...	169
<i>Figura 121.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ” Variante: estructura extensible. Elaboración propia (2020). ...	169
<i>Figura 122.</i> Algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ”. Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).	170
<i>Figura 123.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).	172
<i>Figura 124.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).	172
<i>Figura 125.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación de sistemas topológicos estructurales con <i>Kangaroo</i> ” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).	173
<i>Figura 126.</i> Algoritmo de <i>Grasshopper</i> “sistemas de agregación compleja con <i>Wasp</i> ”. Variante A: reglas de agregación y conexión generadas por la computadora. Elaboración propia (2020).	174
<i>Figura 127.</i> Algoritmo de <i>Grasshopper</i> “sistemas de agregación compleja con <i>Wasp</i> ”. Variante B: reglas de agregación y conexión generadas por el usuario. Elaboración propia (2020).	174
<i>Figura 128.</i> Resultado de algoritmo de <i>Grasshopper</i> “sistemas de agregación compleja con <i>Wasp</i> ”. Variante A: reglas de agregación y conexión generadas por la computadora. Ver	

animación en el siguiente vínculo: https://youtu.be/d154dbPt-6Y . Elaboración propia (2020).	177
<i>Figura 129.</i> Resultado de algoritmo de <i>Grasshopper</i> “sistemas de agregación compleja con <i>Wasp</i> ”. Variante B: reglas de agregación y conexión generadas por el usuario. Ver animación en el siguiente vínculo: https://youtu.be/d154dbPt-6Y . Elaboración propia (2020).	178
<i>Figura 130.</i> Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de https://www.spaumx.com/casa-crv	180
<i>Figura 131.</i> Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de https://www.spaumx.com/casa-crv	180
<i>Figura 132.</i> Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de https://www.spaumx.com/casa-crv	181
<i>Figura 133.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	182
<i>Figura 134.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	184
<i>Figura 135.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	184

<i>Figura 136.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).....	185
<i>Figura 137.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	187
<i>Figura 138.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	187
<i>Figura 139.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	188
<i>Figura 140.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).	188
<i>Figura 141.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).	190
<i>Figura 142.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i> ”. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).	190

<i>Figura 143.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de iluminación natural interior con <i>Honeybee</i> ”. Variante A: factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).....	192
<i>Figura 144.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de iluminación natural interior con <i>Honeybee</i> ”. Variante A: factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).....	195
<i>Figura 145.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de iluminación natural interior con <i>Honeybee</i> ”. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).....	195
<i>Figura 146.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de iluminación natural interior con <i>Honeybee</i> ”. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).	198
<i>Figura 147.</i> Algoritmo de <i>Grasshopper</i> “simulación para análisis de confort térmico espacial interior con <i>Honeybee</i> ”. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).....	202
<i>Figura 148.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de confort térmico espacial interior con <i>Honeybee</i> ”. La imagen muestra las zonas interiores de confort térmico de acuerdo con la sensación de calor. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).	205
<i>Figura 149.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de confort térmico espacial interior con <i>Honeybee</i> ”. La imagen muestra las zonas interiores de confort térmico de acuerdo con la sensación de frío. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).	206

<i>Figura 150.</i> Resultado de algoritmo de <i>Grasshopper</i> “simulación para análisis de confort térmico espacial interior con <i>Honeybee</i> ”. La imagen muestra la temperatura operativa interior anual. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).....	206
<i>Figura 151.</i> Algoritmo de <i>Grasshopper</i> “optimización evolutiva con Algoritmos Genéticos (AG)”. Variante: minimizar incidencia de radiación solar anual sobre superficie. Elaboración propia (2020).....	208
<i>Figura 152.</i> Resultado de algoritmo de <i>Grasshopper</i> “optimización evolutiva con Algoritmos Genéticos (AG) mediante <i>Galapagos</i> ”. Variante: minimizar incidencia de radiación solar anual sobre superficie. Ver animación en el siguiente vínculo: https://youtu.be/eoK3zD0A4WY . Elaboración propia (2020).....	211
<i>Figura 153.</i> Imagen frontal de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.	217
<i>Figura 154.</i> Imagen panorámica de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.	217
<i>Figura 155.</i> Imagen interior de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.	218
<i>Figura 156.</i> Metodología de diseño arquitectónico tradicional: imagen de resultados del taller de Diseño Arquitectónico 1. Tercer semestre, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California. Elaboración propia (2020).	223
<i>Figura 157.</i> Metodología de diseño arquitectónico Algorítmica Paramétrica (AP): imagen de resultados del curso optativo Nociones de Arquitectura y Urbanismo Paramétrico. Quinto semestre en adelante, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California. Elaboración propia (2020).	227
<i>Figura 158.</i> Ejemplar de ficha de análisis y evaluación metodológica, primera parte (análisis dicotómico). Elaboración propia (2021).	230

<i>Figura 159.</i> Ejemplar de ficha de análisis y evaluación metodológica, segunda parte (análisis inter). Elaboración propia (2021).....	232
<i>Figura 160.</i> Relación de enfoques de diseño manifestados en la realización del proyecto “Casa Estudio <i>Loft</i> ” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).	234
<i>Figura 161.</i> Relaciones cognitivas fenomenológicas y semiológicas manifestados en la realización del proyecto “Casa Estudio <i>Loft</i> ” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	235
<i>Figura 162.</i> Relaciones físicas manifestadas en la realización del proyecto “Casa Estudio <i>Loft</i> ” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	236
<i>Figura 163.</i> Relación de códigos de utilidad-función, belleza-forma manifestados en la realización del proyecto “Casa Estudio <i>Loft</i> ” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	237
<i>Figura 164.</i> Relación de carácter estructural propuesto para el proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	239
<i>Figura 165.</i> Adaptabilidad ambiental para el proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).	240
<i>Figura 166.</i> Relación de estrategias de adaptabilidad ambiental para el proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	240

<i>Figura 167.</i> Relación de implementos para efectuar estrategias de adaptabilidad ambiental para el proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).....	241
<i>Figura 168.</i> Aproximación a tendencias de la Arquitectura AP de los estudiantes que cursaron la clase optativa NAUP (2020-1) para el desarrollo del proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California. Elaboración propia (2021).	243
<i>Figura 169.</i> Relación de estrategias algorítmicas para efectuar el proyecto “Casa Estudio <i>Loft</i> ” en la ciudad de Mexicali, Baja California, por los estudiantes la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).	243
<i>Figura 170.</i> El Uróboros, la serpiente que come su propia cola (símbolo de la mitología nórdica), como referencia a la recursividad. Fuente: “Uróboros” (2020) [Imágen]. Recuperado de https://es.wikipedia.org/w/index.php?title=Ur%C3%B3boros&oldid=126403838 . Dominio público (2020).....	247
<i>Figura 171.</i> El Uróboros: recursividad entre los sentidos y la razón, entre el algoritmo y el operador. Elaboración propia (2020).	248

ÍNDICE DE TABLAS

TABLA 1	93
Matriz de congruencia metodológica del segundo objetivo particular de investigación.	93
TABLA 2	114
Descripción generacional del nombre de lista de información {0,1,0}.	114
TABLA 3	120
Tabla descriptiva para construcción de algoritmo: geometría múltiple en superficie subdividida.....	120
TABLA 4	124
Tabla descriptiva para construcción de algoritmo: elementos atractores.	124
TABLA 5	129
Tabla descriptiva para construcción de algoritmo: elemento atractor en superficie subdividida.....	129
TABLA 6	132
Tabla descriptiva para construcción de algoritmo: retícula en diagonal (<i>Diagrid</i>) a partir de una superficie.	132
TABLA 7	135
Tabla descriptiva para construcción de algoritmo: superficie generada a partir de función matemática.....	135
TABLA 8	138
Tabla descriptiva para construcción de algoritmo: superficie generada a partir de puntos aleatorios.....	138

TABLA 9.....	141
Tabla descriptiva para construcción de algoritmo: superficie generada a partir de función matemática gráfica.....	141
TABLA 10.....	145
Tabla descriptiva para construcción de algoritmo: ejemplo de diseño paramétrico de envolvente de estadio deportivo.	145
TABLA 11.....	150
Tabla descriptiva para construcción de algoritmo: ejemplo de diseño paramétrico de rascacielos.	150
TABLA 12.....	163
Síntesis descriptiva de algoritmo: transformación topológica de una superficie con <i>Weaverbird</i>.....	163
TABLA 13.....	165
Relación de las principales ecuaciones matemáticas y parámetros a considerar para trabajar con la herramienta algorítmica <i>Kangaroo</i>.....	165
TABLA 13.....	167
Síntesis descriptiva de algoritmo: simulación de sistemas topológicos estructurales con <i>Kangaroo</i>. Variante: estructura extensible.....	167
TABLA 14.....	170
Síntesis descriptiva de algoritmo: simulación de sistemas topológicos estructurales con <i>Kangaroo</i>. Variante: superficie suspendida “película de jabón”.	170
TABLA 15.....	175
Síntesis descriptiva de algoritmo: sistemas de agregación compleja con <i>Wasp</i>.....	175

TABLA 16.....	181
Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C. (SPAU).	
Hermosillo, Sonora, México. 2015-2017. Relación de características de edificación	
residencial para análisis y simulación (reubicación por diseño a ciudad de Mexicali,	
B.C., México).	
	181
TABLA 17.....	182
Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre	
una superficie con <i>Ladybug</i>. Variante A: simulación de radiación solar anual	
acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.....	
	182
TABLA 18.....	185
Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre	
una superficie con <i>Ladybug</i>. Variante B: estudio de incidencia solar anual de Casa	
RV, SPAU (2017). Aplicación: Mexicali, B.C.	
	185
TABLA 19.....	189
Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre	
una superficie con <i>Ladybug</i>. Variante C: simulación de radiación solar anual de una	
superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de	
rascacielos”.	
	189
TABLA 20.....	191
Relación de las principales ecuaciones matemáticas y parámetros a considerar para	
trabajar simulaciones de iluminación interior con la herramienta algorítmica	
<i>Honeybee</i>.	
	191
TABLA 21.....	192

Síntesis descriptiva de algoritmo: simulación para análisis de iluminación natural interior con <i>Honeybee</i>. Variante A: simulación de factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.....	193
TABLA 22.....	195
Síntesis descriptiva de algoritmo: simulación para análisis de iluminación natural interior con <i>Honeybee</i>. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.	196
TABLA 23.....	199
Relación de las principales ecuaciones matemáticas y parámetros a considerar para trabajar simulaciones de flujo de calor con la herramienta algorítmica <i>Honeybee</i>..	199
TABLA 24.....	203
Síntesis descriptiva de algoritmo: simulación para análisis de confort térmico espacial interior con <i>Honeybee</i>. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C.	203
TABLA 25.....	209
Síntesis descriptiva de algoritmo: optimización evolutiva con Algoritmos Genéticos (AG). Variante: minimizar incidencia de radiación solar anual sobre superficie.	209
TABLA 26.....	213
Matriz de congruencia metodológica del tercer objetivo particular de investigación.	213
TABLA 27.....	216
Características de ubicación del sitio a desarrollar proyecto: casa estudio <i>loft</i>.	216
TABLA 28.....	219

Relación de espacios a diseñar para proyecto casa estudio <i>loft</i> en la ciudad de Mexicali.....	219
TABLA 29.....	220
Encuadre del taller de Diseño Arquitectónico 1. Tercer semestre, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California.....	220
TABLA 30.....	224
Encuadre del curso optativo Nociones de Arquitectura y Urbanismo Paramétrico. Quinto semestre en adelante, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California.....	224

ÍNDICE

ALGORITMOS Y ARQUITECTURA: UNA APORTACIÓN AL DISEÑO	i
<i>ALGORITHMS AND ARCHITECTURE: A CONTRIBUTION TO DESIGN.....</i>	<i>ii</i>
ÍNDICE DE FIGURAS	iii
ÍNDICE DE TABLAS.....	xxiii
I. PLANTEAMIENTO DEL PROBLEMA	1
1.1. Antecedentes	3
1.2. Preguntas de investigación	6
1.2.1. Pregunta principal.....	6
1.2.2. Preguntas secundarias.....	6
1.3. Justificación	6
1.4. Objetivos de investigación	9
1.4.1. Objetivo general	9
1.4.2. Objetivos particulares.....	9
II. MARCO TEÓRICO	11
2.1. Algoritmos y la Arquitectura Algorítmica	11
2.2. Arquitectura Paramétrica	15
2.3. Posicionamiento histórico de la Arquitectura Algorítmica-Paramétrica.....	17
2.3.1. La visión morfológica del diseño arquitectónico computacional.....	19
2.3.2. Esculturalismo y Parametricismo como discursos morfológicos de la era digital.	22
2.3.3. El enfoque tecnológico y político del siglo XXI.....	26

2.4.4. La era post-digital del diseño arquitectónico computacional.	36
2.5. Contexto disciplinar: arquitectura digital y la academia	44
2.6. Teorías y metodologías de la Arquitectura Algorítmica-Paramétrica.....	48
2.6.1. El Parametricismo de Patrik Schumacher.....	49
2.6.2. La Arquitectura Algorítmica de Kostas Terzidis.	53
2.6.3. Innovación y fabricación digital en Arquitectura.	56
2.7. Aspectos fenomenológicos y semiológicos como parámetros cualitativos de diseño Algorítmico Paramétrico	64
2.7.1. Análisis descriptivo de los aspectos fenomenológicos en la Arquitectura.	67
2.7.2. Análisis descriptivo de los aspectos semiológicos en la Arquitectura.....	76
2.7.3. Análisis de la condición semiológica y fenomenológica de la Arquitectura contemporánea.....	78
2.7.4. Sinergia entre los enfoques fenomenológicos y semiológicos de la Arquitectura de Kengo Kuma.	81
2.7.5. Implementación de aspectos fenomenológicos y semiológicos en la Arquitectura computacional.....	85
2.8. Aspectos de análisis fisiológico arquitectónico como parámetros cuantitativos de diseño Algorítmico Paramétrico	88
III. METODOLOGÍA.....	91
3.1. Introducción a <i>Grasshopper</i> como plataforma de programación visual	94
3.1.1. Elementos sintácticos de un algoritmo.	95
3.1.2. Listas de información.	106

3.1.3. Árbol de información.	111
3.2. Estrategias algorítmicas básicas en <i>Grasshopper</i> para aplicación en diseño arquitectónico.	119
3.2.1. Geometría múltiple en superficie subdividida.....	119
3.2.2. Elementos atractores.....	123
3.2.3. Elemento atractor en superficie subdividida.	128
3.2.4. Reticula en diagonal (<i>Diagrid</i>) a partir de una superficie.	132
3.2.5. Superficie generada a partir de función matemática.	135
3.2.6. Superficie generada a partir de puntos aleatorios.....	138
3.2.7. Superficie generada a partir de función matemática gráfica.	140
3.2.8. Ejemplo de diseño paramétrico de envolvente de estadio deportivo.	145
3.2.9. Ejemplo de diseño paramétrico de rascacielos.....	149
3.3. Desarrollo metodológico de diseño arquitectónico Algorítmico Paramétrico	153
3.3.1. Construcción metodológica cualitativa de diseño arquitectónico AP	153
3.3.2. Construcción metodológica cuantitativa de diseño arquitectónica AP.....	157
3.3.3. Metodología de diseño arquitectónico AP.....	159
3.4. Ejemplos de aplicación de la metodología de diseño arquitectónico AP.....	160
3.4.1. Transformación topológica de una superficie con <i>Weaverbird</i>	161
3.4.2. Simulación de sistemas topológicos estructural con <i>Kangaroo</i>	165
3.4.3. Sistemas de agregación compleja con <i>Wasp</i>	173
3.4.4. Simulación para análisis de radiación solar sobre una superficie con <i>Ladybug</i>	179
3.4.5. Simulación para análisis de iluminación natural interior con <i>Honeybee</i>	191

3.4.6. Simulación para análisis de confort térmico espacial interior con <i>Honeybee</i>	198
3.4.7. Optimización evolutiva con Algoritmos Genéticos (AG) mediante <i>Galápagos</i> .	
.....	207
IV. APLICACIÓN	212
4.1. Aplicación metodológica: descripción de proyecto arquitectónico.....	215
4.1.2. Descripción del tema.	215
4.1.3. Programa arquitectónico y condiciones.....	218
4.2. Aplicación de metodología de diseño arquitectónico tradicional: Taller de Diseño Arquitectónico 1	219
4.3. Aplicación de metodología de diseño AAP: Nociones de Arquitectura y Urbanismo Paramétrico.	224
4.4. Instrumentos de análisis y evaluación metodológica	228
4.4.1. Primera parte del instrumento de análisis y evaluación metodológica.....	228
4.4.2. Segunda parte del instrumento de análisis y evaluación metodológica.	231
4.5. Resultados	233
4.5.1. Análisis y evaluación de los aspectos cualitativos de la forma y el espacio.	235
4.5.2. Análisis y evaluación de los aspectos cuantitativos de la forma y el espacio. ...	238
4.5.3. Análisis descriptivo de las tendencias y estrategias algorítmicas aplicadas.....	242
V. DISCUSIÓN.....	244
VI. CONCLUSIONES.....	249
REFERENCIAS.....	254

I. PLANTEAMIENTO DEL PROBLEMA

Según Terzidis (2006) una de las preocupaciones en el ámbito de la Arquitectura a nivel internacional en la segunda parte del siglo XX es la pérdida de control de los arquitectos sobre sus propios diseños por el traslado del uso de herramientas manuales a las computacionales. Considera que con el uso de programas *Computer Aided Design* (CAD) el diseñador no llega a conocer todo el alcance que las computadoras tienen, por lo que hace una distinción en la manera de utilizarla según sus capacidades y las clasifica en: “computarización y computación”.

La primera clasificación (modo más usado), es la capacidad que la computadora tiene de procesamiento, organización y representación de información existente. La segunda forma, menos explorada, está relacionada con su capacidad de cálculo dirigido a la generación de información que aún no se conoce.

En el mismo orden de ideas, Reas y McWilliams (2010) contrastan también dos formas de utilizar los programas computacionales en el diseño arquitectónico:

El uso de programas computacionales en la Arquitectura y las artes se puede separar en dos categorías: producción y concepción. En la primera categoría, la computadora es usada con el fin de producir una idea preconcebida. En la segunda, participa en el desarrollo de la forma. (2010, p. 25).

Un uso computacional de “producción” o “computarización” al iniciar un proyecto, cuando la conciencia de su formación final es difusa, puede traer complicaciones en su resolución ya que en ese momento aún no se conocen los datos que un programa CAD requiere para representar un objeto como son: dimensiones, área, volumen, características geométricas, topológicas, etc. (ver Figura 1).

Sin embargo, un uso de “concepción” o “computación” implica ejercer una comunicación con la computadora a través de un lenguaje para conocer ampliamente sus

capacidades de cálculo, eficiencia y precisión. La programación digital es el medio para desarrollar ese diálogo, y permite a la computadora proceder ante los parámetros, las instrucciones o los algoritmos que construyen la solución de un problema de diseño (ver Figura 2). “La programación computacional implica inminentemente escribir algoritmos” (Shiffman, 2015, p. xvi).

También, Ervin (2018) concluye que la Arquitectura tiene tres distintos modos de interacción con la computación: “1. usando *software* - como herramienta, para la automatización, 2. Programando - como un lenguaje, por algoritmos, 3. Involucrándose con el Internet de las Cosas (*Internet of Things*, IoT) - como un medio, para el incremento: desde lo estático a lo dinámico e interactivo” (citado en Cantrell y Mekies, 2018, p. 112).

Según lo anterior, es posible encontrar que la disciplina de la Arquitectura tradicionalmente interactúa con la computación principalmente en el primer modo, por lo que las dos formas restantes, que son menos exploradas, desaprovechan el alcance que la tecnología digital puede ofrecer como auxiliar en la metodología de diseño arquitectónico.

Específicamente en el segundo modo de uso, que tiene relación con la clasificación de “concepción” de Reas y McWilliams (2010) y la capacidad de “computación” mencionada por Terzidis (2006), es en donde la computadora tiene una función más amplia; es decir, no sólo trabaja como una herramienta, sino también interactúa como un agente colaborador que puede dar respuesta a problemas complejos de diseño.

Un mayor dominio en la comprensión de problemas complejos abre camino a evaluar y sistematizar esquemas que brinden optimización en el proceso de diseño. En un sentido amplio, la optimización implica procesos de evaluación y diagnóstico que tienen el objetivo de obtener mejores resultados en menor tiempo.

El desarrollo algorítmico es el principal instrumento de la programación digital, un enfoque paramétrico en los procesos de diseño facilita la comprensión y la construcción de

algoritmos en el uso de la computadora; de tal manera que, el arquitecto en esta práctica no solo desempeña un rol de usuario, sino también el de programador para generar sistemas que ayuden en la resolución optimizada de un proyecto complejo.

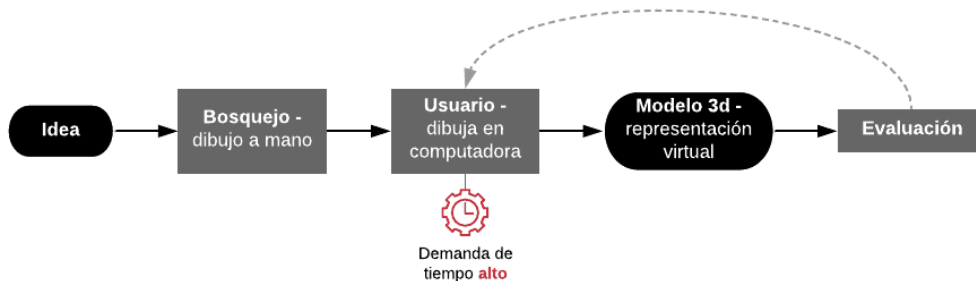


Figura 1. Proceso de diseño con el uso de la computadora a manera de “producción” o “computarización”; se aprovecha la computadora como una herramienta representacional. Elaboración propia (2019).

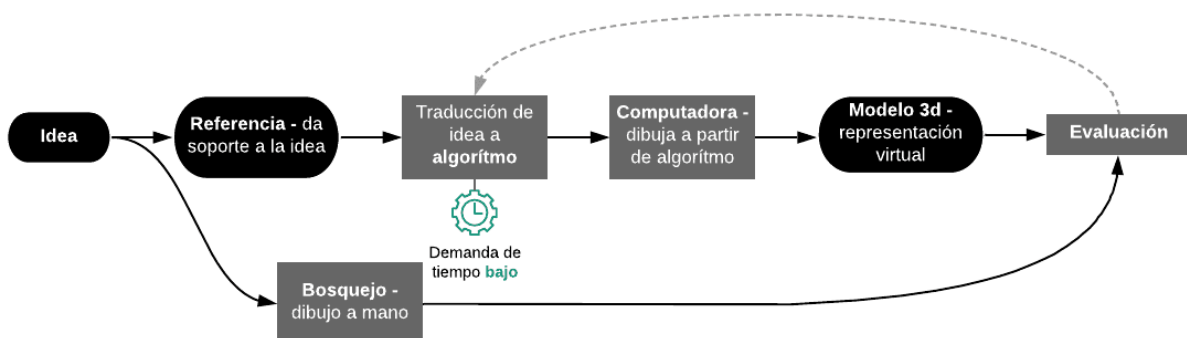


Figura 2. Proceso de diseño con el uso de la computadora a manera de “concepción” o “computación”; se aprovecha la computadora como inteligencia auxiliar en la generación de ideas y objetos de diseño. Elaboración propia (2019).

1.1. Antecedentes

No se encuentra una línea histórica marcada y documentada con relación al origen de la Arquitectura con enfoque algorítmico y paramétrico desde la perspectiva computacional. Aunque existen alusiones del pensamiento paramétrico aplicado en la metodología de diseño arquitectónico que datan del movimiento moderno a través de los arquitectos Antoni Gaudí y

Frei Otto, considerados por Burry (2016) como *Proto-parametricistas*, es hasta la última etapa de la posmodernidad cuando, paralelamente, el avance tecnológico en torno a los medios de comunicación e informáticos comenzaron a tener una presencia más fuerte en la sociedad a nivel global.

El uso y los avances en torno a la computadora marcaron pauta para proyectar nuevos modelos de trabajo en todas las disciplinas; por tanto, es allí donde el pensamiento paramétrico se vuelve fundamental y se torna también en uno algorítmico. Según Frampton (2014):

Los diversos fenómenos que acompañan a la globalización están estrechamente asociados al índice siempre creciente de la comunicación telemática y al incremento constante de los vuelos intercontinentales. Como consecuencia de ello, el ejercicio profesional de la Arquitectura es actualmente tan global como local, cómo podemos juzgar a partir de los célebres arquitectos internacionales que cada vez actúan en todo el mundo. (2014, p. 349).

Así, es posible posicionar el uso computacional en la Arquitectura a partir de la caracterización taxonómica que Frampton (2014) propone sobre la Arquitectura en la era de la globalización como *Morfológica*, donde la manifestación de un grupo de arquitectos tiende a emular la estructura de la forma biológica y botánica. Los avances en la tecnología digital de la posmodernidad permitieron explorar esas inquietudes que giran en torno a la figura natural más que cultural.

La Arquitectura de Frank O. Gehry, Ben van Berkel, Daniel Libeskind, Greg Lynn, Hani Rashid, y Zaha Hadid, es representativa de esta clasificación *Morfológica*. Inicialmente la mayoría de estos personajes exploraron las bondades de programas computacionales CAD como el medio para el desarrollo formal de sus proyectos; sin embargo, estos partían de métodos y diagramas analógicos que después eran trabajados digitalmente con el objetivo de

reproducir estas formas. Es Greg Lynn quien desarrolla un discurso en torno al algoritmo informático a favor de un enfoque biomórfico en la Arquitectura, con un origen totalmente digital (Eisenman, 2011).

Lynn evoca el análisis hecho por D'Arcy Thompson de los principios comunes de la forma animal como el fundamento para dar prioridad a la curvilinealidad sobre la linealidad y, por tanto, favorecer el *morphing* o metamorfosis digital a priori de la figura en superficies polimórficas, a las que denomina “glóbulos” (*blobs*). (Frampton, 2014, p. 363).

En un periodo de 1980 a 2010, surgen dos fuertes visiones con una teoría y metodología amplia sobre el uso de la computadora con un enfoque algorítmico-paramétrico en la Arquitectura; esto, a través de dos representantes importantes en el tema: Patrik Schumacher (2011, 2012) y Kostas Terzidis (2006).

El primero busca posicionar la arquitectura paramétrica en un nivel estilístico o paradigmático, mediante el manifiesto hecho en el 2008 para la Bienal de Venecia llamado *Parametricism as Style* (2009) donde hace una descripción de las cualidades y características formales que los productos arquitectónicos deberían poseer a través del uso de técnicas de Diseño Paramétrico (DP), y declara éste como el camino hacia la gestación de un nuevo estilo. A dicho estilo lo nombra como *Parametricismo*.

Considera que sería el sucesor del movimiento moderno, que agradece su génesis a los nuevos avances en la tecnología digital, y que es capaz de desarrollar una estética de fluidez, diferenciación y complejidad, con aparente mayor sintonía a la encontrada en la naturaleza. Se puede contrastar que en esta visión existe un mayor apego a la estética del producto y a la novedad.

En contraste, la visión de Terzidis (2006) parte de revisar las raíces etimológicas de la palabra diseño, para así cuestionar la existencia de la novedad y si ésta tendría que ser el

espíritu de las manifestaciones de Arquitectura Paramétrica (AP). Apoyado en la filosofía pre-Socrática, menciona que todo conocimiento, proceso o producto generado tiene precedente; es decir, puede encontrarse una raíz u origen, y así quizá pueda entenderse como emergente.

Por tanto, la noción del descubrimiento o invención, en lugar de innovación, apunta a que el diseño y los procedimientos algorítmicos se dirigen hacia lo desconocido o, mejor dicho, a lo todavía no encontrado. Así pues, Terzidis no posiciona la Arquitectura AP en un paradigma o estilo, sino lo entiende como un proceso en el quehacer arquitectónico relacionado con el uso de la tecnología digital.

1.2. Preguntas de investigación

En el planteamiento de este trabajo surgen como punto de partida de la investigación las siguientes preguntas:

1.2.1. Pregunta principal.

¿Qué se requiere para que los arquitectos, a través de definiciones algorítmicas y paramétricas, puedan generar soluciones arquitectónicas?

1.2.2. Preguntas secundarias

¿Cuál es el beneficio de utilizar definiciones algorítmicas para generar soluciones arquitectónicas?

¿En qué consiste una metodología de diseño arquitectónico algorítmico?

¿En qué circunstancias puede ser utilizada la metodología de diseño arquitectónico algorítmico en el quehacer arquitectónico?

1.3. Justificación

A nivel internacional, la Arquitectura desarrollada con una metodología Algorítmica-Paramétrica (AP) se ha manifestado con mayor avance en el hemisferio norte, oscila principalmente entre tres visiones: una con mayor apego a la exploración morfológica de los

objetos a diseñar, otra con mayor preocupación en los avances tecnológicos que la era digital ha ofrecido a la disciplina, y una tercera que añade un carácter político y crítico al enfoque tecnológico.

En México actualmente, estos enfoques han sido explorados de manera acotada y con poca profundidad. Aunque hay instituciones en la enseñanza de la arquitectura donde se instruye la programación digital y el pensamiento paramétrico aplicado en el diseño, es aún necesario ampliar su investigación. Esta realidad, posiciona su aprendizaje principalmente a textos e investigaciones de origen extranjero, donde el tema se encuentra más avanzado.

Es necesario desarrollar, aplicar, traducir o contextualizar más este conocimiento en México, para aumentar las alternativas en torno a la investigación y el desarrollo arquitectónico con un acercamiento mayor a los avances tecnológicos y la experimentación.

El proceso de diseño instruido tradicionalmente en las escuelas de Arquitectura parte de una herencia que conoce ampliamente el uso de herramientas análogas aplicadas en la realización de un objeto arquitectónico. La actualidad demanda una revisión a este paradigma metodológico, en vías de ser consistentes con la realidad contemporánea.

El profesional de la Arquitectura contemporánea requerirá implementar en su conocimiento el dominio de las herramientas computacionales más avanzadas con el objetivo de dirigir su teoría y práctica acorde a lo que según Wujec (2017) llama la cuarta revolución industrial: “Nos estamos adentrando a la cuarta revolución industrial. La cual no se encuentra dirigida por una sola fuerza, sino por una combinación de tecnologías, sensores, algoritmos y robótica” (2017, p. 30).

La postura no busca que la tecnología digital sustituya al hombre en el proceso de diseño, sino busca posicionarla como un complemento. La computadora representa cálculo, eficiencia y precisión, su utilidad puede ser considerada efectiva en la parte intermedia de la

trayectoria entre el surgimiento de la idea (su esquematización o bosquejo) y la construcción física.

Francis Aish (2017), quien dirige *The Applied Research + Development Group* (ARD) del despacho de Arquitectura londinense *Foster + Partners*, menciona que:

Esas herramientas (programas paramétricos *Building Information Modelling* -BIM-) pueden ser muy transformacionales. Sin embargo, no deberían ser sobrevaloradas. Las herramientas nos permiten hacer cosas muy interesantes, pero estas no van a resolver mágicamente todos los problemas por nosotros. Uno necesita aún ser un buen diseñador. (citado por Wujec, 2017, p. 101).

Otro aspecto, no menos importante, es el compromiso de la disciplina con el medio ambiente natural y la sociedad, en el marco de los fenómenos que acontecen mundialmente a causa del cambio climático, es necesario estar en sintonía con las acciones humanas planteadas por la Organización de Naciones Unidas (ONU), en la Agenda 2030 sobre el Desarrollo Sostenible, como una oportunidad para mejorar la vida de todos.

Es por ello, que las estrategias metodológicas desarrolladas en esta investigación, con relación a la aplicación de los avances tecnológicos digitales en la industria de la Arquitectura y Construcción, se posicionan de manera pertinente con los “17 *Objetivos de Desarrollo Sostenible*” que son parte de la agenda.

Específicamente como una aportación al objetivo nueve: “*Industria, Innovación e Infraestructura*”, en la búsqueda de optimización de los métodos de fabricación y construcción hacia una “*Producción y Consumo Responsable*” planteado por el objetivo doce; de una disciplina con un rol fundamental para cumplir el objetivo once: “*Ciudades y Comunidades Sostenibles*”.

Así, se puede encontrar referencias como *Parametric Design for Architecture* (Jabi, 2013) o *Tooling, Pamphlet Architecture 27* (Aranda y Lasch, 2006) que introducen a la

arquitectura paramétrica y funcionan como herramienta para la práctica y desarrollo de algoritmos que dan como resultado diferentes exploraciones formales con utilidad en el diseño.

Sin embargo, la explicación de dichas exploraciones se enfoca únicamente en la geometría; se muestran a través de clasificaciones topológicas y no están contextualizadas en una metodología de diseño, en estrategias de análisis sistemático o simulación para la evaluación de una problemática arquitectónica natural o artificial específica.

Una excepción a lo mencionado anteriormente es el libro *AAD, Algorithms-aided Design: Parametric Strategies Using Grasshopper* (Tedeschi, 2014) en el que se pueden encontrar no solo resultados formales, sino estrategias más amplias dirigidas a encontrar soluciones en torno a la arquitectura y su relación con sistemas estructurales y ambientales.

Otra referencia importante para el desarrollo del presente trabajo es el libro *Atlas of Digital Architecture* (Hirschberg et al., 2020), que incluye un amplio repertorio de estrategias, métodos, herramientas, terminologías y conceptos relacionados con el uso de la tecnología digital en la Arquitectura. De tal manera, que estos dos últimos recursos bibliográficos son considerados referencias más cercanas a los objetivos de este trabajo de investigación.

1.4. Objetivos de investigación

1.4.1. Objetivo general

Desarrollar una herramienta teórico-práctica de definiciones algorítmicas que auxilie a arquitectos, como guía introductoria didáctica, en la generación de soluciones arquitectónicas de forma, espacio y orden.

1.4.2. Objetivos particulares

1. Analizar los aportes del desarrollo de definiciones algorítmicas para el diseño de objetos arquitectónicos en el período de 1980 a 2021.

2. Definir una metodología de diseño arquitectónico algorítmico y contrastar con las tradicionales, se toma como caso de estudio y aplicación la comunidad estudiantil de la FAD (UABC).

3. Identificar condiciones de pertinencia para la aplicación de la metodología de diseño arquitectónico algorítmico a través de instrumentos de comprobación, se considera a estudiantes del taller de Diseño Arquitectónico 1 y la clase optativa Nociones de Arquitectura y Urbanismo Paramétrico como sujetos de estudio.

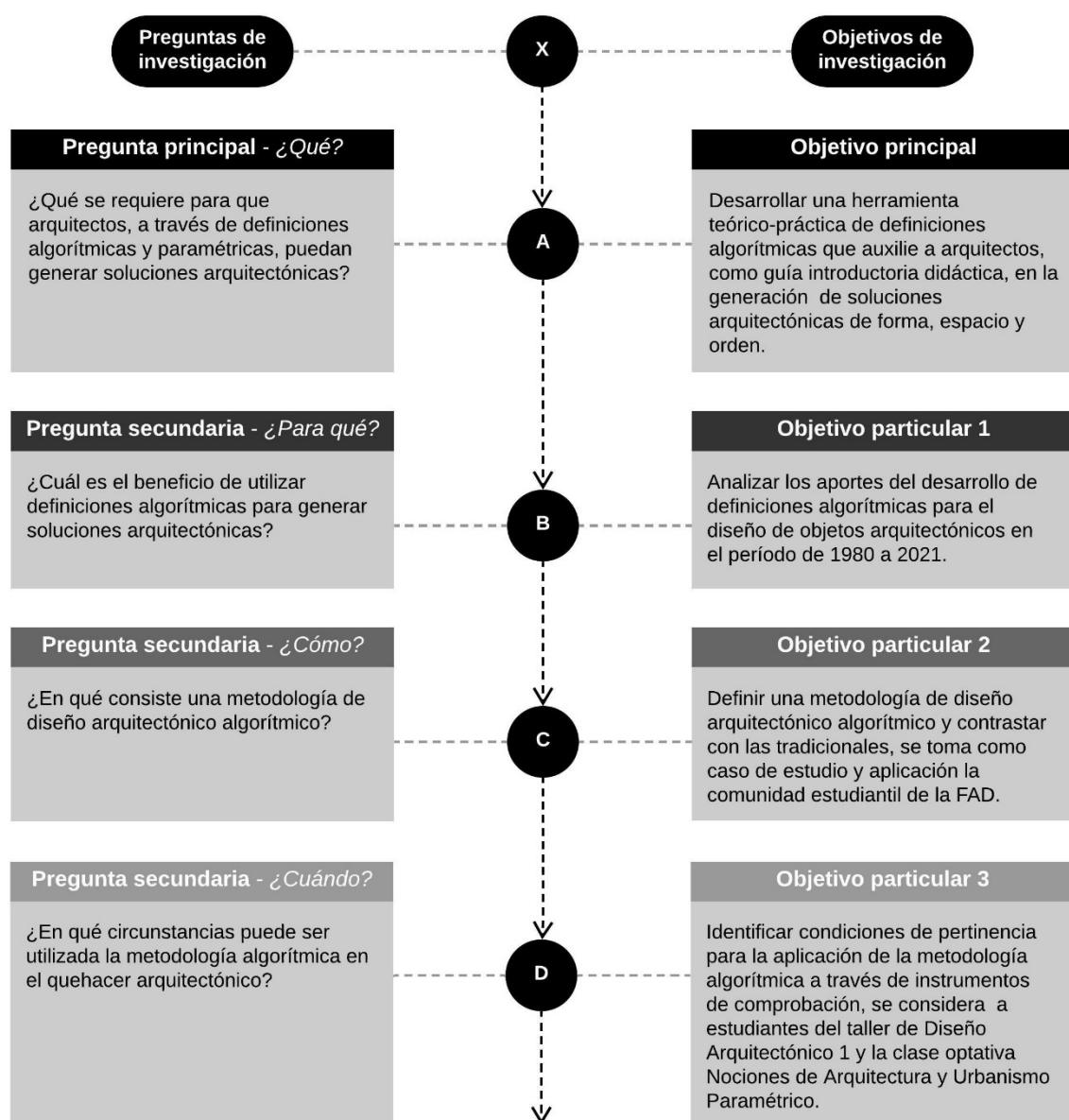


Figura 3. Estructura de preguntas y objetivos de investigación. Elaboración propia (2019).

II. MARCO TEÓRICO

Ante el objetivo principal de desarrollar una herramienta algorítmica de diseño arquitectónico, la posición de la presente investigación es utilizar la computadora en el proceso de diseño arquitectónico a manera de programación digital, es decir se establece en el segundo nivel que concluye Ervin (citado por Cantrell y Mekies, 2018), o en la categoría de “concepción” según Reas y McWilliams (2010), también entendida como “computación” por Terzidis (2006).

En virtud de optimizar los procesos de la producción de un proyecto y privilegiar la eficiencia en la creación de este, explorar esta segunda categoría implica el desarrollo de algoritmos para así constituir a través del lenguaje una comunicación amplia con la computadora; por tanto, es importante definir qué es un algoritmo, y qué implica trabajar con un enfoque paramétrico.

2.1. Algoritmos y la Arquitectura Algorítmica

Se considera algoritmo un procedimiento para aproximarse o solucionar un problema en un finito número de pasos, con el uso de operaciones lógicas (Terzidis, 2006). En programación computacional se entiende como una secuencia de pasos para ejecutar una tarea (Shiffman, 2015).

Si bien, el uso de algoritmos no es exclusivo de los sistemas computacionales, es allí donde su uso es fundamental, ya que éstos no solamente tienen la función de describir los pasos del proceso para la solución del problema, sino también la de comunicarlos a un tercer agente. Es decir, un algoritmo sirve como un traductor o mediador entre el pensamiento humano y el procesador de una computadora (ver Figura 4).

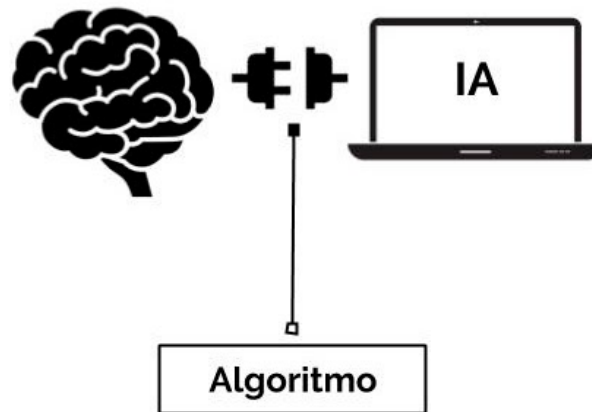


Figura 4. El algoritmo como traductor o mediador entre el pensamiento humano y el procesador de una computadora. Elaboración propia (2019).

Este hecho es fundamental ya que es el medio para desvanecer la desconexión entre el diseñador y el computador al diseñar, y comunicar el procedimiento de solución de un problema a través del pensamiento humano descrito en forma algorítmica (Terzidis, 2006).

Si bien, los programas CAD son desarrollados a través de algoritmos, sus interfaces en el objetivo de facilitar su uso no permiten la manipulación directa de los mismos, sino que éstos se encuentran englobados en una serie de instrucciones predefinidas y preparadas para el uso de la persona; es así como estas instrucciones interactúan aisladas e independientes unas con otras.

Trabajar algorítmicamente (ver Figura 5), implica tener conciencia de que: “(1) un algoritmo puede ser escrito de diferentes maneras; (2) un algoritmo requiere de suposiciones; (3) un algoritmo incluye decisiones; (4) un algoritmo complejo puede ser separado en piezas modulares” (Reas y McWilliams, 2010, p. 13).

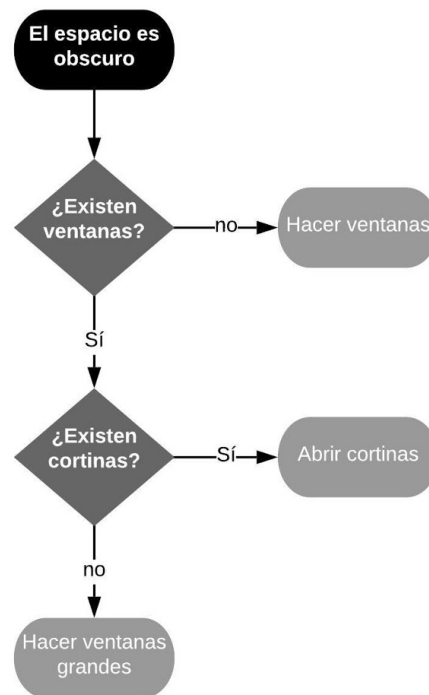


Figura 5. Ejemplo de algoritmo expresado en diagrama de flujo. Elaboración propia (2019).

Por lo tanto, el desarrollo algorítmico con aplicación en la arquitectura brinda la oportunidad de generar variantes de diseño y de hacer una descripción de las ideas en una definición algorítmica, la cual tiene como fin encontrar y transmitir a la computadora las rutas de solución a los problemas arquitectónicos, a través de modelos tridimensionales, simulación y procesos de cálculo. Terzidis (2006) define a la Arquitectura Algorítmica como “Algotectura” y en términos más amplios Caetano et al. (2020) describe el concepto como:

El diseño algorítmico es un paradigma de diseño que usa algoritmos para generar modelos y, por tanto, puede ser considerado también como generativo. Sin embargo, en diseño algorítmico, existe una correlación entre el algoritmo y el modelo generado, proveyendo así trazabilidad y permitiendo al usuario identificar las partes del algoritmo que generaron una parte dada del modelo. En este sentido, en el diseño algorítmico, el algoritmo es isomórfico al modelo. (2020, p. 9).



Figura 6. Serpentine Gallery Pavilion de Toyo Ito, proyecto construido (Londres, 2002). Fuente: Ito, T. (2002). Serpentine Gallery Pavilion 2002 [Imágen]. Recuperado de http://www.toyo-ito.co.jp/WWW/Project_Descript/2000-/2000-p_08/2000-p_08_en.html. Copyright [2002] © Nacása & Partners Inc.



Figura 7. The Morning Line de Aranda & Lasch, proyecto construido (Sevilla, 2008). Fuente: Aranda, B. y Lasch, C. (2008). *The Morning Line* [Imágen]. Recuperado de <http://arandalasch.com/works/the-morning-line/>.

2.2. Arquitectura Paramétrica

El término paramétrico es consecuente de la palabra parámetro el cual procede de raíces griegas; etimológicamente se encuentra compuesta por el prefijo *para* (junto a, a lado de, contra de) y la palabra *metron* (medida, instrumento para medir), cuyo significado refiere a un dato o factor que se toma como referencia para analizar o valorar un problema.

Según Jabi (2013) “el diseño paramétrico es un proceso basado en el pensamiento algorítmico que permite la expresión de parámetros y reglas que, juntas, definen, codifican y clarifican la relación entre las intenciones y respuestas de diseño” (2013, p. 201). Por tanto, es sistematizar una serie de parámetros e instrucciones entendidas como variables, que juntas desarrollan un algoritmo, y permiten verificar de manera interactiva la consecuencia de manipular un dato en el objeto arquitectónico resultante. En esencia, según Caetano et al. (2020):

Es un enfoque de diseño basado en el uso de parámetros para describir un conjunto de diseños ... el diseño paramétrico (DP) también se encuentra relacionado con el paradigma *Building Information Modeling* (BIM), ya que usa conceptos de DP como geometría asociativa y relaciones topológicas que establecen dependencias a través de diferentes elementos de diseño. (2020, pp. 7, 11).



Figura 8. Mercedes-Benz Museum de UNStudio, proyecto construido (Stuttgart, 2006). Fuente: van Berkel, B. (2006). Mercedes-Benz Museum [Imágen]. Recuperado de <http://www.unstudio.com/en/page/12482/mercedes-benz-museum>. Copyright [2006] © Eva Bloem



Figura 9. Queen Alia International Airport de Foster + Partners, proyecto construido (Amman, 2012). Fuente: Foster, N. (2012). Queen Alia International Airport [Imágen]. Recuperado de <https://www.fosterandpartners.com/projects/queen-alia-international-airport/>. Copyright [2012] ©

Foster + Partners

2.3. Posicionamiento histórico de la Arquitectura Algorítmica-Paramétrica

Es difícil encontrar una ruta nítida hacia los antecedentes históricos de la arquitectura paramétrica y su estrecha relación con el desarrollo algorítmico. Sin embargo, existen referencias importantes que han sido determinantes en la visión de generar proyectos con la priorización del parámetro como el elemento detonador de respuestas arquitectónicas.

Una de las primeras alusiones del pensamiento paramétrico, aún no computacional, sino a través de una metodología de diseño arquitectónico en donde las herramientas para desarrollarla son análogas, datan del movimiento moderno a través de los arquitectos Antoni Gaudí y Frei Otto, ambos considerados por Burry (2016) como *Proto-parametricistas*.

En el marco del desarrollo de la Segunda Guerra Mundial, el británico matemático y científico, Alan Turing, establece la fundación de la computadora moderna en el año 1937 en su artículo seminal de “Números Computables”. El hecho de que una máquina tuviera la facultad de completar una tarea humana en el menor tiempo posible significó para sus contemporáneos un concepto totalmente nuevo (Cantrell y Mekies, 2018).

Poco después, en 1939 el arquitecto italiano Luigi Moretti acuñó por primera vez el término “Arquitectura Paramétrica”. Según Tedeschi (2014), Moretti fue también quien inmediatamente comprendió el potencial de la computadora aplicada en el diseño:

En su investigación ‘Las relaciones entre las dimensiones dependen de varios parámetros’ culminaron en una innovativa exposición de sus modelos de estadios... los parámetros diseñados por Moretti fueron asociados a los ángulos de vista y la viabilidad económica en esos proyectos: la forma final fue generada para calcular pseudo isocurvas, que intentaron optimizar vistas desde cada posición en el estadio. (2014, p. 20).

Otro suceso importante en torno al uso computacional en la arquitectura fue la invención del programa *Sketchpad* en 1963 por el científico en computación americano Ivan Sutherland, entendida como la primera máquina de comunicación gráfica y el primer programa interactivo *Computer Aided Design (CAD)*. Tedeschi (2014) considera que tanto Moretti como Sutherland fueron pioneros y marcaron una revolución en las técnicas de diseño arquitectónico, al innovar en las herramientas utilizadas por los arquitectos.

Sin embargo, también ejerció una importante influencia el ingeniero francés Pierre Bézier, ya que es quien desarrolla en 1966 el primer sistema comercial CAD que no solamente tenía la capacidad de manipular formas simples, sino también una clase de curva llamada *B-splines* (Wujec, 2017). Este programa corría en computadoras centrales y fue principalmente utilizado en la industria automotriz.

El avance tecnológico en torno a los sistemas computacionales, así como la accesibilidad comercial de los mismos comenzaron a tener una presencia más fuerte en la sociedad a nivel global. La influencia de *Sketchpad* abre paso en 1982 al programa computacional con mayor éxito comercial: *AutoCAD*, programa que hasta la fecha es altamente utilizado en la profesión. Éste hecho marcó pauta para proyectar nuevos modelos de trabajo con el uso computacional en varias disciplinas afines al diseño arquitectónico, ingeniería y construcción.

Wujec (2017) identifica que la relación entre el diseño y la computación, se desarrolla en cuatro eras (Figura 10): (1) la primera deriva de la invención de los programas CAD, los cuales esencialmente imitan la acción de dibujar, (2) la segunda se relaciona con el avance de los programas CAD para producir, visualizar y manipular modelos en tercera dimensión, (3) la tercera corresponde al momento en el que fue posible agregar propiedades físicas a esos modelos para poder ser usados en simulaciones; nuevos algoritmos permiten la interrelación

de los objetos, (4) nace a partir de la cuarta revolución industrial y es la era en el que las máquinas tienen la capacidad de aprender.

Si bien, las cuatro eras anteriores describen el avance del CAD, es hasta la cuarta era cuando realmente se le puede asignar a la computadora la propiedad de asistir al diseño. En las primeras tres eras, las computadoras asisten a representar y documentar. En palabras de Wujec (2017), “el diseño se encontraba, en gran medida, en la imaginación del diseñador. La computadora necesitaba ser instruida para representar la idea de la imaginación a una forma legible en la máquina” (2017, p. 88).

En la cuarta era, la computadora muestra una real inteligencia ya que combina algoritmos, reconocimiento de patrones, redes neuronales, Inteligencia Artificial, diseño generativo y computación distribuida; para pasar de ser una herramienta representacional a socios creativos que ayudan a explorar y tomar decisiones de diseño (Wujec, 2017).

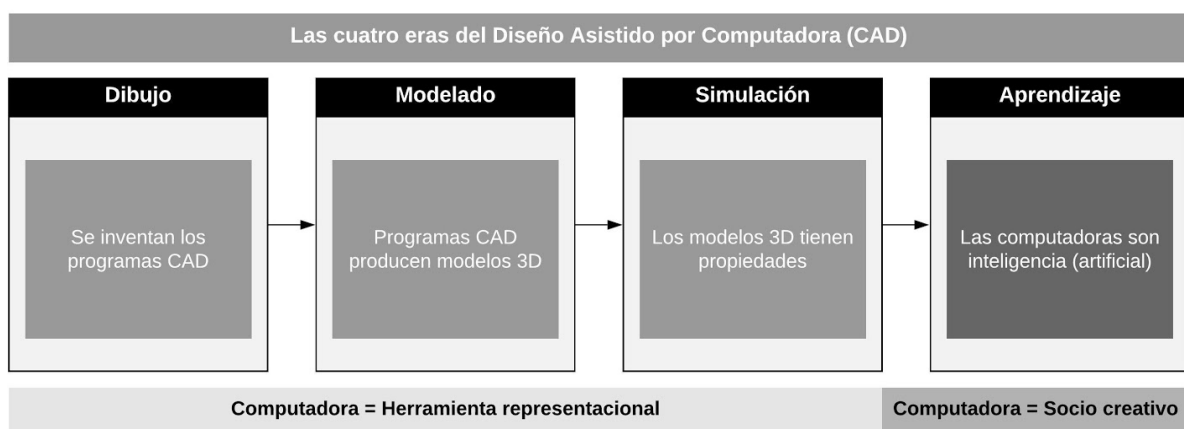


Figura 10. Las cuatro eras de la relación entre diseño y computación: Diseño Asistido por Computadora (CAD), según Wujec (2017). Elaboración propia (2019).

2.3.1. La visión morfológica del diseño arquitectónico computacional.

Es posible posicionar el uso computacional a nivel internacional en la arquitectura a partir de la caracterización taxonómica que Frampton (2014) propone sobre la arquitectura

en la era de la globalización como *Morfológica*, en el período que corresponde de 1980 al año 2010.

La manifestación de un grupo de arquitectos a finales del siglo XX en la escena europea como: Zaha Hadid, Ben van Berkel, Wolf D. Prix, Hani Rashid, así como en la norteamericana: Frank O. Gehry, Eric Owen Moss, Thom Mayne, Greg Lynn; tienden a emular la estructura de la forma biológica y natural gracias a los avances en la tecnología digital que permitieron explorar esas inquietudes con programas de *Computer Aided Design* (CAD) como medio para el desarrollo formal de sus proyectos. En palabras de Lynn (1998), “romper con la simetría no es una pérdida sino un incremento en organización dentro de un sistema abierto, flexible y adaptable” (1998, p. 12).

Sin embargo, los arquitectos mencionados anteriormente partían de métodos y diagramas analógicos que después eran trasladados y trabajados digitalmente, con el objetivo de reproducir sus formas (Eisenman, 2011). Es Greg Lynn (1998) quien desarrolla un discurso en torno al algoritmo informático a favor de un enfoque biomórfico en la arquitectura, con un origen totalmente digital.

La tecnología digital es capaz de construir imágenes intermediarias entre dos puntos fijos, resultando en una transformación suavizada. Esos efectos suavizados calculan las figuras intersticiales entre las figuras fijas. Además, el proceso de *morphing* es suficientemente flexible a los estados intermedios que son posibles. (Lynn, 1998, p. 123).

Cabe mencionar, que en todos estos casos tiene un mayor peso las transformaciones de la forma del producto a diseñar que las variables determinantes en la configuración de un proyecto, como lo son el clima, la topografía, los materiales, el programa arquitectónico y la función. La manera, por tanto, tiene un enfoque principalmente *escultórico* al momento de desarrollar una obra de arquitectura.

Carpo (2017) reconoce el momento explicado anteriormente, específicamente a partir de 1990, como *primer giro digital* de la Arquitectura; cuando los arquitectos adoptan los cambios en las herramientas digitales antes que cualquier otra disciplina, y ven en ellas una oportunidad de generar productos *personalizados* en masa sin generar un costo añadido, y no *estandarizados* como hasta entonces era el paradigma de producción y mercado a nivel internacional. “Arquitectos -no todos, pero si los mejores- adoptaron las herramientas digitales y abrazaron el cambio digital antes que cualquier otro negocio, industria o profesión creativa” (Carpo, 2017, p. 3).

Posteriormente, ya en siglo XXI, Patrik Schumacher (2011, 2012) -socio principal en *Zaha Hadid Architects*- busca posicionar la arquitectura paramétrica computacional en un nivel estilístico o paradigmático, mediante el manifiesto hecho en el 2008 para la Bienal de Venecia llamado *Parametricism as Style* (2009) donde hace una descripción de las cualidades y características formales que los productos arquitectónicos deberían poseer a través del uso constante de técnicas de diseño paramétrico como el camino hacia la gestación de un nuevo estilo al cual llama “Parametricismo”.

Schumacher (2009, 2011, 2012) considera al *Parametricismo* como el sucesor del movimiento moderno; agradece su génesis a los nuevos avances en la tecnología digital, que le dan la capacidad de desarrollar una estética de fluidez, continuidad, diferenciación y complejidad, con una mayor sintonía a la encontrada en la naturaleza, en donde no encontramos líneas rectas.

Si bien, su visión reconoce las estrategias paramétricas como un mecanismo importante en el uso computacional para el diseño de objetos arquitectónicos, su manifiesto tiene un fuerte apego a la búsqueda de la novedad en la estética del producto. Esta es una inquietud cercana a las motivaciones morfológicas de tipo esculturalista que le antecedieron.

2.3.2. Esculturalismo y Parametricismo como discursos morfológicos de la era digital.

Primeramente, al arquitecto Peter Eisenman (2011) se le atribuye anticipar un cambio cultural y técnico que engendrará un nuevo estilo arquitectónico basado en curvas y superficies suavizadas (Carpo, 2017). También, es considerado un precursor en el desarrollo de procesos de personalización en masa y heterogeneidad de objetos. “*Eisenman Architects* adoptó las primeras tecnologías de computacional personal accesibles para explorar conceptos extraídos de la biología como la adaptación y la morfogénesis, así como de la filosofía de Gilles Deleuze” (Linder et al., 2019, p. 24).

Es el mismo arquitecto Eisenman (2011) quien considera que Greg Lynn (1998) propone diagramas arquitectónicos donde el origen formal parte de lógicas computacionales, es decir, de las operaciones propias que tienen lugar en el tiempo a través de un algoritmo informático.

Es así, que es posible distinguir a Lynn como el punto de partida en la manipulación de algoritmos en el diseño computacional bajo el discurso de emular organizaciones biológicas con un origen totalmente digital y sin antecedente o registro de información determinante para el desarrollo de la forma arquitectónica (Eisenman, 2011).

Por otro lado, Frampton (2014) considera a Lynn como el principal teórico morfológico de la “figura”, que evoca los principios comunes de la forma animal como medio para fundamentar la prioridad de la curvilinealidad y la metamorfosis digital para favorecer las superficies polimórficas a las que llama *BLOBS* (glóbulos). “Lynn se opuso diametralmente a la derivación de la arquitectura a partir de puntos fijos en el espacio - geométrico, material, espacial- que existían en un ideal conjunto de coordenadas cartesianas” (Linder et al., 2019, p. 24).

La postura de Lynn, aunque opera a través de algoritmos, es realmente escultórica y no algorítmica, ya que se encuentra dirigida a valorar las características formales resultantes de un proceso morfogénico computacional, y son ajenas a ser producto responsivo de las problemáticas arquitectónicas. Según Bouman (1998):

Lynn prefiere una arquitectura experimental en la cual conceptos son desarrollados y expresados a través de lo que Deleuze llamaría “máquinas abstractas”. Esas son técnicas que revelan las fortalezas de un programa dado sin inmediatamente instrumentalizarlas en un diseño concreto ... La gran pasión de Greg Lynn es la forma. Sin embargo, desde el punto de vista de esa pasión él hace arquitectura amorfa”. (citado en Lynn, 1998, p. 12).

Así, Greg Lynn (1998) desarrolló un planteamiento inexplorado y diferente a las manifestaciones arquitectónicas pertenecientes a las última década del siglo XX, donde el uso computacional como auxiliar en el diseño era únicamente representacional a través del dibujo digital.

Los estudios a profundidad de los recursos computacionales aplicados en el desarrollo arquitectónico, así como los diagramas arquitectónicos de Lynn en torno al *BLOB*, constituyen una de las principales directrices del *primer giro digital en la Arquitectura*, en donde los métodos causales derivados de modelos matemáticos diferenciales son importantes para la resolución morfológica del carácter complejo y diferenciado del manejo de geometrías curvilíneas y amorfas.

Lynn reconoce el papel fundamental desempeñado por paradigmas tan potencialmente morfológicos como la invención del cálculo diferencial y por la evidencia de la indeterminación dinámica en la naturaleza, revelada gracias a los modelos matemáticos tanto en las ciencias puras como en las aplicadas, en campos como la

biología, la embriología, la virología, la geología, la aerodinámica, la cibernética y la ingeniería en estructuras. (Frampton, 2014, p. 363).

Cabe mencionar que pocos años después, en el 2002, Frank Gehry funda la empresa *Gehry Technologies* con el fin de desarrollar avances tecnológicos en torno a programas paramétricos de edificación del tipo *Building Information Modeling* (BIM), que permitieron un progreso en el ámbito de proyecto colaborativo interdisciplinario, pero no lograron ser usados más allá del carácter representacional con el que se utilizaban las herramientas computacionales en el ámbito arquitectónico de ese entonces.

Ya iniciado el siglo XXI, ante la influencia latente de las teorías y metodologías de Lynn, son exploradas las plataformas algorítmicas y de programación computacional en la arquitectura de una manera más amplia, así como también comienza a tomar mayor relevancia el diseño a través de definiciones paramétricas.

Aún el estilo del *BLOB*, también conocido como el estilo del *spline* o la racionalización digital, se convirtieron en el sello distintivo de la primera era digital de los años 90. Actualmente ambas tendencias, tanto la técnica como la estilística, a menudo van bajo la rúbrica del Parametricismo; En aquel entonces, ambos cayeron en tiempos difíciles cuando a principios del nuevo milenio, estalló la burbuja del internet. (Carpo, 2017, p. 4).

Es hasta el 2008, cuando el diseño paramétrico es discutido en el medio arquitectónico internacional de manera más amplia con el manifiesto de Patrik Schumacher (2009, 2011, 2012) *Parametricism as Style*, y es allí cuando se cuestiona si el *Parametricismo* establecería el inicio de una nueva manera de hacer y transmitir Arquitectura.

Schumacher (2009, 2011, 2012) es uno de los exponentes más importantes de la arquitectura computacional en la segunda década del siglo XXI, ya que fue quien planteó un

primer escenario sólido en torno al diseño a través del parámetro digital como medio relevante para la generación morfológica de un sistema arquitectónico.

Su visión obedece a las principales preocupaciones de la era digital en la primera década del siglo XXI; poner a prueba todos estos alcances computacionales para desarrollar sistemas complejos con geometría curvilínea, fluida, de continuidad y diferenciación constante, en la búsqueda de un nuevo paradigma a través de una naciente estética; una filosofía en consonancia con los objetivos modernos del siglo pasado.

Esta postura, a diferencia de Lynn, sí toma en cuenta información relacionada con la problemática arquitectónica, como los parámetros que alimentarán la generación de un sistema formal y espacial que moldeara un objeto arquitectónico, y que no es producto del desarrollo autónomo de un algoritmo computacional aislado. Sin embargo, este método parametricista puede ser clasificado como *Morfológico*, ya que considera la forma resultante como la parte de mayor valor en el objetivo de consolidar una nueva estética estilística.

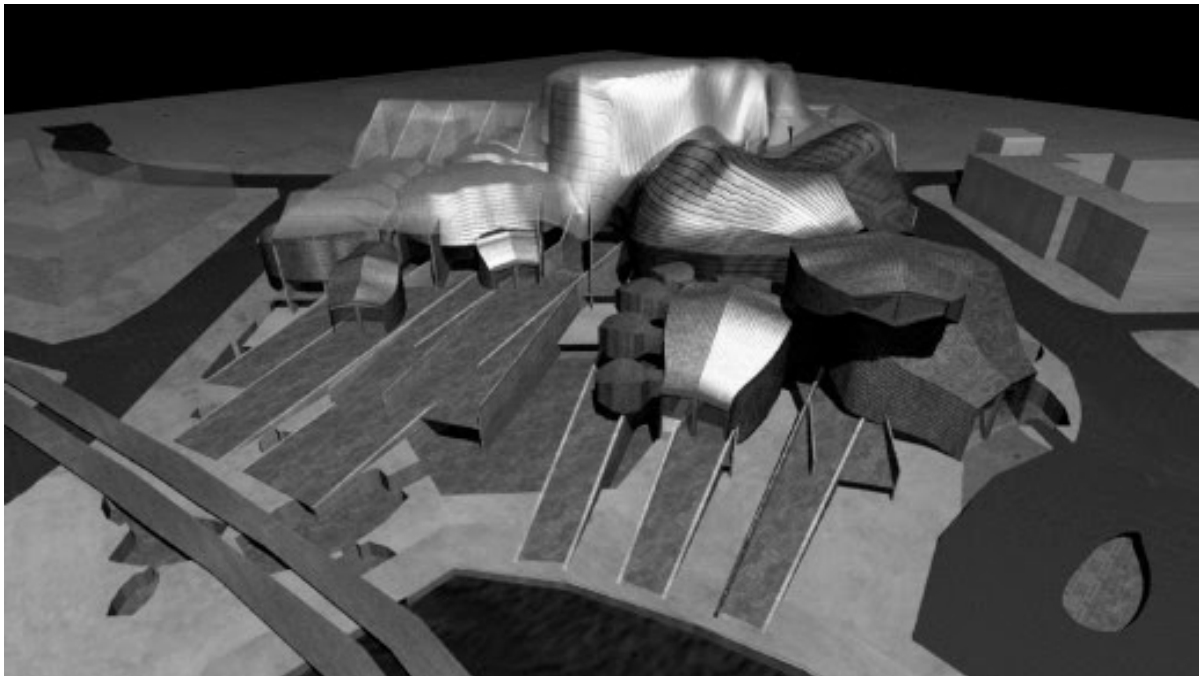


Figura 11. Cardiff Bay Opera House de Greg Lynn, proyecto para concurso (Gales, 1994). Fuente: Lynn, G. (1994). *Cardiff Bay Opera House* [Imagen render]. Recuperado de <http://glform.com/buildings/cardiff-bay-opera-house/>. Copyright [1994] © Greg Lynn FORM



Figura 12. Heydar Aliyev Centre de Zaha Hadid Architects, proyecto construido (Azerbaijan, 2012).

Fuente: Hadid, Z. (2012). *Heydar Aliyev Centre* [Imágen]. Recuperado de <https://www.zaha-hadid.com/architecture/heydar-aliyev-centre/>. Copyright [2012] © Iwan Baan

2.3.3. El enfoque tecnológico y político del siglo XXI.

Paralelo a la visión de Schumacher (2009, 2011, 2012), el siglo actual muestra una discusión nutrida y diversa con relación al diseño arquitectónico paramétrico. En contraste al enfoque estilístico del Parametricismo, surgen discursos que consideran más importante la aportación *tecnológica* de los sistemas computacionales y sus herramientas que los objetivos morfológicos en el diseño arquitectónico. Este enfoque aprovecha el acelerado avance tecnológico de los recursos digitales, y los ve como un medio de transformación en los sistemas culturales, creativos y de producción.

Con el fin de describir los momentos clave en los que la tecnología ha cambiado masivamente la forma de producir cosas, Wujec (2017) explica las cuatro revoluciones industriales acontecidas:

La primera nace en el siglo XVIII con la máquina de vapor ... la segunda emerge en la última etapa del siglo XIX ... cuando se impulsa la producción en masa ... Una tercera se apoderó de los años 70, impulsada por las herramientas digitales ... Ahora estamos entrando a una cuarta revolución, la cual no es manejada por una sola fuerza, sino por una combinación de tecnologías, incluyendo sensores, algoritmos y robótica. (2017, p. 30).

Históricamente la tecnología transforma el avance de todos los oficios y profesiones, incluyendo la Arquitectura; sin embargo, según Carpo (2017) ésta tiende a tardar en adoptar los cambios tecnológicos. Considera que aún en la primera etapa del modernismo, los arquitectos no iban en sintonía con los avances que la humanidad ya había desarrollado, y regresaban al uso de las viejas estructuras de poste y dintel de la antigüedad clásica, a pesar de que la ingeniería Romana ya había generado arcos y bóvedas, y la arquitectura Gótica construido sorprendentes torres y pináculos de gran altura.

La edad de oro del modernismo del siglo XX representa ser el momento en el que los arquitectos van por primera vez al ritmo del mundo industrial. La Arquitectura, adopta o sigue el ejemplo de la tecnología; sin embargo, no la genera hasta el primer giro digital de los años 90 mencionado anteriormente.

De esta forma, se le atribuye antes que cualquier otra disciplina, la idea de abandonar la *producción estandarizada en masa* para trasladarse a la *personalización digital en masa* (Carpo, 2017). “El traslado del Post-Fordismo en la Arquitectura se caracteriza claramente por este cambio de la estandarización en masa a la personalización en masa de la segunda mitad del siglo XX” (Linder et al., 2019, p. 24)

Esta última causa, fue la que impulsó de forma significativa al diseño paramétrico a nivel internacional y se consolidó a través del desarrollo teórico y metodológico propuesto por Schumacher (2009, 2011, 2012) con el Parametricismo.

El Taylorismo y Fordismo no fue inventado por arquitectos, éstos simplemente siguieron su ejemplo ... Para bien o para mal, la personalización en masa fue nuestra idea (de arquitectos): fue desarrollada, perfeccionada, probada y conceptualizada en un puñado de escuelas de arquitectura de Europa y Estados Unidos en los años 90. (Carpo, 2017, pp. 2, 4).

Como parte de las motivaciones de la geometría diferenciada del primer giro digital de la Arquitectura, y no tanto por la continuidad formal suavizada del Parametricismo, Terzidis (2006) parte de revisar las raíces etimológicas de la palabra diseño, para así cuestionar la existencia de la novedad y si ésta tuviera que ser el espíritu de las manifestaciones de Arquitectura Paramétrica (AP). Apoyado en la filosofía pre-Socrática, menciona que todo conocimiento, proceso o producto generado tiene precedente o raíz, y por tanto lo entiende como emergente en vez de novedoso.

A través de esta idea destaca la importancia algorítmica de las definiciones paramétricas y el carácter difuso que comparte con el proceso de diseño en busca de respuestas hacia lo desconocido o lo todavía no encontrado. Terzidis (2006) no posiciona la arquitectura paramétrica, en su enfoque *Algorítmico*, como un paradigma o estilo, sino como un proceso en el quehacer arquitectónico con el uso de la tecnología digital.

Carpo (2017) distingue que, aunque las computadoras de ese tiempo ya trabajaban de manera eficaz, la informática era aplicada bajo la lógica de un suministro limitado de datos, debido a que eran aún considerados artículos costosos. Esta lógica del “*Small Data*” es la que caracterizaba la forma de trabajar de esa primera era digital de la Arquitectura.

Conforme avanza el tiempo, la tecnología digital también lo hace a un ritmo cada vez más acelerado, y el primer giro digital de la arquitectura termina en el momento que la información comienza, gracias al internet, a proliferar tanto como el acceso de las personas a las computadoras personales, las cuales progresivamente tienen mayor capacidad de

almacenamiento. Esto abre camino al concepto de “*Big Data*” (mencionado por primera vez por Erik Larson en 1989) que se refiere a la capacidad técnica lograda para almacenar y procesar una cantidad creciente de datos a un costo cada vez más bajo.

Esta evolución tecnológica, está en proceso de cambiar los procesos de producción y la manera en el que se desenvuelven todas las disciplinas del siglo XXI. En Arquitectura el cambio sucede a partir del 2010; poco después de que *Gmail* dejará de ser una versión *beta*, y haya forjado en los usuarios un criterio de búsqueda y no de clasificación de información como un medio más coherente de desplazamiento en el *Big Data*.

Así, a partir de ese año se da el paso al *segundo giro digital de la Arquitectura* (Carpo, 2017). Este suceso representó una evolución significativa en la Inteligencia Artificial (IA), con repercusión no sólo en la Arquitectura, sino en todas las áreas de conocimiento.

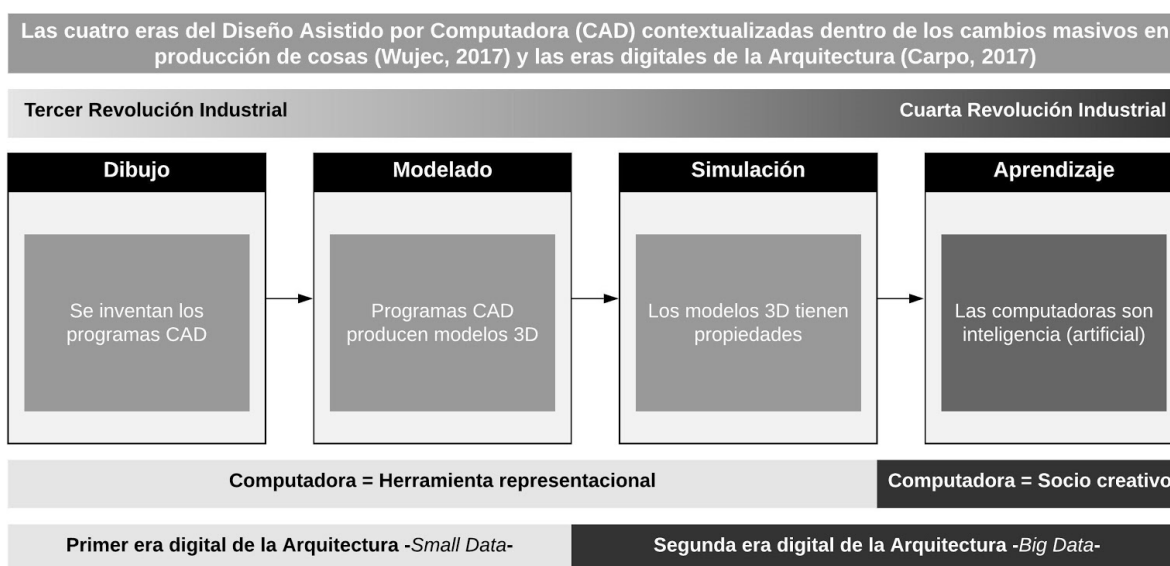


Figura 13. Diagrama descriptivo de las cuatro eras de la relación entre diseño y computación: Diseño Asistido por Computadora (CAD), contextualizado dentro de las cuatro Revoluciones Industriales según Wujec (2017) y las eras digitales de la Arquitectura según Carpo (2017). Elaboración propia (2019).

Carpo (2017, 2019) sostiene que dicha repercusión ha cambiado el método científico; existe un traslado de la ciencia moderna impulsada por la Revolución Industrial y caracterizada por una lógica matemática que tiene como fin determinar relaciones de causa y efecto a través de un limitado número de factores (*Small Data*), a una nueva que aprovecha las virtudes de la IA revolucionada por el latente *Big Data* que permite resolver problemas complejos a través de simulaciones producidas por la repetición interminable de operaciones simples en un periodo corto de tiempo.

Es posible intuir en esta aproximación una afinidad epistemológica con los principios rizomáticos de Deleuze y Guattari (2002) de los sujetos nómadas que manifiesta una postura contraria a la ciencia tradicional:

Una posible posición del sujeto que quedaría descrita por los principios de organización rizomáticos -de conexión y heterogeneidad, de multiplicidad, de ruptura asignificante, de cartografía y de calcomanía-, contrapuestos a los clásicos modelos arborescentes o piramidales, del tipo causa-efecto, implícitos a las formulaciones científicas y filosóficas tradicionales. (Abalos, 2017, p. 148).

Esta nueva lógica marca pauta a una metodología de diseño donde las simulaciones computacionales son fundamentales, en vías de solucionar problemas arquitectónicos complejos relacionados con la forma, el espacio y su adaptabilidad con el entorno. En otras palabras, los nuevos métodos se encuentran en la adopción de un carácter *heurístico* computacional, para dejar atrás el modelo que caracterizó la modernidad apoyada en lógicas *causales* producto de la necesidad de ecuaciones matemáticas y la abstracción por la falta de información.

Las computadoras pueden trabajar de esa manera, y nosotros no, porque el ser humano no puede ejecutar una gran cantidad de cálculos cualquier cantidad de tiempo

práctico. A diferencia de los humanos, la computadora no necesita ciencia para comprimir y acortar sus operaciones basadas en números. (Carpo, 2019, p. 89).

El método heurístico basado en IA implica procesos de simulación generativa para la resolución de problemas, la optimización y desempeño en el diseño a través de sistemas algorítmicos evolutivos (ciclos de prueba y error) en un corto periodo de tiempo.

La importancia algorítmica destacada por Terzidis (2006) en el diseño paramétrico de inicios del presente siglo, toma un carácter pertinente al enfoque heurístico del segundo giro digital de la Arquitectura. Las principales estrategias algorítmicas producto de este enfoque, tales como las búsquedas estocásticas, sistemas-L, autómatas celulares o fractales; abren camino al concepto de *Diseño Generativo (DG) en la Arquitectura*. Según Caetano et al. (2020):

Algunos autores definen el DG como un proceso que principalmente se refiere a técnicas evolucionarias tanto para la creación, como para los procesos de producción de las soluciones de diseño; sin embargo, otros no restringen el DG con los procesos evolucionarios, considerándolo un enfoque de diseño basado en procesos algorítmicos basados en reglas que generan múltiples, y posibles, soluciones complejas. (2020, p. 8).

Las aportaciones del enfoque algorítmico, la evolución de los sistemas computacionales, la IA, el avance en los medios de producción y la robótica del segundo giro digital de la Arquitectura (Carpo, 2017, 2019) o la cuarta era de Diseño Asistido por Computadora (Wujec, 2017), motivan a arquitectos como Achim Menges (2008; 2008, 2015; 2020), Neri Oxman (2015), Gramazio y Kohler o Marc Fornes a explorar las áreas de oportunidad abiertas por estas innovaciones.

A través de la investigación -cualitativa, cuantitativa, experimental- y diseño generativo, desarrollan proyectos avanzados que son ejemplares en la optimización de

desempeño de su materialidad y cualidades constructivas, para apuntalar la vanguardia en el diseño y cambiar el paradigma de los procesos de fabricación y edificación. Este enfoque *Performativo* es producto del método heurístico, la simulación y los avances en cuanto al alcance realizativo de la tecnología. “El enfoque debe estar en explorar nuevas posibilidades y utilizar los medios de nuestro tiempo para tener en cuenta la complejidad de los aspectos cualitativos y cuantitativos que componen un edificio” (Menges y Knippers, 2020, p. 8).

Automatización, biomimética, fabricación digital, impresión en tercera dimensión, materiales experimentales, auto-organización o auto-ensamble, son algunas de las estrategias que se exploran como nuevas metodologías en los procesos de construcción de objetos. En palabras de Menges (2015) tiene como objetivo investigar, “la reciprocidad entre el material y lo computacional en Arquitectura, sus múltiples y profundas interrelaciones con tecnología, biología y cultura” (2015, p. 7).

Por otra parte, la influencia de los últimos desarrollos arquitectónicos en relación con la tecnología digital ha motivado a otro grupo de arquitectos a alinearse con la crítica a los métodos causales modernos y al sistema político capitalista neoliberal que caracterizaron la segunda mitad del siglo XX y la primera década del siglo XXI, que tuvieron una condición arquitectónica de geometría diferenciada, fluida y ortodoxamente continua, con el predominio formal de la curvilinealidad derivada del uso de geometría basada en las *splines*.

La postura crítica y *política* va directamente en contra de las inquietudes escultóricas de los años 90 que evolucionaron hasta llegar a la cúspide parametricista de Schumacher (2009, 2011, 2012). Se posiciona en sintonía con la teoría de Carpo (2017, 2019) relacionada con la segunda era digital de la arquitectura, con un enfoque arquitectónico constituido por elementos separados, sincopados y repetidos serialmente para generar sistemas complejos generados, producidos y ensamblados, también bajo métodos heurísticos de simulación digital, sistemas de automatización y robótica.

Este enfoque adopta la denominación de *Discrete* (Retsin, 2019) o “Discreto”, desde su connotación de “separado” o “distinto” según la RAE, a partir de la primera aproximación de Carpo (2017) con el término *Discreteness* (Discreción). “‘*Discreteness*’ es una noción que proviene de las ciencias, que se refiere a lo que es individual y separado. Es lo opuesto a lo continuo, a lo que es ininterrumpido y fluido” (Retsin, 2019, p. 8).

Derivado de estos vocablos, Morel (2019) llama a esta visión *Discretism* (Discretismo), con el fin de alinearlo con su contraparte paradigmática *Parametricism*, en el artículo *The Origins of Discretism: Thinking Unthinkable Architecture* de la revista *Architectural Design* 89(2).

De esta forma, los arquitectos Gilles Retsin (2019; 2019), Philippe Morel (2019) o José Sánchez (2019b, 2020) bajo la idea del *Discretismo*, buscan dar respuesta con la tecnología digital más avanzada, la crítica y un enfoque *político* a la crisis económica del 2008. Se posicionan en un escenario donde los elementos que conforman el sistema arquitectónico son de repetición serial compleja, con el objetivo de generar formaciones más oportunas a las necesidades de aprovechamiento de materiales asequibles, más endeble o provenientes de la reutilización o desperdicio.



Figura 14. Research Pavilion 2011 de ICD/ITKE University of Stuttgart, proyecto liderado por Achim Menges y construido (Stuttgart, 2011). Fuente: Menges, A. (2011). ICD/ITKE Research Pavilion 2011 [Imágen]. Recuperado de <http://www.achimmenges.net/?p=5123>. Copyright [2011] © ICD/ITKE University of Stuttgart.



Figura 15. BUGA Wood Pavilion 2019 de ICD/ITKE University of Stuttgart, proyecto liderado por Achim Menges y construido (Heilbronn, 2019). Fuente: Menges, A. (2019). ICD/ITKE Research Pavilion 2019 [Imágen]. Recuperado de <http://www.achimmenges.net/?p=20987>. Copyright [2019] © ICD/ITKE University of Stuttgart.



Figura 16. Tallinn Architecture Biennale Pavilion de Gilles Retsin, proyecto construido (Tallin, 2017).

Fuente: Retsin, G. (2017). *Tallinn Architecture Biennale Pavilion* [Imágen]. Recuperado de

<https://www.retsin.org/Tallinn-Architecture-Biennale-Pavilion>.



Figura 17. Bloom, 2012 Olympic Games, pabellón personalizable de José Sanchez y Alisa Andrasek,

proyecto construido (Londres, 2012). Fuente: Sanchez, J. y Andrasek A. (2012). *Bloom* [Imagen].

Recuperado de <https://www.plethora-project.com/bloom>.

2.4.4. La era post-digital del diseño arquitectónico computacional.

A diferencia de la arquitectura vigente de Lynn (1998) y Schumacher (2009, 2011, 2012), los últimos enfoques sobre la arquitectura computacional se desligan de las exploraciones morfológicas promovidas en el siglo XX, y encuentran mayor significado a las aportaciones tecnológicas de las herramientas digitales, así como las oportunidades de automatización e IA, que se ha desarrollado de manera acelerada en los últimos años, gracias al fenómeno del *Big Data*.

Kostas Terzidis (2006) hace un análisis de las capacidades que puede brindar la computadora como auxiliar en el diseño, contrasta la “computación” y la “computarización” como dos maneras en las que puede ser utilizada. Mientras la primera es relacionada con el cálculo y la resolución de problemas difusos, la segunda se refiere al potencial de organización, procesamiento y representación de información ya preconcebida; estas últimas capacidades son utilizadas más intensamente en el quehacer arquitectónico a través de programas *Computer Aided Design* (CAD).

Este análisis, fue importante para sustentar el valor que encuentra en el algoritmo digital como el medio para explorar la capacidad de “computación” y para demostrar que esta forma es la única en la que la computadora realmente puede asistir en el proceso de concepción del diseño. Por lo que Terzidis (2006) clasifica las estrategias algorítmicas como recursos de diseño, y destaca su aplicación en la Arquitectura. Algunas de estas estrategias son: multi-booleanos, búsquedas estocásticas, fractales, autómatas celulares (y “el juego de la vida”), hibridación, *morphing* o recursión.

Contrario a Lynn (1998), que también explora el algoritmo como herramienta de diseño, el énfasis que le da Terzidis (2006) se apoya en considerarlo un medio para resolver la problemática arquitectónica, y no como un fin para definir la figura.

Aunque Terzidis (2006) coincide con Schumacher (2009, 2011, 2012) en que las definiciones paramétricas derivan del pensamiento algorítmico, difiere en considerar los resultados como una novedad estilística. Para el primero la Arquitectura Paramétrica es un proceso y no un producto con características definidas.

Impulsados por la evolución de la IA y la robótica, arquitectos como Achim Menges (2008; 2008, 2015; 2020), Neri Oxman (2015), Gramazio Kohler, Marc Fornes o de manera más reciente Skylar Tibbits, dirigen el discurso tecnológico a un nivel avanzado de fabricación, desarrollan innovaciones constructivas a través de destacar las oportunidades de optimización y desempeño ofrecidos por los sistemas algorítmicos generativos y heurísticos. “A medida que la computación se vuelve más poderosa y accesible, los algoritmos serán entrenados para participar en cada aspecto de creación” (Wujec, 2017, p. 139).

La investigación de estos exponentes en las propiedades de los materiales y el desarrollo de metodologías performativas aportan a cambiar el paradigma de hacer las cosas, así transgreden la personalización digital en masa con sistemas de producción automatizada e integración de capacidades de auto-organización con propiedades inherentes de auto-ensamble en los objetos y materiales.

Materiales de alta ingeniería, materiales biológicos, y nanomateriales han ampliado el rango de lo que los diseñadores pueden usar ... Las cualidades fundamentales de los materiales (rendimiento, estética e impacto, intuitivamente entendidas y aplicadas por los fabricantes) ahora se están volviendo computables. (Wujec, 2017, pp. 205, 206).

Este enfoque se interesa en la reciprocidad entre los sistemas naturales y los computacionales, a fin de estructurar características biomiméticas en los productos que ayuden a que la Arquitectura obtenga propiedades de mejor adaptación con el entorno en el que existe. Neri Oxman (2015) considera que:

Actualmente el diseño digital ha progresado al punto en donde podemos lograr una alta resolución espacial y complejidad material en la generación de forma y fabricación. En combinación con la emergencia en las áreas de investigación como la biología sintética, procedimientos *bottom-up* de diseño - tanto digitales como físicos - están ayudando a la transición del diseño biológicamente inspirado a la ingeniería biológicamente diseñada. Este cambio en práctica requiere nuevos métodos que ofrecen plantillas *top-down* empleadas para ensamblar entidades de formación *bottom-up* en una manera escalable y paralela. (2015, p. 107).

El valor Performativo encontrado en el diseño computacional apunta hacia un entendimiento fenomenológico de creación que no se queda en lo digital, sino que requiere ser traducido materialmente a lo físico, aunque su creación provenga de un método *top-down*, de caja negra o heurístico digital. Esta visión contrasta con los exponentes anteriores, donde sus principales exploraciones giran alrededor de las cualidades computacionales virtuales.

Desde el final de la primera década del siglo XXI, otras manifestaciones arquitectónicas se posicionan firmemente en la llamada era post-digital o también segunda era digital de la Arquitectura. Cantrell y Mekies (2018) consideran que la Arquitectura hoy en día ha mostrado una maduración, al alejarse de la complejidad que parece no llevar a ningún lado en la década pasada. “Post-2008, los arquitectos, instituciones y curadores se evacuaron de lo digital, encontrando refugio en una variedad de nuevas neo-tendencias y obsesiones, que van desde la austeridad-chic al neo-postmodernismo, desde la ontología orientada a objetos hasta el llamado post-digital” (Retsin, 2019, p. 7).

Los arquitectos Gilles Retsin (2019; 2019), Philippe Morel (2019) o José Sánchez (2019b, 2020) han desarrollado esta discusión en la publicación *Discrete* de la revista inglesa *Architectural Design* (Retsin, 2019) a través de una manifestación arquitectónica discontinua de carácter tectónico esquelético o fragmentado con articulaciones sincopadas, que acepta la

repetición de un elemento estándar, como cajas dispuestas a manera de *voxels* en un sistema de crecimiento abierto y orgánico; un criterio contrario a la continuidad fluida y curvilínea de la estética parametricista.

Patrik Schumacher ha reclamado que la arquitectura ha entrado a una nueva época estilística de Parametricismo ... Neri Oxman y Greg Lynn celebran la ‘disolución de la tectónica’ a medida que el tejido de la construcción del diseño y la Arquitectura se ha vuelto más fluido y continuo, eliminando la necesidad de partes, donde la materia pueda ser controlada a un grado infinitesimal de precisión ... En oposición ...

Discrete rompe con una tradición de personalización en masa de una simple edificación, tratando de reconstruir los bienes comunes a través del conocimiento distribuido y la producción de diseño. (Sanchez, 2019b, p. 25).

También es posible posicionar este enfoque dentro de un ámbito *político*, y no solo como una expresión *tecnológica*, ya que aboga por una economía de escala en la manera de utilizar los elementos que servirán de insumo en la producción arquitectónica, bajo una estrategia de agregación consecutiva.

La última década, después de la crisis financiera de 2008, ha visto una reacción violenta contra la arquitectura digital. Después de aproximadamente dos décadas de optimista y celebrada experimentación, se entendió de repente que lo digital en la Arquitectura estaba afiliado al neoliberalismo problemático que causó la crisis.

(Retsin, 2019, p. 7)

El objetivo es hacer un replanteamiento en la utilización de los avances computacionales aplicados en el diseño arquitectónico, de una manera más prudente con la crisis económica a través de lo que Sánchez (2019b, 2020) describe como una economía de escala, es decir generar repeticiones complejas automatizadas de materiales asequibles, endebles o provenientes de la reutilización o desperdicio.

Discrete architecture se encuentra inevitablemente vinculada con lo social y la posibilidad de que un mayor número de pequeñas empresas coexistan. El paradigma de la tectónica Discreta privilegia las unidades autónomas, partes que no son miembros subsidiarios de un entero. Las partes pueden ser recombinadas en múltiples permutaciones identificando una condición tectónica abierta. (Sanchez, 2019b, p. 24).

Es importante destacar que el replanteamiento mencionado anteriormente con relación a la tecnología avanzada se relaciona con el enfoque y no tiene la intención de alejarse de ella. Inclusive, una de las búsquedas de este enfoque es diluir las brechas entre (a) el proceso de diseño, (b) la generación del modelo digital, (c) el sistema de (d) fabricación y (e) la materialidad misma, al grado de convertir todos estos pasos en un solo proceso de autoproducción efectuado por las propiedades del mismo material diseñado, dentro del marco de la automatización y todo lo que ello implica (tecnología digital, IA, robótica).

Morel (2019), distingue que ahora se opera el registro de todo el conocimiento humano en fracciones de segundos gracias al método de discretización generalizada de información de las computadoras, y dirige así a la humanidad a una era de efectividad. Enfatiza que la computadora no realiza cálculos a la manera humana, sino que es esta última manera la que los evalúa de acuerdo su criterio a partir de que son producidos, sin embargo, declara que la Arquitectura del futuro será generada por una superinteligencia artificial, que tiene una condición híbrida entre la humana y la artificial.

Las similitudes que tal vez existan entre la Arquitectura de la Discreción y la humana no deberían engañarnos. De hecho, esto es lo que el idealismo arquitectónico ‘neo-neo racionalista’ (el post-digital) y la arquitectura política o biomimética neo-positivista hace, la última sólo reproduce la inteligencia de la naturaleza en la imagen de un antropofornismo renacentista ... Cualquier arquitectura futura será generada

masivamente por procedimientos computacionales que ya son el resultado de una (proto) superinteligencia. (Morel, 2019, p. 19).

Es posible concluir que la aproximación del Discretismo es afín a los diagramas automáticos producidos por lógicas computacionales que se manejaban desde Lynn (sin el concepto de una superinteligencia artificial de por medio); sin embargo, el discurso más allá de esta coincidencia es una visión totalmente diferente.

En primera instancia el resultado formal es fundamentalmente diferente y contrario, ya que el Discretismo no admite la fluidez y la curvilinealidad. Desde esta perspectiva, el enfoque no solo contrasta con los fines del Esculturalismo y el Parametricismo de la era digital de la arquitectura, sino que también se contrapone a las motivaciones biomiméticas de la aproximación Performativa de la era post-digital.

Sin embargo, coincide con este último enfoque en la metodología, ya que utiliza los procedimientos generativos y heurísticos para la creación, así como los sistemas avanzados en robótica para la producción, con la diferencia que el Discretismo visualiza que los métodos aditivos de fabricación, como la impresión en tercera dimensión y el ensamble automatizado, son los que representan la segunda era digital de la Arquitectura (Carpo, 2017). Por consecuencia, es evidente la afinidad con el enfoque Algorítmico de Terzidis (2006), que se desenvuelve en las dos eras.

La discusión acontecida en las últimas tres décadas en torno a la Arquitectura computacional refleja contrastes a partir del significado interpretativo de los parámetros inherentes del proceso de diseño; es decir, los valores en los que se apoyan estos importantes exponentes del escenario arquitectónico de la vanguardia tecnológica para darle sentido a las herramientas digitales que utilizan. Con relación a esta idea, Cantrell y Mekies (2018) sugieren que mientras las estrategias de diseño computacional abren nuevos paradigmas y

oportunidades de diseño, al final se trata de los valores añadidos y de un mejor desempeño en los proyectos.

Hoy en día, el paradigma paramétrico se está convirtiendo en una interpretación profunda del significado de los parámetros. Utilizando la lógica de los números, los arquitectos pueden aplicar una metodología de diseño integrada para descubrir datos específicos sobre aspectos como el rendimiento de los materiales, los métodos de fabricación y el comportamiento humano. (Yuan, 2016, p. 99).



Figura 18. Gemini de Neri Oxman, proyecto construido (Paris, 2014). Fuente: Oxman, N. (2014).

Gemini [Imágen]. Recuperado de <http://neri.media.mit.edu/projects/details/gemini>. Fotografía: Michel Figuet. *Copyright* [2018] © Neri Oxman



Figura 19. Mínima | Maxima World Expo Pavilion 2017 de TheVeryMany (Marc Fornes), proyecto construido (Astana, Kazakhstan, 2017). Fuente: Fornes, M (2017). *Mínima | Maxima World Expo Pavilion 2017* [Imagen]. Recuperado de <https://theverymany.com/world-expo-17-astana-kz-1>. Fotografía: NAARO. *Copyright* [2017] © NAARO.

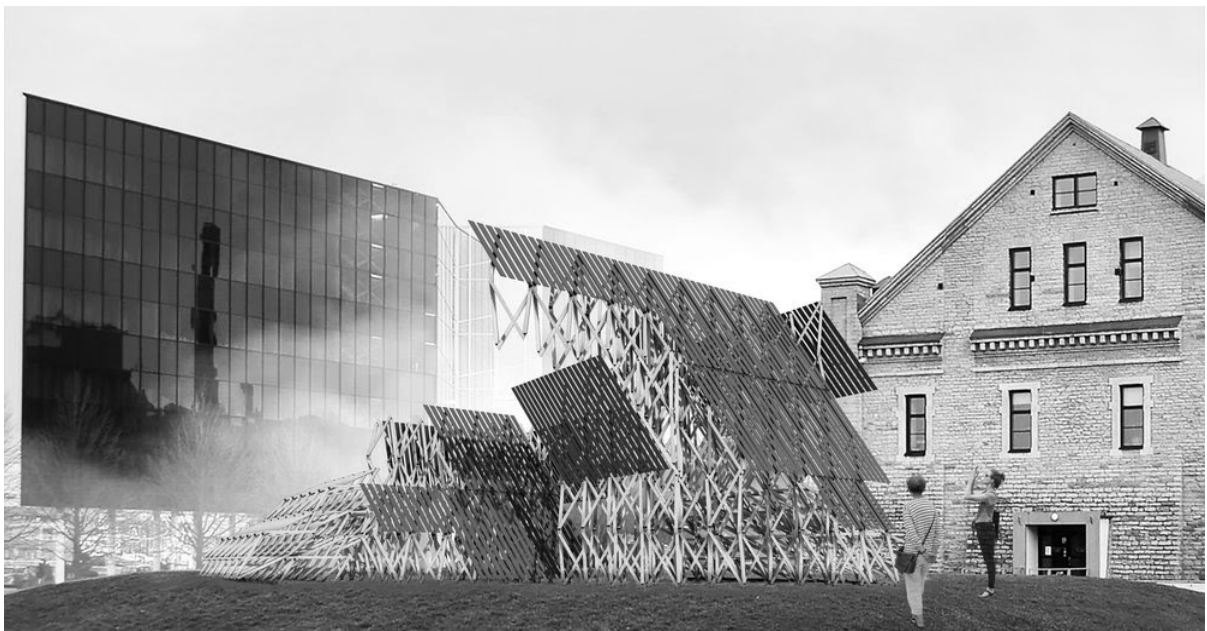


Figura 20. Combinational Nest, Tallin Architecture Biennale Pavilion de José Sánchez, proyecto (Tallin, 2019). Fuente: Sánchez, J. (2019a). *Combinational Nest* [Render]. Recuperado de <https://www.plethora-project.com/combinatorial-nest>.



Figura 21. Nuremberg Concert Hall de Gilles Retsin, Proyecto (Nuremberg, 2018). Fuente: Retsin, G. (2018). Nuremberg Concert Hall [Render]. Recuperado de <https://www.retsin.org/Nuremberg-Concert-Hall>.

2.5. Contexto disciplinar: arquitectura digital y la academia

Las manifestaciones arquitectónicas descritas anteriormente y relacionadas con uso de la tecnología computacional, definen un diagrama en el que se estructura un contexto disciplinar, el cual, se caracteriza por tres maneras de apreciar las ventajas de utilizar los medios digitales en los procesos metodológicos de diseño: (1) el morfológico, (2) el tecnológico, (3) y el político.

El primero, se dirige a explotar las capacidades computacionales con el fin de generar posibilidades formales para un proyecto; el principal objetivo se inclina hacia el *resultado*. El segundo, tiene mayor interés en explorar los avances en las técnicas y herramientas para así evaluar las metodologías de diseño arquitectónico, esta visión se encuentra más apegada al *proceso*. El tercer enfoque, añade a los objetivos del segundo un *carácter crítico* y una

preocupación por que estos sean más consistentes con la realidad ambiental, económica y política de la segunda década del siglo XXI.

Al seguir el enfoque morfológico, es posible verificar que a partir de ella se derivan dos posturas que se desarrollaron en consonancia al progreso de la tecnología digital, a la vez que estructuraron las bases teóricas para consolidar metodologías donde las lógicas causales desarrolladas por la ciencia moderna tienen mayor fuerza en el proceso de diseño. Dichas posturas son (1.1) *el esculturalismo*, (1.2) y *el parametricismo*.

Por otro lado, de la visión tecnológica se encaminan dos enfoques: (2.1) la algorítmica, (2.2) *la performativa*; las cuáles se desenvuelven con base a metodologías principalmente algorítmicas generativas y estrategias heurísticas computacionales para ser aplicadas en el diseño, así como robótica para la fabricación de objetos.

Por último, el enfoque político ha sido manifestado por (3.1) *el discretismo*, el cual comparte las estrategias metodológicas apegadas a las lógicas heurísticas y de automatización (que implica IA y robótica) del enfoque performativo.

El objetivo de esta visión es diluir las brechas existentes entre los aspectos virtuales y materiales en los procesos de (a) generación y diseño, (b) producción y (c) distribución a través de sistemas avanzados de auto-ensamble. Todo ello bajo una agenda crítica arquitectónica de adaptación a las crisis económicas y ambientales de inicios del siglo XXI, a través del uso de materiales con propiedades de economía de escala.

Paralelamente, dicha estructura genera una red de intercambio cultural internacional importante, ya que los arquitectos exponentes de estas manifestaciones se desenvuelven como académicos en las escuelas e institutos con mayor influencia a nivel internacional. Esto permite consolidar, a través de la educación, un estado del arte cada vez más nutrido, que permite racionalizar, explorar y transformar nuevas metodologías con mayor coherencia a las aportaciones tecnológicas en la arquitectura (ver Figura 22).

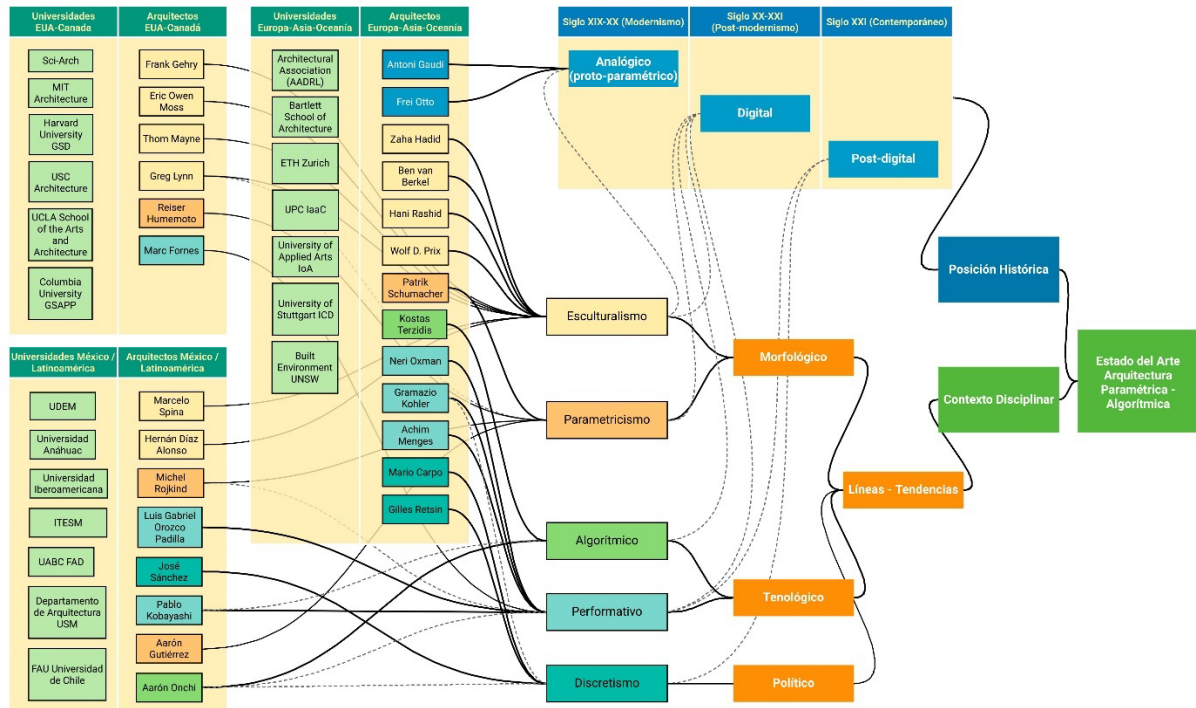


Figura 22. Estructura del estado del arte de la Arquitectura Algorítmica Paramétrica. Elaboración propia (2019).

De estas instituciones destacan en Europa, la *Architectural Association* de Inglaterra y su Laboratorio de Investigación en el Diseño (*Design Research Laboratory, AADRL*), del mismo país la *Bartlett School of Architecture*, de Suiza la Escuela Politécnica Federal de Zurich (ETH), de Alemania el Instituto de Diseño y Construcción Computacional (ICD) de la Universidad de Stuttgart, de Austria la Universidad de Artes Aplicadas de Viena y su Instituto de Arquitectura (IoA), así como la Universidad Politécnica de Cataluña y su Instituto de Arquitectura Avanzada (IaaC) de España.

En Norteamérica sobresalen la *Southern California Institute of Architecture* (*Sci_Arch*), localizada en la ciudad de Los Ángeles, de la cual también son importantes la Universidad del Sur de California (USC) y la Universidad de California, Los Ángeles (UCLA). También en el norte de Estados Unidos se encuentran instituciones de prestigio internacional como la Universidad de Harvard y su *Graduate School of Design* (GSD) o la

Universidad de Columbia y la *Graduate School of Architecture, Planning y Preservation* (GSAPP), o el Instituto de Tecnología de Massachusetts, Arquitectura (MIT Architecture).

En México, aunque las investigaciones relacionadas con la temática son acotadas, destacan por tener programas donde se instruye la arquitectura algorítmica-paramétrica, principalmente en posgrado y educación continua: la Universidad de Monterrey (UDEM), el Instituto Tecnológico de Monterrey (ITESM), la Universidad Iberoamericana de México y Tijuana, así como la Universidad Anáhuac. Todas estas instituciones comparten la característica de ser privadas, por tanto, la oferta es limitada en las escuelas que son públicas.

El posicionamiento geográfico de estas tendencias, marcan de manera contundente una hegemonía de los países correspondientes al hemisferio norte del planeta, en transmitir y aplicar este conocimiento. La causa puede estar relacionada con la accesibilidad que tienen, por su economía, a los recursos tecnológicos de vanguardia (ver Figura 23).

La estructura del contexto disciplinar descrita en este trabajo, así como los diagramas mostrados, no buscan ser determinista con los sucesos estructurados con relación al diseño arquitectónico computacional, sino que trata de transmitir aproximaciones que sirvan a orientar la comprensión de un contexto disciplinar contemporáneo, que hasta el momento sigue en transformación y consolidación.

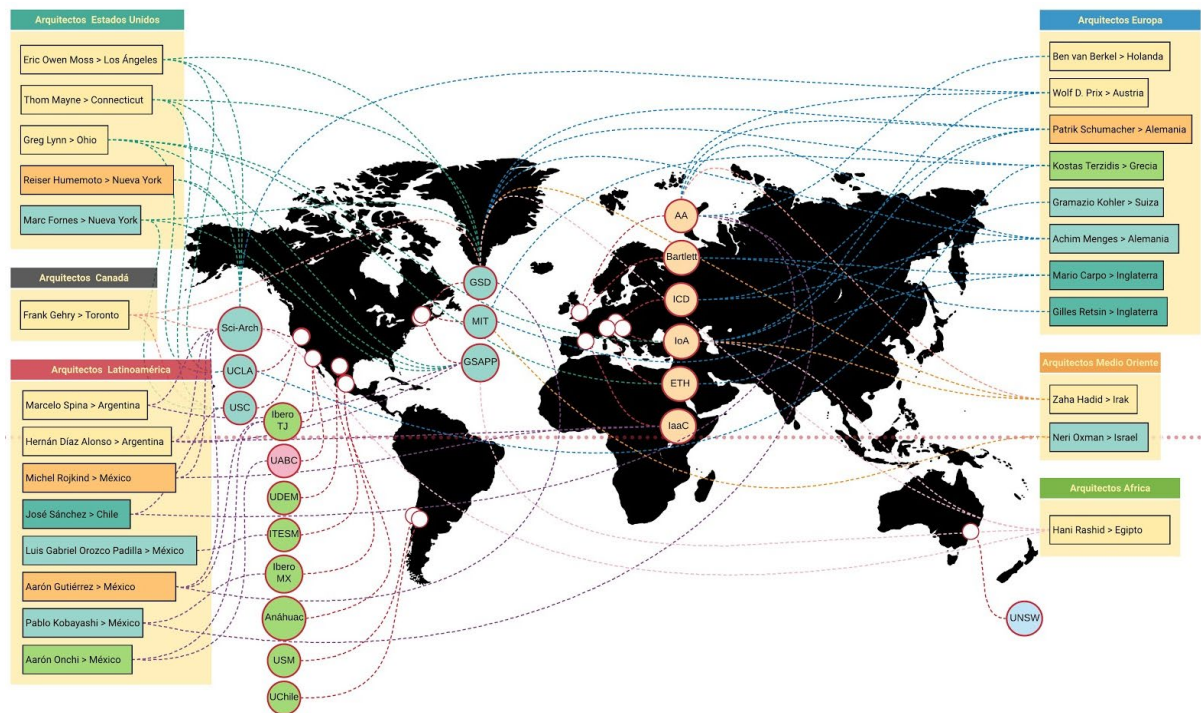


Figura 23. Posicionamiento geográfico de la Arquitectura Algorítmica Paramétrica. Elaboración propia (2019).

2.6. Teorías y metodologías de la Arquitectura Algorítmica-Paramétrica

El objetivo particular número dos de este trabajo de investigación, tiene como fin definir una metodología de diseño arquitectónico que sirva al objetivo general de desarrollar una herramienta teórico-práctica de definiciones algorítmicas para la solución de problemas arquitectónicos.

La estructura del estado del arte previo muestra una red de manifestaciones arquitectónicas computacionales con diferente enfoque metodológico; destacan así los contrastantes objetivos de arquitectos con una visión morfológica, los que tienen una perspectiva tecnológica y los que añaden un carácter político. En este orden de ideas, sobresalen de estas posturas tres metodologías que tienen un desarrollo amplio y un planteamiento teórico consolidado.

2.6.1. El Parametricismo de Patrik Schumacher.

Se fundamenta en un marco teórico establecido en dos libros llamados *Autopoiesis of Architecture Vol.I: a new framework for architecture* (2011) y *Autopoiesis of Architecture Vol.II: a new agenda for architecture* (2012). En ellos Schumacher, parte de una arquitectura entendida como un sistema complejo Deleuziano (2002, 2017), con influencia en teorías que clasifica en tres tipos: (1) problematizantes, (2) generativas, (3) analíticas-predictivas.

De las teorías problematizantes destaca la desarrollada por Christopher Alexander en su libro: *Notes on the Synthesis of Form* (2002), de las generativas manifiesta especial apoyo en Greg Lynn (1998) y su teoría del *BLOB*, así como considera la analítica-predictiva más importante de Bill Hillier llamada teoría del *Space Syntax*.

La teoría de la *Autopoiesis of Architecture* de Schumacher (2011), busca reunir las características de las tres teorías mencionadas anteriormente, en el objetivo de generar lo que llama una “Super Teoría” con propiedades de flexibilidad en transformación y robustez en contenido. Así, esta teoría acuña su nombre debido a la propiedad de autoproducción que pueden ser encontrados en los sistemas biológicos naturales, y que tiene como objetivo reinterpretar en un sistema arquitectónico.

Es por ello, que para Schumacher (2009, 2011, 2012) la Arquitectura debe ser entendida como un sistema de (a) operaciones, (b) procesos y (c) estructuras de comunicación, que consoliden una agenda para calificar condiciones de organización y articulación en el proceso de diseño, para de esta forma ejercer relaciones físicas (en relación a la organización) y relaciones cognitivas (relacionadas con la articulación). Las primeras relaciones analizan los patrones y las cognitivas distinguen condiciones fenomenológicas (de percepción) y semiológicas (de significación).

Las decisiones de diseño, producto de (a) operaciones comunicativas, evalúan las características cualitativas de un proyecto a partir de su código de utilidad (con relación a la

función), su código de belleza (con relación a la forma) y su código de novedad (con relación a la originalidad). El objetivo es reunir la mayoría de estas características en el desarrollo proyectual.

Esta evaluación conduce a un sistema de funciones, que relacionan la Arquitectura con la sociedad a partir de la investigación académica, como medio de experimentación a manera de laboratorio, con el fin de cumplir con dos tareas: (1) buscar en la sociedad problemas arquitectónicos y definir informes, (2) proliferar nuevos repertorios formales en conjunto con la exploración de nuevas técnicas de modelado.

Bajo este planteamiento, Schumacher funda el programa *Design Research Laboratory* (AADRL) en la *Architectural Association* (AA) en donde se experimenta la aplicación del método Parametricista con objetivos de complejidad dinámica y estrategias de simultaneidad, afiliación múltiple, fluidez, *space of becoming*, matriz de organización, red ampliamente acoplada y difuminado.

Bajo el método de llevar la forma a un programa heurístico, para así analizar las capacidades de asequibilidad de la función, y así determinar los lineamientos de esta, la AADRL presenta una agenda con los siguientes pasos de desarrollo arquitectónico, con un carácter experimental:

(1) Precedente, se hace un análisis de proyectos construidos o no construidos, los cuales son base para la identificación de lógicas. (2) Exploración, se procede experimentos físicos a través de sistemas de auto-organización morfogénica. (3) Sinopsis, se realizan informes de los resultados de las exploraciones morfogénicas. (4) Tesis, se desarrolla la formulación de hipótesis de investigación. (5) Extremismo, se formulan distintivas (al extremar) de la tesis de investigación de diseño. (6) Totalización, realización “final” del proyecto de diseño.

La teoría de la *Autopoiesis of Architecture* se clasifica el diagrama en dos tipos: (1) el ordinario (considerado predecible), y (2) el extraordinario (entendido como complejo Deleuziano); ambos, pueden ser del tipo métrico o paramétrico, éste último desarrollado bajo la programación digital, alfanumérica o GUI (*Graphic User Interface*) (ver Figura 24).

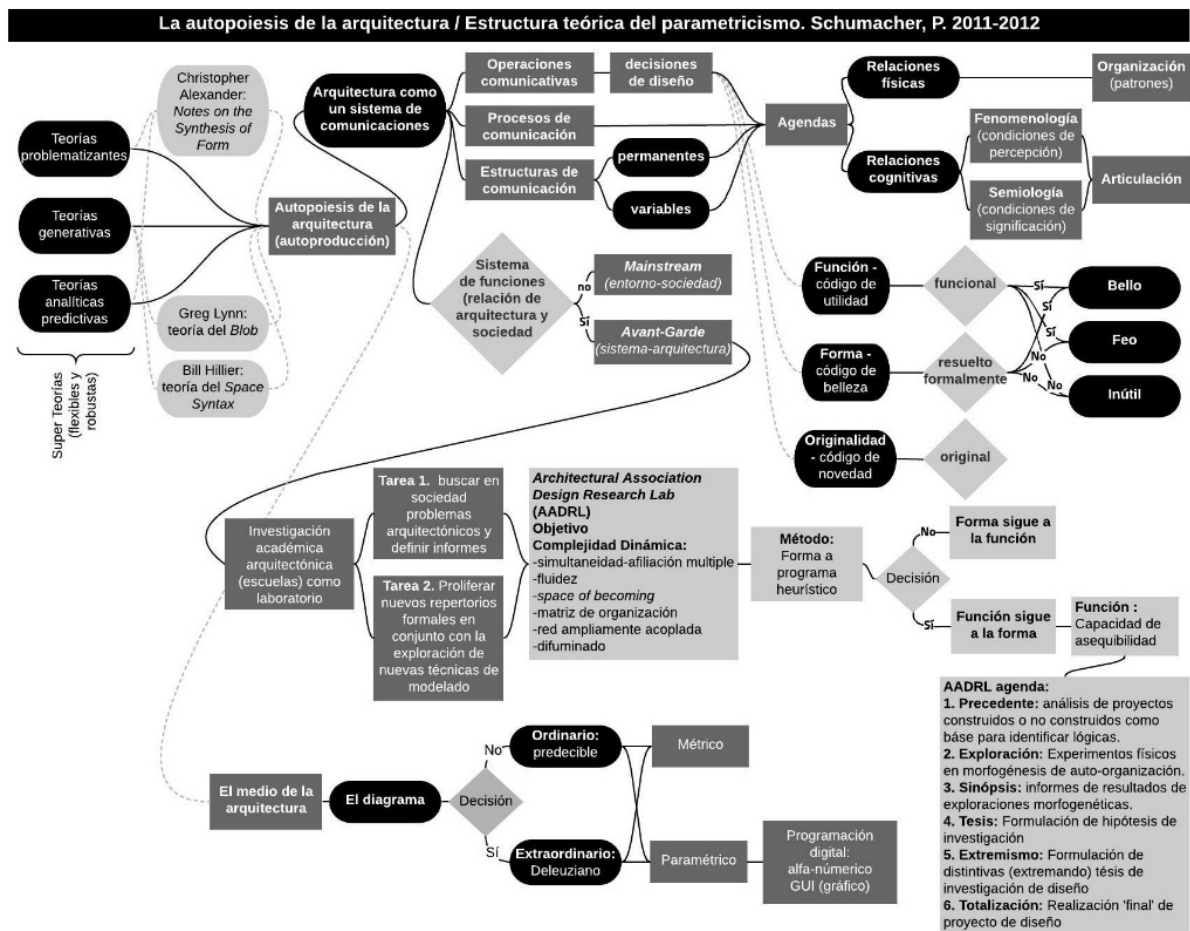


Figura 24. Estructura del marco teórico de la *Autopoiesis of Architecture* de Schumacher, P. (2011, 2012). Elaboración propia (2019).

Esta estructura teórica, establece los fundamentos para el desarrollo metodológico del Parametricismo, el cual considera como “nuevos primitivos” de ejecución todas aquellas estrategias que sean: sistemas dinámicos (como: *hair, cloth, blobs, metaballs*, entre otros), adaptativos, interactivos, con el uso de geometrías *splines, NURBS* y subdivisiones, reactivos a “atractores” y resonancia vía *Scripts*.

Como anteriormente se muestra, en el esquema teórico de la Autopoiesis de la Arquitectura, el parametricismo lleva la forma a un programa heurístico de exploraciones morfogénicas a través de sistemas de auto-organización. Estas exploraciones son guiadas por un sistema de principios funcionales y formales, en el cual se encuentran identificados unos que son negativos, entendidos como tabúes y otros que son positivos, reconocidos como dogmas (ver figura 25).

Los principios positivos funcionales son: (a) todas las funciones son actividades - escenarios eventuales paramétricos, (b) todos los espacios, actividades y eventos se comunican unos con otros. Por tanto, los negativos son: (a) evitar estereotipos funcionales rígidos, (b) evitar zonificaciones funcionales segregadas.

Los principios positivos formales son: (a) todas las formas deben ser suavizadas (deformación inteligente = información), (b) todos los sistemas deben ser diferenciados (graduales, umbrales, con singularidades), (c) todos los sistemas deben ser independientes (correlacionales). Así, los principios negativos son: (a) evitar formas rígidas (falta de maleabilidad), (b) evitar repetición simple (falta de variedad), (c) evitar collage de elementos aislados desvinculados (falta de orden).

También, es posible encontrar que las principales agendas del Parametricismo son: (a) la inter-articulación de subsistemas paramétricos, (b) la acentuación paramétrica, (c) la figuración paramétrica, (d) la semiología paramétrica, (e) la responsividad paramétrica, (f) la ecología paramétrica y (g) el urbanismo paramétrico.

Aunque, sea utilizado un método heurístico para la búsqueda de la forma, esta se encuentra principalmente enfocada en el desarrollo morfogénico a través de sistemas de auto-organización guiada por una visión biomórfica del producto arquitectónico. Es decir, los principios formales y funcionales que condicionan la búsqueda morfogénica dirigen el resultado a una lógica principalmente causal.

Las propiedades formales que son directrices para el reconocimiento funcional del espacio derivan de la inteligencia humana a través de operaciones sintéticas o matemáticas, basadas en el cálculo diferencial, que dan sentido a las formas biológicas o naturales y son ejecutadas o simuladas por la computadora, sin ser producto de sus procesos de inteligencia (repetición infinita de operaciones simples y limitadas, discontinuas o granulares).

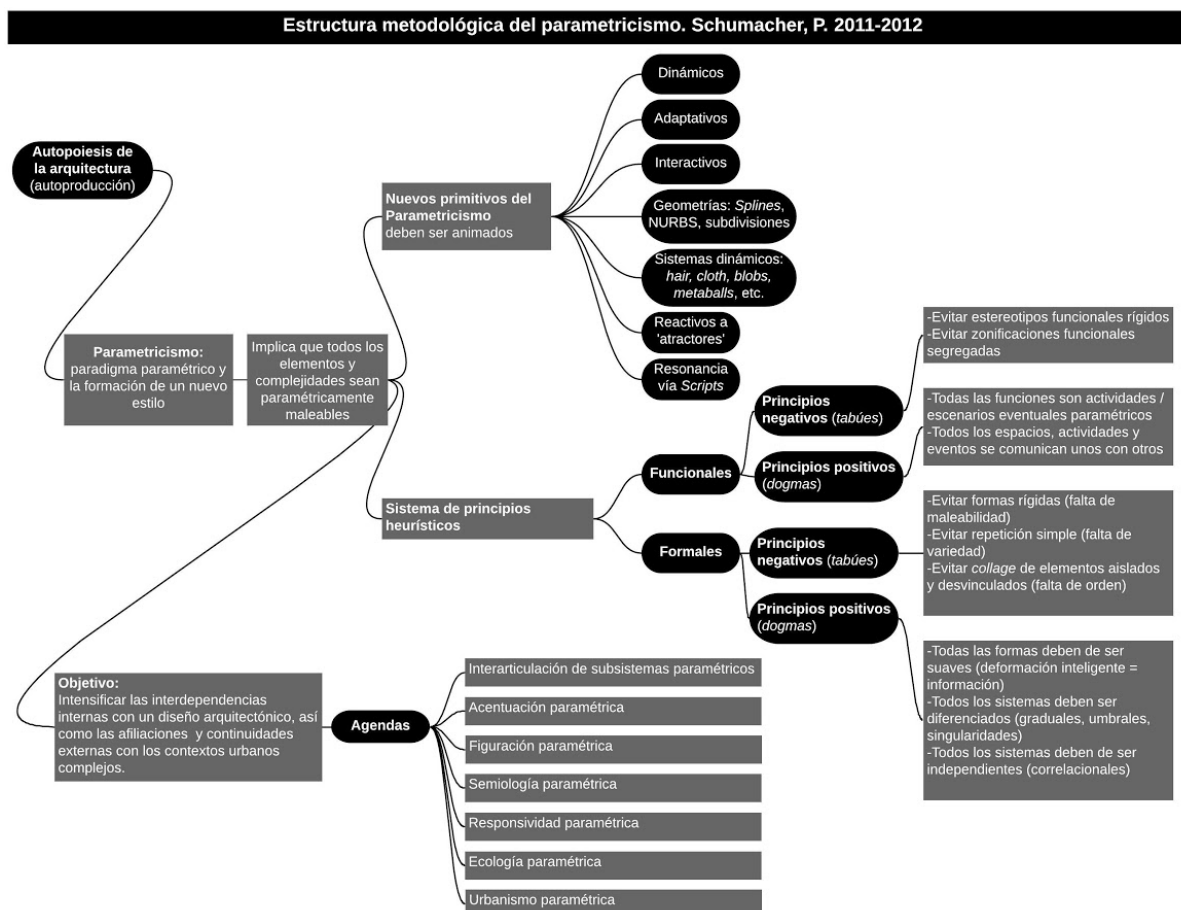


Figura 25. Estructura metodológica del Parametricismo de Schumacher, P. (2009, 2012). Elaboración propia (2019).

2.6.2. La Arquitectura Algorítmica de Kostas Terzidis.

Terzidis (2006) establece en el libro *Algorithmic Architecture* el desarrollo teórico para una metodología arquitectónica con el uso de definiciones algorítmicas. A diferencia de

Schumacher (2009, 2011, 2012), este método es considerado como un proceso emergente, sin fines paradigmáticos o estilísticos.

Considera que la arquitectura algorítmica se dirige hacia lo desconocido (analiza los sistemas difusos) y reconoce la inteligencia computacional como una ajena a la humana. Presenta el desarrollo algorítmico como un medio de comunicación amplia con la computadora para aprovechar las capacidades de su inteligencia en vías de extender las posibilidades de dar soluciones a problemas de diseño arquitectónico.

Tiene como principios: (a) visualizar la computadora no solo como herramienta, sino como un compañero de trabajo, con el fin de extender las capacidades humanas en torno a la solución de problemas. (b) Entender a la computadora como una entidad de inteligencia artificial ajena al pensamiento humano. (c) Apoyado en la filosofía pre-Socrática, cuestiona la existencia de la novedad. (e) Considera que todo conocimiento, proceso o producto generado tiene precedente. (f) Por tanto, la arquitectura algorítmica no busca la novedad. (g) La arquitectura algorítmica no se evalúa por su apariencia. (h) La arquitectura algorítmica se define por el proceso, el cual es emergente.

Los elementos básicos para el desarrollo de algoritmos y *Scripts* son: (1) variables: *int, float, string, vector, etc.*, (2) operaciones aritméticas y lógicas, (3) repetición, (4) *arrays* (colecciones), (5) objetos geométricos, (6) transformaciones, (7) atributos, (8) transformación secuencial, (9) multi-Booleanos, (10) búsquedas estocásticas, (11) fractales, (12) autómatas celulares, (13) hibridación.

La metodología aborda dos estrategias algorítmicas para procesos arquitectónicos: *Amphiboly* y *Periplocus*. La primera explora las nociones de ambigüedad, ambivalencia y equivocación utilizados en los esquemas computacionales (prueba y error). También investiga las posibles maneras en las que ciertas características humanas pueden ser expresadas a través de síntesis algorítmicas de formas.

La segunda analiza la complejidad artificial, que considera no se encuentra basada en significados cuantitativos, y tiene el fin de mostrar cómo estructuras pueden emerger de simples algoritmos, en donde la belleza recae en la articulación de pocos elementos intrincados.

Las aplicaciones que pertenecen a la estrategia de *Amphiboly* son: (a) *(Syn) biosis* basada en la analogía algorítmica de una interacción biológica entre un huésped y un parásito (de beneficio y perjuicio) para ejercer un tipo de simbiosis donde dos (o más) organismos de diferentes especies que viven en cercana proximidad a otro.

(b) *En (dia) meso*² hace relación a la dualidad de la identidad de una forma transformada bajo la función de *morphing*. La cual se caracteriza por una forma híbrida en la que existe remanente de los arquetipos que la componen y transformaron.

(c) *(Syn) diasis*³ es la aplicación algorítmica de la arquitectura booleana, la cual es un proceso acumulativo que resulta en ensamblajes complejos al combinar elementos progresivamente en una mayor complejidad. La combinación de elementos sucede a través de las operaciones de configuración (booleanas) *union (OR)*, *intersection (AND)* y *complementation (NOT)*.

Las aplicaciones de la estrategia *Periplocus* son: (a) Recursión, para arquitectura es un proceso ontológico que involucra la formación existencial de estructuras infinitamente anidadas a través de la auto-replicación. (b) Rotaciones sutiles, en arquitectura la iteración es empleada como un recurso de ordenamiento que produce patrones repetitivos. (c) Búsquedas estocásticas, es un proceso en el cual elementos son dispuestos en ubicaciones aleatorias en el espacio, después son evaluados contra un conjunto de restricciones para ser aceptados si cumple satisfactoriamente con ellas. (d) Distribución programática, una búsqueda estocástica es utilizada como el algoritmo central para distribuir espacios programáticos dentro de un sitio dado.

Estas estrategias heurísticas impulsaron posteriormente los métodos generativos, donde la inteligencia artificial, es utilizada para desarrollar posibilidades arquitectónicas a través de operaciones evolutivas (de prueba y error), en periodos cortos de tiempo (ver figura 26).

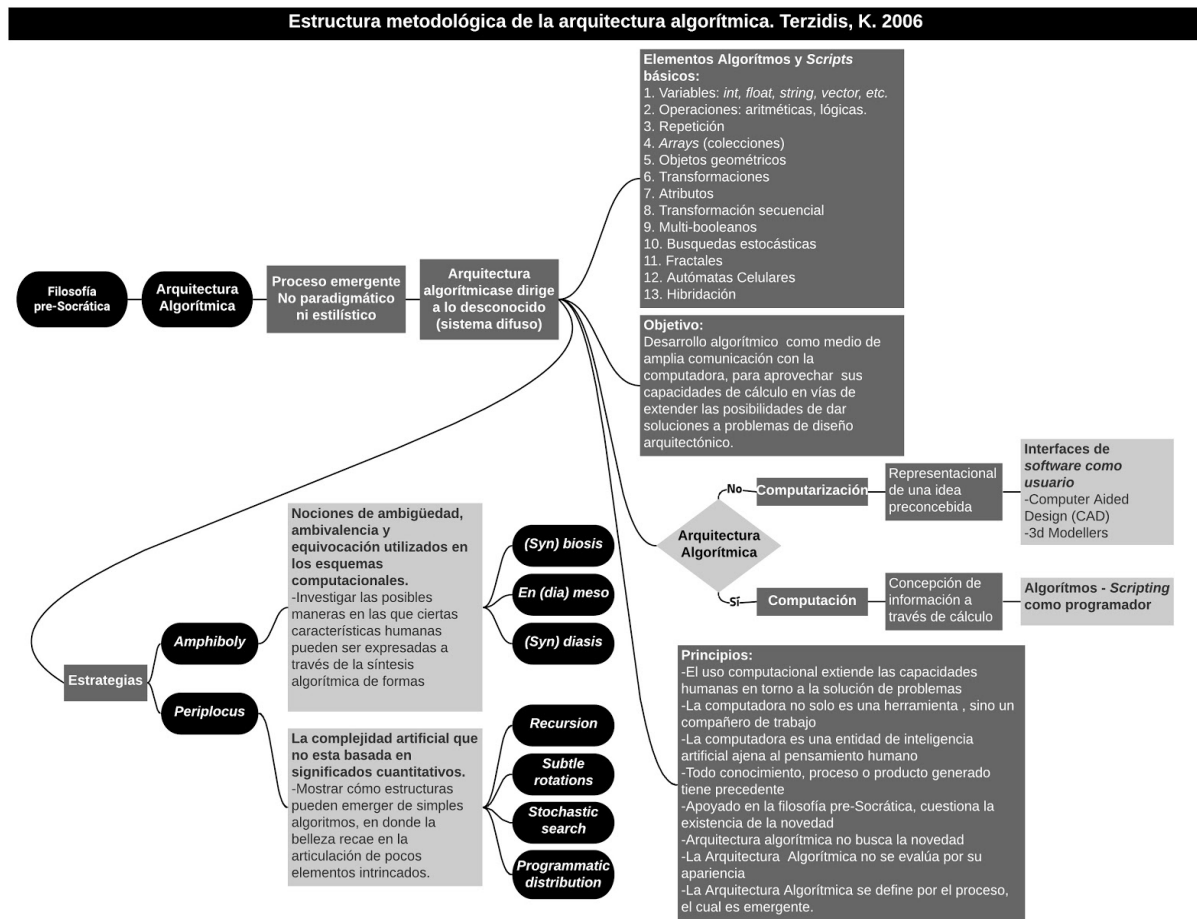


Figura 26. Estructura metodológica de la Arquitectura Algorítmica de Terzidis, K. (2006).

Elaboración propia (2019).

2.6.3. Innovación y fabricación digital en Arquitectura.

Las manifestaciones arquitectónicas más destacadas dentro de la línea tecnológica, en su enfoque performativo, se encuentran representadas por arquitectos como Achim Menges (2008; 2008, 2015; 2020), Gramazio Kohler, Neri Oxman (2015), Marc Fornes, Skylar Tibbits, entre otros.

El proceso metodológico inicia con investigación de tipo cualitativo, cuantitativo y experimental, seguido por procesos morfogenéticos (*form-finding*) mediante sistemas biomiméticos y auto-organización, análisis de optimización y desempeño a través de métodos heurísticos computacionales que incluyen: estrategias algorítmicas, algoritmos genéticos o análisis de elementos finitos, para posteriormente pasar por el proceso de producción con robótica avanzada.

La experiencia de estos exponentes, no solo se encuentra involucrada con una metodología de diseño, sino que representan vanguardia en procesos de construcción y generación de nuevos materiales. La innovación en los procedimientos de fabricación son un aspecto importante, a fin de promover métodos de creación para el futuro.

El aspecto correspondiente a la investigación sigue una estructura (ver figura 27) en la que, según Aksamija (2016, p. 176): (1) se identifica el problema a investigar, (2) se conduce a una revisión literaria, (3) se identifican brechas de conocimiento, (4) se identifican métodos de investigación: cualitativos, cuantitativos o experimentales, (4.1) se ejecuta el estudio, (4.2) se hace recauda información y analiza resultados (4.3) se hacen conclusiones e implementaciones.

Los métodos de investigación correspondientes a la estructura anterior tienen un enfoque de innovación de servicio, en el que se analizan aspectos de tipo cualitativo que incluyen la observación y realización encuestas. Posteriormente se procede con la innovación en los productos y procesos con métodos cuantitativos que incluyen simulaciones y modelado, para después proceder con el sistema experimental que incluye prototipado y evaluación a través de *testing*.

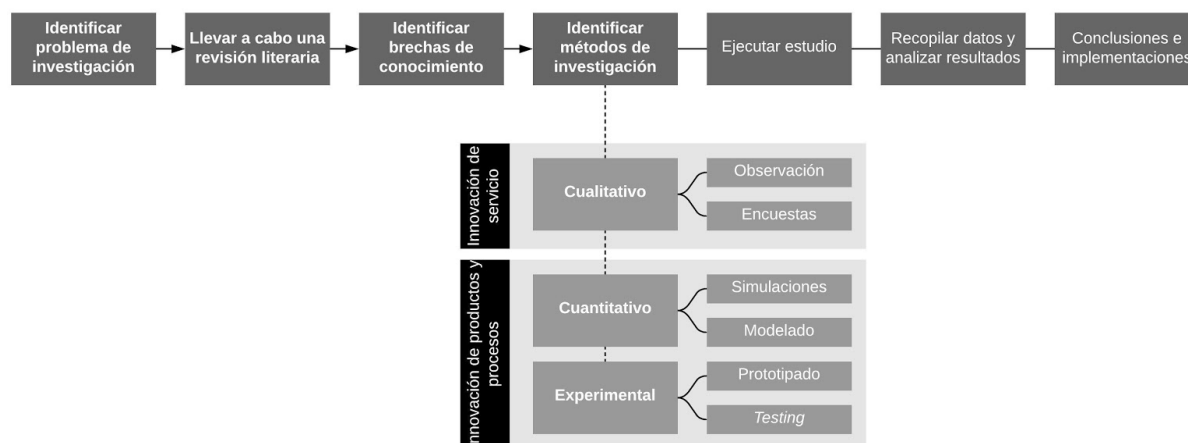


Figura 27. Aksamija, A. (2016, p. 176). Proceso y método de investigación performativo [diagrama].

Modificación: elaboración propia (2020).

Como complemento a la estructura descrita anteriormente (ver figura 28), en el segmento cualitativo de innovación de servicio, actualmente la observación puede ser aumentada a través del uso de sensores, los cuales tienen la capacidad de ser receptores de información a partir de los fenómenos del mundo físico, y ser enviada de forma inmediata al mundo digital.

Los sensores (activos y pasivos) tienen la capacidad de registrar imágenes, temperatura, estrés físico, movimiento, presión atmosférica, radiación infrarroja, entre otras cosas (Wujec, 2017). Estas propiedades aumentan la percepción humana y contribuyen a que la información cualitativa, sea producto de un mayor rango de datos cuantitativos adquiridos, para que la apreciación humana sea más certera en el momento de tomar decisiones.

Los sensores digitales están aumentando y expandiendo la percepción humana, ampliando nuestra habilidad para percibir las cosas... A medida que los sensores mejoren en contarnos sobre los cambios del mundo. Con cada mejora en la resolución, perciben cualidades en un objeto que antes eran invisibles. (Wujec, 2017, pp. 37, 39).

En el segmento de innovación de productos y procesos de los métodos de investigación, las simulaciones son realizadas principalmente a través de métodos cuantitativos de tipo *top-down* o heurísticos (de caja negra).

Destaca el uso de estrategias algorítmicas como sistemas-L, autómatas celulares, fractales, búsquedas estocásticas, etc., así como procedimientos de optimización y desempeño tales como algoritmos genéticos, que tienen un amplio rango de uso que van desde optimización geométrica y topológica hasta la adaptabilidad al ambiente y energética, o los análisis y modelado de elementos finitos que tienen un uso principalmente dirigido al desempeño estructural.

La parte experimental de la metodología, por lo contrario, sigue una estrategia *bottom-up* y se encuentra conformada por innovación en los sistemas de prototipado, los cuales se desenvuelven principalmente a través de la robótica -desde la más elemental, hasta la más avanzada -.

Con aplicaciones de fabricación automatizada de tipo sustractivo o aditivo como: impresoras 3D, máquinas CNC (de corte, fresado, moldeo, vertido, soldar) de 1 a 5 ejes, brazos robóticos (de 6 ejes), equipados con sensores o dispositivos personalizados. También esta fase involucra el desarrollo de nuevos materiales, tanto estáticos o dinámicos con propiedades de movimiento o auto-ensamble. En palabras de Wujec (2017):

Nosotros siempre hemos usado las herramientas físicas para magnificar nuestras habilidades para dar forma a los objetos. Ahora las herramientas robóticas aceptan instrucciones digitales y se han vuelto expertos para los actos físicos de fabricación, sin cansarse, aburrirse, y usualmente sin cometer errores... equipados con sensores y un cerebro de aprendizaje automático, los robots finalmente se volverán brillantes, receptivos, fuertes, rápidos e inteligentes. (2017, pp. 146, 148).

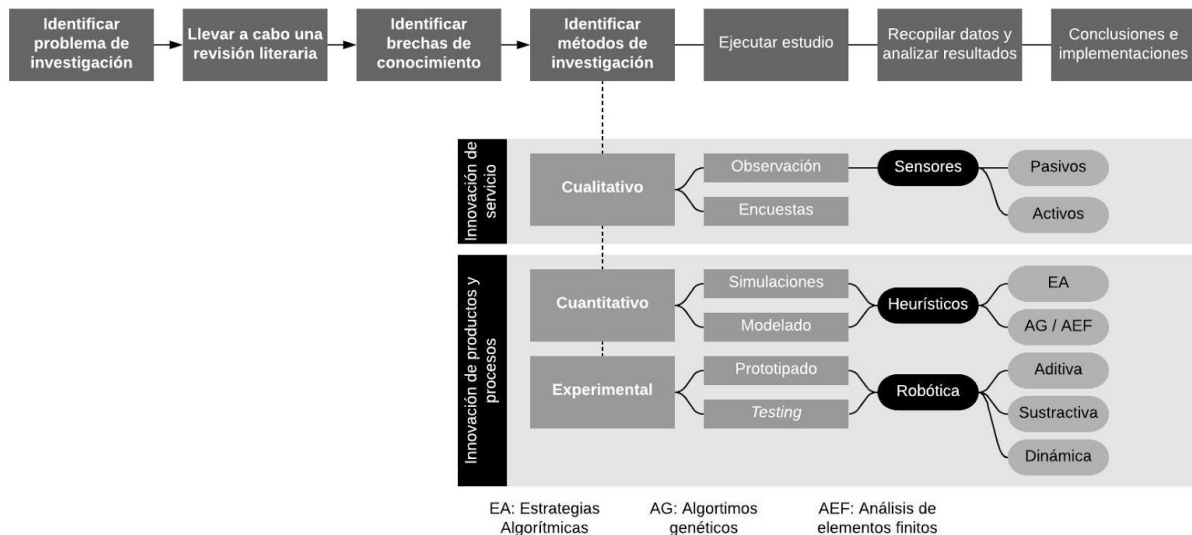


Figura 28. Complemento a proceso y método de investigación performativo de Aksamija (2016, p. 176) [diagrama]. Elaboración propia (2020).

El proceso de investigación se integra a las prácticas de diseño como una rama de la estructura organizacional de una firma de diseño arquitectónico, para consolidar una relación entre los valores de la firma y las prácticas de investigación a fin de innovación arquitectónica (ver figura 29).

Según Aksamija (2016) existe una gran cantidad de maneras de operación entre la investigación y las prácticas de diseño en una firma. Reconoce tres modelos en los que habitualmente se diferencian estas relaciones: (1) interno, (2) externo e (3) híbrido, a consecuencia de diversos factores como el tamaño de la firma u organización, su dedicación a la investigación y desarrollo, al financiamiento, etc. El carácter de estos tres modelos es descritos por Aksamija (2016) a continuación:

Las prácticas de investigación interna son financiadas internamente por la firma de diseño o incluidas como parte de los servicios de esta, donde los investigadores son empleados de la firma ... Las prácticas de investigación externa constituyen participación con socios externos, donde las empresas de diseño pueden financiar las actividades de varios centros de investigación o universidades ... El modelo híbrido

constituye una mezcla de los dos tipos anteriores. Por ejemplo, alguna firma de diseño se encuentra estableciendo organizaciones separadas sin fines de lucro, dedicadas a la investigación, pero están fuertemente influenciadas por los valores, estrategias y proyectos de la empresa. (2016, p. 175).

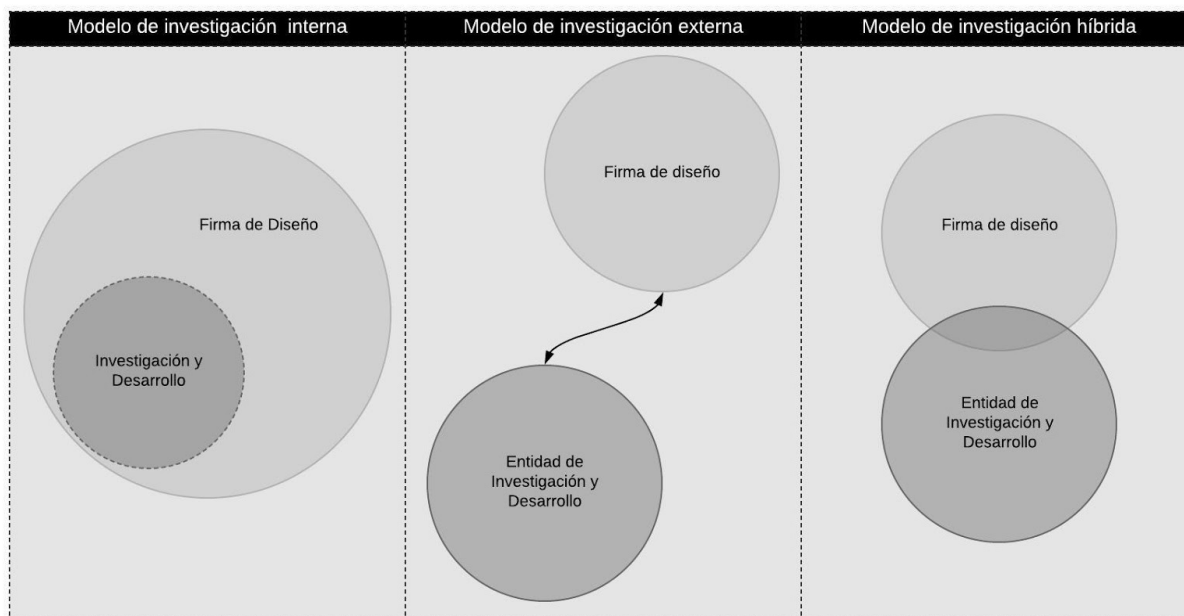


Figura 29. Aksamija, A. (2016, p. 175). Modelos de prácticas de investigación performativa -relación con las firmas de diseño arquitectónico- [diagrama]. Modificación: elaboración propia (2020).

Un ejemplo de relación entre una entidad de investigación y una firma de diseño, de acuerdo con los modelos anteriores, es el proyecto de investigación realizado en el año 2012 por la firma de Arquitectura Perkins+Will y la Universidad de Cincinnati (ver Figura 30). El proyecto tenía como objetivo investigar métodos de diseño paramétrico, análisis de optimización y desempeño, simulación y fabricación, con el fin de desarrollar componentes de fachada, para el reúso adaptable de un edificio construido durante la década 10 del siglo XX en Chicago (Aksamija, 2016, p. 83).

El proceso de diseño sigue una metodología bajo el modelo de investigación externa en la que primeramente (a) se recolecta información; posteriormente se desarrollan modelos de diseño digitales para generar (b) el diseño conceptual con programas como: *Rhinoceros*®,

Revit®, *3d Max*® o *Maya*®, (c) se hacen simulaciones: *Ecotect*, equivalentes o aplicaciones personalizadas, (d) se procede con la etapa de “búsqueda de la forma” (*form finding*) paramétricamente con: *Grasshopper*, *Maya Scripts*, *Dynamo* o aplicaciones (*plug-in*) personalizadas de *Revit*, después comienza el proceso de (e) fabricación (a través de modelos digitales): impresión 3d, corte láser, fresado CNC, y así generar prototipos de (f) objetos físicos: ensamblado.

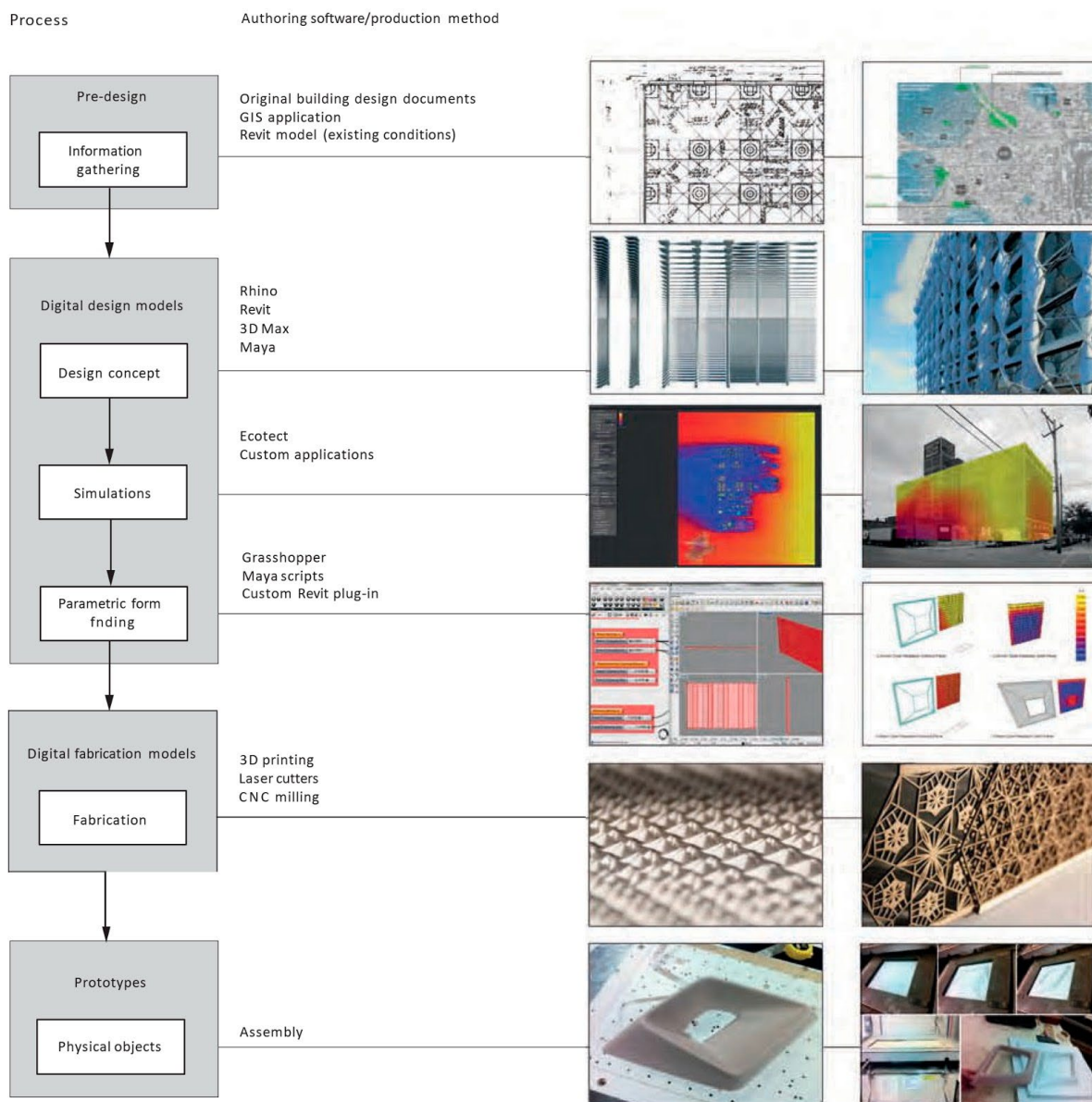


Figura 30. Aksamija, A. (2016, p. 81). Proceso de proyecto de Investigación, de Perkins+Will y la Universidad de Cincinnati, 2012 [diagrama].

Existen por tanto cuatro pasos generales por los que pasa el proceso de fabricación digital en la arquitectura (ver Figura 31), según Wujec (2017) en la actualidad el hacer cosas implica (1) capturar: proceso principalmente cualitativo para la obtención de información, (2) computar: proceso de diseño y métodos cuantitativos de simulación y modelado en 3D con herramientas digitales como recurso, (3) crear: paso en el que se lleva el diseño a la producción o fabricación mediante sistemas de robótica y automatización, (4) componer, la producción ágil de objetos permite desarrollar nuevos materiales con propiedades de movimiento o auto-organización.

Los primeros dos pasos (enlistados anteriormente) se ubican en el mundo digital o virtual, mientras que los últimos son parte del mundo físico. La dinámica de interrelación entre ambos mundos permite en el presente, tener mayor dominio en la resolución de problemas complejos arquitectónicos y una mayor capacidad para generar productos cada vez más adaptables a nuestro entorno.

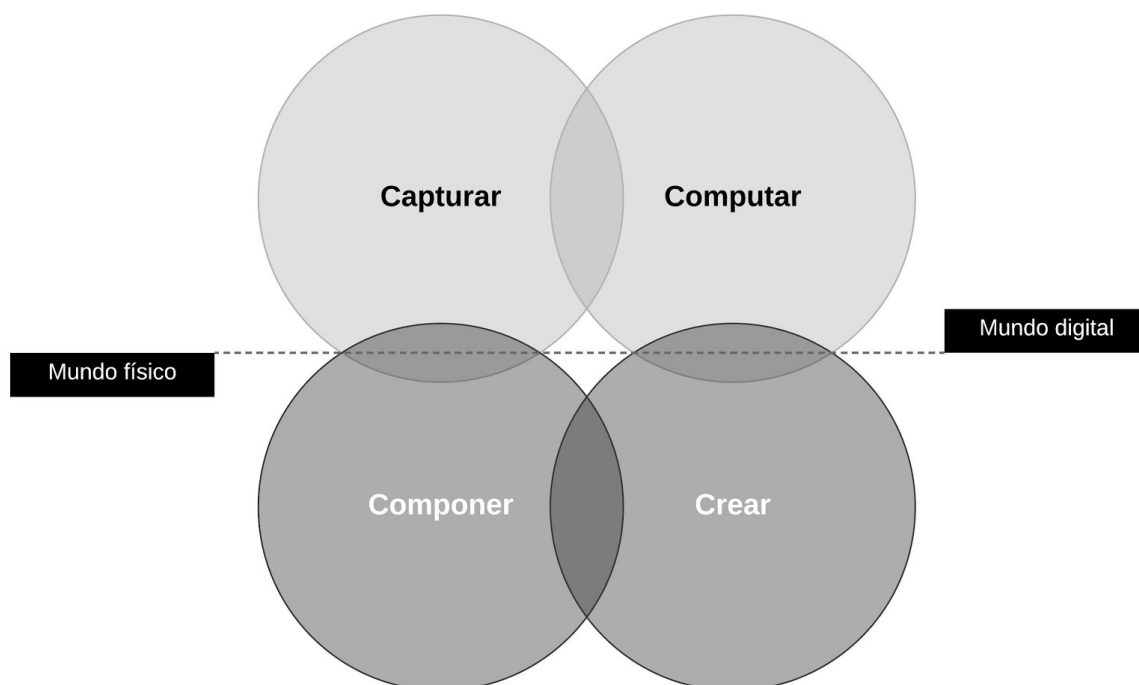


Figura 31. Wujec, T. (2017, p. 31). Pasos generales de la fabricación digital [diagrama].

Modificación: elaboración propia (2020).

De manera más extensa es posible distinguir en el libro *Digital Fabrication in Architecture* de Dunn (2012) un proceso para el desarrollo de fabricación digital de tres pasos (ver Figura 32): (a) generación, es donde se aplican las técnicas de producción de las formas a través de recursos digitales; (b) integración, se utiliza la vanguardia en máquinas de fabricación al utilizar diferentes procesos; (c) estrategias, se determinan métodos de fabricación para la producción específica de las demandas del diseño.

De estos tres pasos, derivan distintas estrategias de aplicación normalmente utilizadas en cada rubro. El esquema contempla una contextualización con el estado del arte de la Arquitectura Algorítmica Paramétrica.

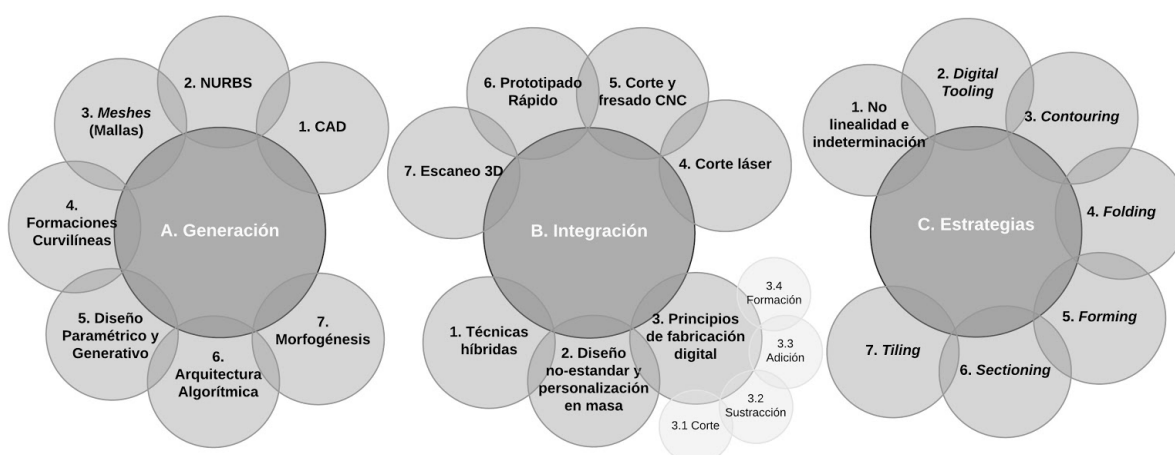


Figura 32. Los tres pasos del proceso de fabricación digital en la arquitectura, según el libro *Digital Fabrication in Architecture* (Dunn, 2012). Elaboración propia (2019).

2.7. Aspectos fenomenológicos y semiológicos como parámetros cualitativos de diseño Algorítmico Paramétrico

La Arquitectura (occidental) a lo largo de su historia, ha mantenido una constante discusión entre los aspectos cualitativos y cuantitativos que la conforman. Esa tensión esencialmente ha sido producto del enfoque en el que se apoya la sociedad para dar sentido a todas las cosas que se encuentran alrededor, y a la perspectiva desde la cual se concibe y

construye todo producto hecho por el hombre. Este escenario, tiene un carácter dialéctico y retórico epistemológico, que recae naturalmente en las concepciones filosóficas de las distintas épocas del desarrollo humano.

El punto de partida a esta discusión tiene origen en la conciencia y posturas alrededor del tema de la relación del hombre con las cosas; las cuales promovieron el establecimiento de las principales familias de la teoría filosófica clásica: el idealismo y el materialismo.

Mientras Platón situaba a los objetos como producto de la razón: “los objetos del conocimiento se reducen a esencias universales que son exclusivamente producto de la mente” (citado por Bloomer y Moore, 1983, p. 34), Aristóteles los comprendía como substancias compuestas de materia y forma que hacen la realidad: “yo, por otra parte, valoro el sentido del tacto” (citado por Bloomer y Moore, 1983, p. 34).

En este sentido, y con una aproximación mayor al espacio y la belleza como unos de los principales valores de análisis de la Arquitectura, Platón concibió al primero a través de la geometría, la cual consideraba como su ciencia (idealismo), mientras que Aristóteles posteriormente desarrolló la teoría del “lugar” (topos) y definió al espacio como la suma de todos los lugares, con propiedades cualitativas y dinámicas de dirección (materialismo) (Norberg-Schulz y Schulz, 1971).

En relación a la belleza, el debate en la antigua Grecia no fue diferente, según Bloomer y Moore (1983), “comenzó a debatirse el tema de la procedencia de las leyes que gobiernan nuestro sentido de belleza. Este debate llevaba necesariamente a intentar descubrir cuáles eran las funciones del cuerpo humano y de su aparato sensorial” (1983, p. 35).

La tensión entre estas dos posiciones filosóficas ha sido reiterativa, sin embargo, a partir de la ilustración, se manifiesta enfocada en los conceptos de la razón contra los sentidos. Las motivaciones racionalistas de esa época impulsaron el escepticismo en la

fiabilidad de los sentidos y a desconfiar de todo aquello que no fuera producto del pensamiento.

A partir de la fundación de las academias científicas, los debates entre los conceptos de “belleza” y la “función” emergen en la época Neoclásica y adquieren mayor intensidad en el Modernismo, así que se convierten en un referente de discusión de la vida arquitectónica postmoderna, y se mantienen aún latentes (o heredados) en la contemporaneidad.

No obstante, es posible encontrar en la arquitectura de Frank Lloyd Wright, Alvar Aalto o Louis Kahn, proyectos modernos que manifiestan un equilibrio latente y sinérgico entre estos dos valores aparentemente contrarios.

A partir de las últimas décadas del siglo XX, como remanente de la tensión entre los sentidos y la razón, esta dicotomía ha adquirido en la creación arquitectónica, el carácter cualitativo de valorar los aspectos fenomenológicos o semiológicos del espacio.

Es decir, la discusión se enfoca en dar mayor peso a los estímulos sensoriales generados por la experiencia del espacio arquitectónico ya creado (Materialismo), o en considerar más importante los estímulos de la mente y las relaciones del significado, que tienen su génesis en las ideas, en el concepto arquitectónico o en el diagrama, de una Arquitectura que aún no es realizada, que tiene una condición proyectada, virtual o inclusive digital-computacional (Idealismo).

El presente apartado analizará estos aspectos, a partir de referentes importantes en el tema como, Norberg-Schulz (1999; 1971), Bloomer y Moore (1983), Peter Zumthor (2011, 2016), Josep María Montaner (2015) y Peter Eisenman (2011).

Paralelamente, los debates actuales en relación a la metodología de diseño arquitectónico, también evalúan en un marco más cuantitativo, si los productos que aprovechan la vanguardia de la tecnología computacional y robótica, deban surgir a partir de lógicas causales derivadas de la síntesis matemática y racional dominante en el movimiento

moderno y la primer era digital de la Arquitectura, donde era necesario resolver problemas con poca información, o a través de métodos más intuitivos de búsqueda heurística de la inteligencia artificial y el *Big Data*, como auxiliares en la producción arquitectónica (Carpo, 2017; Retsin, 2019).

Ante esta realidad contemporánea de analizar los valores dicotómicos anteriores, también es posible identificar en la actualidad expresiones arquitectónicas importantes con un carácter equilibrado o híbrido, donde las concepciones racionales o intuitivas relacionadas con la interacción del hombre y los objetos, van de la mano en un comportamiento sinérgico y recursivo.

Es el caso de la obra de arquitectos contemporáneos como Herzog & DeMeuron, Sou Fujimoto, Toyo Ito, y en especial Kengo Kuma (2010) que el presente trabajo analizará algunas de sus manifestaciones arquitectónicas como caso de estudio.

2.7.1. Análisis descriptivo de los aspectos fenomenológicos en la Arquitectura.

El aspecto fenomenológico tiene un carácter filosófico y epistemológico fundamentalmente materialista, aunque mantiene una esencia humanista heredada desde las manifestaciones platónicas renacentistas y neoclásicas, se basa en el existencialismo y realismo específico.

Este enfoque se centra en la experiencia espacial y la repercusión que esta tiene en los sentidos, con una atención especial en el sentido háptico y la envolvente corporal como el parámetro de referencia para el análisis de las coordenadas psicofísicas, que funcionan como directrices en el entendimiento del espacio.

Es posible encontrar afinidades de referencia cruzada, compleja o Deleuziana con el modernismo y el postmodernismo; ya que, al tener un fundamento materialista, se enfoca en la realidad material y la comprobación de la experiencia espacial más que en la estructuración

de conceptos; sin embargo, es guiada por aspectos cualitativos de la intuición y las apreciaciones sensoriales, que se aproximan a las motivaciones postmodernas.

También, a partir de este hecho, mantiene un carácter racional con una referencia humanista cercana al enfoque moderno. Según Montaner (2015), las bases, teorías y principales exponentes arquitectónicos son:

La tradición del realismo ha evolucionado hacia el realismo específico de la fenomenología de los filósofos Edmund Husserl, Merleau-Ponty, Edith Stein y Gaston Bachelard y del existencialismo de Martin Heidegger y Hannah Arendt. Ello se ha conceptualizado en las teorías de Juhani Pallasmaa y Alberto Pérez-Gómez, y se ha expresado en las obras de Steven Holl, Glenn Murcutt, Peter Zumthor, Elizabeth Diller y Ricardo Scofidio, Tod Williams y Billie Tsien, entre otros. (2015, p. 52)

Es importante agregar a los anteriores referentes, las aportaciones teóricas de Norberg-Schulz (1999; 1971), así como las de Bloomer y Moore (1983), como una referencia relevante en la consolidación teórica de los aspectos fenomenológicos en la arquitectura. Ellos también posicionan a los arquitectos Alvar Aalto y Luis Barragán, como manifestantes pioneros de una Arquitectura que pone atención a los detalles que promuevan el estímulo sensorial.

Una descripción a fondo de las construcciones teóricas de estos últimos exponentes, pudiera ser estudiada a partir de considerar el análisis de los aspectos topológicos del espacio cognoscitivo y artístico de Norberg-Schulz (1999; 1971) y los aspectos de estímulos en los sistemas sensoriales de la experiencia espacial de Bloomer y Moore (1983).

2.7.1.1. Aspectos topológicos del espacio cognoscitivo y artístico de Norberg-Schulz.

Para el entendimiento de los aspectos topológicos, primero debe tenerse como referencia la definición de los distintos espacios que entran en juego en la percepción humana, los cuales son según Norberg-Schulz (1971): (a) el espacio pragmático, de la acción

física, una función del instinto innato que es expresada y comunicada a través de relaciones espaciales como arriba y abajo, delante y detrás, derecha e izquierda; (b) el espacio cognoscitivo, del mundo físico; (c) el espacio artístico, o creado para expresar la estructura del mundo imaginario; (d) y el espacio abstracto, perteneciente a las relaciones lógicas.

El espacio cognoscitivo descrito anteriormente, se encuentra conformado por (a) el espacio perceptivo y (b) el espacio existencial; el primero se refiere a la orientación inmediata y el segundo forma la imagen estable del ambiente que rodea al hombre, también de este último derivan dos aspectos que permiten describirlo a detalle: (a) el aspecto abstracto, que consta de esquemas de índole *topológica* y *geométrica*, y (b) el aspecto concreto que se refiere a la captación de “elementos circundantes” como: el paisaje rural, ambiente urbano, edificios y elementos físicos.

El espacio artístico, paralelamente se encuentra conformado por (a) el espacio expresivo, el cual se encuentra en manos de personas especializadas como constructores, arquitectos y planificadores; y (b) el espacio estético, estudiado por teóricos de la Arquitectura y filósofos. Ambos forjan una construcción abstracta para su descripción, y juntos consolidan el *espacio arquitectónico*.

Según lo anterior, como parte del espacio existencial, el análisis de la topología es fundamental en la comprensión de los esquemas de organización y orientación espacial, antes de determinar las condiciones geométricas que determinarán una morfología con carácter más específico (ver Figura 33). Según Tedeschi (2014):

La topología es el estudio de la relación entre las partes geométricas sometidas a deformación. A diferencia de la geometría plana, el análisis topológico no requiere mediciones métricas o angulares; en cambio, el estudio de la topología se basa en la comparación de figuras. (2014, p. 411).

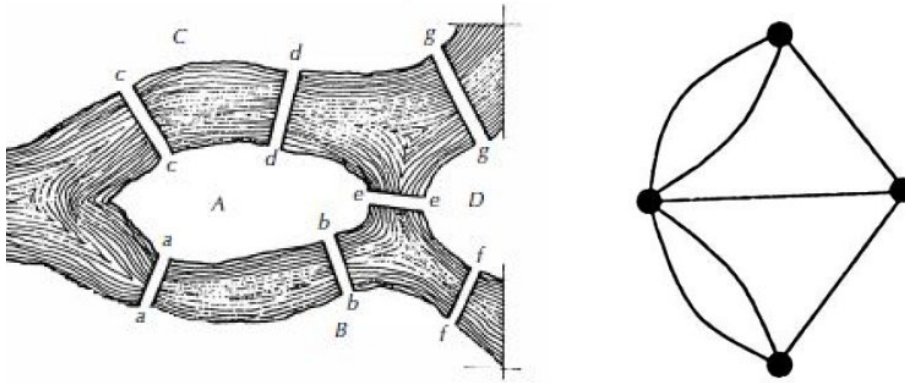


Figura 33. Stadler, M. M. (2002, p. 66). Diagrama de los siete puentes de Königsberg (izquierda) y Grafo de Euler como demostración topológica a los siete puentes de Königsberg (derecha) [diagrama].

Es posible describir para el estudio topológico (ver Figura 37) los esquemas elementales de organización como: (a) centros o lugares (proximidad), (b) direcciones o caminos (continuidad) y (c) áreas o regiones (cerramientos o cercados). “La topología no trata de distancias, ángulos y áreas permanentes, sino que está basada en relaciones tales como proximidad, separación, sucesión, clausura (interior-exterior) y continuidad”. (Norberg-Schulz y Schulz, 1971, p. 20).

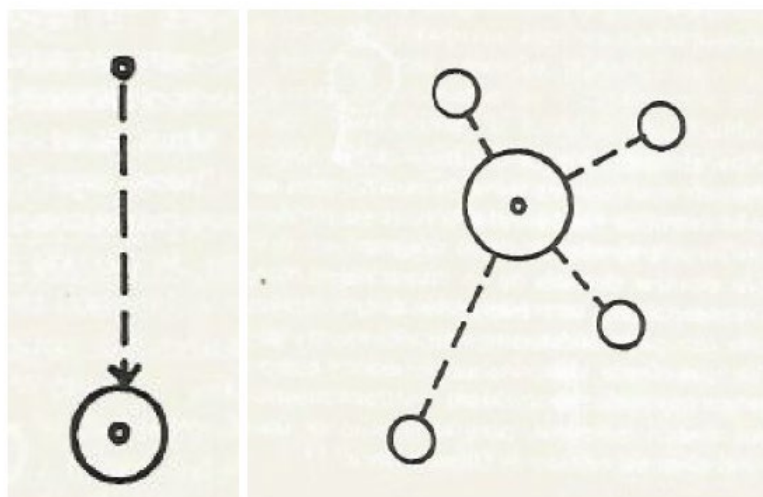


Figura 34. Norberg-Schulz, C. (1971, pp. 21, 22). Esquema topológico del *centro* y el *lugar* de las propiedades elementales del espacio existencial [diagrama].

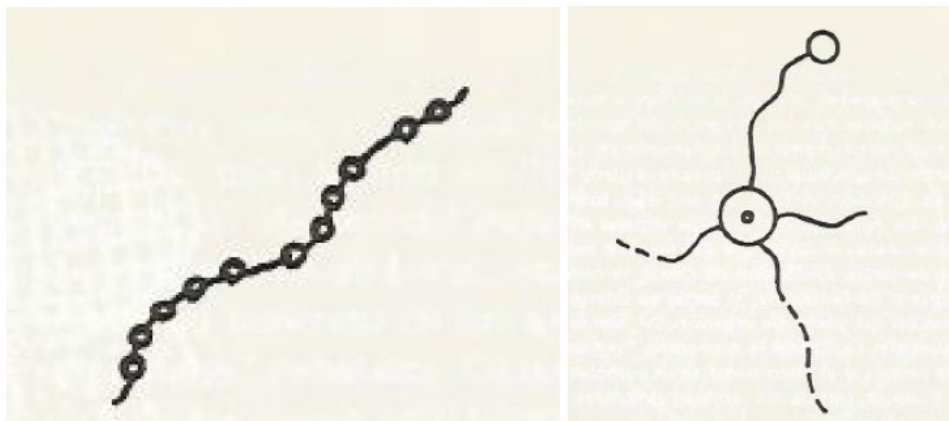


Figura 35. Norberg-Schulz, C. (1971, pp. 26, 27). Esquema topológico de la *dirección* y el *camino* de las propiedades elementales del espacio existencial [diagrama].

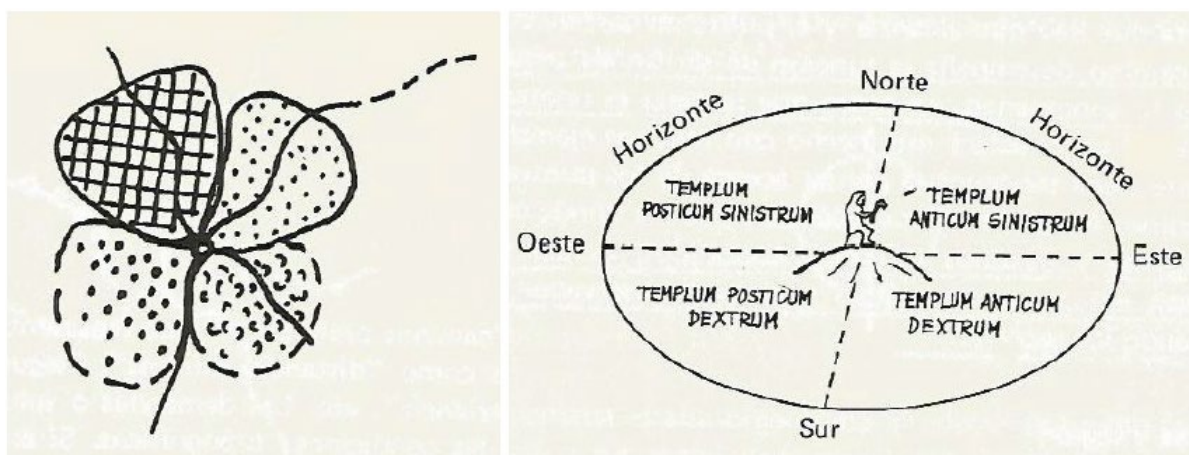


Figura 36. Norberg-Schulz, C. (1971, p. 28). Esquema topológico del *área* y la *región* de las propiedades elementales del espacio existencial [diagrama].

De la misma forma los elementos del espacio arquitectónico son: (a) lugar o nodo, se relaciona con los centros o lugares del espacio existencial, y se define como focos estratégicos en los que existen uniones de caminos o puntos de referencia externos al observador, y su función es indicar límites o direcciones; (b) camino y eje, con afinidad a las direcciones o caminos, se define como los canales a través de los cuales se mueve el observador, el eje organizador representa una dirección simbólica que unifica cierto número de elementos entre sí; (c) región y distrito, relacionado con las áreas o regiones, son las áreas

que el observador puede penetrar mentalmente y son reconocibles por tener algún carácter de identificación común.

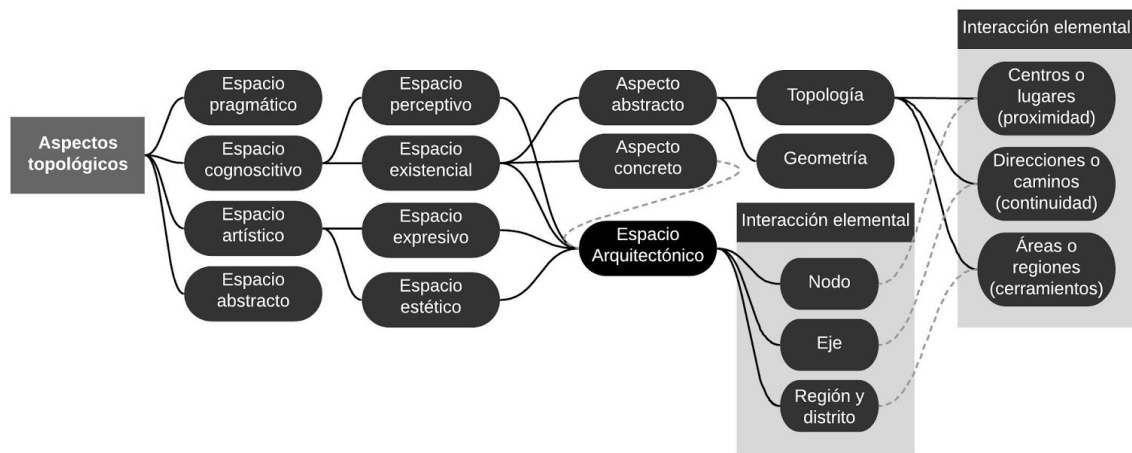


Figura 37. Aspectos topológicos del espacio cognoscitivo y artístico de Norberg-Schulz (1999; 1971).
Elaboración propia (2020).

2.7.1.2. Estímulos en los sistemas sensoriales de la experiencia espacial de Bloomer y Moore.

Para el análisis de los estímulos sensoriales de la experiencia espacial (Figura 38), en primera instancia, es importante definir la empatía como la transmisión de emociones o sentimientos acontecida a partir de la experiencia de los objetos. La empatía tiene un carácter meramente sensorial y no racional, se relaciona con el cuerpo y todos sus sentidos, y no específicamente con la vista (o con la mente).

El filósofo Robert Vischer ... se refiere a ella como sentimiento más que como proceso mental ... Al observar que también la experiencia de los objetos absolutamente abstractos podía despertar sentimientos ... avanzó la hipótesis de que la empatía con los objetos tiene lugar cuando se proyecta sobre ellos emociones personales. (Bloomer y Moore, 1983, p. 39).

A partir de la noción de la cualidad empática, la problemática de una desconexión entre el diseñador y su diseño adquiere mayor sentido. El debate relacionado con el desapego del diseñador con la percepción del espacio toma lugar en la ilustración, ya que es en ese momento cuando el sentido de la vista adquiere mayor relevancia sobre el resto de los sentidos y abre camino a la posibilidad de promover, por tanto, una falta de empatía sistemática con el espacio arquitectónico.

Como consecuencia del mecanismo fenomenológico de la empatía, es posible distinguir tres tipos de magnitudes al momento de percibir el espacio de una edificación, según Scott (1970):

En cualquier edificio pueden distinguirse tres cosas: la magnitud que realmente tiene (medida mecánica), la magnitud que parece tener (medida visual), y la sensación de magnitud que produce (medida corporal). Las dos últimas han sido muchas veces confundidas, pero la única que posee valor estético es la sensación de magnitud.

(citado por Bloomer y Moore, 1983, p. 40)

Así, es a través del conjunto de sistemas sensoriales (sistema visual, sistema auditivo, sistema gusto-olfativo, sistema de orientación y sistema háptico) propuestos por J. J. Gibson (citado por Bloomer y Moore, 1983), que la empatía es transmitida hacia las condiciones del espacio arquitectónico.

Es decir, la falta de empatía en el diseño de un espacio se produce por únicamente valorar el sistema visual, en lugar de considerar más importante el sistema de orientación y el háptico, que hacen trabajar al resto sinérgicamente.

El trabajo sinérgico de los sistemas sensoriales como principio organizativo, es determinado perceptualmente por la envolvente corporal, que marca una frontera entre el espacio personal, y el extra personal. “Se trata de una especie de barrera inestable sometida a

la acción de los acontecimientos que producen tanto en su interior como en el exterior de la misma” (Bloomer y Moore, 1983, p. 55).

La percepción de los estímulos de la envoltura corporal es guiada por las *coordenadas psicofísicas* como *parámetros fenomenológicos*: arriba-abajo, delante-detrás, a la derecha-a la izquierda, así como a lo que se encuentra en el centro (el cuerpo). La noción de estas coordenadas puede adquirir una concepción ontológica:

Las coordenadas delante/detrás y derecha/izquierda son las que llevan más directamente a la definición de las cuatro coordenadas polares, Norte, Sur, Este y Oeste, mientras que los tres niveles verticales hacen referencia a las trilogías místicas, que en la cultura occidental están constituidas por el paraíso, la tierra y el infierno. (Bloomer y Moore, 1983, p. 52,53).

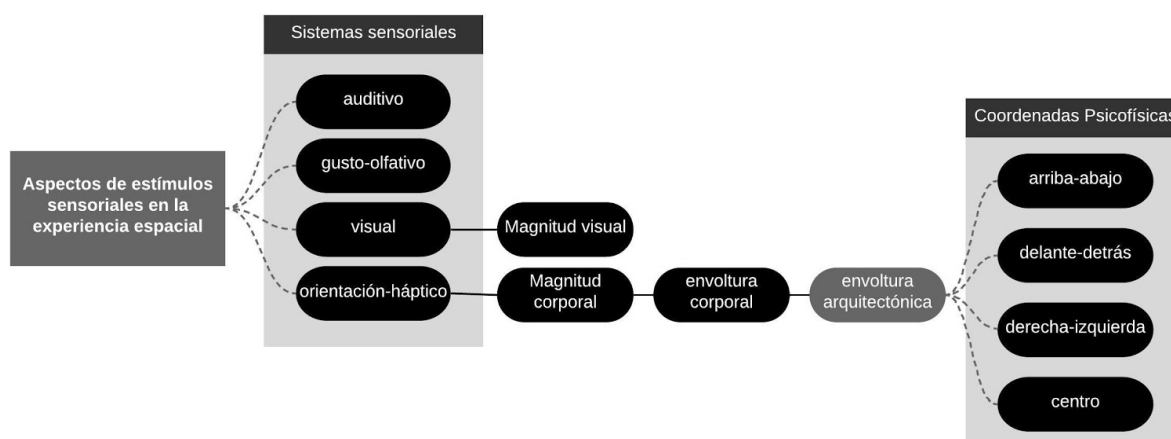


Figura 38. Estímulos en los sistemas sensoriales de la experiencia espacial de Bloomer y Moore (1983). Elaboración propia (2020).

2.7.1.3. La importancia de las cosas concretas y las atmósferas de Zumthor.

Las apreciaciones fenomenológicas, recientemente se han visto alimentadas (y reforzadas) por la manera de entender la arquitectura de Peter Zumthor (2011, 2016). Sus contribuciones teóricas se posicionan como parte de los argumentos de la contemporaneidad y tienen, por tanto, una connotación diferente a las de Norberg-Schulz (1999; 1971) y

Bloomer y Moore (1983), que son afines a las motivaciones de dinamismo, movimiento y complejidad arquitectónica del postmodernismo.

Con un posicionamiento más cercano a las ideas reduccionistas del movimiento moderno, Zumthor aboga por las cosas concretas y su naturaleza *per se*, con una aproximación a las características de la tendencia artística y arquitectónica minimalista. Aboga por una relación directa y materialista entre el hombre y el objeto, sin ninguna estructuración conceptual de por medio, más que la acontecida por la experiencia espacial y los estímulos sensoriales afectados por la creación de una atmósfera espacial.

La realidad de la arquitectura es lo concreto, lo convertido en forma, masa y espacio, su cuerpo. No hay ninguna idea fuera de las cosas ... La arquitectura es siempre una materia concreta; no es abstracta, sino concreta ... Con mi trabajo contribuyo a que aparezcan circunstancias reales y creo determinadas atmósferas en el espacio que hacen que se despierten nuestros sentimientos. (Zumthor, 2016, p. 37,66,85).

Así, considera el proceso de diseño arquitectónico como una tensión entre el sentimiento y la razón, y le da al acto de proyectar el significado de inventar (o descubrir). Su metodología gira en torno a la suposición de imágenes que son producto de la ausencia, la melancolía y el recuerdo. Por otro lado, comprende a la belleza como una sensación, en el que la razón desempeña un papel subordinado (Figura 39).

El proceso de proyecto se basa en un continuo juego conjunto de sentimiento y razón. Por un lado, los sentimientos, las preferencias, las nostalgias y los deseos que emergen y que quieren cobrar forma deben examinarse por medio de una razón crítica. Del otro, el sentimiento dice si las reflexiones abstractas concuerdan entre sí. (Zumthor, 2016, p. 21).



Figura 39. Pabellón Swiss Sound (Hanover, 2000). Fuente: Zumthor, P. (2000). *Pabellón Swiss Sound* [Imágen]. Recuperado de <https://es.wikiarquitectura.com/edificio/pabellon-swiss-sound/>.

2.7.2. Análisis descriptivo de los aspectos semiológicos en la Arquitectura.

Contrario al enfoque fenomenológico, los aspectos semiológicos en la Arquitectura tienen un carácter filosófico y epistemológico idealista (platónico), basados en la razón, en el desarrollo de las ideas y la construcción de conceptos a través del manejo de los diagramas como principal herramienta.

Colin Rowe primero me enseñó a ver aquello que no estaba presente en un edificio ... quería que viera las ideas implícitas en aquello que estaba físicamente presente. En

otras palabras, que me preocupara menos por lo que el ojo ve -lo óptico- y más por lo que ve la mente -lo visual-. (Eisenman, 2011, p. 16).

Sin embargo, paradójicamente parten de referencias romanticistas, organicistas y surrealistas, que posteriormente se posicionan en referencias provenientes del empirismo radical del pragmatismo estadounidense y del expresionismo arquitectónico del siglo XX. Se valen de los símbolos y de los iconos para desarrollar un lenguaje de diagramas y metáforas complejas, con una clara influencia de los desarrollos epistemológicos de Deleuze y Guattari (2002, 2017).

Se trata de una Arquitectura en la que, como en los sueños, se superponen tiempos, materias y capas; unas obras que se relacionan con el entorno revelando estratos y elevando voladizos, y en el que cada detalle, a la manera del Romanticismo y el surrealismo, se individualiza, se independiza, toma vida y se repiensa de una manera nueva. (Montaner, 2015, p. 27).

Es posible encontrar entre sus principales exponentes, a arquitectos como Frank O. Gehry, Thom Mayne (Morphosis), los experimentos futuristas de NOX, FOA con una notable herencia de Archigram y Future Systems; la arquitectura fragmentada de Peter Eisenman (2011), Bernard Tschumi y Rem Koolhaas, que a la vez fueron influencia importante de los estudios de Arquitectura: Zaha Hadid Architects, MVRDV, UNStudio (Ben van Berkel), BIG (Bjarke Ingels).

La propuesta de arquitecturas hechas como *collage* de fragmentos, teorizada por Colin Rowe y Peter Eisenman a partir de la década de 1970, ha seguido desarrollándose hasta la actualidad. Tiene su máximo exponente en la prolífica obra de OMA, estudio liderado por Rem Koolhaas, y se expresa en gran parte de la Arquitectura holandesa actual. (Montaner, 2015, p. 66).

Esta arquitectura de diagramas, conceptos y símbolos han sido una influencia importante para la eclosión de la arquitectura computacional y su avance hacia las teorías y metodologías de la Arquitectura Paramétrica y Algorítmica.

2.7.3. Análisis de la condición semiológica y fenomenológica de la Arquitectura contemporánea.

A partir del previo análisis, es posible distinguir en la Arquitectura contemporánea una amplia reminiscencia del enfoque semiológico en su metodología de diseño, con una herencia proveniente de la ilustración a partir del establecimiento hegemónico de la razón sobre los sentidos en el desarrollo tecnológico y constructivo.

Descartes, citado por (Bloomer y Moore, 1983, p. 35), uno de los representantes más importantes de ese movimiento, introdujo el sistema de coordenadas cartesianas u ortogonales espaciales; a partir de ello, consolidó una lógica que exigía significados objetivos, racionalizados bajo el yugo de la deducción y no de los sentidos (Bloomer y Moore, 1983; Norberg-Schulz y Schulz, 1971).

Es prácticamente la agenda racionalista la que marcó las directrices, no sólo para el Neoclasicismo, sino el camino hacia la modernidad. Bajo esta perspectiva, la razón y la lógica adquieren un valor mayor sobre los conocimientos no racionales provenientes de la percepción sensorial.

Es así como este paradigma cimbra las bases de la ciencia moderna causal, fundamentada en la síntesis matemática y diferencial, que brindaron mayor jerarquía e influencia social a las disciplinas ingenieriles, militares e industriales de la modernidad, así como las relacionadas con la computación en la primera etapa de la era digital contemporánea (Carpo, 2017).

A pesar de que el objetivo de la ciencia moderna es afín a la posición materialista aristotélica del “ver para creer”, como una metodología donde la comprobación es

fundamental para la determinación de una verdad, se sustenta en una lógica no solo racionalista, sino también humanista, donde el camino para llegar a esa verdad es estructurado por la razón, las ideas, el pensamiento o la mente. Es decir, esta última afinidad se apega a la visión platónica idealista de comprensión de las cosas que se encuentran alrededor.

Estas relaciones paradójicas en el debate entre estos dos enfoques existen de manera constante. Por ejemplo, en un sistema social, económico y político de tipo imperialista, como lo fue la época medieval, sustentado por la religión y la nobleza, donde la espiritualidad y la teología jugaban un papel importante en las estructuras sociales desde una clara perspectiva idealista; con relación al arte, la postura era totalmente diferente, pues las manifestaciones eran intuitivas, con mayor cercanía al sentido háptico y mayor apego al cuerpo, a su escala como principio ordenador.

Fue entonces, cuando las cuestiones divinas y humanas encarnadas por la aristocracia y la iglesia se vieron desafiadas por los planteamientos de los ingenieros, militares e industriales, cuya influencia crecía. El paso desde lo que había sido la presencia del cuerpo humano como principio organizador -divino- de la Arquitectura a otro tipo de organización más mecánica de la misma se vio decisivamente impulsado por las argumentaciones (matemáticas) de Galileo. (Bloomer y Moore, 1983, p. 27)

Así, es posible verificar que estas relaciones dicotómicas se han reiterado de manera cíclica hasta la fecha. La época clásica era impulsada por la razón, sembró las bases de la filosofía y el arte era desarrollado bajo la matemática; en contraste la edad media, que se desarrolló a partir de la espiritualidad, se impulsó artísticamente por la intuición y los sentidos.

Posteriormente el humanismo renacentista retoma la razón y el arte bajo la lógica de la precisión abstracta matemática; luego, tanto el manierismo como el barroco abrazan de

nuevo a la creatividad espontánea, instintiva y corpórea. Es en el Neoclasicismo donde se desarrollan las bases de la democracia y la ciencia moderna, luego el Romanticismo busca romper ese paradigma a través de recuperar la importancia de los sentimientos, la pasión y la intuición; un conjunto de valores que parecieran alimentar y ser afines al carácter contestatario de la vida postmoderna.

Es importante mencionar que esta dicotomía cíclica no tiene un carácter lineal, sino es una estructura de mayor complejidad que muestra claras paradojas y contradicciones que hacen de su apreciación, un sistema rizomático más próxima a la epistemología y filosofía de Deleuze y Guattari (2002).

El siglo XX comienza con transformaciones importantes en los modelos de vida de las personas a partir del avance tecnológico que la revolución industrial fomentó. Este periodo marco las pautas necesarias para que los medios de comunicación evolucionarán e impulsarán el desarrollo científico. El conjunto de estos fenómenos abrió paso a que se retomara la hegemonía del enfoque racional en un nivel radicalizado y paradigmático, que tuvo como cúspide el movimiento moderno.

Bajo la modernidad, el método científico se estableció como el modelo base para el desarrollo tecnológico. A pesar de que los medios de comunicación y la transferencia de información evolucionaron en ese lapso de tiempo, no fue hasta el inicio del siglo XXI que fueron lo suficientemente avanzados para tener un carácter de accesibilidad inmediata y global, gracias a la tecnología digital y lo que hoy en día es conocido como el *Big Data* (Carpo, 2011, 2017).

Esta situación, condicionó el método científico de la modernidad a trabajar con poca información para resolver problemas complejos. Así, el racionalismo se mantuvo hegemónico sobre cualquier método sensitivo, intuitivo y heurístico, ya que era fundamental trabajar bajo la lógica matemática, abstracta y causal (Carpo, 2011, 2017, 2019).

En términos de Arquitectura y Arte, la situación no fue diferente, el movimiento moderno fue el principal paradigma del siglo XX. A pesar de que el periodo culminó con movimientos contestatarios, la racionalidad fue la principal directriz para el desarrollo creativo del momento.

Ante esa racionalidad dominante, los aspectos semiológicos de la Arquitectura adquieren un valor importante en los principales exponentes de del movimiento moderno, los cuales siguen latentes en las manifestaciones arquitectónicas contemporáneas. El posmodernismo, como el principal antagonista del movimiento moderno, trajo de vuelta la necesidad de reflexionar sobre el valor de los aspectos fenomenológicos en la Arquitectura del siglo XXI.

Martin Heidegger, Michael Foucault y Jacques Derrida han expuesto que el pensamiento y la cultura de la modernidad no solo han continuado con el privilegio histórico de la vista, sino que han fomentado sus tendencias negativas. Cada uno a su manera han considerado el dominio de la vista en la era moderna como claramente diferente al de épocas anteriores. (Pallasmaa, 2014, p. 25)

Los avances en la tecnología digital, así como el *Big Data* consecuente, también han permitido revalorar estrategias heurísticas e intuitivas para la resolución de problemas complejos. Ya que la computadora, la inteligencia artificial y la robótica, se han convertido en medios autónomos importantes, que tienen la capacidad de recuperar información para resolver tareas inmediatamente, sin afectación en la economía de tiempo y costo (Carpo, 2017, 2019; Linder et al., 2019; Sanchez, 2020).

2.7.4. Sinergia entre los enfoques fenomenológicos y semiológicos de la Arquitectura de Kengo Kuma.

Como se expone anteriormente, al margen de la fricción existente entre los enfoques fenomenológicos y semiológicos en el desarrollo de la concepción arquitectónica, es posible

distinguir arquitectos que han incorporado en su visión de diseño espacial estos dos aspectos de manera equilibrada y sinérgica, con una clara relación retroalimentativa entre los enfoques aplicados en la producción de sus propuestas arquitectónicas.

En este sentido, arquitectos modernistas-posmodernistas como Frank Lloyd Wright, Alvar Aalto y Louis Kahn destacan por relacionar la razón y los sentidos de manera ejemplar e inclusive con cualidades canónicas, tal como manifiesta Peter Eisenman (2011) en su libro *Diez edificios canónicos: 1950-2000*.

En la contemporaneidad, arquitectos como Herzog & DeMeuron, Sou Fujimoto, Toyo Ito, o Kengo Kuma (2010), muestran con su Arquitectura que los aspectos fenomenológicos y semiológicos son recursivos e integrales, de tal forma que ambas consideraciones epistemológicas consolidan de manera holística el diseño del espacio de un producto arquitectónico. Por otro lado, todos estos últimos arquitectos mencionados, con excepción de Kengo Kuma (2010), abrazan la arquitectura computacional sin problema alguno y de manera enérgica.

En el seguimiento de esta lógica, un ejemplo claro y amplio de esta sinergia arquitectónica aplicada de los enfoques debatidos, puede ser apreciada en la arquitectura de Kengo Kuma (2010), que retoma del pasado aspectos técnico-constructivos artesanales heredados de la tradición japonesa, y los reinterpreta en manifestaciones contemporáneas arquitectónicas.

Tal reinterpretación refleja una clara inquietud por integrar elementos arquitectónicos que, por un lado, sean representativos y análogos a símbolos tanto constructivos como culturales con un latente carácter histórico (semiología); y por otro, reflejen a través la combinación de elementos arquitectónicas, organización y manejo de los materiales, reminiscencias de carácter cognoscitivo y existencial, relacionados con la tradición japonesa y su añoranza (fenomenología).

Kengo Kuma (2010), recuerda con sus proyectos la Arquitectura artesanal e intuitiva, con un carácter de autoconstrucción, y la refleja a través de la disposición de elementos consecutivamente agregados de materiales blandos, frágiles y accesibles, que buscan constreñir la necesidad de una lógica técnica e industrializada guiada por la matemática y la ciencia (Figura 40-43).

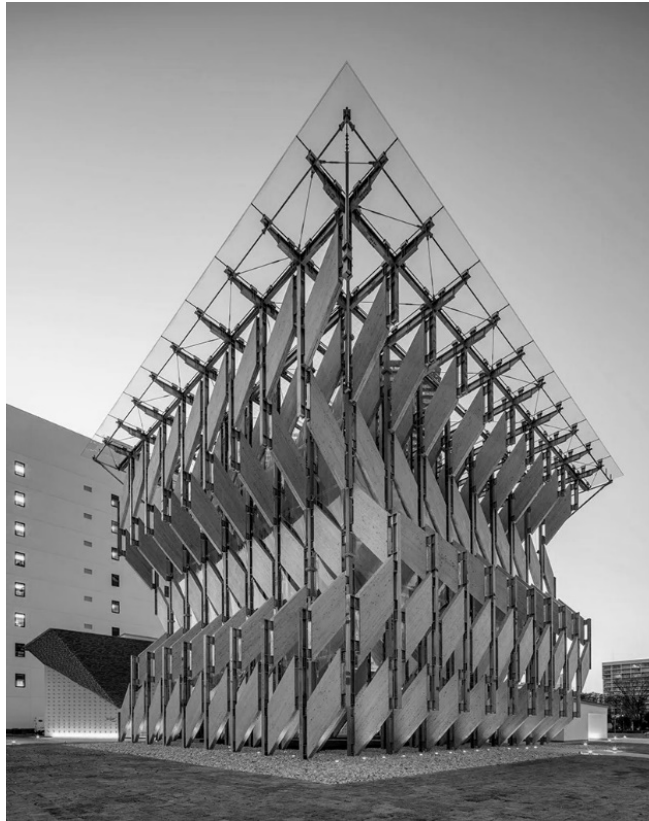


Figura 40. Pabellón CLT Park Harumi (Tokio, 2019). Fuente: Kuma, K. (2019a). *CLT Park Harumi* [Imágen]. Recuperado de <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-02.jpg>. Copyright [2019] © Kawasumi • Kobayashi Kenji Photograph Office.



Figura 41. Pabellón *CLT Park Harumi* (Tokio, 2019). Fuente: Kuma, K. (2019b). *CLT Park Harumi* [Imágen]. Recuperado de <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-05.jpg>. Copyright [2019] © Kawasumi • Kobayashi Kenji Photograph Office.



Figura 42. Pabellón *Climbable Wooden Pavilion* (Paris, 2015). Fuente: Kuma, K. (2015a). *Climbable Wooden Pavilion* [Imagen]. Recuperado de https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_11.jpg.



Figura 43. Pabellón *Climbable Wooden* (Paris, 2015). Fuente: Kuma, K. (2015b). *Climbable Wooden Pavilion* [Imagen]. Recuperado de https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_31.jpg.

2.7.5. Implementación de aspectos fenomenológicos y semiológicos en la Arquitectura computacional

Terzidis (2006) abre camino al desarrollo teórico de su libro *Algorithmic Architecture*, y expresa que, a partir del traslado del uso de herramientas manuales a las computacionales en la Arquitectura, existe una preocupación en este el ámbito a nivel internacional por la posible pérdida de control de los arquitectos sobre sus propios diseños. Este planteamiento se dirige hacia una problemática de desconexión entre el diseñador y sus proyectos.

Aunque la respuesta a este problema la dirige a la manera de usar la inteligencia artificial de la computadora; profundizar en estas desconexiones y discernirlas tiene importancia en el desarrollo metodológico de la Arquitectura computacional.

Esta problemática, por tanto, se aproxima a una cuestión de desapego que puede recaer en distintos enfoques relacionados con el proceso de diseño y el uso de las tecnologías computacionales.

Un enfoque se relaciona con la falta de apego de la realidad arquitectónica desde la perspectiva de la experiencia física o material, con una connotación fenomenológica; es decir, una desconexión sensorial, especialmente háptica, que trae como consecuencia una falta de empatía espacial al momento de generar las variables tectónicas que repercuten psicofísicamente en el usuario de un proyecto.

Las imágenes por ordenador tienden a aplanar nuestras magníficas, multisensoriales, simultáneas y sincrónicas capacidades de imaginación al convertir el proceso de proyecto en una manipulación visual pasiva, un viaje retiniano. El ordenador crea una distancia entre el hacedor y el objeto, mientras que el dibujo a mano, así como trabajar con maquetas, colocan al proyectista en un contacto háptico con el objeto o espacio. (Pallasmaa, 2014, p. 14).

Otro enfoque, derivado del anterior, se relaciona con la metodología arquitectónica y el carácter semiológico que puede tener, cuando se considera el diagrama como una herramienta fundamental en el proceso creativo y expresivo de diseño, con el fin de dar un fuerte valor a las ideas o conceptos para dar respuesta a los problemas.

Una visión idealista y predominantemente racional puede promover resultados alienados con el cuerpo humano, con probabilidades inconscientes de falta de apego a la realidad arquitectónica.

Los diagramas se han convertido en emblemas del cambio de siglo ... Su versatilidad ha jugado a su favor, pero sus tentaciones conceptuales, abstractas y simplificadoras pueden conducir a procesos y resultados forzados ... podemos valorar tanto su versatilidad y su capacidad de transmitir ideas, actividades, intenciones y complejidades como sus riesgos de caer en el formalismo y la arbitrariedad. (Montaner, 2015, p. 91).

Estos enfoques, que evidencian una problemática con la arquitectura computacional, abren un debate en el que la solución depende del nivel logrado de equilibrio y sinergia entre los aspectos fenomenológicos y semiológicos anteriormente descritos en el desarrollo del espacio arquitectónico.

Una aproximación metodológica consolidada, en la que se incorpore el análisis de los aspectos fenomenológicos y semiológicos como consideraciones importantes en el proceso de diseño arquitectónico con enfoque Algorítmico Paramétrico (AP), es el propuesto por Patrik Schumacher (2011, 2012) en los títulos *Autopoiesis of Architecture Vol.I: a new framework for architecture* y *Autopoiesis of Architecture Vol.II: a new agenda for architecture*.

Schumacher, como se expone anteriormente, describe una agenda para calificar *condiciones de organización y articulación* en el proceso de diseño, para de esta forma ejercer relaciones físicas y cognitivas que analizan *los patrones* y las condiciones *fenomenológicas* (de percepción) y *semiológicas* (de significación).

Por otro lado, es posible discernir una afinidad entre las manifestaciones arquitectónicas de este enfoque vanguardista Discretismo y las del arquitecto Kengo Kuma (2010), a pesar de que existe una brecha amplia de pensamiento y criterio entre ambos. Kuma rechaza firmemente la experimentación computacional, tecnológica y científica, ya que va en contra de su filosofía y visión de la Arquitectura; el Discretismo ve la tecnología digital y los avances en la robótica, como el medio en el que se desenvuelve su producción arquitectónica.

A pesar del paralelismo, hay una cercanía clara en la tectónica resultante de sus productos arquitectónicos. También en sus discursos existe un enemigo común: la modernidad industrial y la lógica matemática diferencial como método para la resolución de problemas. Kuma aboga por recuperar los aspectos intuitivos de los procesos artesanales, y el Discretismo se apoya en las lógicas heurísticas para la solución de problemas a través de la automatización (Linder et al., 2019; Retsin, 2019; Sanchez, 2019b, 2020), que implica

procesos de inteligencia artificial y robótica como métodos característicos de la segunda era digital de la arquitectura (Carpo, 2011, 2019).

En este sentido, las características del discurso de diseño del Discretismo pueden promover a través de sus métodos computacionales, la construcción eficiente y optimizada de elementos distribuidos consecutivamente mediante un sistema de agregación, que aporten en una relación sinérgica de los aspectos fenomenológicos y semiológicos, tal como lo demuestra la Arquitectura de Kengo Kuma (2010).

2.8. Aspectos de análisis fisiológico arquitectónico como parámetros cuantitativos de diseño Algorítmico Paramétrico

Uno de los elementos importantes en la Arquitectura, que demanda complejidad, y que es fundamental para el entendimiento del espacio es la envolvente arquitectónica. Ya que no solo define los límites espaciales, sino es un agente que se desenvuelve como un órgano que establece las directrices de forma y funcionamiento.

Un análisis desde el aspecto fisiológico arquitectónico de la envolvente arquitectónica brinda una oportunidad para generar estrategias metodológicas de optimización arquitectónica. El objetivo es identificar las variables oportunas para el análisis de los aspectos espaciales cuantitativos que aporten los procesos Algorítmicos y Paramétricos (AP). Estas variables funcionarán como parámetros que servirán al análisis y simulación de aspectos estructurales, así como a la optimización y desempeño del espacio con relación al medio ambiente.

El análisis de los aspectos estructurales se desarrolla con estrategias de optimización a través del manejo de simulaciones computacionales de sistemas de resorte de partículas; es decir, sistemas de cálculo y visualización de físicas. Las principales características para evaluar su optimización son (1) los aspectos geométricos y (2) los aspectos topológicos de la forma. La toma de decisiones de diseño de estos aspectos, pueden ser analizados a través dos

estrategias algorítmicas de evaluación con una plataforma de programación visual, funcional u orientada a objetos.

La primera estrategia consiste en un modelo (1) *causal*, basado en lógicas matemáticas - diferenciales para la aplicación de simulaciones computacionales, a fin de visualizar resultados complejos en tiempo real, de acuerdo con los parámetros introducidos, en modelos dinámicos en tercera dimensión. Posteriormente, a efecto de dar respuesta a las problemáticas manifestadas en la simulación, se procede a aplicar estrategias paramétricas de optimización y desempeño.

La segunda estrategia es el modelo (2) *heurístico*, a través de búsquedas generativas optimizadas producto de la inteligencia artificial, que desarrollan procesos evolutivos de prueba y error hasta llegar a resultados que cumplan con los parámetros definidos por el diseñador, se utilizan simulaciones de Algoritmos Genéticos (AG) o de Análisis de Elementos Finitos (AEF), este último para una búsqueda geométrica-topológica estructural. Según Santamouris (2007, p. 93) los principales parámetros de los AG son: (a) tamaño de la población, (b) tasa de cruce, (c) tasa de mutación, (d) brecha generacional, (e) tipo de cromosoma.

El estudio de *optimización* integra diseño y decisiones estructurales, para crear soluciones óptimas dentro de los parámetros establecidos. Estas soluciones son a menudo estructurales y tienen como objetivo reducir la cantidad de material, abarcar claros mayores y aprovechar las latencias estructurales. La optimización también se puede aplicar al programa, función y forma del edificio. (Tedeschi, 2014, p. 405).

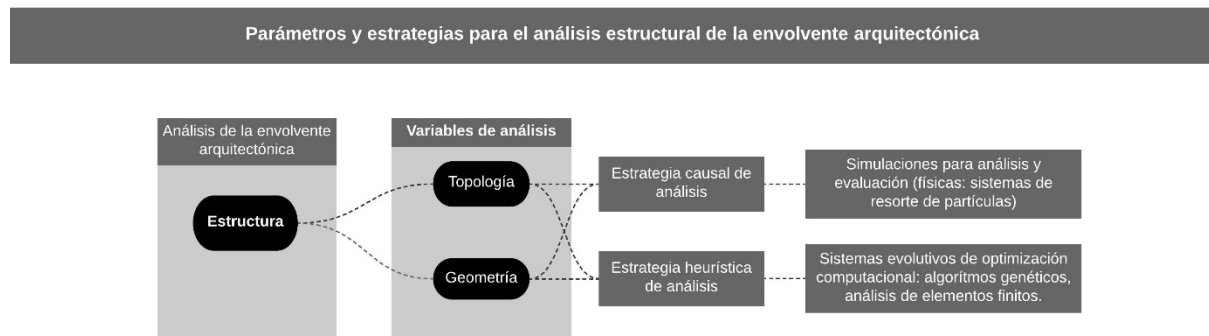


Figura 44. Parámetros y estrategias para el análisis estructural de la envolvente arquitectónica.

Elaboración propia (2020).

Los aspectos estructurales de geometría y topología son determinantes como estrategias de búsqueda de la forma (*form-finding*) de un proyecto, y condicionan la morfología de la envolvente arquitectónica. De manera posterior o paralela, la envolvente puede ser analizada para medir sus capacidades de adaptabilidad con el medio ambiente.

Desde esta perspectiva, el análisis debe considerar características de (1) geometría y sus respectivas variables de (1.1) compacidad, (1.2) área de extensión, (1.3) orientación, (1.4) ángulos de inclinación y (1.5) permeabilidad; así como también, sus (2) propiedades térmicas, que incluye la (2.1) transmitancia térmica (valor U) y la (2.2) absorptividad.

Como ejemplo, las estructuras de análisis fisiológicas desarrolladas en este trabajo de investigación se enfocarán en la *radiación solar* como mecanismo de transferencia de calor incidente en el espacio. Dentro de esta perspectiva las unidades de estudio involucradas con este mecanismo son: Watts (W) para la radiación solar directa, Watts/m² (W/m²) para la radiación solar global y °C para la temperatura superficial.

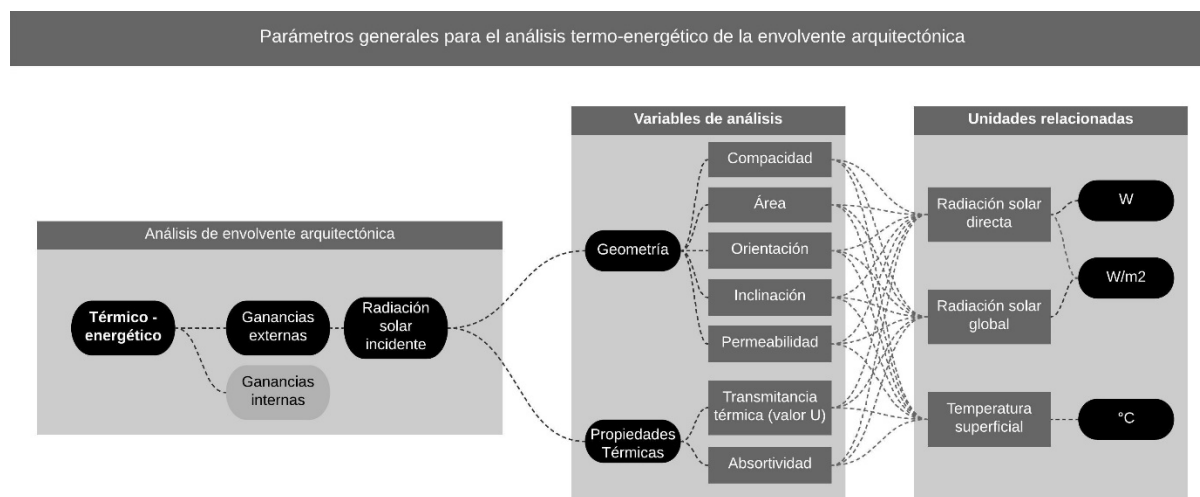


Figura 45. Diagrama de parámetros generales para el análisis termo-energético de la envolvente arquitectónica. Elaboración propia (2019).

Al tomar como parámetro de referencia la radiación solar incidente sobre la superficie de la envolvente arquitectónica, esta será también analizada mediante las dos estrategias algorítmicas de evaluación (causal y heurística) explicadas anteriormente.

Varias aplicaciones computacionales están disponibles en el campo del diseño ambiental; Sin embargo, la mayoría ... aportan pequeñas contribuciones para explorar soluciones de diseño holístico. Debido a las dificultades para realizar cambios durante las etapas avanzadas de modelado / diseño, los diseñadores suelen confiar en el juicio personal deducido de la experiencia, sobre el análisis. De este modo, se reducen las posibilidades de explorar un gran conjunto de diseños con información ambiental. El uso combinado de programas ambientales y paramétricos puede guiar la generación de formas desde los bocetos iniciales hasta el diseño final. (Tedeschi, 2014, p. 441).

III. METODOLOGÍA

Las teorías y metodologías expuestas anteriormente en torno a la Arquitectura Algorítmica Paramétrica (AAP) son las principales referencias para la construcción

metodológica del objetivo particular número dos del presente trabajo de investigación, el cual consiste en *definir una metodología de diseño arquitectónico Algorítmico Paramétrico (AP)*.

Las características del diseño de la estructura metodológica propuesta se muestran en la siguiente matriz de congruencia metodológica (ver Tabla 1). La clasificación de características de análisis metodológico presentes en la matriz, fueron retomadas de referentes en el desarrollo de metodologías de investigación con relación a la disciplina de Arquitectura, los cuales son: Groat y Wang (2002), Hernández Sampieri et al. (2014), Barrantes Echavarría (1999) y Martínez Zarate (1991).

La investigación por realizar tiene un carácter *mixto* según Hernández Sampieri et al. (2014), ya que se desarrolla en dos líneas de investigación que son de tipo *cuantitativo* y *cuantitativo*. El objetivo particular de investigación se evalúa a detalle en la matriz a partir de su (a) enfoque, (b) sujetos u objetos de estudio, (c) variables que utilizará, (e) métodos que aplicará y (f) técnicas que utilizará.

Tiene un enfoque no-experimental de tipo arquitectónico, tecnológico digital-virtual y tecnológico performativo. Los sujetos de estudio serán (1) los estudiantes de arquitectura, (2) profesionistas de la disciplina arquitectónica con interés en la tecnología, (3) profesionistas de la disciplina con interés en la programación digital, así como (4) estudiantes y profesionistas afines a la disciplina de diseño con interés en la tecnología digital. Las variables por utilizar serán del tipo (1) inter, por que ejercerá comparaciones entre parámetros de análisis de un mismo tipo (cualitativos o cuantitativos); y (2) dicotómica, porque también contrastará de manera cruzada valores entre los parámetros cualitativos y cuantitativos.

Los métodos por aplicar serán combinados, se distinguen: (1) simulación y modelado en tercera dimensión, (2) aspectos cualitativos de análisis, (3) aspectos cuantitativos de análisis, así como (4) argumentaciones lógicas que demanda la elaboración de algoritmos.

Se utilizarán técnicas que implican (1) instrumentos de aplicación cualitativa y cuantitativa, (2) gráficos y (3) tablas de información cualitativa y cuantitativa, (4) gráficos descriptivos, (5) instrumentos algorítmicos, (6) modelos virtuales, (7) simulaciones virtuales.

Tabla 1

Matriz de congruencia metodológica del segundo objetivo particular de investigación.

Objetivo de investigación		
Definir una metodología de diseño arquitectónico algorítmico y contrastarlo con las tradicionales, se toma como caso de estudio y aplicación la comunidad estudiantil de la FAD (UABC).		
Enfoque	Variables	Técnicas
1. No experimental: -Arquitectónico -Tecnológico digital-virtual -Tecnológico performativo	1. Inter 2. Dicotómica	1. Instrumentos de aplicación cualitativa / cuantitativa 2. Gráficos de información cualitativa y cuantitativa 3. Tablas de información cualitativa / cuantitativa 4. Gráficos descriptivos 5. Instrumentos algorítmicos 6. Modelos virtuales 7. Simulaciones virtuales
Sujeto	Metodología	
1. Estudiantes de Arquitectura 2. Profesionistas de la disciplina de Arquitectura con interés en la tecnología 3. Profesionistas de la disciplina de Arquitectura con interés en la programación digital 4. Estudiantes y profesionistas afines a la disciplina de diseño con interés en la tecnología digital	Estrategias combinadas: 1. Simulación y modelaje 2. Cualitativa 3. Cuantitativa 4. Argumentación lógica	

Nota. Elaboración propia (2019). Clasificación de las características de análisis metodológico retomadas de Groat y Wang (2002), Hernández Sampieri et al. (2014), Barrantes Echavarría (1999) y Martínez Zarate (1991).

La metodología de diseño arquitectónico AP propuesta en esta investigación se encuentra constituida en tres partes. Primeramente, se hace una (1) introducción a Grasshopper como plataforma de programación visual como el medio a través del cual se

orientarán las herramientas y estrategias propuestas en el trabajo. Posteriormente, se (2) expondrán estrategias algorítmicas básicas en *Grasshopper* para aplicación en diseño arquitectónico. Después, se (3) presentará el desarrollo de la metodología a proponer, la cual se encontrará conformada por dos líneas de análisis: una cualitativa y otra cuantitativa. Finalmente, se mostrarán (4) ejemplos de aplicación de la metodología de diseño arquitectónico AP.

3.1. Introducción a *Grasshopper* como plataforma de programación visual

En Arquitectura, la plataforma integrada al programa CAD *Rhinoceros*®, llamada *Grasshopper* es uno de los medios más sólidos para acercar al arquitecto, y a las disciplinas afines al diseño, al uso de algoritmos orientados al diseño de objetos, así como al amplio universo de la programación computacional sin necesariamente tener conocimientos o habilidades previas. En palabras de Hansen Phillips (2018):

Actualmente, *Grasshopper*, un editor de programación visual para el programa modelador en tercera dimensión *Rhino3D*, es la principal herramienta usada por los arquitectos paisajistas, tanto para el análisis del sitio, como para el diseño paramétrico.” (citado por Cantrell y Mekies, 2018, p. 208).

Por tanto, *Grasshopper* es un entorno de programación computacional visual (*Graphical User Interface*, GUI) que funciona como un editor de algoritmos para construir modelos paramétricos a través de diagramas de flujo. Según Anton y Tănase (2016):

Grasshopper permite a arquitectos e ingenieros a desarrollar algoritmos para diseño edificaciones basados en sistemas de reglas paramétricas. Las posibilidades de programación e interoperabilidad con otros programas de análisis de *Grasshopper* son extendidos por una serie de implementos (*plugins*) dedicados al análisis energético, tales como *GECO*, *Diva* para *Rhino* y *Ladybug*. (2016, p. 12).

A partir de la versión 6.0 del programa CAD *Rhinoceros*®, la plataforma *Grasshopper* se encuentra inmediatamente integrada y no hace falta implementarla posterior a la instalación d *Rhino3D*. El origen de esta plataforma data del año 2008, y a los pocos años se convirtió en uno de los entornos más amplios de programación visual dirigida al diseño de objetos a nivel internacional.

Grasshopper, un complemento (*plug-in*) para *Rhinoceros*®, es un editor basado en nodos desarrollado por David Rutten para Robert McNeel & Associates. Creado en el 2007 para *Rhino 4.0* -originalmente llamado como Explicit History- fue renombrado como *Grasshopper* en el 2008. A los pocos años el complemento (*plug-in*) ganó una basta comunidad de usuarios y desarrolladores que incluye estudiantes, académicos, y profesionales. (Tedeschi, 2014, p. 33).

3.1.1. Elementos sintácticos de un algoritmo.

la construcción de un algoritmo puede ser comparado con el proceso en el que se construye el enunciado en un texto. Un enunciado se encuentra compuesto por dos partes, (1) *el sujeto* y (2) *el predicado*, las cuales tienen sentido a partir de dos tipos de palabras que se desenvuelven como las más importantes de manera correspondiente a cada parte. El *sustantivo* como la palabra fundamental del sujeto determina el “algo” que, a partir del *verbo* como el elemento más importante en el predicado, se encuentra en espera de realizar una “acción”, para producir otro “algo”.

Así, es posible comprender la entrada de información (*input*) de un algoritmo como el equivalente al *sustantivo* en una oración, que a partir de una *instrucción (verbo)* generará una acción que dará como resultado una salida de información (*output*), que funciona también otro *sustantivo*, el cual se encontrará disponible para ser conectado a otra *instrucción*. En programación computacional, a los *input* y *outputs* (sustantivos) se les reconoce como *parámetros* y a las *instrucciones (verbos)* como *funciones*.

Un *parámetro* es un dato o factor que se toma como referencia para analizar o valorar un problema. En programación puede ser entendida como una *variable*, la cual es declarada en espera de asignarle un dato o valor según se requiera. Ese dato asignado o “almacenado” en parámetro (*input*) será el recurso para la *acción* o *función* que tendrá la misión de generar otro parámetro (*output*), o transformar el anterior (*input*), para que obtenga como resultado otras características (*output*).

Esta relación convergente entre parámetros y funciones van construyendo de manera concisa una secuencia de pasos que tienen como fin ejecutar o resolver una tarea. Dicha secuencia, a través de los elementos sintácticos mencionados anteriormente, puede ser expresada a través texto (*scripting, coding, etc.*), o mediante gráficos, comúnmente representados con diagramas de flujo o flujogramas (*Graphical User Interface, GUI*).

Según Tedeschi (2014), las principales propiedades de un algoritmo son: “(a) un algoritmo es un conjunto preciso (no ambiguo) de *instrucciones* definidas ... (b) Un algoritmo espera un conjunto definido de entradas de información (*inputs*) ... (c) Un algoritmo genera una salida de información (*output*) bien definida” (2014, p. 23).

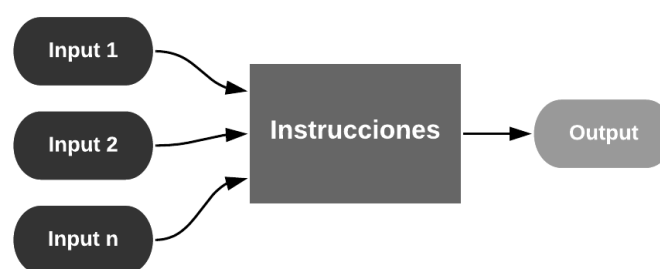


Figura 46. Representación esquemática de los elementos sintácticos de un algoritmo. Recuperado de AAD *Algorithmis-Aided Design: Parametric Strategies Using Grasshopper*, Tedeschi (2014, p. 23). Adaptación y modificación: elaboración propia (2020).

Grasshopper representa la construcción de algoritmos con diagramas de flujo, los cuales relacionan parámetros (*inputs* y *outputs*) con funciones (instrucciones) a través de la interconexión de elementos gráficos llamados “*componentes*”. Dicha conexión se hace reglamentariamente de izquierda a derecha, y es este el sentido en el que se transmite la información y se determina el flujo de trabajo.

Los *componentes* de *Grasshopper* se distinguen principalmente entre dos tipos: (1) parámetros, (2) funciones, y son el equivalente a los elementos sintácticos que construyen un algoritmo.

Los primeros tienen la función específica de almacenar información, y son clasificados de acuerdo con el tipo de dato que se le asigna (datos alfanuméricos -*strings*-, exclusivamente numéricos, ya sea fraccionarios -*float numbers*-, enteros -*integer numbers*-, o geometría 3d como puntos, líneas, curvas NURBS, superficies NURBS, sólidos primitivos, mallas -*meshes*-, etc.). Los segundos, tienen la misión de desarrollar acciones generativas o de transformación.

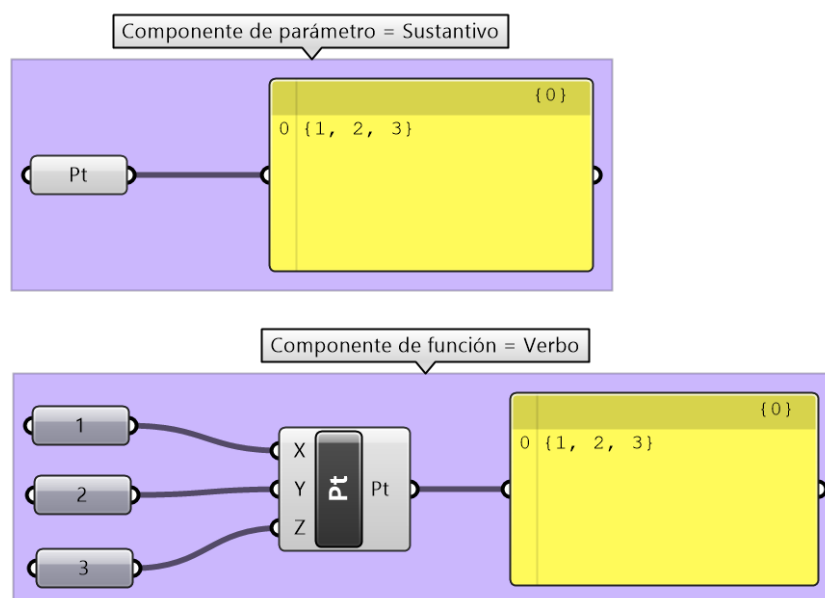


Figura 47. Elementos sintácticos de *Grasshopper*: el parámetro y la función (componente).

Elaboración propia (2020).

Los *componentes de función* de *Grasshopper*, tienen una estructura determinada por tres columnas. La del extremo izquierdo representa la cantidad de parámetros *inputs* incrustados dentro del *componente de función*, y pueden ser manipulados a través del botón *click* derecho del *mouse*, o pueden ser sustituidos al conectar un parámetro externo del mismo tipo. La columna del centro representa la acción a realizar por la función.

La del extremo derecho representa la cantidad de parámetros *outputs* incrustados y resultantes de la acción, que están en espera de ser conectados, como parámetro externo, a otro *componente de función*.

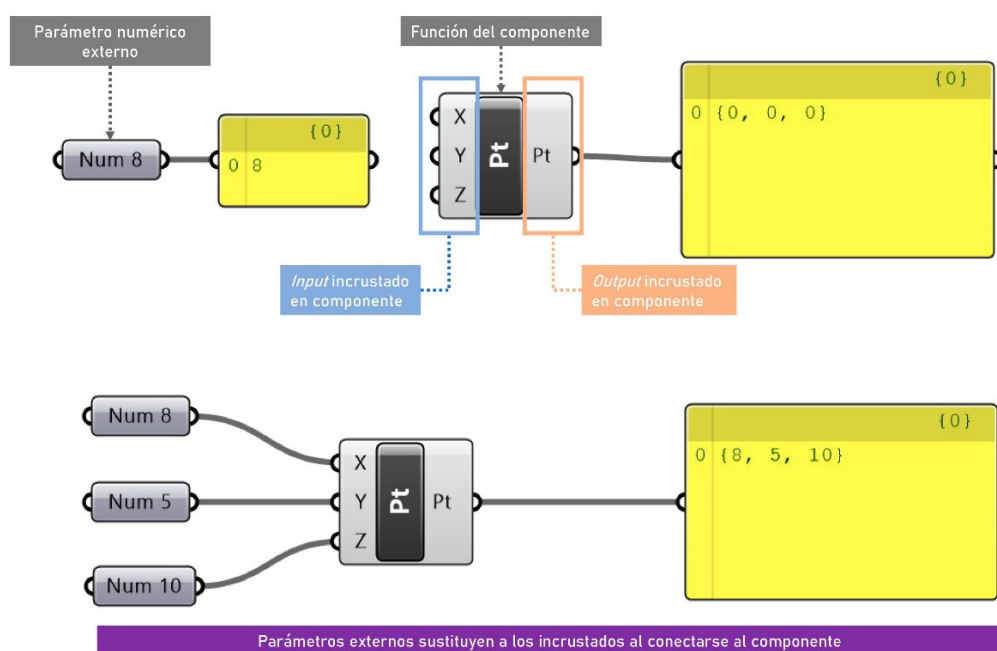


Figura 48. Estructura de componentes de parámetro y de función en *Grasshopper*. Elaboración propia (2020).

La acción del componente de función de *Grasshopper* puede producir, de acuerdo con sus características, resultados de información y previsualizaciones geométricas en tercera dimensión en el área de trabajo de *Rhinoceros®*.

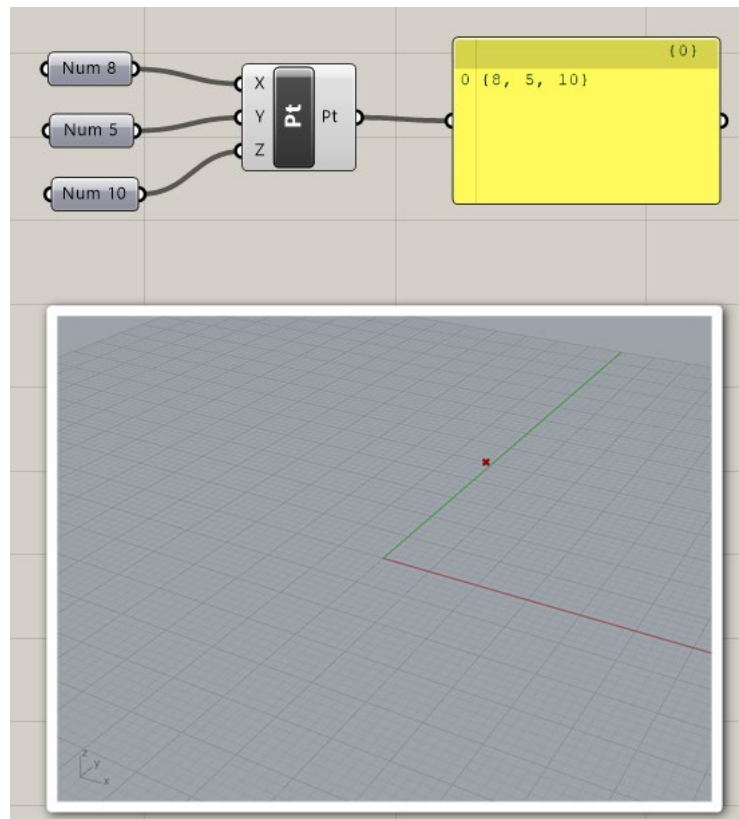


Figura 49. Previsualización geométrica de un punto generado en *Grasshopper*. Elaboración propia (2020).

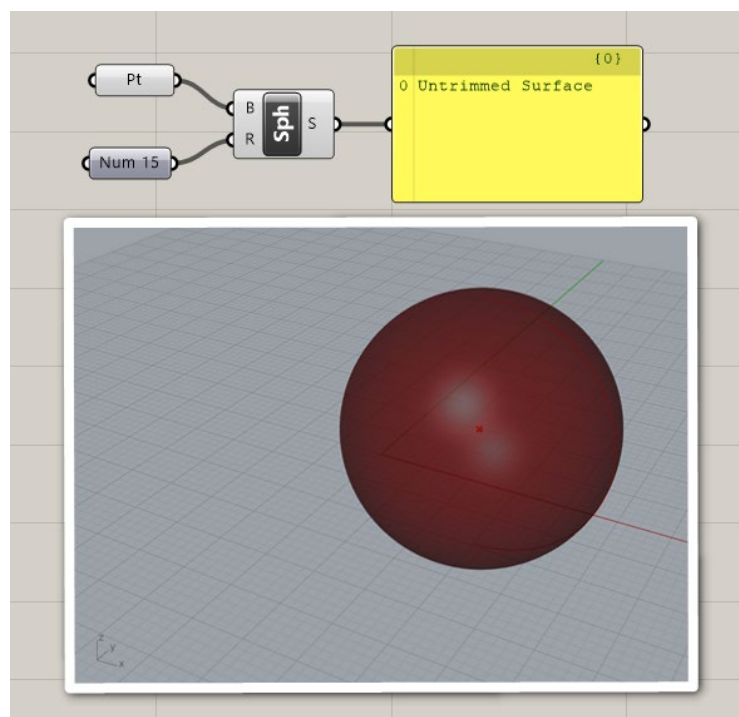


Figura 50. Previsualización geométrica de una esfera generada en *Grasshopper*. Elaboración propia (2020).

La interfaz de *Grasshopper* consiste en cuatro renglones donde se engloba todas las herramientas necesarias para la construcción de algoritmos (ver Figura 51). El renglón (A) corresponde a la barra de operaciones del sistema operativo *Windows*®, donde a través de botones se despliegan las operaciones que ayudan al desempeño adecuado al programa, dentro de estas operaciones encontraremos: abrir, guardar, generar nuevo archivo, exportar, preferencias, deshacer y rehacer, copiar, cortar, pegar información, entre otras operaciones.

El renglón (B) obedece al “listón” (*ribbon*) de botones representados a través de iconos que enlistan los componentes de *Grasshopper*, éstos se encuentran agrupados en pestañas, que contienen subgrupos de componentes. Tanto los grupos como los subgrupos se estructuran de acuerdo con la clasificación de elementos para construir algoritmos.

La primera pestaña obedece a los *componentes de parámetros* (que funcionarán como *inputs*), en el resto de las pestañas se encuentran todas las *funciones* disponibles en la plataforma. Todos los componentes se arrastran al área de trabajo (renglón D), para comenzar a trabajar.

El renglón (C) dispone herramientas de visualización, tanto para el área de trabajo de *Grasshopper*, como para la de *Rhinoceros*. Y finalmente el renglón (D) es el área de trabajo donde se construyen los algoritmos a través de los componentes que se arrastran desde el renglón (B).

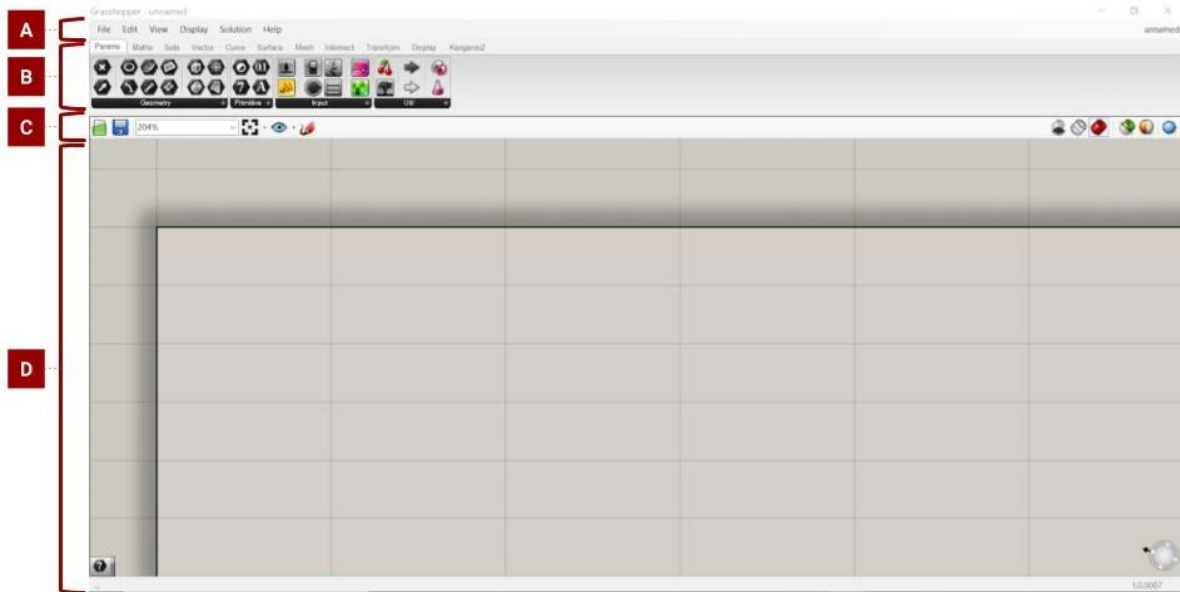


Figura 51. Elementos de Interfaz de *Grasshopper*. Elaboración propia (2020).

Los componentes dispuestos en el área de trabajo se interconectan al arrastrar con el botón *click* izquierdo del *mouse* los extremos representados con una bola blanca de los *inputs* y *outputs* de cada componente. El orden de conexión siempre debe ser de *output a input*, nunca al revés.

Una conexión sustituye a otra que se quiera conectar. Para evitar esto y poder hacer conexiones múltiples, debe arrastrarse la conexión y presionar a la vez el botón *Shift* del teclado. Para lograr desconexiones, se debe arrastrar la conexión junto al botón *Ctrl* del teclado. Otra manera de hacer desconexiones es presionar el botón derecho del *mouse* en la opción de *Disconnect*.

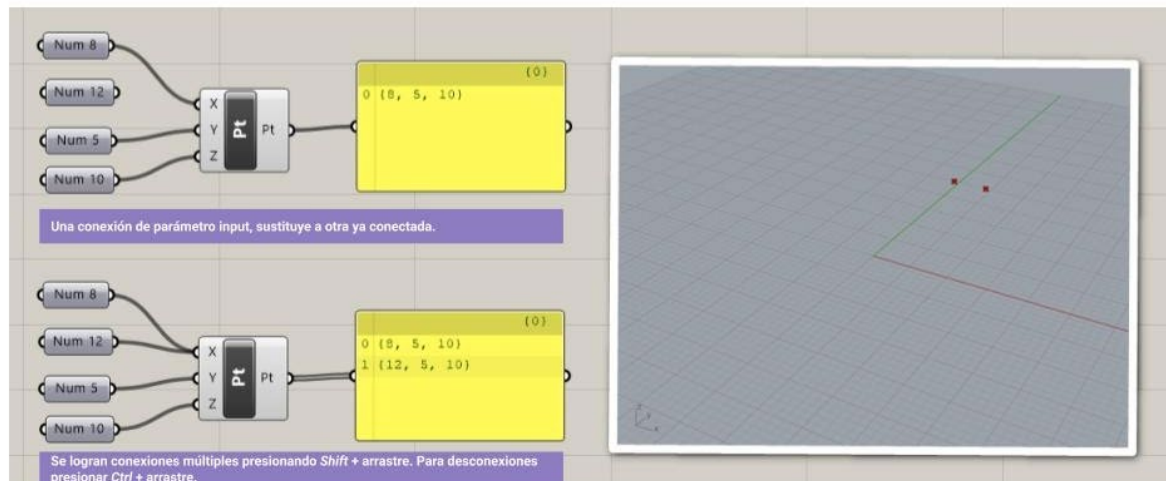


Figura 52. Métodos de conexión de componentes de parámetros y de función en *Grasshopper*.

Elaboración propia (2020).

Los colores en los componentes de *Grasshopper* representan distintos estados en la construcción de un algoritmo. Los colores existentes son los siguientes: (1) *gris*, representa que los componentes trabajan adecuadamente, ya que cuentan con la información necesaria para funcionar, también significa que arroja una previsualización de geometría en el área de trabajo de *Rhinoceros*® (ver Figura 53).

(2) *Gris oscuro*, representa un buen funcionamiento del componente, pero por su clasificación no arroja previsualización de geometría en el área de trabajo de *Rhinoceros*®, también puede significar que la previsualización (en caso de que la clasificación si arroje geometría) fue desactivada (ver Figura 54). (3) *Gris opaco*, significa que el componente no se encuentra activo, y esto sucede cuando el operador deshabilita su funcionamiento (botón derecho de *mouse* en cada componente, opción *Disable-Enable*) (ver Figura 55).

(4) *Verde*, es el color que muestra un componente seleccionado por el operador. Esta acción también cambia el color de las geometrías que se encuentren previsualizados en el área de trabajo de *Rhinoceros*® (ver Figura 56). (5) *Amarillo*, representa una advertencia para

el operador; dicha advertencia es producida por la incapacidad del componente de funcionar, debido a la falta de información (*inputs*) necesaria para lograrlo (ver Figura 57).

(6) *Rojo*, representa un error de lógica en el componente y una posible falta de funcionamiento. Dichos errores se producen por varias situaciones correspondientes al manejo de información (por ejemplo: clasificación equívoca de *inputs* establecidos, como conectar un *parámetro externo* de clasificación curva en un parámetro *input* de clasificación superficie) (ver Figura 58).

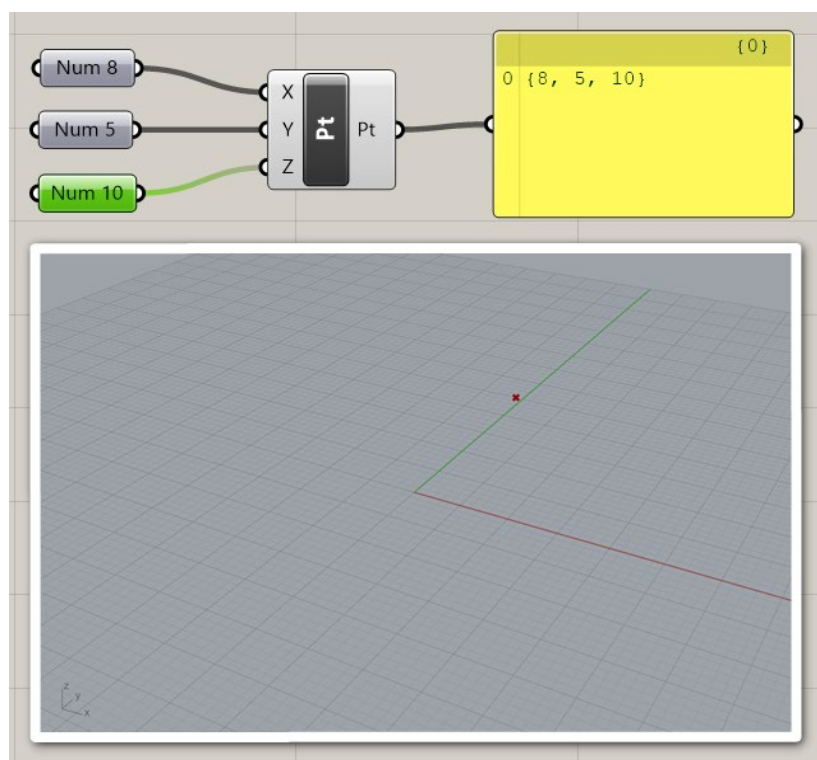


Figura 53. Estado de componente de *Grasshopper* en color gris. Elaboración propia (2020).

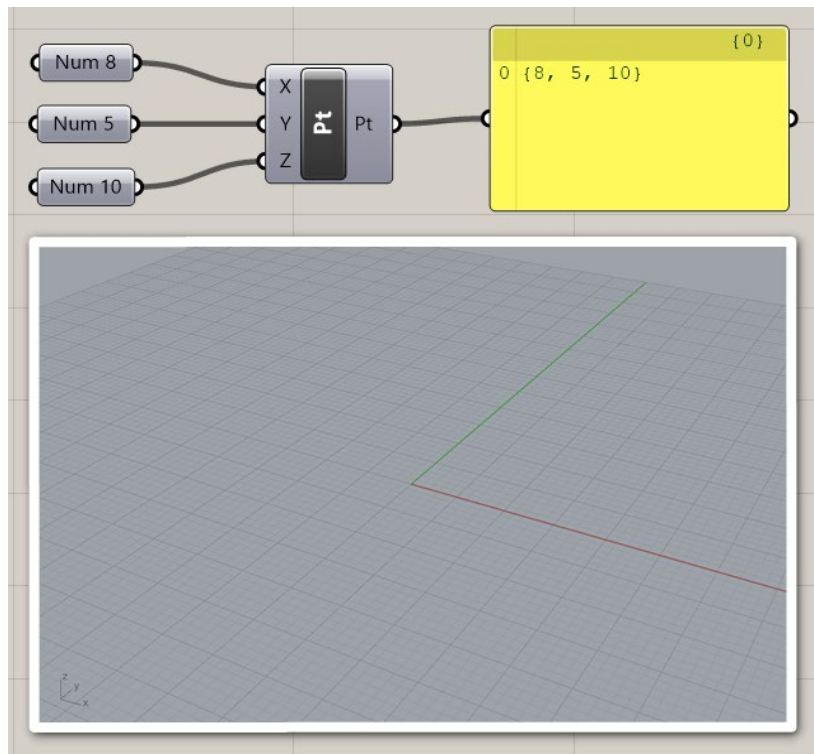


Figura 54. Estado de componente de *Grasshopper* en color gris oscuro. Elaboración propia (2020).

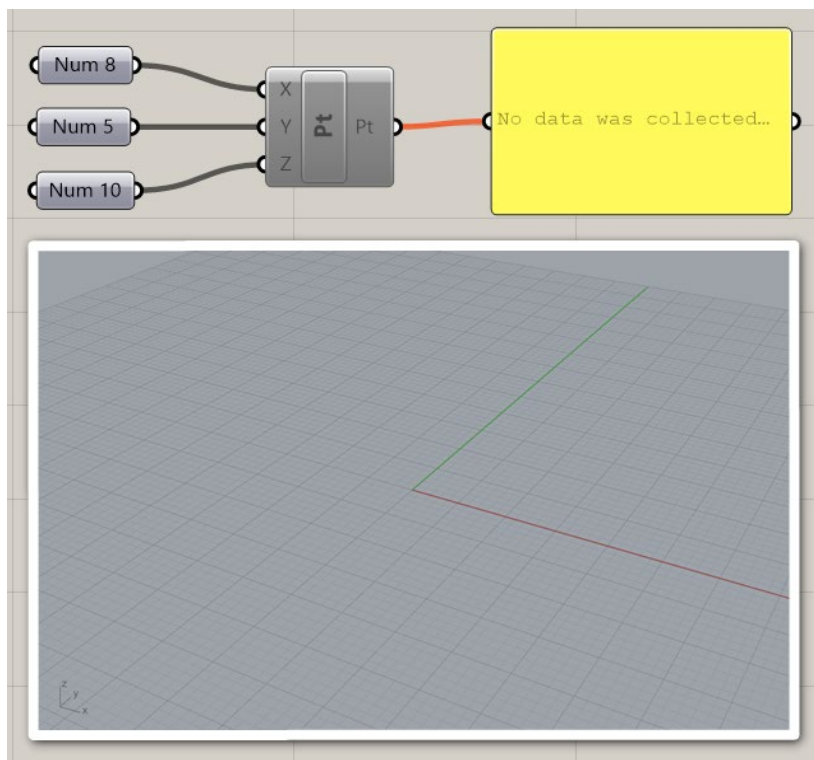


Figura 55. Estado de componente de *Grasshopper* en color gris opaco. Elaboración propia (2020).

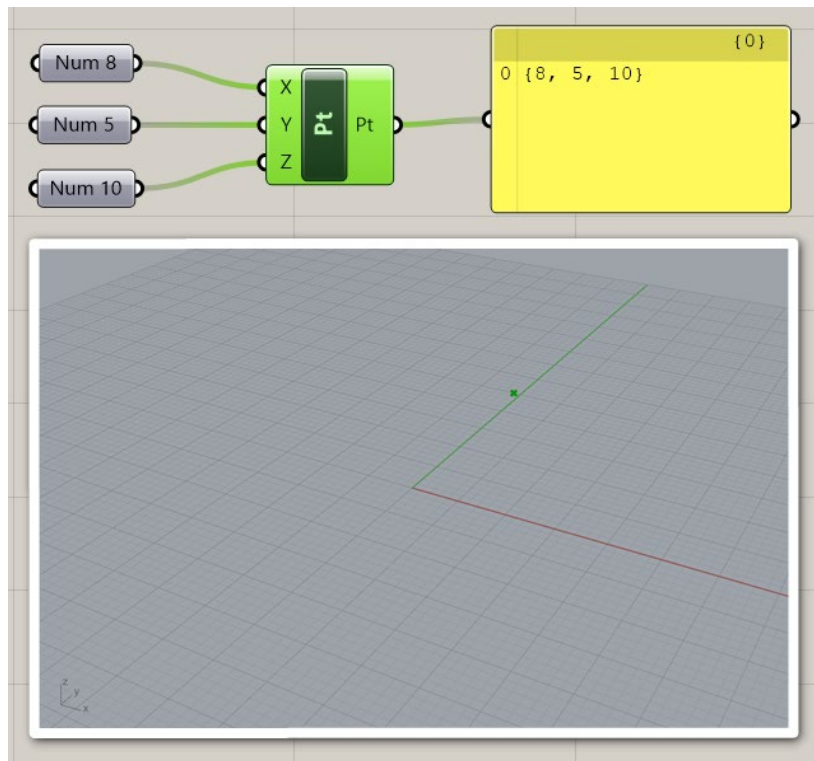


Figura 56. Estado de componente de *Grasshopper* en color verde. Elaboración propia (2020).

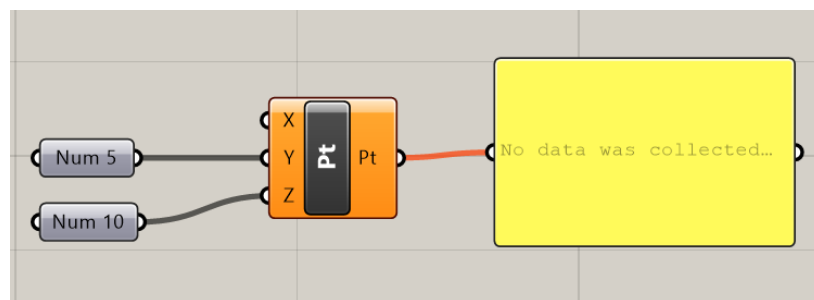


Figura 57. Estado de componente de *Grasshopper* en color amarillo. Elaboración propia (2020).

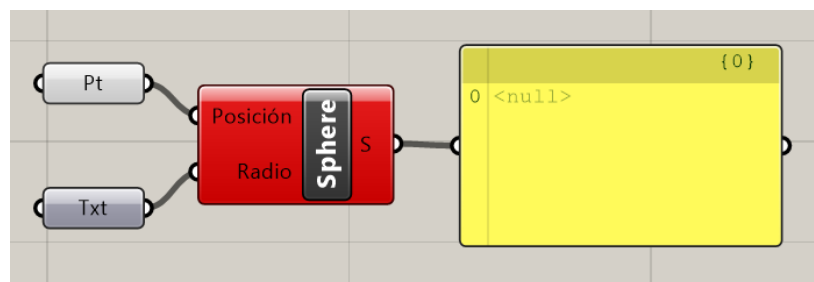


Figura 58. Estado de componente de *Grasshopper* en color rojo. Elaboración propia (2020).

3.1.2. Listas de información.

Las listas de información se producen a partir de guardar más de un dato en un componente de parámetro de *Grasshopper*. Cada lista de información etiqueta cada elemento con el valor de orden que tiene según su disposición. Ejemplo: 0.-Elemento, 1.-Elemento, 2.-Elemento, etc. (ver Figura 59).

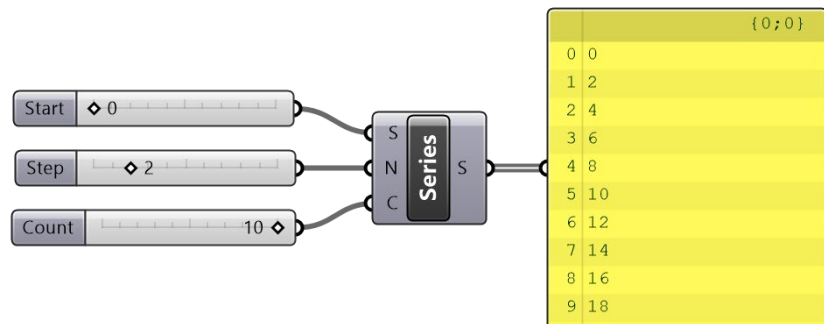


Figura 59. Ejemplo de lista de información en componente de *Grasshopper*. Elaboración propia.

Al momento de relacionarse más de una lista de información y converger en un componente de función, puede suceder que una o varias de estas tenga una longitud mayor que las otras. Existen, por tanto, listas de información más grandes que otras al momento de compararse. En *Grasshopper* la *lista larga* de datos prevalece o domina con respecto a las demás. Cuando esto sucede, los últimos elementos de las *listas de información cortas* se reproducen para acompañar y no dejar fuera los elementos extra que determinan que una lista de información sea larga.

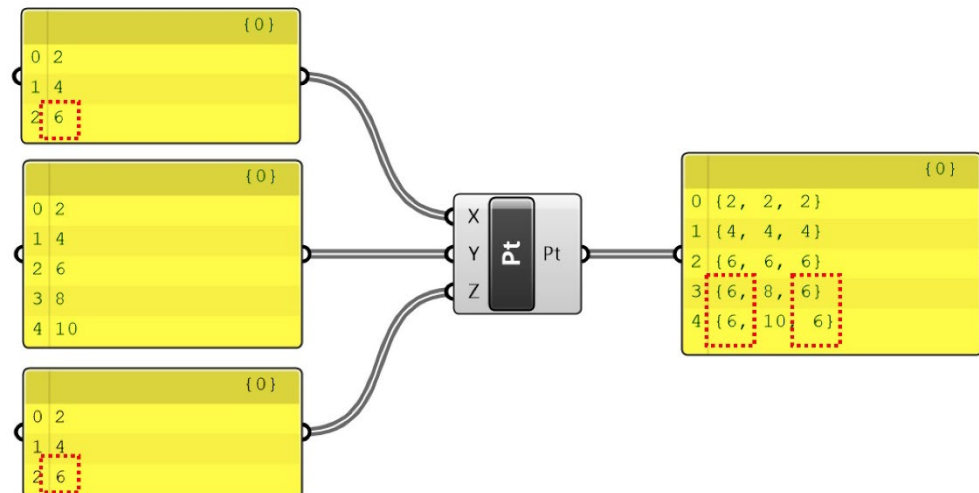


Figura 60. Relación entre listas de información convergentes en un componente de función. La lista larga prevalece sobre las más cortas. Los últimos elementos de las listas cortas se reproducen para complementar la larga. Elaboración propia (2020).

En *Grasshopper* es posible hacer que domine la lista más corta; por tanto, los elementos adicionales de la *lista larga* son eliminados u omitidos en el parámetro de *output* del componente de función. También se puede lograr que todos los elementos, que son parte de cada lista (no importa si es corta o larga), interactúen de manera cruzada. Para lograr este objetivo puede utilizarse el componente de función *Shortest List, Largest List* o *Cross Reference* como filtro entre las listas y la función a realizar.

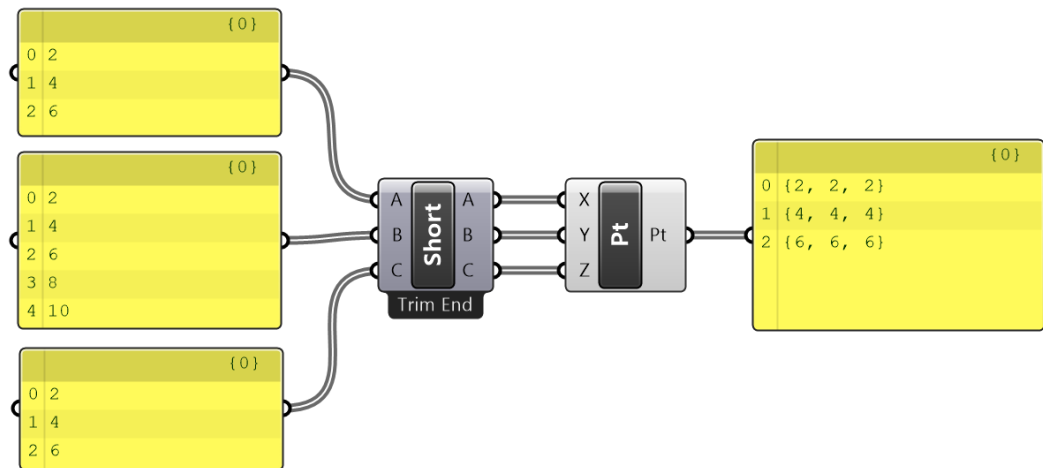


Figura 61. Ejemplo de uso de componente *Shortest List* como filtro para hacer que prevalezca la lista corta. Los valores excedentes de la lista larga son omitidos. Elaboración propia (2020).

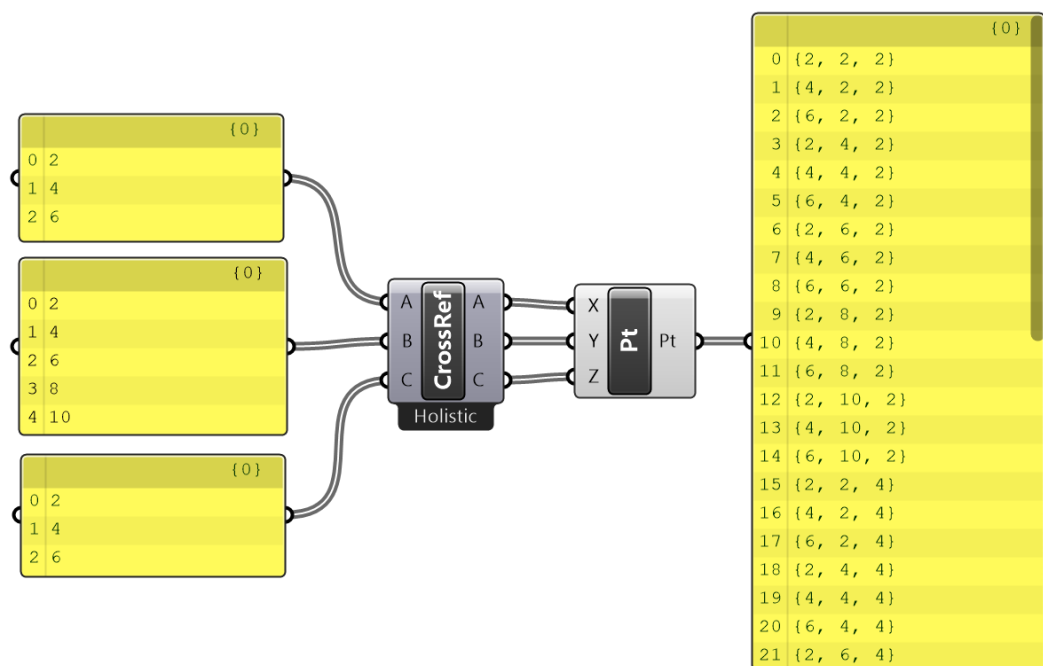


Figura 62. Ejemplo de uso de componente *Cross Reference* como filtro para hacer que todos los elementos de las listas interactúen en todas sus posibilidades de relación. Elaboración propia (2020).

De igual manera en la pestaña de *Sets* de la sección de *Ribbon* de la interfaz de *Grasshopper*, en el apartado de *List*, se pueden encontrar más componentes que sirven como filtros para modificar las listas de información, los cuales son: *Insert items*, *List Item*,

Partition List, Reverse List, Sort List, Sub List, Item Index, List Length, Replace Items, Shift List, Split List. A continuación, se muestran algunos ejemplos para ver y analizar los cambios en el flujo de información.

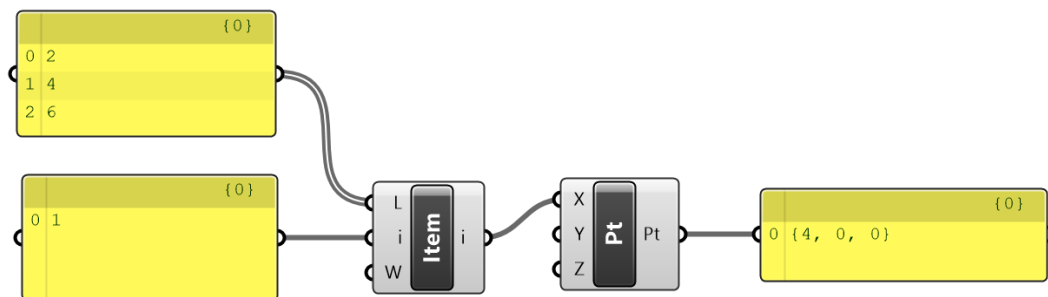


Figura 63. Ejemplo de uso de componente *List Item* como filtro para seleccionar elementos específicos de la lista. Elaboración propia (2020).

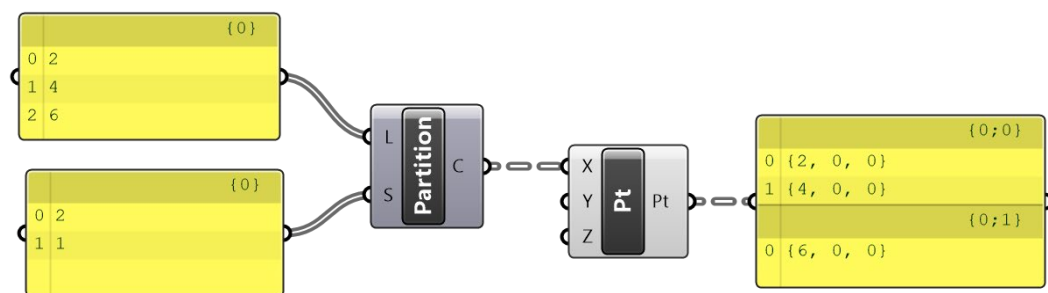


Figura 64. Ejemplo de uso de componente *Partition List* como filtro para dividir una lista de información. Elaborar propia (2020).

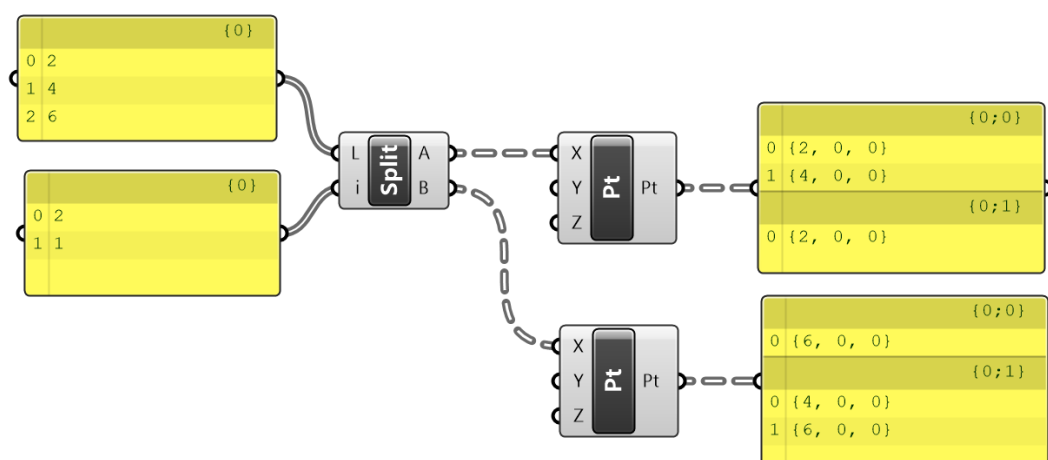


Figura 65. Ejemplo de uso de componente *Split List* como filtro para seccionar una lista de información. Elaborar propia (2020).

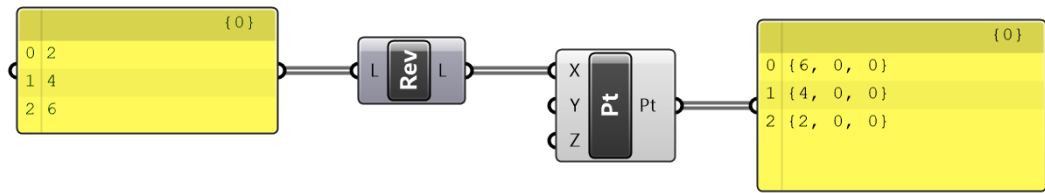


Figura 66. Ejemplo de uso de componente *Reverse List* como filtro para revertir el orden de la lista de información. Elaboración propia (2020).

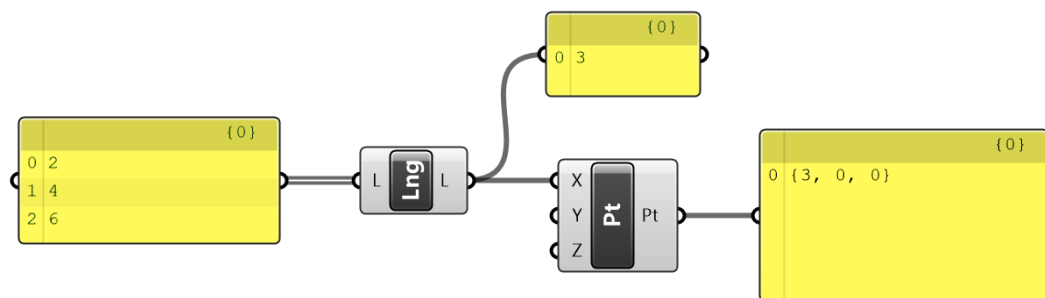


Figura 67. Ejemplo de uso de componente *List Length* como filtro para determinar la longitud de la lista de información. Elaboración propia (2020).

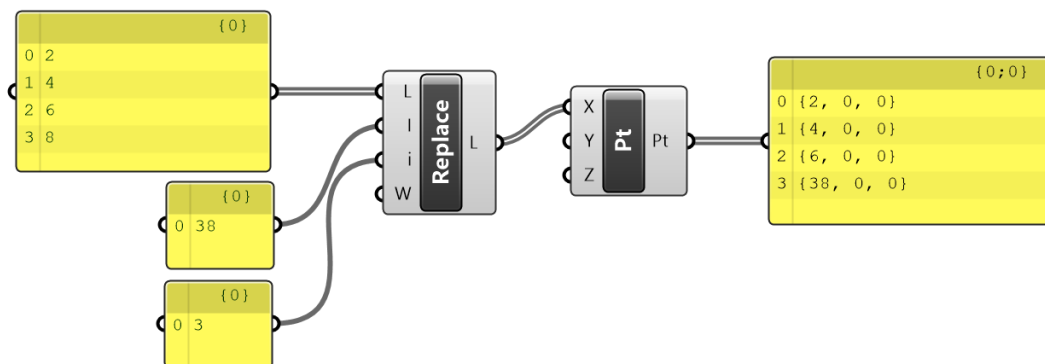


Figura 68. Ejemplo de uso de componente *Replace Items* para reemplazar algún elemento dentro de la lista de información. Elaboración propia (2020).

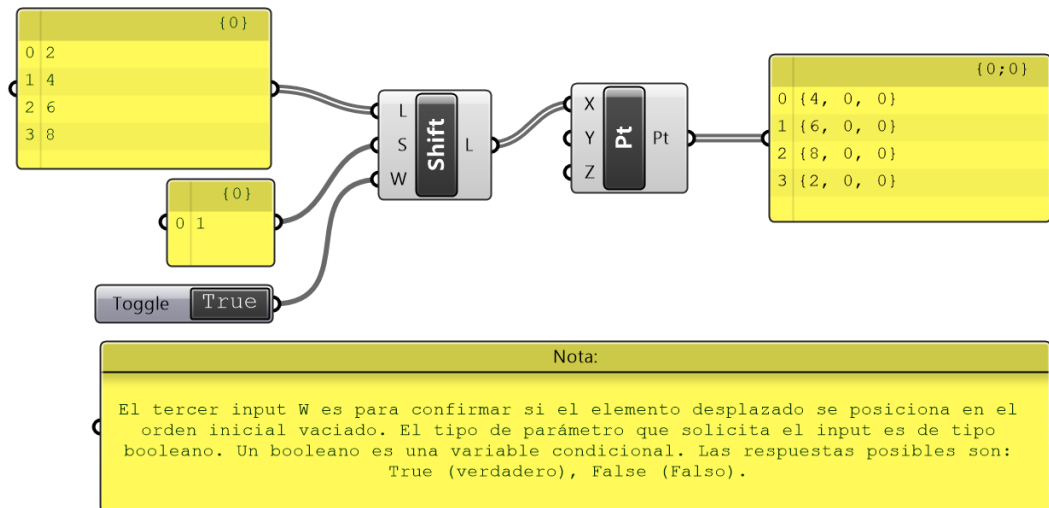


Figura 69. Ejemplo de uso de componente *Shift List* para desplazar el orden de los elementos de la lista de información. En el componente se determina la magnitud de desplazamiento y si los elementos desplazados se recuperan en las primeras posiciones vaciadas. Elaboración propia (2020).

3.1.3. Árbol de información.

El árbol de información se produce cuando un *componente (función) genera varias listas de información almacenadas en un mismo parámetro*, al suceder esto, *Grasshopper* organiza la información correspondiente a cada lista generada en una estructura jerárquica y ramificada de tipo arbóreo.

Toda lista de información, aparte de los datos contenidos y sus respectivas etiquetas de orden, tiene un nombre el cual se representa con unos valores entre corchetes en la esquina superior derecha de cada lista. La imagen de abajo (ver Figura 70) muestra como ejemplo una lista de información que tiene el siguiente nombre: {0;0}.

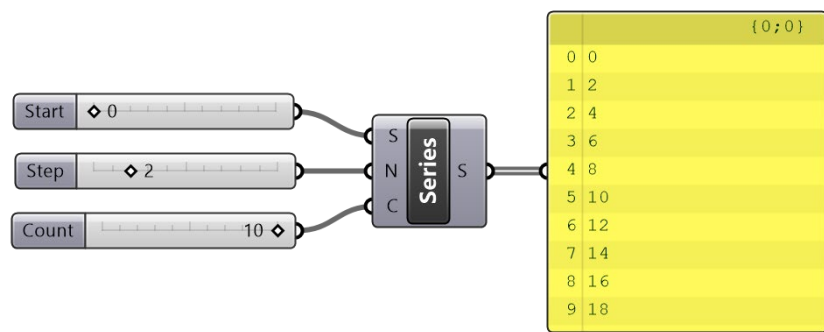


Figura 70. Ejemplo de lista de información de *Grasshopper*, como parámetro output de un componente. En la parte superior derecha se muestra el nombre de la lista: {0;0}. Elaboración propia (2020).

Si un componente (o función) genera varias listas de información y fluyen por un mismo componente, estas se representarán con sus respectivos nombres con el objetivo de diferenciar una de otra. También la conexión cambiará de apariencia, en lugar de ser continua, ahora se mostrará con apariencia discontinua, esto significa que por allí están fluyendo varias listas de información.

El siguiente ejemplo (ver Figura 71) muestra cómo el componente *Partition List* divide una lista de información en dos, al dividirse cada una de ellas se representa con su respectiva etiqueta {0;0} y {0;1} que la diferencia. Cada una de ellas posee su información correspondiente. También la imagen muestra el cambio de apariencia en la conexión a manera discontinua.

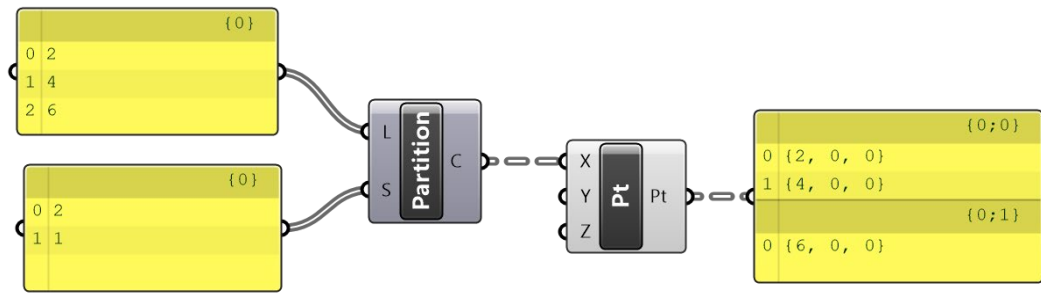


Figura 71. Ejemplo de uso de componente *Partition List* como filtro para dividir una lista de información. El componente genera dos listas de información que fluyen a través del parámetro *output*. Elaborar propia (2020).

Cuando en un parámetro fluye una sola lista de información, en *Grasshopper* se le llama a esta: “información plana” o *Flat, Flatten*. Cuando en un parámetro fluyen varias listas de información, en *Grasshopper* se le llama a esta: “información engendrada” o *Graft*. Significa que las distintas listas se encuentran organizadas de manera ramificada; conforme más componentes engendran más listas de información, se estructuran ramificaciones de información de manera consecutiva y cada vez más complejas. La estructura en la que se organiza el árbol es la siguiente (ver Figura 72):

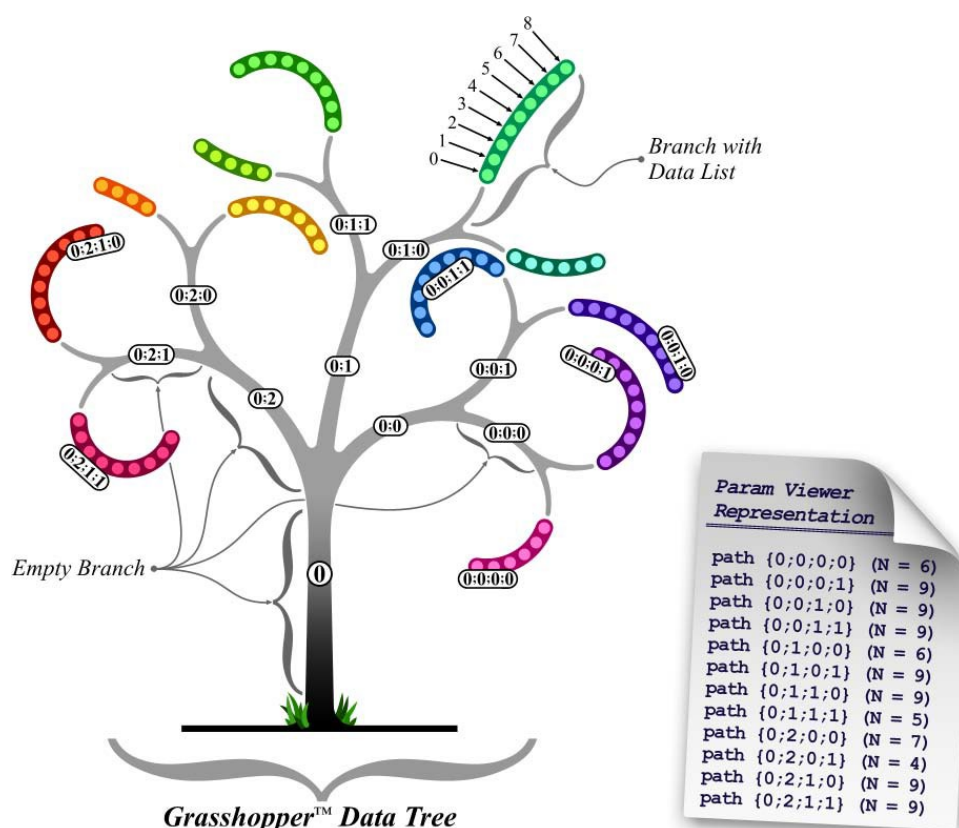


Figura 72. Estructura del árbol de información en Grasshopper. Recuperado de *Grasshopper Primer: for version 0.6.0007 - Spanish Edition*, Payne e Issa (2009, p. 36).

Cada etiqueta o nombre de lista, presenta valores entre corchetes los cuales representan el origen de la información, así como el nivel generacional que tiene. El nivel generacional va de izquierda a derecha. Por ejemplo, la siguiente tabla (ver Tabla 2) describe el nombre de la siguiente etiqueta: {0,1,0}

Tabla 2

Descripción generacional del nombre de lista de información {0,1,0}.

Número	Descripción
1	El <i>cer</i> de la izquierda es el remanente de la etiqueta de origen (el tronco) y es por tanto el nombre de la generación más vieja.
2	El valor <i>uno de la parte media</i> , es una de las ramas (o lista) que posteriormente un componente generó a partir del tronco.
3	Por último, el valor <i>cer</i> de la derecha es la generación más nueva, esto significa que es una lista consecuente de la <i>rama con etiqueta uno</i> , en la cual otro componente generó varias listas más.
4	La estructura puede crecer de manera infinita conforme el algoritmo crezca.

Número	Descripción
5	Cada rama, o lista de información, naturalmente contiene varios datos almacenados. Todo esto fluye por un solo componente.
6	Cada posición extra hacia la derecha representa una generación más añadida. Por ejemplo, si se compara una rama llamada {0;0;0} con otra llamada {0;0;0;0}; significa que, en la última, más componentes intervinieron en desarrollar más generaciones de listas que en la primera rama mostrada. Por lo que, la primera lista y la segunda no son iguales.

Nota. Elaboración propia (2020)

Es importante tener en cuenta que, en un algoritmo una rama (o lista de información) no puede interactuar con otra que tenga diferente, generación, etiqueta o nombre. Por ejemplo: la rama llamada {0,1} no puede interactuar con otra llamada {0,0,1} porque los elementos que contiene cada lista no pueden vincularse unos a otros. *En Grasshopper solo pueden interactuar los elementos que son parte de una misma lista.*

Por tanto, es importante siempre verificar, si en el algoritmo que se realiza la información fluye *Flatten* o *Graft*, ya que de no verificar cómo la información progresa, pueden generarse errores lógicos. En caso de que la información en un algoritmo vaya *Graft* y se quiera utilizar dos parámetros en un componente, es importante también revisar el nombre y generación de la lista para verificar si pueden interactuar entre sí o no.

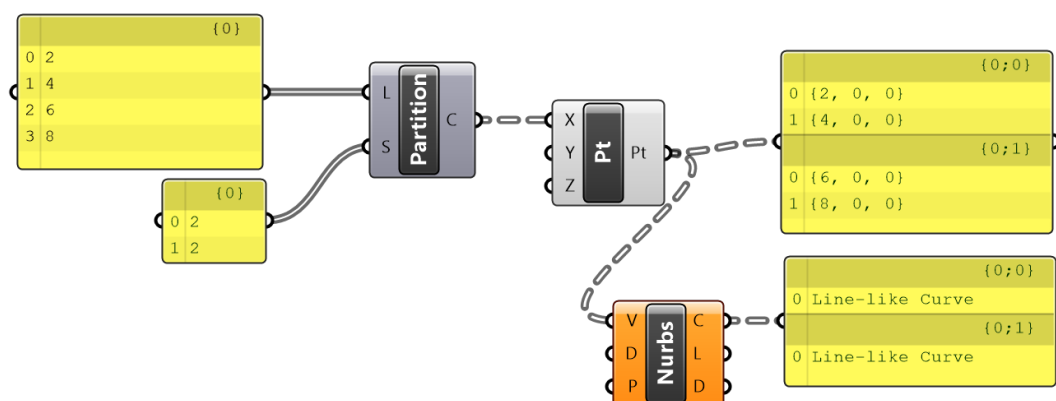


Figura 73. Ejemplo de árbol de información estructurado por dos listas de datos (ramas). Los elementos de cada lista (rama) solo pueden interactuar entre sí. Los elementos de diferentes listas (ramas) no pueden interactuar. Elaboración propia (2020).

La imagen superior, muestra el componente *Partition List* que divide una lista de información en dos listas de datos que funcionan como parámetros y se conectan en el *Input X* del componente *Construct Point* para generar puntos. Al introducir el parámetro *output* de los puntos generados (cuatro divididos en dos ramas o listas llamadas $\{0;0\}$ y $\{0;1\}$) en el componente *Nurbs Curve* para generar una curva, es posible observar que el componente se queda en color amarillo, esto es porque no cuenta con los datos necesarios para hacer una curva, ya que se requieren al menos tres puntos para poder generarla.

Aunque se tiene cuatro puntos, no es posible hacer una sola curva porque no puede interactuar los puntos de la rama $\{0;0\}$ (que son dos) con los de la rama $\{0;1\}$ que son dos más. Así que el componente genera dos líneas, una que une los dos elementos de la primera rama, y otra que une los dos elementos de la segunda.

Es posible en *Grasshopper* cambiar la condición de un parámetro *Graft* a *Flatten*, es decir romper las ramas para que todos los elementos que contengan sean parte de una sola lista, y se combinen. Para ello, se debe utilizar el componente *Flatten Tree* como filtro entre la lista de información y el componente posterior. *Flatten Tree*, así como otras utilidades exclusivas para manipular el árbol de información pueden ser encontradas en la pestaña de *Sets / apartado Tree / de Grasshopper* (ver Figura 74).

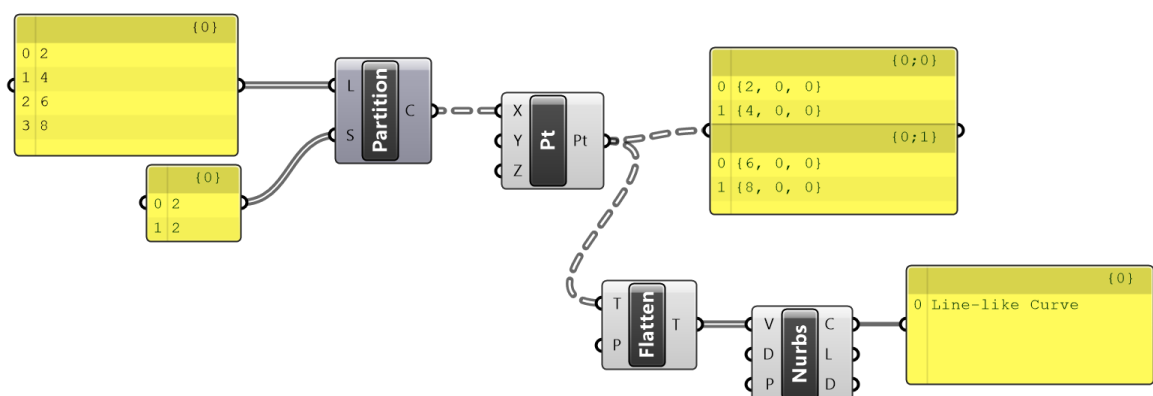


Figura 74. Ejemplo de uso de componente *Flatten Tree* como filtro para romper los grupos de listas de datos en una sola que contenga todos los parámetros. Elaborar propia (2020).

Es posible en *Grasshopper* cambiar la condición de un parámetro *Flatten* a *Graft*, es decir posicionar en ramas los elementos que anteriormente se encuentran combinados en una sola. Para ello, se debe utilizar el componente *Graft Tree* como filtro entre la lista de información y el componente posterior. *Graft Tree*, así como otras utilidades exclusivas para manipular el árbol de información pueden ser encontradas en la pestaña de *Sets / apartado Tree / de Grasshopper* (ver Figura 75).

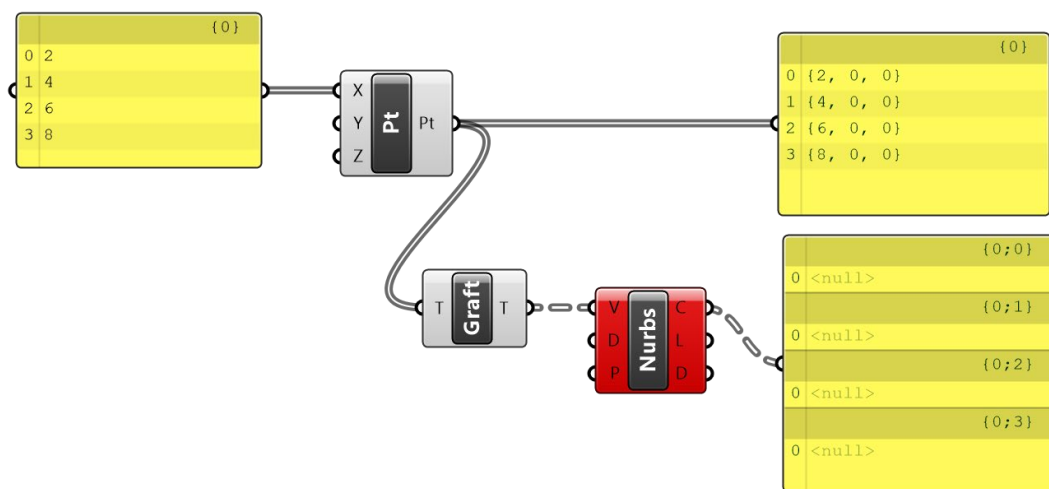


Figura 75. Ejemplo de uso de componente *Graft Tree* como filtro para agrupar los datos en varias listas (ramas) que contengan todos los parámetros. Elaborar propia (2020).

En el ejemplo de la imagen superior, una lista de información llamada $\{0\}$ de cuatro valores, produce puntos a partir del componente *Construct Points*. Posteriormente al utilizar el componente *Graft* la lista se parte en cuatro ramas que contiene un elemento cada una de ellas ($\{0;0\}$, $\{0;1\}$, $\{0;2\}$ y $\{0;3\}$). El valor de la izquierda en la etiqueta de cada rama representa su origen, es decir, de la lista que proviene que se llama $\{0\}$ y el valor de la derecha, representa la nueva generación o rama.

Después, al ser conectado el parámetro de salida (*output T*) del componente de *Graft*, al parámetro de entrada (*input V*) del componente de *Nurbs Curve*, éste último se pone en

rojo porque, aunque fluyen 4 puntos en la lista de información, cada uno de ellos pertenece a rama diferente, por ello no es posible generar curva alguna con un punto por lista.

Otras opciones que ofrece el apartado *Tree* de la pestaña *Sets* en *Grasshopper*, son las de utilizar componentes como filtro para seleccionar en un algoritmo una lista o rama completa con el componente llamado *Tree Branch* (ver Figura 76), o seleccionar un elemento de una rama específica con *Tree Item* (ver Figura 77).

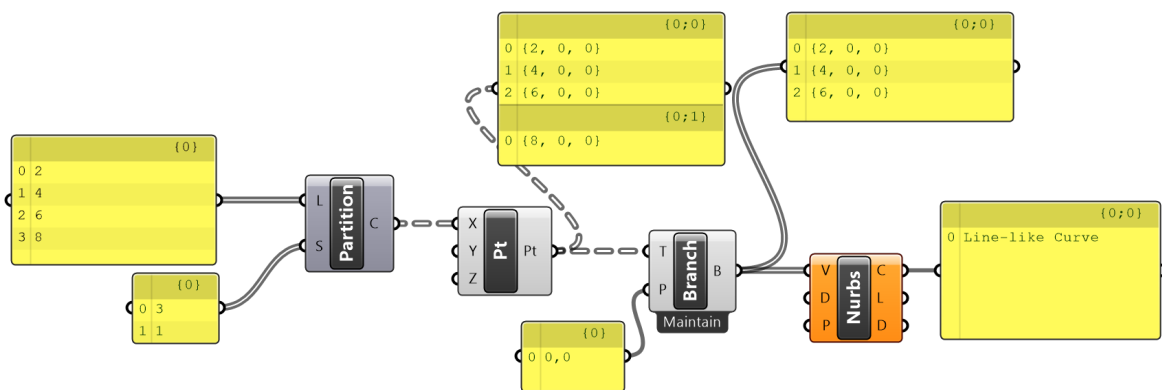


Figura 76. Ejemplo de uso de componente *Tree Branch* como filtro para seleccionar una lista (rama) de un conjunto de ramas (árbol) que fluyan a través de un componente. Elaborar propia (2020).

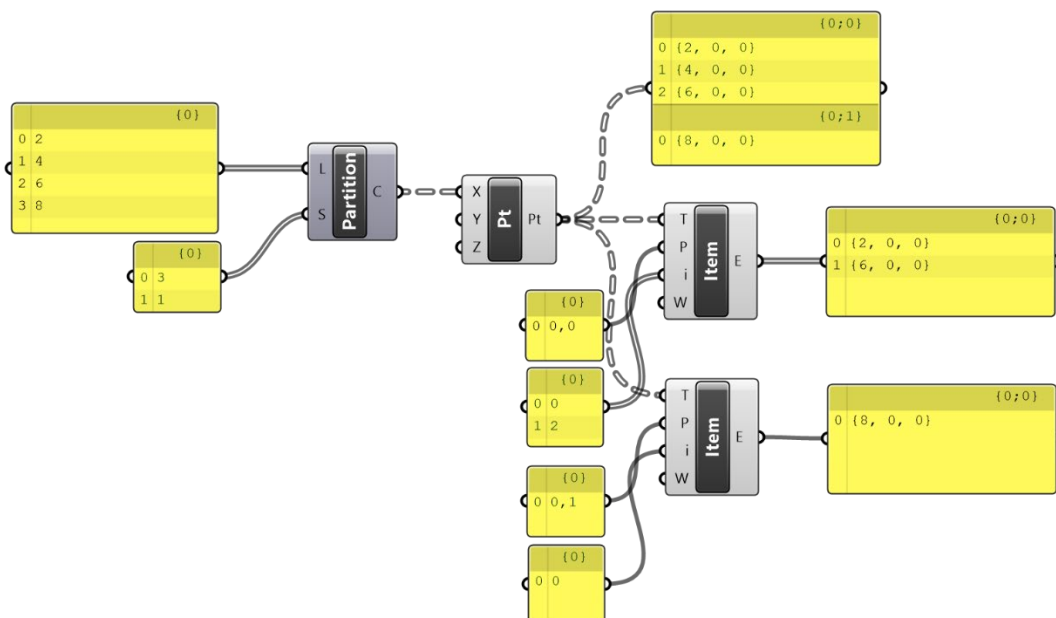


Figura 77. Ejemplo de uso de componente *Tree Item* como filtro para seleccionar uno o varios datos específicos de una lista (rama), en un conjunto de ramas (árbol) que fluyan a través de un componente. Elaborar propia (2020).

3.2. Estrategias algorítmicas básicas en *Grasshopper* para aplicación en diseño arquitectónico.

El presente apartado tiene como objetivo mostrar algunas estrategias básicas algorítmicas en *Grasshopper* que generan sistemas geométricos y topológicos con utilidad en el proceso de diseño de un proyecto arquitectónico.

La descripción algorítmica es mostrada a través de una descripción introductoria al objetivo de cada estrategia, seguido de una tabla con los pasos a seguir para la elaboración del algoritmo en cuestión y finalmente se mostrará un ejemplo en imagen de la geometría o sistema topológico producido por el algoritmo de *Grasshopper* completo.

Las estrategias que se describirán en esta sección del trabajo de investigación son: (1) geometría múltiple en superficie subdividida, (2) elementos atractores, (3) elemento atractor en superficie subdividida, (4) retícula en diagonal (*Diagrid*) a partir de una superficie, (5) superficie generada a partir de función matemática, (6) superficie generada a partir de puntos aleatorios, (7) superficie generada a partir de una función matemática gráfica, (8) ejemplo de diseño paramétrico de envolvente de estadio deportivo, (9) ejemplo de diseño paramétrico de rascacielos.

3.2.1. Geometría múltiple en superficie subdividida.

El objetivo del presente algoritmo es desarrollar una superficie, subdividirla y convertir cada subdivisión en una caja. Posteriormente se dibujará una geometría la cual tendrá como fin multiplicarse constantemente en la superficie, se sustituirá cada caja producto de la subdivisión por la geometría realizada (ver Figura 78 y Tabla 3).

Paso	Descripción
7	Posteriormente convertir cada subdivisión de superficie generada en una caja utilizar el componente de función <i>Surface Box</i> , para este objetivo se recurre a las divisiones de dominio anteriormente generadas, y se asigna una altura a cada caja con un parámetro de número o con un <i>Number Slider</i> .
8	A partir del parámetro de geometría donde se referenció el objeto dibujado en <i>Rhinoceros</i> ®, generar una caja de referencia con el componente de función <i>Bounding Box</i> , con el objetivo de parametrizar el ancho, largo y alto del objeto.
9	Sustituir las cajas generadas a partir de la subdivisión de superficie, por el objeto referenciado en el parámetro de geometría con el componente de función <i>Box Morph</i> . Se utilizará como recurso el parámetro de caja de referencia, el cual corresponde al output del <i>Bounding Box</i> , así como las cajas objetivo, que son las generadas por <i>Surface Box</i> .

Nota. Elaboración propia (2020).

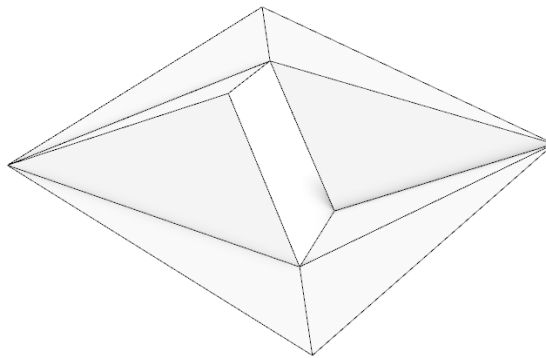


Figura 79. Geometría inicial para multiplicar en superficie a través de algoritmo de *Grasshopper*.

Elaboración Propia (2020).

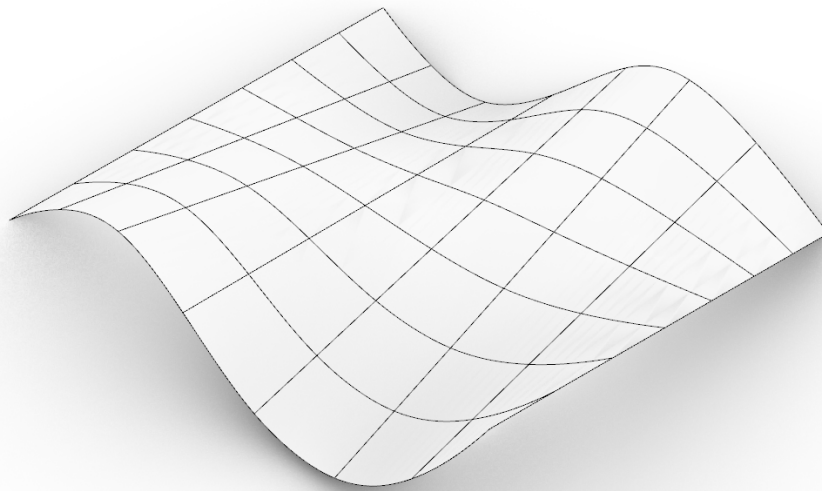


Figura 80. Superficie inicial como referencia para aplicación en algoritmo de *Grasshopper*.

Elaboración propia (2020).

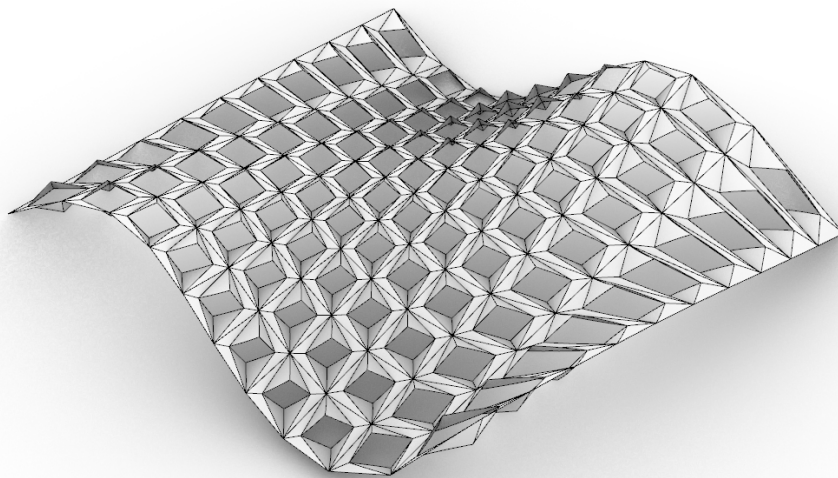


Figura 81. Resultado de algoritmo de *Grasshopper*: “geometría múltiple en superficie subdividida”.

Elaboración propia (2020).

3.2.2. Elementos atractores.

El objetivo del presente algoritmo es diferenciar las propiedades de escala de un conjunto de objetos con relación directamente proporcional a la lejanía de éstos con respecto a otra geometría, la cual funcionará como referencia y será nombrada como “elemento atractor” (ver Figuras 82-84 y Tabla 4).

Este ejercicio arquitectónicamente puede ser útil, tanto para generar membranas arquitectónicas como celosías. En escalas más grandes puede diferenciar alturas de edificios a partir de algún límite que funcione como elemento atractor.

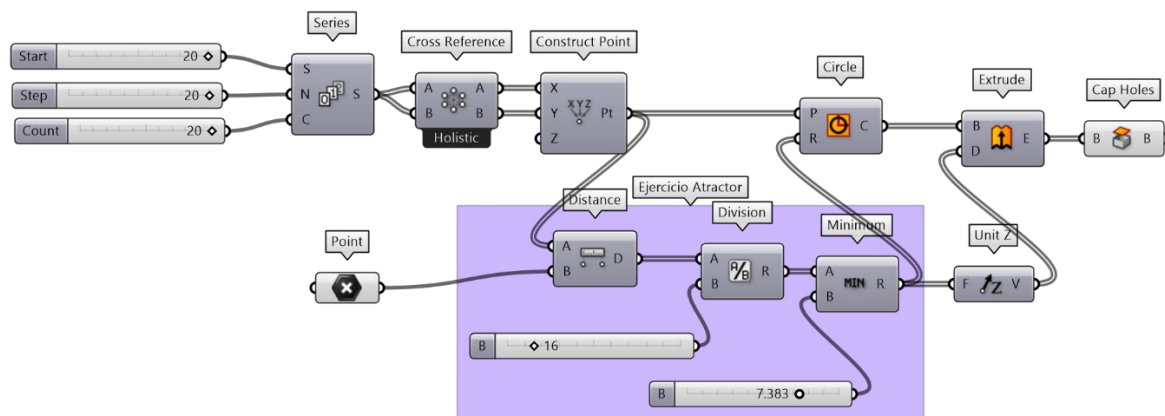


Figura 82. Algoritmo de Grasshopper para estrategia básica “elementos atractores”. El ejemplo muestra el parámetro *Point* como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).

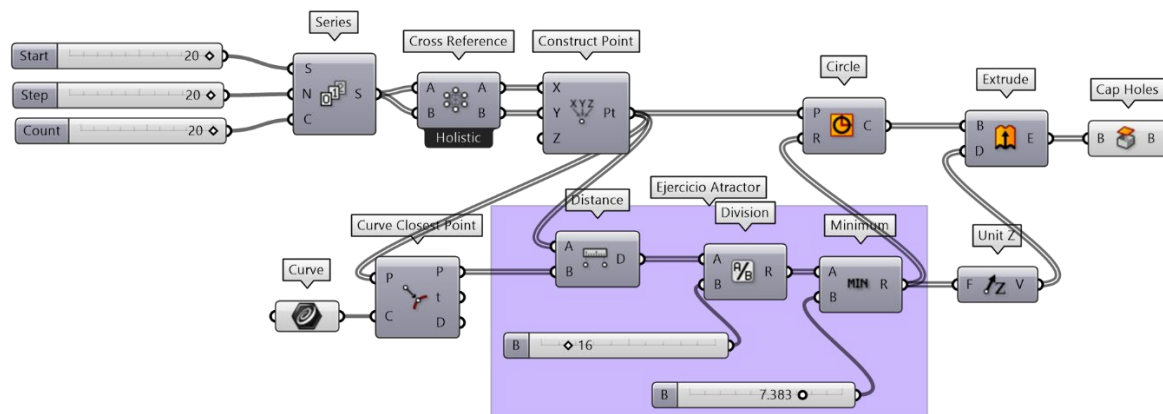


Figura 83. Algoritmo de *Grasshopper* para estrategia básica “elementos atractores”. El ejemplo muestra el parámetro *Curve* como elemento atractor y círculos como objetos a multiplicar.

Elaboración propia (2020).

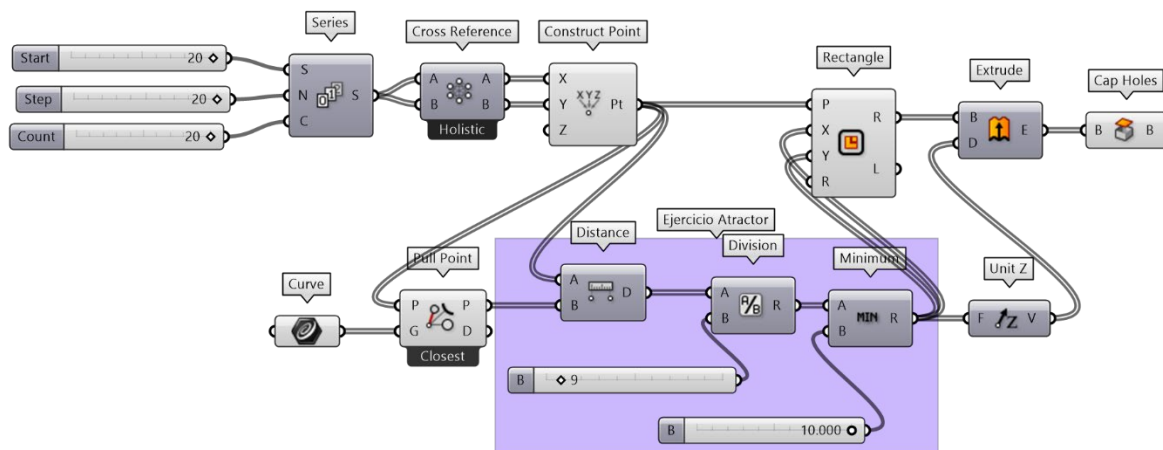


Figura 84. Algoritmo de *Grasshopper* para estrategia básica “elementos atractores”. El ejemplo muestra varios parámetros *Curve* como elementos atractores y rectángulos como objetos a multiplicar.

Elaboración propia (2020).

Tabla 4

Tabla descriptiva para construcción de algoritmo: elementos atractores.

Sección	Paso	Descripción
1. Generar una retícula de puntos a partir de una lista de información	1.1	Extraer componente de función de <i>Series</i> para desarrollar una serie numérica. Establecer el número inicial, el tamaño de paso y cantidad de pasos correspondientes para así generar la serie.
	1.2	Construir puntos con el componente de función <i>Construct Points</i> .

Sección	Paso	Descripción
	1.3	Utilizar la lista de información generada por la serie numérica, para generar puntos en las coordenadas X e Y .
	1.4	Utilizar el filtro de referencia cruzada (<i>Cross Reference</i>), para que las listas de información que entran en juego se relacionen completamente; una variable con todas las variables posibles.
2. Generar círculos a partir de los puntos desarrollados por la retícula.	2.1	Extraer componente de función <i>Circle</i> para desarrollar círculos a partir de los parámetros de círculos existentes.
	2.2	Determinar un radio correspondiente al implementar un parámetro <i>Number</i> o un parámetro dinámico <i>Number Slider</i> en el <i>input R</i> .
3. Sustituir el valor único establecido como radio (ya sea con parámetro <i>Number</i> o con <i>Number Slider</i>), por una relación de distancias a partir de un <i>elemento atractor</i> .	3.1	Dibujar elemento atractor: dibujar directamente en <i>Rhinoceros®</i> un punto.
	3.2	Referenciar el punto dibujado en <i>Rhinoceros®</i> con un parámetro <i>Point</i> .
	3.3	Tomar distancias correspondientes al elemento atractor (punto referenciado) con respecto a los puntos de la retícula. Utilizar función <i>Distance</i> .
	3.4	Dividir con función <i>Division</i> los valores de distancias resultantes del componente de función <i>Distance</i> en X cantidad de veces.
	3.5	Conectar la salida (<i>output</i>) de las distancias divididas en la entrada (<i>input</i>) de radio de la función <i>Circle</i> .
	3.6	Jugar con el <i>punto o elemento atractor</i> , moviéndolo directamente en <i>Rhinoceros®</i> , así como los valores de división (de las distancias).
4. Desarrollar mayor control en el incremento de radios, utilizar un filtro que controle los valores numéricos. El objetivo es controlar su incremento hasta cierto parámetro.	4.1	Utilizar función <i>Minimum</i> para ser utilizado como un filtro numérico. La función determinará hasta qué valor numérico los radios se incrementarán, al comparar una lista de números (conjunto de distancias divididas) con respecto a un número mínimo que funcionará como tope de incremento.
	4.2	Determinar el número mínimo que funcionará como parámetro tope de incremento. Utilizar un parámetro <i>Number</i> o un parámetro dinámico <i>Number Slider</i> y conectar en <i>input B</i> de función <i>Minimum</i> .
5. Agrupar todo aquello que sirve para determinar los radios.	5.1	Seleccionar <i>Distance</i> , <i>Division</i> , <i>Minimum</i> , presionar botón de <i>scroll</i> de <i>mouse</i> y seleccionar icono de <i>Group</i> .
	5.2	Establecer el nombre de “Ejercicio atractor” al grupo, presionar botón derecho del <i>mouse</i> en el recuadro del grupo, y establecer el nombre en el primer renglón.
6. Sustituir el punto como elemento atractor por una curva atractor.	6.1	Eliminar el parámetro de <i>Point</i> que funciona como punto atractor.
	6.2	Eliminar punto dibujado directamente en <i>Rhinoceros®</i> .
	6.3	Dibujar directamente en <i>Rhinoceros®</i> una curva.
	6.4	Extraer un parámetro <i>Curve</i> y referenciar la curva dibujada en <i>Rhinoceros®</i> .
	6.5	Extraer función de <i>CurveCP</i> (<i>Curve Closest Point</i>) para evaluar los puntos más cercanos a la nueva curva atractor. Los recursos para la función son el <i>output</i> de los puntos que producen la retícula y la curva atractor.
	6.6	Conectar el <i>output</i> de puntos de <i>CurveCP</i> en el <i>input</i> de <i>Distance</i> , donde estaba anteriormente conectado el punto atractor que ya no existe.

Sección	Paso	Descripción
	6.7	Jugar con la <i>curva atractora</i> y mover; también con los parámetros de la división de distancias y el valor mínimo que funciona como límite.
7. Utilizar dos o más curvas atractores.	7.1	Borrar los datos referenciados en el parámetro <i>Curve</i> donde estaba referenciada la curva dibujada en <i>Rhinoceros®</i> (<i>clear values</i>).
	7.2	Dibujar otra curva directamente en <i>Rhinoceros®</i> (ahora se tienen dos).
	7.3	Re-asignar en el parámetro <i>Curve</i> existente las dos curvas.
	7.4	Sustituir la función <i>Curve Closest Point</i> por <i>Pull Point</i> .
	7.5	Conectar los <i>inputs</i> correspondientes.
	7.6	Jugar con ambas curvas y moverlas.
8. Diferenciar otros objetos con los resultados obtenidos por el ejercicio atractor.	8.1	Es posible utilizar los componentes de función que desarrollan el ejercicio atractor (el <i>output Minimum</i>), como recurso para diferenciar otros valores de otras funciones. Por ejemplo, la altura de un conjunto de cilindros.
9. Invertir lógica. Decremento de escala de objetos.	9.1	Si se quiere invertir la lógica; es decir, entre más lejos se encuentren los objetos de la retícula con respecto a los elementos atractores, menor será la escala de estos, se puede lograr si se invierte el parámetro conectado como divisor con el parámetro conectado como dividendo en la función de <i>Division</i> .

Nota. Elaboración propia (2020).

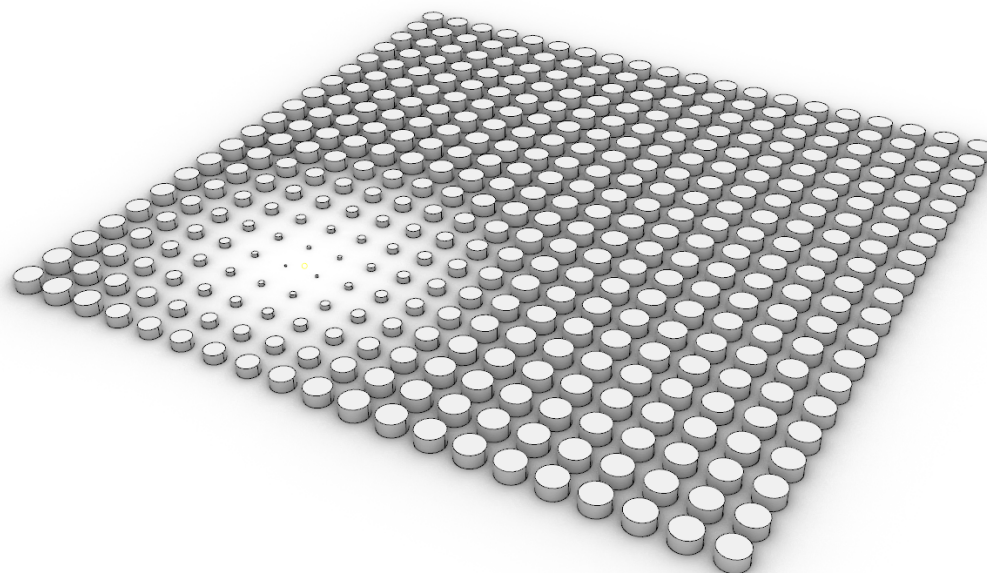


Figura 85. Resultado de algoritmo de *Grasshopper*: “elementos atractores”. El ejemplo es producto del parámetro *Point* como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).

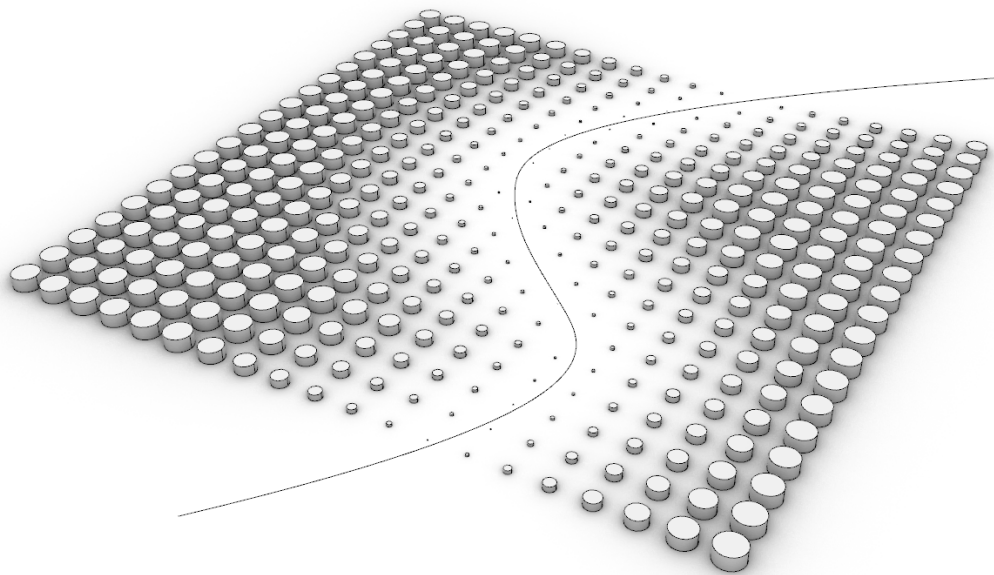


Figura 86. Resultado de algoritmo de *Grasshopper*: “elementos atractores”. El ejemplo es producto del parámetro *Curve* como elemento atractor y círculos como objetos a multiplicar. Elaboración propia (2020).

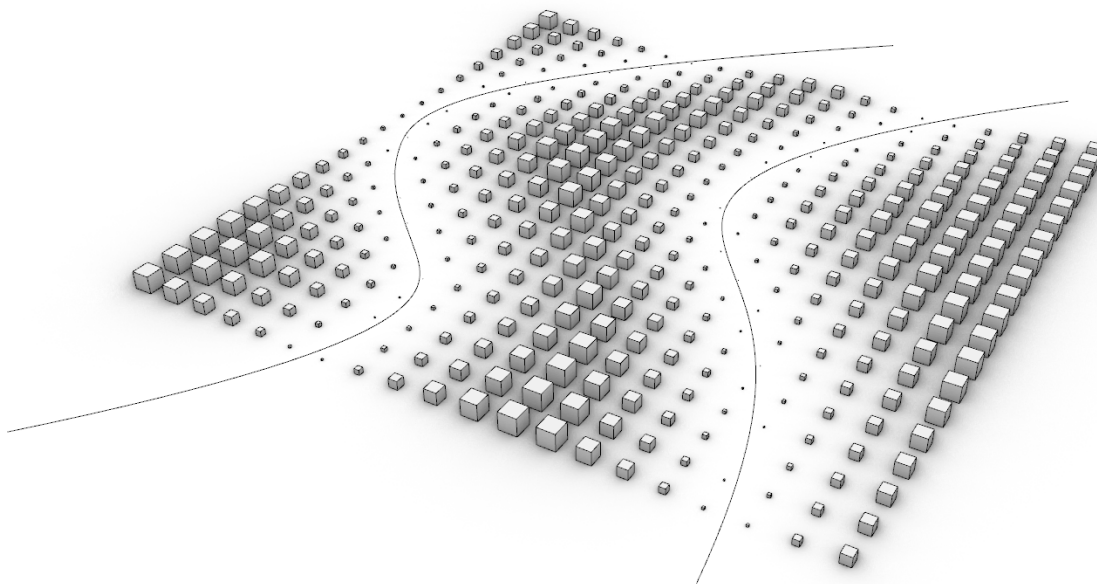


Figura 87. Resultado de algoritmo de *Grasshopper*: “elementos atractores”. El ejemplo es producto de varios parámetros *Curve* como elementos atractores y rectángulos como objetos a multiplicar. Elaboración propia (2020).

3.2.3. Elemento atractor en superficie subdividida.

El objetivo de la presente estrategia algorítmica es subdividir una superficie existente y generar aberturas a partir de su subdivisión de manera diferenciada. La magnitud de la subdivisión será en función directamente proporcional a la lejanía existente con respecto a un elemento atractor, el cual será proyectado en la superficie, con el fin de siempre presentar una localización directa del atractor en la misma independientemente de su ubicación.

El algoritmo combina criterios de las estrategias básicas número: (1) “geometría múltiple en superficie subdividida” y (2) “elementos atractores”. El objetivo es utilizar la superficie a generar como recurso arquitectónico en elementos verticales como muros, celosías, membranas, o en elementos horizontales como losas, pérgolas y techos (ver Figuras 88, 89 y Tabla 5).

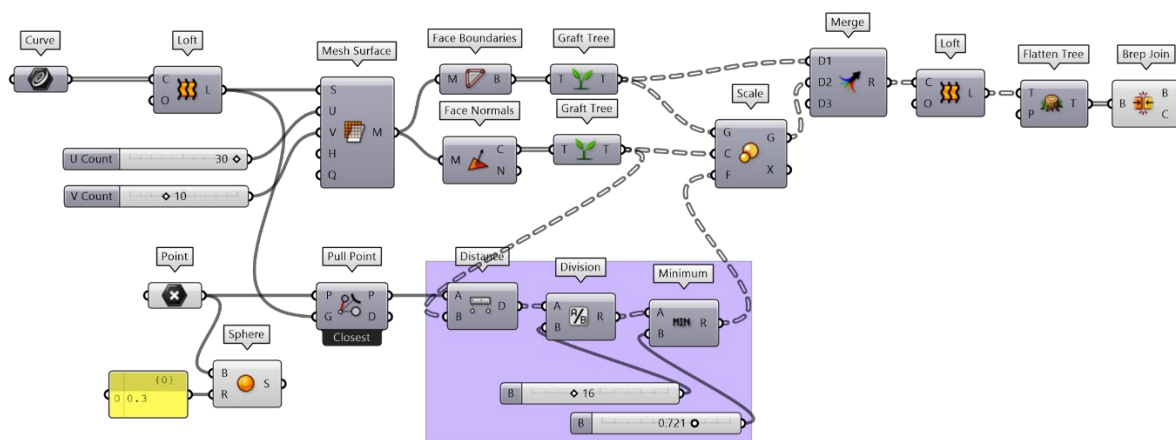


Figura 88. Algoritmo de *Grasshopper* para estrategia básica “elemento atractor en superficie subdividida”. El ejemplo muestra el parámetro *Point* como elemento atractor con influencia en superficie. Elaboración propia (2020).

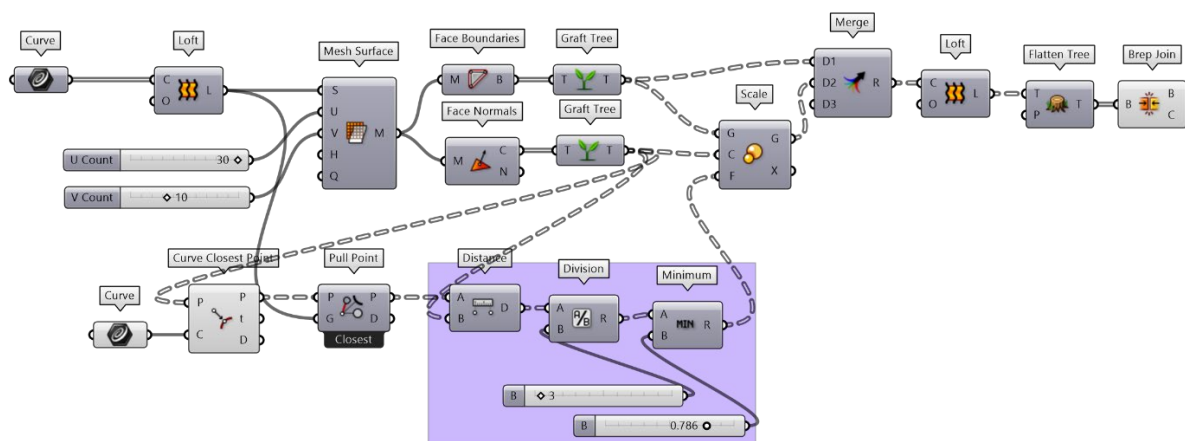


Figura 89. Algoritmo de *Grasshopper* para estrategia básica “elemento atractor en superficie subdividida”. El ejemplo muestra el parámetro *Curve* como elemento atractor con influencia en superficie. Elaboración propia (2020).

Tabla 5

Tabla descriptiva para construcción de algoritmo: elemento atractor en superficie subdividida.

Sección	Paso	Descripción
1. Establecer una superficie	1.1	Referencia en <i>Grasshopper</i> una superficie generada directamente en <i>Rhinoceros</i> ® a través de un parámetro <i>Surface</i> o establecer los elementos necesarios para generar en <i>Grasshopper</i> una superficie.
2. Convertir superficie de tipo NURBS a <i>Mesh</i>	2.1	La estrategia requerirá primeramente convertir la superficie NURBS a <i>Mesh</i> , con subdivisiones manipulables en dirección U y V. Utilizar componente de función <i>Mesh Surface</i> , conectar la superficie inicial en el <i>input S</i> correspondiente y determinar la subdivisión con un parámetro <i>Number</i> o un <i>Number Slider</i> conectado en los <i>inputs U</i> y <i>V</i> consecutivamente.
3. Extraer contorno de cada subdivisión producida	3.1	Extraer el contorno de cada cara producida por la subdivisión de la <i>Mesh</i> . Utilizar componente <i>Face Boundaries</i> y conectar la malla (<i>Mesh</i>) resultante al <i>input M</i> correspondiente.
4. Extraer centros de cada subdivisión producida	4.1	Extraer el centro de cada cara producida por la subdivisión de la <i>Mesh</i> . Utilizar componente <i>Face Normals</i> y conectar la malla (<i>Mesh</i>) resultante al <i>input M</i> correspondiente. El componente ofrecerá como <i>output</i> los centros y los vectores normales de cada cara.
5. Escalar los contornos de cada cara, utilizar sus	5.1	Extraer componente <i>Scale</i> para manipular la escala de abertura de los contornos de cada cara. Conectar el <i>output B</i> del componente <i>Face Boundaries</i> en el <i>input G</i> de la función

Sección	Paso	Descripción
respectivos centros como recurso.		<i>Scale</i> , así como el <i>output C</i> del componente <i>Face Normals</i> en el <i>input C</i> de la función <i>Scale</i> .
	5.2	De manera opcional, utilizar un parámetro <i>Number</i> o un <i>Number Slider</i> para manipular el factor de escala. Para efectos de este objetivo, conectar el parámetro en el <i>input F</i> de la función <i>Scale</i> .
6. Variar factor de escala con elemento atractor	6.1	Sustituir (en caso de que se haya implementado) el parámetro <i>Number Slider</i> conectado al <i>input F</i> de la función <i>Scale</i> por un conjunto de parámetros de ejercicio atractor de punto o curva (ver estrategia algorítmica básica “elementos atractores”).
	6.2	Es importante tener en cuenta que dicho factor no puede ser mayor a 1. Dicho parámetro se determina en el componente <i>Minimum</i> del algoritmo atractor (ver estrategia algorítmica básica “elementos atractores”).
7. Proyectar elemento atractor en superficie	7.1	Proyectar elemento atractor (punto o curva) que se utilizará para manipular las aberturas.
	7.2	En caso de utilizar una curva como elemento atractor utilizar función de <i>CurveCP</i> (<i>Curve Closest Point</i>) para evaluar los puntos más cercanos a la curva atractora (ver paso 6.5 de la estrategia algorítmica básica “elementos atractores”).
	7.3	Utilizar función <i>Surface Closest Point</i> o <i>Pull Point</i> y conectar el <i>output</i> de punto(s) del elemento atractor en el <i>input P</i> del componente <i>Surface Closest Point</i> o <i>Pull Point</i> . Conectar el <i>output</i> de la superficie inicial en el <i>input S</i> del componente <i>Surface Closest Point</i> o en el <i>input G</i> del componente <i>Pull Point</i> .
	7.4	Utilizar el punto proyectado por el componente <i>Surface Closest Point</i> o <i>Pull Point</i> para tomar distancia con respecto a los centros de las caras de la superficie subdividida y dar continuidad al ejercicio atractor. Conectar <i>output P</i> de función <i>Surface Closest Point</i> o <i>Pull Point</i> en <i>input A</i> del componente <i>Distance</i> .
8. Generar superficies entre el contorno de cada cara de malla y los contornos escalados.	8.1	Generar el <i>Loft</i> correspondiente a los contornos de cada cara de la malla y los contornos escalados (producto de la función <i>Scale</i>). De forma opcional se puede usar <i>Merge Tree</i> , para tener control visual del orden en el que se hace la conexión a <i>Loft</i> .
	8.2	Para que el <i>Loft</i> se haga correctamente entre los dos contornos, la información previamente debe ser agrupada en listas de información (<i>Graft Tree</i>); de no ser así, la superficie se generará con un trayecto continuo erróneo que pasa por todos los contornos existentes.
	8.3	Implementar un componente <i>Graft Tree</i> entre el <i>output B</i> de la función <i>Face Boundaries</i> y el <i>input G</i> de la función <i>Scale</i> .
9. Unir todas las caras generadas	8.4	Implementar un componente <i>Graft Tree</i> entre el <i>output C</i> de la función <i>Face Normals</i> y el <i>input C</i> de la función <i>Scale</i> .
	9.1	Para finalizar el algoritmo, debe unirse todas las nuevas caras (con abertura), utilizar <i>Brep Join</i> . Para que la unión se haga correctamente se debe aplanar la información (desagrupar) con <i>Flatten Tree</i> .

Nota. Elaboración propia (2020).

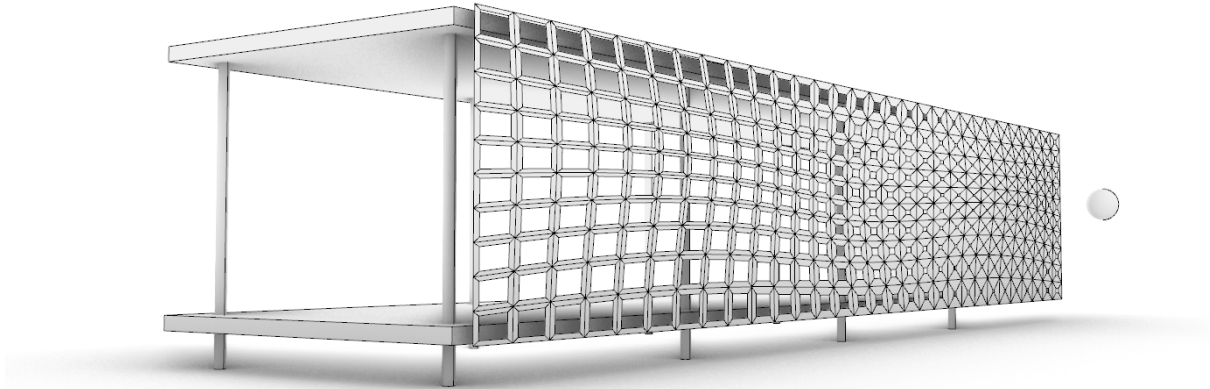


Figura 90. Resultado de algoritmo de *Grasshopper*: “elemento atractor en superficie subdividida”. El ejemplo es producto del parámetro *Point* como elemento atractor. Elaboración propia (2020).

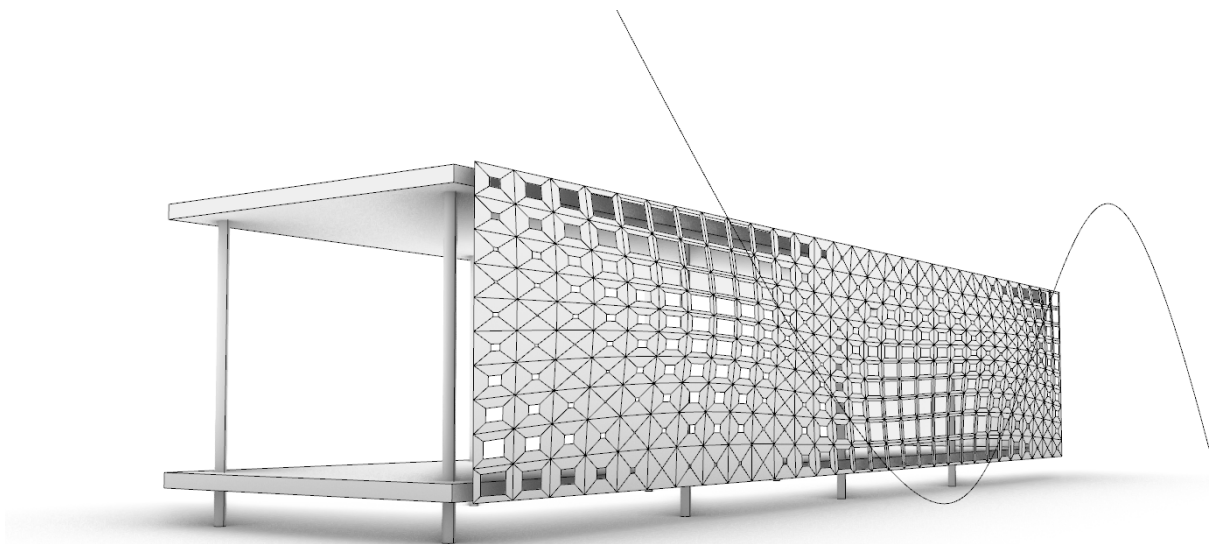


Figura 91. Resultado de algoritmo de *Grasshopper*: “elemento atractor en superficie subdividida”. El ejemplo es producto del parámetro *Curve* como elemento atractor. Elaboración propia (2020).

3.2.4. Retícula en diagonal (*Diagrid*) a partir de una superficie.

El algoritmo expuesto en la presente estrategia genera una retícula de bordes en diagonal a partir de la subdivisión de una superficie NURBS. Al subdividir la superficie, se extrae en el algoritmo los vértices con el fin de generar un conjunto de líneas que se interconecten con los puntos de manera cruzada.

El objetivo es utilizar la retícula a generar como recurso arquitectónico en elementos verticales como celosías, armazones estructurales y membranas, o en elementos horizontales como, pérgolas, techos y armazones estructurales. (ver Figuras 92 y Tabla 6).

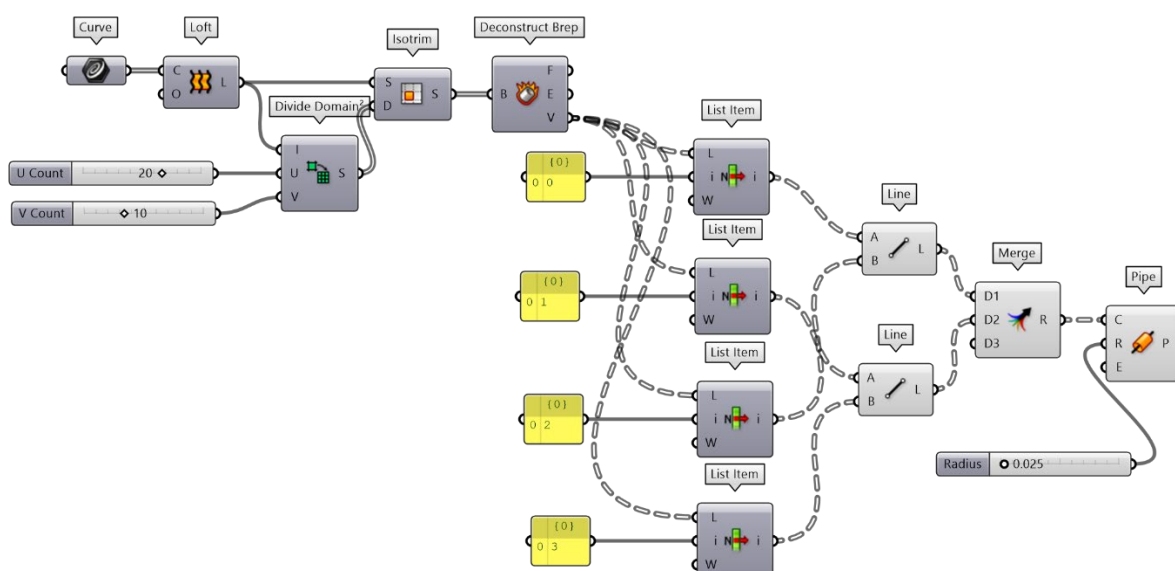


Figura 92. Algoritmo de *Grasshopper* para estrategia básica “Retícula en diagonal (*Diagrid*) a partir de una superficie”. Estrategia retomada de *AAD_Algorithmis-Aided Design: Parametric Strategies Using Grasshopper*, (Tedeschi, 2014, p. 157). Adaptación y modificación: elaboración propia (2020).

Tabla 6

Tabla descriptiva para construcción de algoritmo: retícula en diagonal (*Diagrid*) a partir de una superficie.

Paso	Descripción
1	Dibujar en <i>Rhinoceros</i> ® una superficie con forma a elección puede presentar curvatura. También es posible dibujar dos o más curvas que funcionen como directriz, para posteriormente, utilizar el componente de función <i>Loft</i> en <i>Grasshopper</i> (a partir del parámetro de <i>Curve</i> con el dibujo referenciado).
2	Comenzar un nuevo archivo de <i>Grasshopper</i> y referenciar la superficie dibujada en <i>Rhinoceros</i> ® en un parámetro de <i>Surface</i> . En caso de haber dibujado en <i>Rhinoceros</i> ® dos o más curvas, utilizar parámetro de <i>Curve</i> para referenciarlas.
3	En caso de haber dibujado en <i>Rhinoceros</i> ® dos o más curvas, utilizar componente de función de <i>Loft</i> para generar superficie a partir de estas.
4	Subdividir dominios de la superficie generada a partir del componente de función <i>Divide Domain</i> ² y asignar los correspondientes <i>inputs</i> U y V, utilizar un parámetro numérico o un <i>Number Slider</i> .
5	Proceder a subdividir físicamente la superficie generada por el <i>Loft</i> , con el componente de función <i>Isotrim</i> , utilizar como recurso las divisiones de dominio de la superficie.
6	Deconstruir cada superficie resultante de la subdivisión con el componente de función <i>Deconstruct Brep</i> . Conectar el <i>output S</i> de <i>Isotrim</i> con el <i>input B</i> de <i>Deconstruct Brep</i> .
7	Seleccionar de manera agrupada los vértices resultantes de <i>Deconstruct Brep</i> . Extraer función <i>List Item</i> cuatro veces para seleccionar todos los elementos 0, 1, 2 y 3 por separado, y segregar en grupos los conjuntos de vértices contenidas en las listas de información. Conectar el <i>output V</i> de <i>Deconstruct Brep</i> en el <i>input L</i> de cada <i>List Item</i> extraído. Establecer el índice correspondiente (0,1,2,3) en el <i>input i</i> .
8	Generar líneas con los vértices resultantes de manera alternada. Extraer dos componentes de función <i>Line</i> . Conectar el <i>output i</i> de cada <i>List Item</i> con los <i>inputs A</i> y <i>B</i> de la función <i>Line</i> . Seguir la siguiente secuencia: conectar la función <i>List Item</i> con índice 0 en el <i>input A</i> de la primera función <i>Line</i> y la función <i>List Item</i> con índice 2 en el <i>input B</i> de la primera función <i>Line</i> ; posteriormente, conectar la función <i>List Item</i> con índice 1 en el <i>input A</i> de la segunda función <i>Line</i> y la función <i>List Item</i> con índice 3 en el <i>input B</i> de la segunda función <i>Line</i> .
9	Desarrollar un tubo por cada línea generada. Extraer función <i>Pipe</i> y conectar el <i>output L</i> de cada componente <i>Line</i> con el <i>Input C</i> . Establecer con un parámetro <i>Number</i> o con un parámetro dinámico <i>Number Slider</i> el radio del tubo en el <i>input R</i> del componente <i>Pipe</i> .

Nota. Elaboración propia (2020).

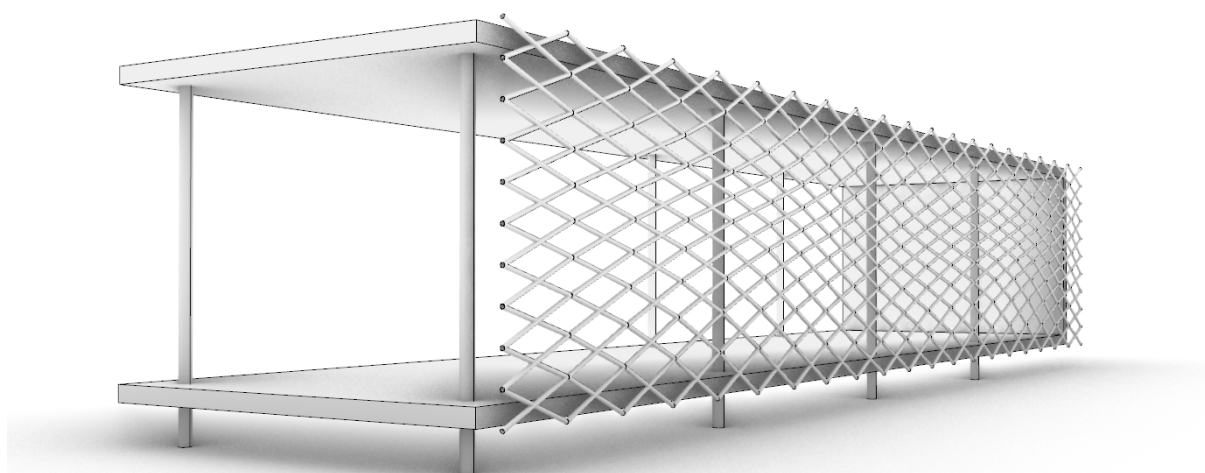


Figura 93. Resultado de algoritmo de *Grasshopper*: “retícula en diagonal (*Diagrid*) a partir de una superficie”, variante A. Elaboración propia (2020).

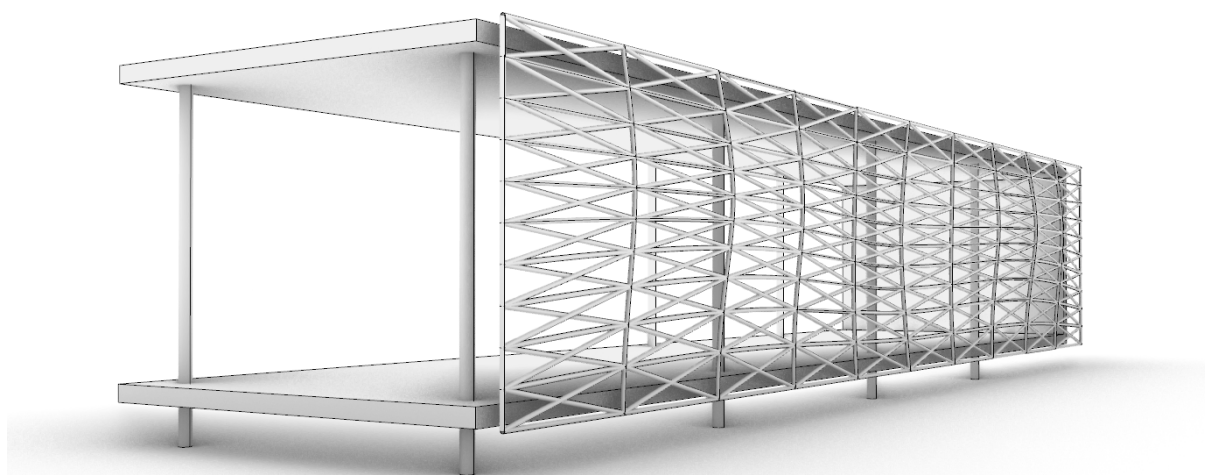


Figura 94. Resultado de algoritmo de *Grasshopper*: “retícula en diagonal (*Diagrid*) a partir de una superficie”, variante B. Elaboración propia (2020).

3.2.5. Superficie generada a partir de función matemática.

El objetivo es generar una superficie a partir de puntos manipulados en ubicación a través de una ecuación matemática. La estrategia parte de generar un conjunto de rangos numéricos que servirán como recurso para la localización de un conjunto de puntos, que posteriormente generarán una superficie.

El algoritmo expuesto sirve de referencia para tener un criterio morfogenético para el desarrollo de superficies a partir de lógicas matemáticas. Las ecuaciones de una, dos o más variables tienen la misión de desempeñarse como un medio morfológico ante un conjunto de parámetros transformables. Las modificaciones en los parámetros tienen una repercusión inmediata en el desarrollo de la forma a través de la ecuación matemática que funciona como un mecanismo de transformación formal (ver Figura 95 y Tabla 7).

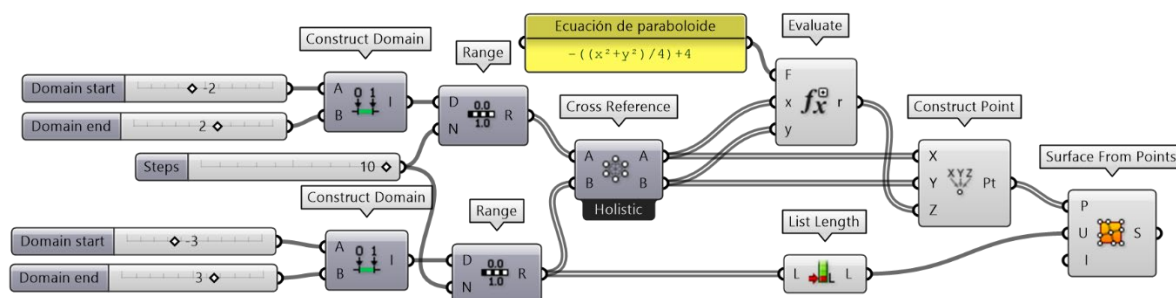


Figura 95. Algoritmo de *Grasshopper* para estrategia básica “superficie generada a partir de función matemática”. Estrategia retomada de *AAD_Algorithmis-Aided Design: Parametric Strategies Using Grasshopper*, (Tedeschi, 2014, p. 105). Adaptación y modificación: elaboración propia (2020).

Tabla 7

Tabla descriptiva para construcción de algoritmo: superficie generada a partir de función matemática.

Sección	Paso	Descripción
1. Generar listas de información a partir de rangos numéricos	1.1	Extraer dos componentes de función <i>Range</i> .
	1.2	Construir un dominio numérico para cada función de rango. Utilizar el componente de función <i>Construct Domain</i> . Establecer con un parámetro dinámico <i>Number Slider</i> el valor mínimo

Sección	Paso	Descripción
		(conectar a <i>input A</i>) y el valor máximo (conectar a <i>input B</i>) a trabajar en el dominio numérico.
	1.3	Conectar el <i>output I</i> de la función <i>Construct Domain</i> en el <i>input D</i> de cada componente <i>Range</i> .
	1.4	Establecer en cuantos pasos se desarrollará cada rango numérico determinado. Conectar un parámetro <i>Number Slider</i> en el <i>input N</i> de cada componente <i>Range</i> .
2. Relacionar todos los datos de las listas de información generadas por los rangos numéricos	2.1	Relacionar todos los datos de las listas de información generadas por los rangos numéricos a través de la función <i>Cross Reference</i> . Conectar el <i>output R</i> del primer componente <i>Range</i> en el <i>input A</i> de <i>Cross Reference</i> . Conectar el <i>output R</i> del segundo componente <i>Range</i> en el <i>input B</i> de la función <i>Cross Reference</i> .
3. Generar una retícula de puntos a través de las listas de información generadas	3.1	Cada lista de información con datos cruzados servirá de recurso para elaborar una retícula de puntos. Extraer componente de función <i>Construct Point</i> .
	3.2	Conectar el <i>output A</i> de la función <i>Cross Reference</i> en el <i>input X</i> de <i>Construct Point</i> .
	3.3	Conectar el <i>output B</i> de la función <i>Cross Reference</i> en el <i>input Y</i> de <i>Construct Point</i> .
4. Aplicar ecuación matemática en las listas de información generadas para manipular la ubicación de las coordenadas Z de los puntos a generar	4.1	Extraer componente de función <i>Evaluate</i> .
	4.2	Implementar la ecuación matemática a trabajar. Ejemplo, ecuación de paraboloides: $((x^2+y^2) / 4) + 4$. Utilizar un parámetro dinámico de panel para escribir y conectar la ecuación en el <i>input F</i> del componente <i>Evaluate</i> .
	4.3	Establecer las variables <i>x</i> e <i>y</i> de la ecuación. Conectar el <i>output A</i> del componente <i>Cross Reference</i> en el <i>input x</i> de <i>Evaluate</i> . Conectar el <i>output B</i> del componente <i>Cross Reference</i> en el <i>input y</i> de <i>Evaluate</i> .
	4.4	Utilizar los datos transformados por la ecuación matemática como valor de coordenada Z para disponer los puntos. Conectar el <i>output r</i> del componente <i>Evaluate</i> en el <i>input Z</i> de la función <i>Construct Point</i> .
5. Convertir el conjunto de puntos en superficie	5.1	Extraer componente <i>Surface From Points</i> .
	5.2	Conectar el <i>output Pt</i> del componente <i>Construct Point</i> en el <i>input P</i> de <i>Surface From Points</i> .
	5.3	Establecer la longitud de la lista como valor U de la superficie. Extraer función <i>List Length</i> . Conectar el <i>output R</i> del segundo componente de <i>Range</i> al <i>input L</i> de la función <i>List Length</i> . Conectar el <i>output L</i> de <i>List Length</i> en el <i>input U</i> de <i>Surface From Points</i> .

Nota. Elaboración propia (2020).

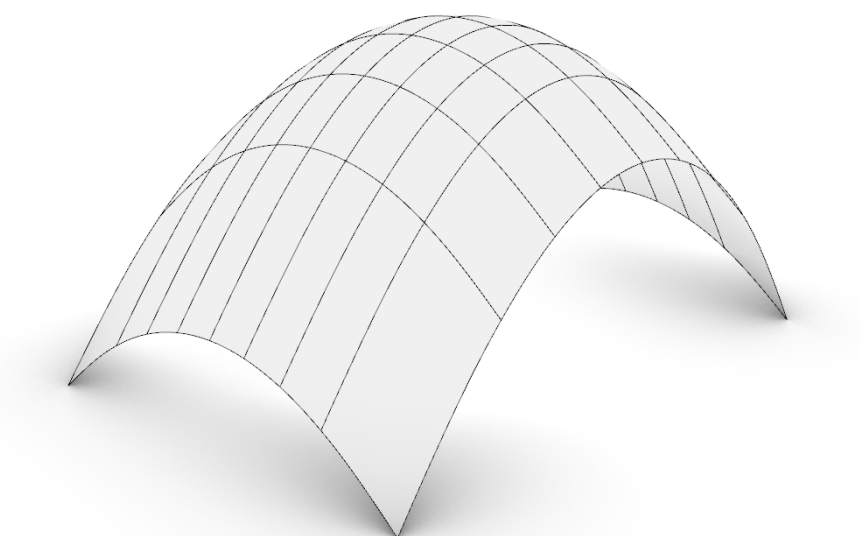


Figura 96. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir de función matemática”, variante A: superficie NURBS. Elaboración propia (2020).

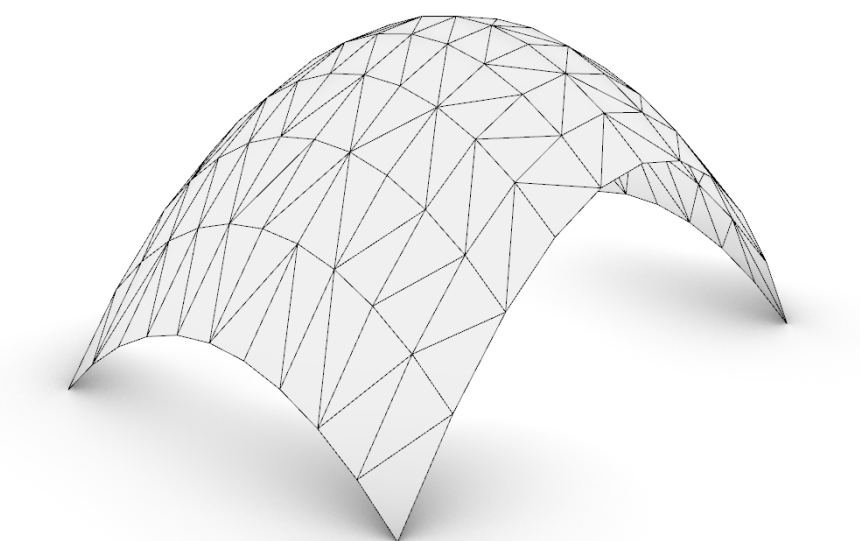


Figura 97. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir de función matemática”, variante B: *Mesh*. Elaboración propia (2020).

3.2.6. Superficie generada a partir de puntos aleatorios.

El algoritmo expuesto en la presente estrategia genera una superficie a partir de puntos manipulados en ubicación a través de una lista de información con valores aleatorios. La estrategia parte de generar una retícula de puntos, los cuales serán movidos de forma aleatoria con el fin de generar una superficie de tipo *Mesh* (ver Figura 98 y Tabla 8).

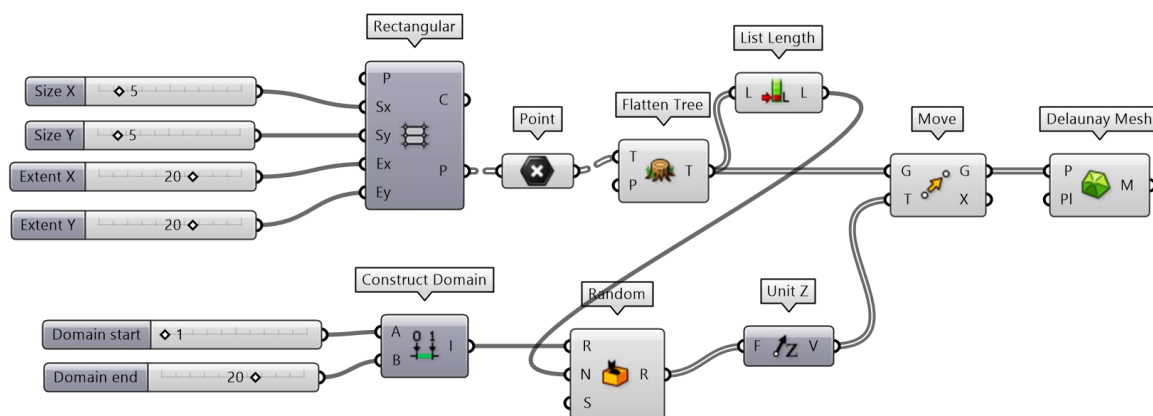


Figura 98. Algoritmo de *Grasshopper* para estrategia básica “superficie generada a partir de puntos aleatorios”. Elaboración propia (2020).

Tabla 8

Tabla descriptiva para construcción de algoritmo: superficie generada a partir de puntos aleatorios.

Sección	Paso	Descripción
1. Generar retícula de puntos	1.1	Extraer una función de retícula, específicamente <i>Rectangular Grid (Vector/Grid)</i> , establecer los correspondientes <i>inputs</i> de tamaño de separación, así como cantidad de puntos en dirección X e Y con un parámetro dinámico <i>Number Slider</i> .
	1.2	Conectar el <i>output</i> de puntos de la función <i>RecGrid</i> en un parámetro <i>Point</i> (el objetivo de este paso tiene carácter únicamente visual, ya que a partir de esta acción se aparecerán en pantalla de <i>Rhinoceros®</i> los puntos de la retícula para mayor control)
	1.3	Quitar <i>preview</i> a función de <i>RecGrid</i> , para de esta manera, ver solamente los puntos de la retícula.
2. Mover puntos de retícula	2.1	Mover los puntos en vector Z, para ello utilizar la función <i>Move</i> , y establecer el correspondiente vector Z de movimiento a través de una función de <i>Unit Z</i> , conectada en el <i>input T</i> correspondiente.

Sección	Paso	Descripción
	2.2	En este paso, es posible conectar un <i>Number Slider</i> , para determinar el movimiento en altura de los puntos, sin embargo, no es el objetivo de la estrategia, la cual, busca mover cada uno de ellos con una altura diferenciada.
3. Mover puntos de forma aleatoria	3.1	Contrario a establecer un valor único, la intención es generar una lista de información con valores diferenciados y que sean calculados por la computadora. por tanto, extraer función de <i>Random (Sets/Sequence)</i> , para generar una lista numérica aleatoria.
4. Parámetros de componente de función <i>Random</i>	4.1	La función de <i>Random</i> solicita tres <i>inputs</i> :
	4.2	Dominio: establecer un dominio numérico que determinará dónde comenzará y terminará la altura en la que se modificarán los puntos, es decir, número aleatorio inicial y final. Para lograr este objetivo, establecer función <i>Construct Domain</i> y determinar en sus <i>inputs</i> con un <i>Number Slider</i> el inicio y el final
	4.3	Número: establecer la cantidad de elementos que entrarán en juego en la dinámica aleatoria, en el caso de esta estrategia, es la cantidad de puntos que se tienen. Así, conectar una función de <i>List Length</i> en la salida del parámetro de <i>Point</i> , con el objetivo de que la computadora cuente dicha cantidad automáticamente.
	4.4	Semilla: el tercer <i>input</i> , consiste en determinar el valor de secuencia, esto permite variar la secuencia aleatoria conforme se cambie el número, de otra forma, solo se visualizará una sola en la lista de información.
5. Generar malla (<i>Mesh</i>) a partir de los puntos trasladados	5.1	Conectar la salida de geometría desplazada (puntos) de la función <i>Move</i> a una función <i>Delaunay Mesh</i> , para generar una superficie malla (<i>Mesh</i>) a partir de los puntos.
	5.2	Al hacer dicha conexión, aparecerá la función <i>Delaunay Mesh</i> en color amarillo, debido a que la malla se genera a través de los puntos con una topología triangular, es decir, busca a partir de cada punto, los más cercanos para generar una <i>Face</i> de malla consecutiva. Por tanto, la función no puede lograrlo porque la información viaja de manera ramificada (<i>Graft</i>), es decir, los puntos se encuentran agrupados en distintas familias o ramas, las cuales no pueden interactuar una con otra.
	5.3	Utilizar el filtro <i>Flatten Tree</i> , para desagrupar las familias de información y todos los datos fluyan en una sola lista o rama.
	5.4	Al “aplanar” la información, es decir, al viajar en una sola lista, se logrará ver como los puntos ahora son desplazados de manera independiente, y no de forma agrupada. <i>Mover magnitud del dominio que determina la altura para poder hacer una malla más legible.</i>
	5.5	Activar en el apartado de la interfaz superior de <i>Grasshopper</i> , la opción <i>Display</i> , y proseguir con la opción de <i>Preview Mesh Edges</i> , para lograr ver en la superficie <i>Mesh</i> de previsualización los bordes topológicos que la componen.

Nota. Elaboración propia (2020).

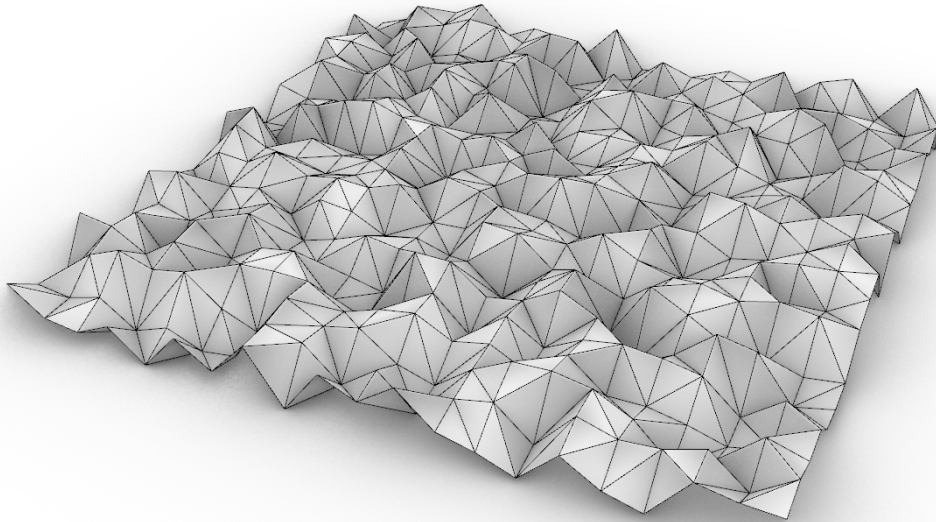


Figura 99. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir de puntos aleatorios”, variante: superficie *Mesh*. Elaboración propia (2020).

3.2.7. Superficie generada a partir de función matemática gráfica.

El objetivo es generar una superficie a partir de puntos trasladados en ubicación a través de una función matemática gráfica. La estrategia parte de generar una retícula de puntos, los cuales serán trasladados mediante una gráfica, que posicionará la lista de información como variables de una ecuación matemática con el fin de generar una superficie de tipo NURBS o *Mesh* (ver Figuras 100, 101 y Tabla 9).

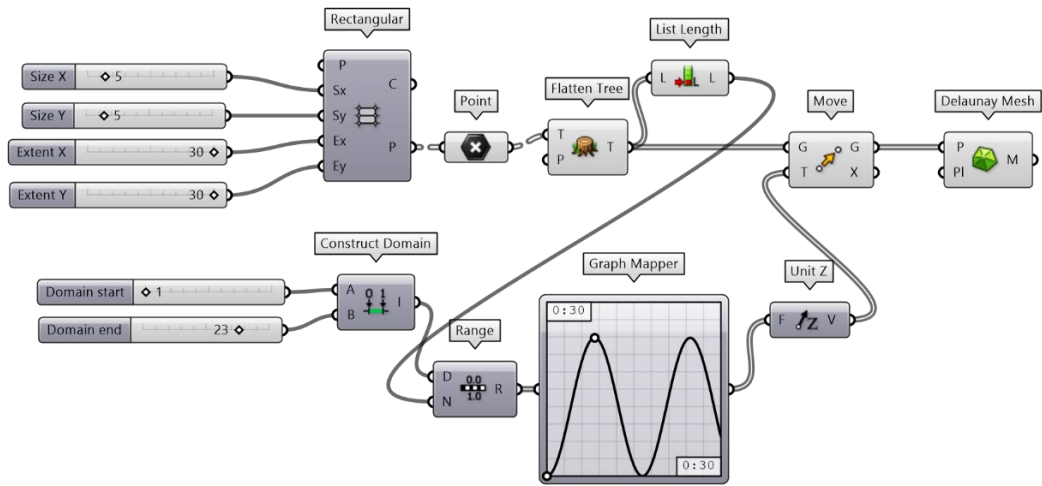


Figura 100. Algoritmo de *Grasshopper* para estrategia básica “superficie generada a partir de función matemática gráfica”, variante A: superficie *Mesh*. Elaboración propia (2020).

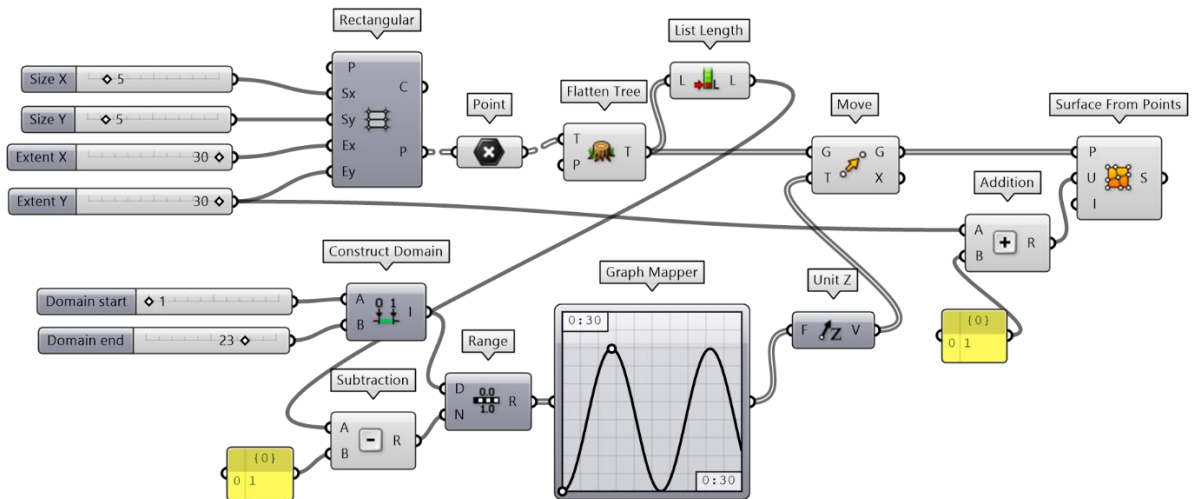


Figura 101. Algoritmo de *Grasshopper* para estrategia básica “superficie generada a partir de función matemática gráfica”, variante B: superficie *NURBS*. Elaboración propia (2020).

Tabla 9

Tabla descriptiva para construcción de algoritmo: superficie generada a partir de función matemática gráfica.

Sección	Paso	Descripción
1. Iniciar con algoritmo de estrategia básica de <i>Grasshopper</i> : <u>superficie generada a partir de puntos aleatorios</u> .	1.1	Utilizar el mismo algoritmo de la estrategia algorítmica básica de <i>Grasshopper</i> : “ <u>superficie generada a partir de puntos aleatorios</u> ”, el cual será modificado.

Sección	Paso	Descripción
2. Modificar algoritmo inicial.	2.1	Eliminar la función de <i>Random</i> y sustituirla por un rango numérico, que realizará una lista de información a partir de rangos con la función <i>Range (Sets/Sequence)</i> .
	2.2	El primer <i>input</i> les solicitará un dominio. Construir uno con función <i>Construct Domain</i> .
	2.3	El segundo <i>input</i> solicitará establecer cuántos puntos (o valores) entran en juego en el algoritmo, conectar la salida del <i>List Length</i> que ya se tiene.
3. Implementar función matemática gráfica	3.1	Extraer parámetro dinámico <i>Graph Mapper</i> , para filtrar los valores resultantes de la lista de información producida por el rango numérico (<i>Range</i>) con una función gráfica matemática.
	3.2	Posicionar el <i>Graph Mapper</i> , entre el rango numérico y el componente <i>Unit Z</i> . Conectar <i>output R</i> de la función <i>Range</i> en el <i>input</i> del parámetro dinámico <i>Graph Mapper</i> . Conectar <i>output</i> del <i>Graph Mapper</i> en el <i>input F</i> del componente <i>Unit Z</i> .
	3.3	Determinar en el parámetro dinámico <i>Graph Mapper</i> con botón derecho del <i>mouse</i> , el tipo de gráfica matemática a trabajar (comenzar con seno / <i>sine</i>).
	3.4	Configurar parámetros de gráfica, presionar doble clic izquierdo del <i>mouse</i> en el parámetro <i>Graph Mapper</i> . Establecer, de 0 a 30 en variable <i>X</i> , y de 0 a 30 en variable <i>Y</i> . (con esto se establece la magnitud aritmética de la gráfica en coordenada <i>X</i> e <i>Y</i>)
	3.5	Modificar la gráfica a gusto, (mover punto blanco de la gráfica). Las transformaciones en la gráfica repercuten en la forma de la malla (<i>Mesh</i>).
	3.6	Es posible elegir entre diferentes tipos de gráficas (funciones matemáticas) para transformar la malla.
	3.7	Elegir suma de senos (<i>Sine Summation</i>) y jugar con la forma de la malla. Elegir una forma a gusto.
4. Cambiar de <i>Mesh</i> a <i>NURBS Surface</i>	4.1	Es posible con este algoritmo trabajar con <i>NURBS Surface</i> en lugar de <i>Mesh</i> .
	4.2	Eliminar función <i>Delaunay Mesh</i> en algoritmo previamente elaborado.
	4.3	Extraer componente <i>Surface from Points</i> , conectar los puntos desplazados producto del <i>Move</i> , en el <i>input</i> correspondiente.
	4.4	Establecer la cantidad de puntos existentes en dirección <i>U</i> para que la computadora pueda establecer un parámetro de referencia de acuerdo con la cantidad de puntos existentes, y así crear la superficie.
5. Automatizar el parámetro de dirección <i>U</i> en la superficie.	5.1	Verificar datos para establecer de manera automatizado el parámetro de dirección <i>U</i> de la superficie.
	5.2	Ejemplo: en la imagen del algoritmo presentado en esta estrategia (ver Figura 101) se tiene en vector espacial <i>X</i> , una cantidad de 30 divisiones, eso significa 31 puntos en esa dirección. Por tanto, es necesario utilizar la salida del <i>Number Slider</i> que se tiene conectado en <i>input Ex</i> , para también ser conectado en el <i>input U</i> de la función <i>Surface from Points</i> , Es importante sumar una unidad al valor (utilizar función de suma - <i>Addition</i> - y usar <i>Panel</i> para agregar valor de 1).
	5.3	Al hacer esto, un error de lógica persistirá en el componente de función <i>Surface from Points</i> que no es perceptible.

Sección	Paso	Descripción
	5.4	<i>El error se encuentra en el conteo de puntos totales existentes, al utilizar <i>List Length</i> como recurso. El conteo arroja <i>X</i> cantidad, pero se debe tener en cuenta que en <i>Grasshopper</i> el 0 cuenta como valor, por tanto, esa cantidad tiene un excedente de 1. Ejemplo: en la imagen del algoritmo presentado (ver Figura 101) se tiene 961 puntos en total, como el valor 0 cuenta, entonces realmente se tiene 962.</i>
	5.5	<i>Para corregir el error, restar un valor a la cantidad longitud de la lista, utilizar la función de resta <i>-Subtraction-</i> y establecer con panel el valor de 1 para ser restado.</i>

Nota. Elaboración propia (2020).

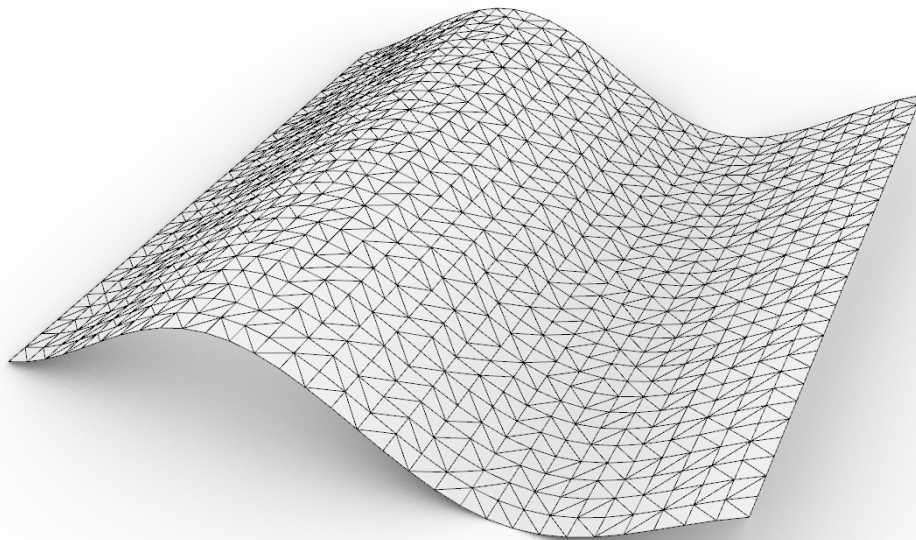


Figura 102. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir función matemática gráfica”, variante A: superficie *Mesh*, función matemática seno. Elaboración propia (2020).

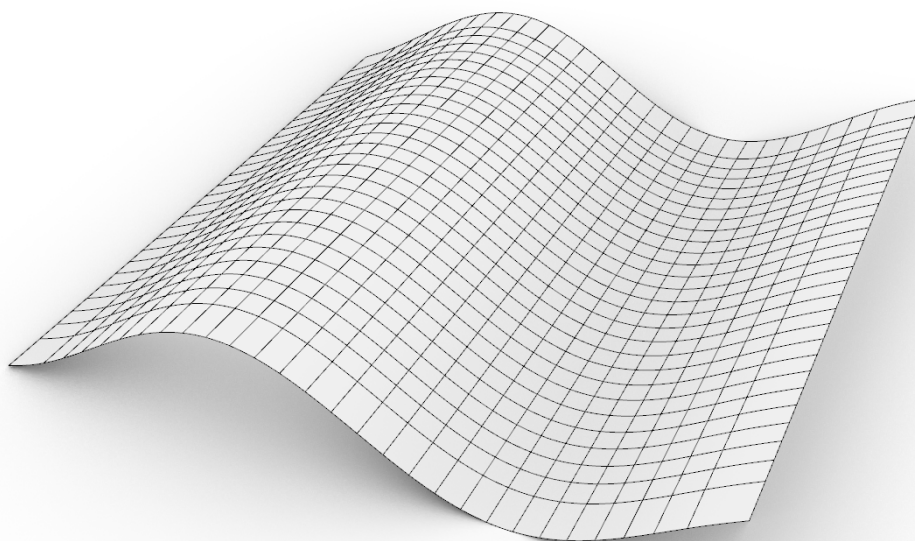


Figura 103. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir función matemática gráfica”, variante B: superficie NURBS, función matemática seno. Elaboración propia (2020).

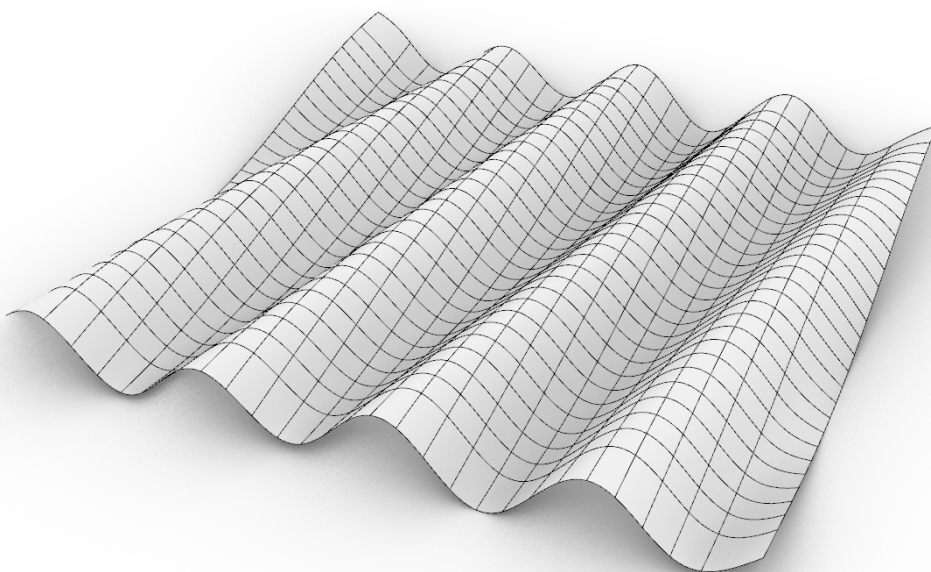


Figura 104. Resultado de algoritmo de *Grasshopper*: “superficie generada a partir función matemática gráfica”, variante C: superficie NURBS, función matemática suma de senos. Elaboración propia (2020).

3.2.8. Ejemplo de diseño paramétrico de envoltorio de estadio deportivo.

El objetivo del presente algoritmo es generar una membrana para un estadio deportivo. Se utilizan como estrategia modificadores de listas de información y una geometría multiplicada y escalada en una superficie subdividida. El algoritmo implementa una combinación de estrategias expuestas en apartados anteriores, por lo que es una definición de *Grasshopper* más amplia y compleja con aplicación arquitectónica. (ver Figura 105 y Tabla 10).

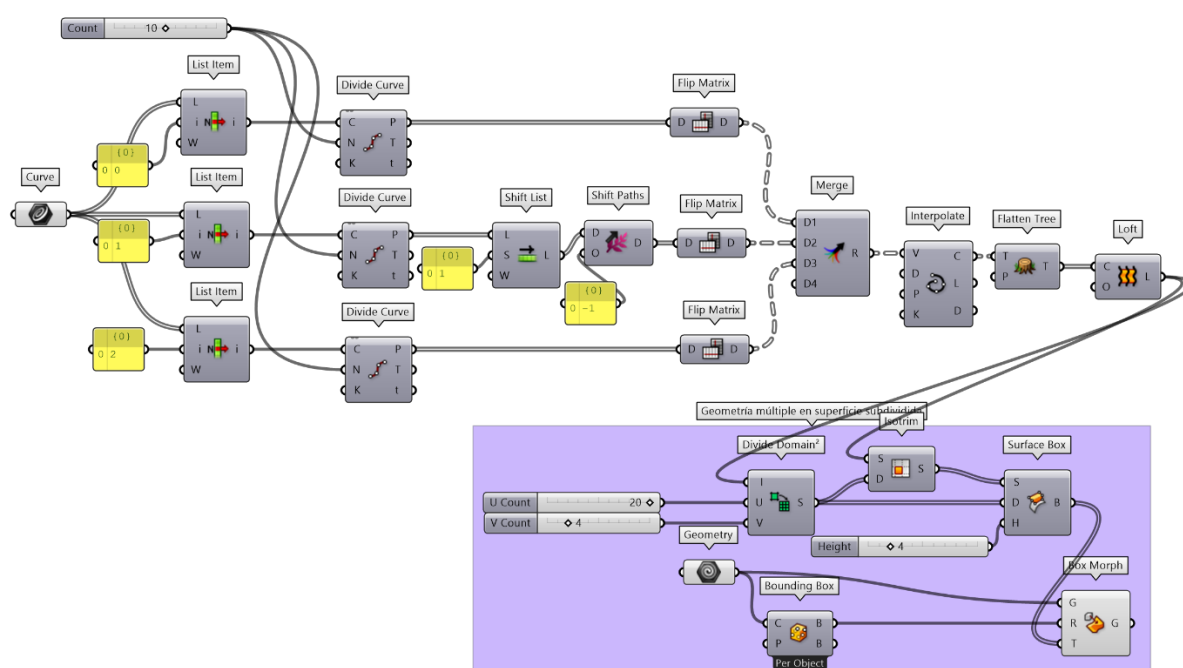


Figura 105. Algoritmo de *Grasshopper* para estrategia básica “ejemplo de diseño paramétrico de envoltorio de estadio deportivo”. Elaboración propia (2020).

Tabla 10

Tabla descriptiva para construcción de algoritmo: ejemplo de diseño paramétrico de envoltorio de estadio deportivo.

Sección	Paso	Descripción
	1.1	Dibujar en <i>Rhinceros</i> ® tres círculos con diferente radio y con el centro en la misma ubicación.

Sección	Paso	Descripción
1. Dibujar elementos geométricos iniciales en <i>Rhinoceros</i> ®	1.2	Mover directamente en <i>Rhinoceros</i> ® dos de los tres círculos en vector <i>Z</i> , la cantidad de unidades necesarias para dar forma a una envolvente de estadio deportivo.
	1.3	Dibujar en <i>Rhinoceros</i> ® una geometría diseñada a elección. (tener en cuenta, que dicha geometría debe ser una polisuperficie unida en todos sus bordes <i>-Join-</i> y no debe contener, de forma traslapada, alguna curva o polilínea.)
2. Referenciar en <i>Grasshopper</i> los elementos dibujados en <i>Rhinoceros</i> ®	2.1	Extraer un parámetro <i>Curve</i> . Referenciar las <i>tres curvas existentes</i> en el parámetro extraído (importante: referenciar las tres curvas y elegirlas de abajo hacia arriba).
3. Separar elementos de la lista de información	3.1	Utilizar función modificadora de listas de información <i>List Item</i> con el fin de separar las tres curvas referenciadas en el parámetro <i>Curve</i> (utilizar un <i>List Item</i> por cada curva).
4. Dividir en segmentos cada curva	4.1	Dividir cada curva separada con la función <i>Divide Curve</i> , y establecer una división de 10 segmentos.
5. Verificar estructura de lista/árbol de información	5.1	Verificar como fluye la información, conectar un <i>Param Viewer</i> al <i>output P</i> de cada <i>Divide Curve</i> .
	5.2	En este paso, se tiene <i>tres</i> listas de información similares de 10 puntos provenientes de las <i>tres circunferencias iniciales</i> .
6. Modificar listas de información para posteriormente generar una superficie con torsión	6.1	Se procederá con modificar la lista de información proveniente de la curva de en medio (<i>Item 1</i>) con una función modificadora de recorrer la lista. Utilizar <i>Shift List</i> y recorrer la información 1 paso.
	6.2	Conectar un <i>Param Viewer</i> en la salida de <i>Shift List</i> , con el objetivo de monitorear como la información fluye.
	6.3	El objetivo de separar las curvas y extraer puntos a través de dividir las, es para posteriormente generar curvas verticales que servirán de recurso para desarrollar una superficie con la función <i>Loft</i> ; sin embargo, el desplazamiento de información que se generó con <i>Shift List</i> , servirá para deformar la superficie a través de torsión.
7. Generar curvas verticales entre los círculos	7.1	Analizar cómo fluye la información. Observar que la <i>curva 0</i> y <i>2</i> van en una familia (o rama) llamada $\{0,0\}$, y la <i>curva 1</i> termina en una familia llamada $\{0,0,0\}$. Situación que generará un problema al momento de realizar las curvas verticales, ya que las tres curvas no podrían relacionarse (sólo la 0 y 2 podrían hacerlo) por la diferencia de nombre (o generación).
	7.2	Renombrar la familia de la <i>curva 1</i> $\{0,0,0\}$, al quitar una generación, ahora se llamará $\{0,0\}$. El método para quitar una generación a la familia es utilizar la función modificadora del árbol de información llamada <i>Shift Paths (Sets/Tree)</i> , e indicar en el <i>input</i> correspondiente, la cantidad de generaciones que se buscan quitar (-1).
	7.3	Los puntos resultantes de cada curva inicial (<i>círculos 0, 1 y 2</i>), vienen en una sola familia, ya que todas se llaman $\{0,0\}$. Al hacer una curva con todos estos puntos como recurso, daría como resultado una curva continua que pasará por todos ellos; y el objetivo, es lograr que las curvas se generen verticalmente.

Sección	Paso	Descripción
	7.4	Agrupar verticalmente los puntos. utilizar la función modificadora del árbol de información llamada <i>Flip Matrix (Sets/Tree)</i> . Esta función cambia verticalmente la estructura de cada lista de información y la convierte en árbol de información, agrupa todos aquellos elementos con la misma etiqueta.
	7.5	A partir de <i>Flip Matrix</i> , los puntos provenientes de cada curva, presentan consecutivamente los nombres $\{0,0\}$, $\{0,1\}$, $\{0,2\}$, etc. Por lo que los tres puntos llamados de la misma forma podrán interactuar y generar una curva (que será vertical).
	7.6	Extraer la función de <i>Interpolate Curve</i> para generar la curva (también se puede usar <i>NURBS Curve</i>). De manera opcional, antes de hacer la curva, converger organizadamente los tres conjuntos de puntos con la función modificadora del árbol de información <i>Merge Tree</i> .
8. Generar una superficie a través de las curvas verticales generadas	8.1	Generar una superficie a través de la función <i>Loft</i> . Utilizar como recurso las curvas verticales. Para que <i>Loft</i> pueda generar la superficie, es necesario convertir la información, que viene en distintas familias, en una sola para que puedan interactuar todas las curvas. Utilizar <i>Flatten Tree</i> para lograrlo.
	8.2	Establecer en el <i>input O</i> de la función <i>Loft</i> la opción <i>Closed Loft</i> para que el final de la superficie generada se conecte con el inicio.
9. Utilizar estrategia algorítmica básica de <i>Grasshopper</i> : <u>geometría múltiple en superficie subdividida</u>	9.1	Utilizar posteriormente el algoritmo: “ <u>geometría múltiple en superficie subdividida</u> ” descrita anteriormente.
	9.2	Utilizar como recurso la geometría dibujada inicialmente.
	9.3	Utilizar función <i>Brep Join (Surface/Util)</i> para soldar o unir todas las piezas que componen la piel resultante.

Nota. Elaboración propia (2020).

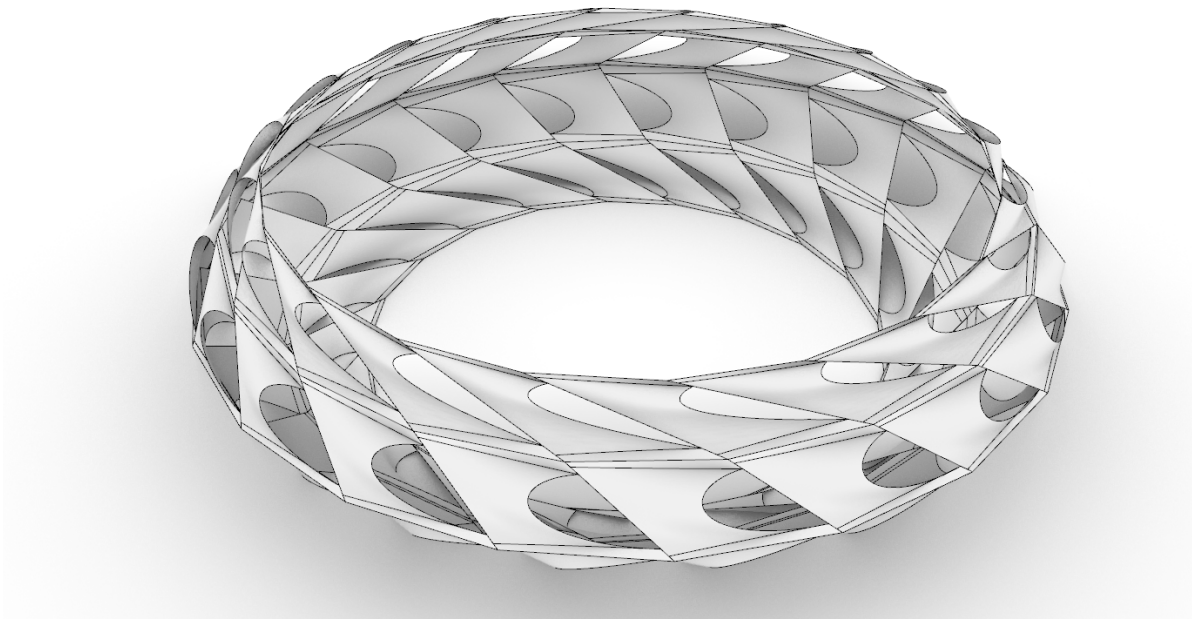


Figura 106. Resultado de algoritmo de *Grasshopper*: “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).

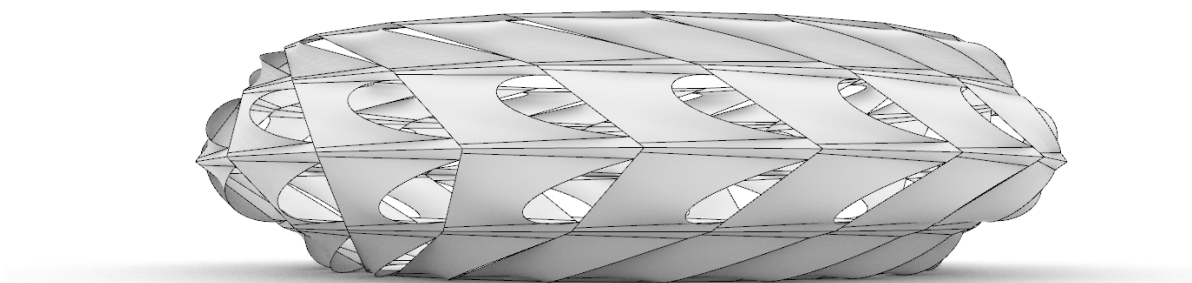


Figura 107. Resultado de algoritmo de *Grasshopper*: “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).

3.2.9. Ejemplo de diseño paramétrico de rascacielos.

El objetivo del presente algoritmo es generar un edificio de altura, el cual pueda ser manipulado en su forma y envoltente a través de manipular diversos parámetros. El ejercicio utilizará estrategias transformación de forma a través de una “función matemática gráfica”, así como alteración en la apertura de la membrana que lo compone con “elementos atractores en superficie subdividida”. También mostrará otra alternativa de diseño de piel a través de la implementación de la estrategia “retícula en diagonal (*Diagrid*) a partir de una superficie”

El algoritmo implementa una combinación de estrategias expuestas en apartados anteriores, por lo que es una definición de *Grasshopper* más amplia y compleja con aplicación arquitectónica. (ver Figuras 108, 109 y Tabla 11).

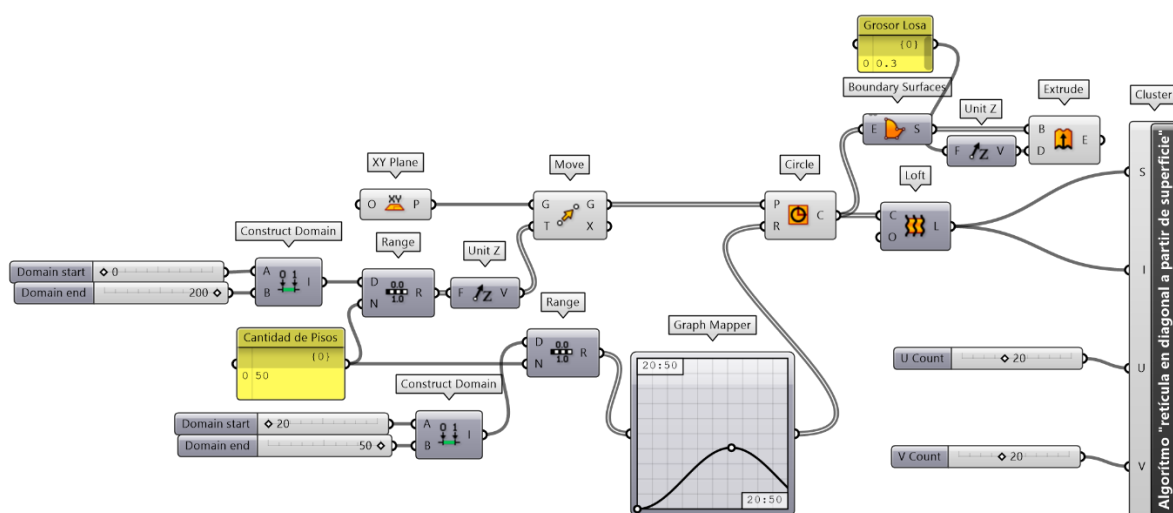


Figura 108. Algoritmo de *Grasshopper* para estrategia básica “ejemplo de diseño paramétrico de rascacielos”, variante A: envoltente desarrollada con estrategia algorítmica “retícula en diagonal (*Diagrid*) a partir de una superficie”. Elaboración propia (2020).

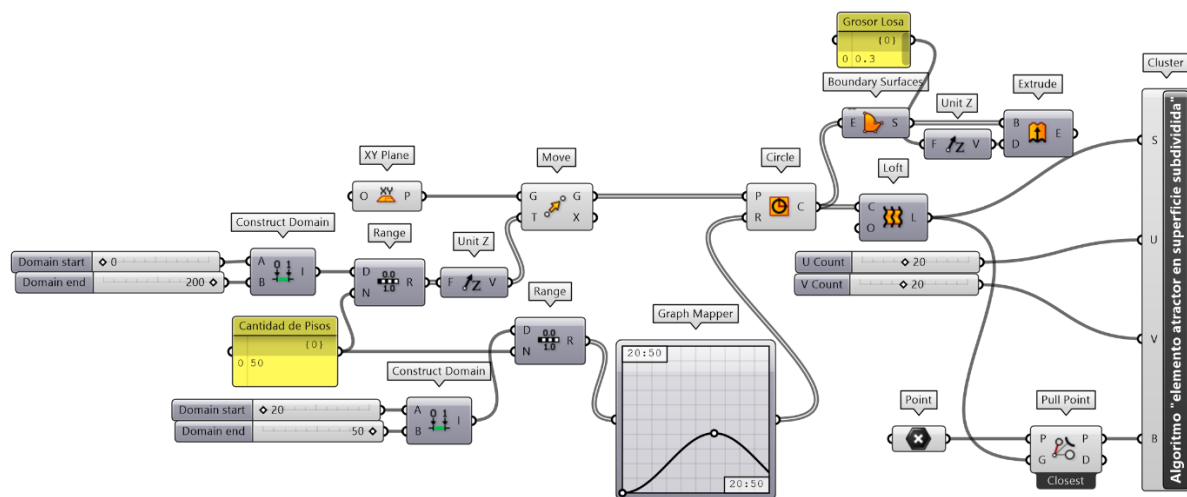


Figura 109. Algoritmo de Grasshopper para estrategia básica “ejemplo de diseño paramétrico de rascacielos”, variante B: envolvente desarrollada con estrategia algorítmica “elemento atractor en superficie subdividida”. Elaboración propia (2020).

Tabla 11

Tabla descriptiva para construcción de algoritmo: ejemplo de diseño paramétrico de rascacielos.

Sección	Paso	Descripción
1. Crear planos de trabajo geométrico como referencia para el desarrollo de losas	1.1	El algoritmo inicia al crear un plano de trabajo (<i>CPlane</i>) que representará una losa del edificio (también puede iniciar al establecer un punto en el espacio). Extraer función <i>XY Plane</i> (o establecer un punto a través de un parámetro de <i>Point</i>).
	1.2	El siguiente paso es generar una lista de información para mover el <i>CPlane X</i> cantidad de veces hasta una altura de 200m. Una manera práctica de hacerlo es utilizar una función de <i>Range</i> para establecer con parámetros la altura mínima y la máxima en la que se establecerán las losas, así como determinar en ese rango numérico, en cuántos pasos (o en este caso losas) se completará la secuencia.
	1.3	Extraer función <i>Move</i> y establecer en el <i>input</i> correspondiente, el vector <i>Z</i> de traslado, con una función de <i>Unit Z</i> .
	1.4	Generar la lista de información que determinará la ubicación de cada plano. Extraer componente de función <i>Range</i> y generar el dominio mínimo y máximo correspondiente (0 a 200), así como determinar en cuántos pisos (losas) se llevará ese dominio.
	1.5	Establecer 50 pisos para que cada uno de ellos este a 4m. de altura, adecuados para un edificio rascacielos.
2. Generar losas	2.1	Establecer el límite de cada losa. Utilizar función <i>Circle</i> para dar forma a cada piso.

Sección	Paso	Descripción
3. Diferenciar tamaño de losas	3.1	Establecer un radio diferenciado (lista de datos) correspondiente a la función de <i>Circle</i> . Se utilizará la estrategia de “ <u>función matemática gráfica</u> ” (ver estrategia “ <u>superficie generada a partir de una función matemática gráfica</u> ”) trabajada anteriormente para ejercer la diferenciación. Establecer en el <i>Graph Mapper</i> un dominio de 20:50 tanto en variable aritmética <i>X</i> , como en <i>Y</i> , en un tipo de gráfica <i>Sine</i> .
	3.2	Definir la forma de la edificación, manipular la gráfica y elegir a gusto una determinada.
	3.3	Generar superficies a partir de las circunferencias creadas. Extraer función <i>Boundary Surfaces</i> y utilizar los círculos como recurso.
	3.4	A partir de este momento las losas se encuentran generadas (puede darse grosor a cada superficie, de manera opcional, utilizar función <i>Extrude</i>).
4. Generar envolvente de edificio	4.1	El paso siguiente es generar la membrana o envolvente del edificio. Extraer función <i>Loft</i> y utilizar las circunferencias como recurso.
	4.2	Quitar temporalmente la visualización de las losas, desactivar <i>Preview</i> en la función <i>Boundary Surfaces</i> , ya que se trabajará a continuación únicamente con la superficie que funciona como envolvente.
5. Estrategias para el diseño de envolvente de la edificación	5.1	Los pasos previos producen una superficie que servirá de parámetro para aplicar una estrategia de diseño para la envolvente de la edificación.
	5.2	Utilizar estrategia algorítmica básica de <i>Grasshopper</i> “ <u>elemento atractor en superficie subdivida</u> ” (ver estrategia en apartado previo).
	5.3	Utilizar estrategia algorítmica básica de <i>Grasshopper</i> “ <u>retícula en diagonal (<i>Diagrid</i>) a partir de una superficie</u> ” (ver estrategia en apartado previo).

Nota. Elaboración propia (2020).

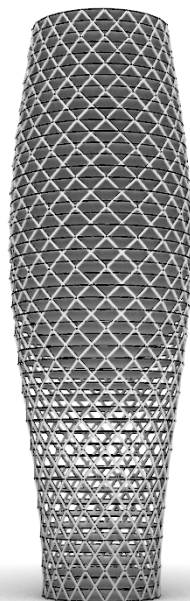


Figura 110. Resultado de algoritmo de *Grasshopper*: “ejemplo de diseño paramétrico de rascacielos”, variante A: envolvente desarrollada con estrategia algorítmica “retícula en diagonal (*Diagrid*) a partir de una superficie”. Elaboración propia (2020).

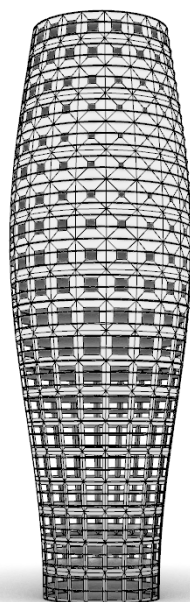


Figura 111. Resultado de algoritmo de *Grasshopper*: “ejemplo de diseño paramétrico de rascacielos”, variante B: envolvente desarrollada con estrategia algorítmica “elemento atractor en superficie subdividida”. Elaboración propia (2020).

3.3. Desarrollo metodológico de diseño arquitectónico Algorítmico Paramétrico

Con el objetivo de desarrollar una *metodología de diseño arquitectónico Algorítmico Paramétrico (AP)* el presente apartado expone una propuesta metodológica de diseño arquitectónico bajo las líneas de análisis de los aspectos cualitativos y cuantitativos como parámetros importantes a considerar en el proceso de diseño arquitectónico.

El desarrollo metodológico es presentado a través de una (a) construcción metodológica cualitativa, y otra (b) cuantitativa de diseño arquitectónico AP; a partir de las cuales, se identificarán las variables que funcionarán como parámetros para el análisis arquitectónico desde una lógica AP.

También, se relacionarán estos parámetros con las principales herramientas algorítmicas de *Grasshopper* que, a través de métodos causales de simulación o métodos de optimización heurística generativa, pueden desarrollarse oportunidades automatizadas de producción arquitectónica desde la perspectiva digital.

3.3.1. Construcción metodológica cualitativa de diseño arquitectónico AP

La vanguardia computacional en la Arquitectura considera los procesos de programación digital como medio para la producción de proyectos arquitectónicos, en especial el segmento relacionado con la programación digital visual. Las principales plataformas de programación visual en la actualidad son *Grasshopper* del programa computacional CAD (*Computer Aided-Design*) *Rhinoceros*®, *Dynamo* del programa computacional BIM *Revit*® de *Autodesk*® y el programa computacional multiplataforma *Vectorworks*®.

De las plataformas mencionadas anteriormente, la más utilizada y ampliamente desarrollada en la actualidad es *Grasshopper*, la cual es un entorno de programación computacional visual (*Graphical User Interface*, GUI) del programa computacional CAD

Rhinoceros®. Funciona como un editor de algoritmos que construye modelos paramétricos a través de diagramas de flujo.

A través de esta plataforma, y de acuerdo con las consideraciones de implementación de los aspectos fenomenológicos y semiológicos de Patrik Schumacher (2011, 2012), así como la agenda del enfoque Discretista en la Arquitectura computacional, es posible proponer un esquema metodológico que sintetice los pasos y variables a tomar en cuenta en el diseño arquitectónico, para el desarrollo de un proyecto a través de un proceso AP (ver Figura 112).

Esta propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP), es un algoritmo que refleja los aspectos a considerar para el proceso de diseño de la forma y el espacio desde la perspectiva fenomenológica y semiológica.

Parte de la visión de Schumacher (2011, 2012) que considera estos aspectos como un sistema de estructuras de comunicación para calificar las relaciones de organización y articulación de la forma y el espacio. También considera los principales elementos teóricos y metodológicos de Norberg-Schulz (1999; 1971) en torno a la topología, y los aspectos de percepción sensorial de Bloomer y Moore (1983) previamente descritos.

Finalmente, el diagrama relaciona estos elementos con las principales herramientas algorítmicas existentes en la plataforma de programación visual *Grasshopper*. Dichas herramientas son distinguidas en dos grupos: (a) aquellas que generan simulaciones para el análisis de problemas mediante el método causal, la solución a estos es a través de estrategias post-racionalizadas con modelos tradicionales abstractos o matemáticos, y (b) las que siguen el método y lógica heurística con sistemas de inteligencia artificial computacional que brindan oportunidades de solución generativa, con fines de optimización y desempeño de los productos arquitectónicos.

En el primer grupo, se encuentran herramientas de simulación y cálculo de elementos estructurales, físicas de partículas y elementos, análisis ambiental y termo-energético. El segundo grupo engloba utilidades para optimización multiobjetivo a través de algoritmos genéticos, análisis de elementos finitos y búsquedas estocásticas.

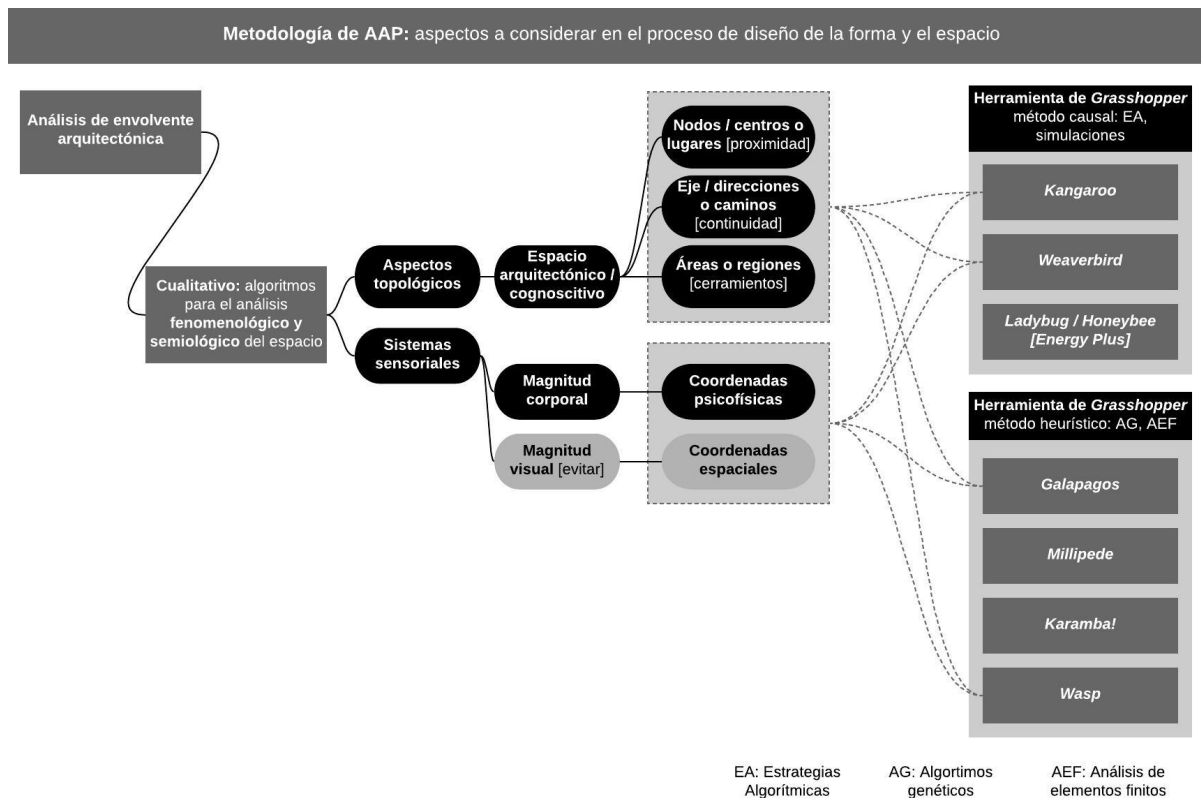


Figura 112. Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema metodológico de análisis cualitativo: aspectos fenomenológicos y semiológicos del espacio. Elaboración propia (2020).

El esquema metodológico se encuentra estructurado como un algoritmo, con el objetivo de identificar rutas de análisis y solución en el proceso de diseño arquitectónico. La propuesta se centra en la envolvente arquitectónica como uno de los parámetros más importantes para determinar el espacio. La envolvente arquitectónica define los límites espaciales, por tanto, un agente que se desenvuelve como un órgano que establece las directrices que consolidan la forma y el funcionamiento.

Para el análisis cualitativo de la envolvente arquitectónica, desde la perspectiva de los aspectos *fenomenológicos* y *semiológicos* del espacio, emergen las variables de diseño *topológicas cognoscitivas-existenciales* y las cualidades formales con relación a los *estímulos de los sistemas sensoriales y mentales*.

Derivan de las variables topológicas cognoscitivas-existenciales las variables: (1) *nodos, centros o lugares* como parámetros de análisis de la *proximidad*, (2) *ejes, direcciones o caminos* como parámetros de *continuidad*, y (3) *áreas o regiones* como parámetros de *cerramiento*. A partir de los referentes psico-sensoriales, es consecuente la *magnitud corporal*, que incluye como parámetro la *envolvente corporal*, en la cual desempeñan un papel importante las *coordenadas psicofísicas*.

Los parámetros de análisis de la propuesta metodológica cualitativa son relacionados con herramientas algorítmicas de *Grasshopper*. Los aspectos *topológicos cognoscitivos-existenciales* pueden ser articulados con *Kangaroo* y *Weaverbird* como herramientas algorítmicas bajo el método causal. Desde la perspectiva metodológica heurística pueden relacionarse con las herramientas *Galapagos* y *Wasp*.

Kangaroo es una herramienta algorítmica con aplicación en el análisis y diseño de aspectos topológicos para el análisis estructural (arquitectónico-constructivo). *Weaverbird*, es una utilidad para la optimización y transformación de aspectos topológicos desde una perspectiva formal-geométrica.

Galapagos es una herramienta que desarrolla resultados evolutivos para la optimización y desempeño mediante simulaciones de Algoritmos Genéticos (AG). *Wasp* es una utilidad para el diseño de elementos discretos, genera sistemas complejos de agregación a través de parámetros de geometría, ubicación de conexiones y orientación. Los sistemas topológicos generados por *Wasp* siguen estrategias de agregación estocástica y agregación impulsada por campos (*field-driven aggregation*).

3.3.2. Construcción metodológica cuantitativa de diseño arquitectónica AP

La propuesta metodológica de diseño AP, en la línea de análisis cuantitativo, se centra también en la envolvente arquitectónica como el parámetro principal para desarrollar el espacio; sin embargo, desde esta perspectiva la ruta de análisis de diseño se enfoca en la resolución de otras variables que son afines a la optimización de los aspectos fisiológicos arquitectónicos: *el análisis termo-energético y la estructura*.

Las variables que conformarán esta ruta de análisis cuantitativo también funcionarán como un algoritmo que incluye parámetros que servirán al análisis y simulación de aspectos estructurales, así como la optimización del espacio con relación a su integración al medio ambiente (ver Figura 113).

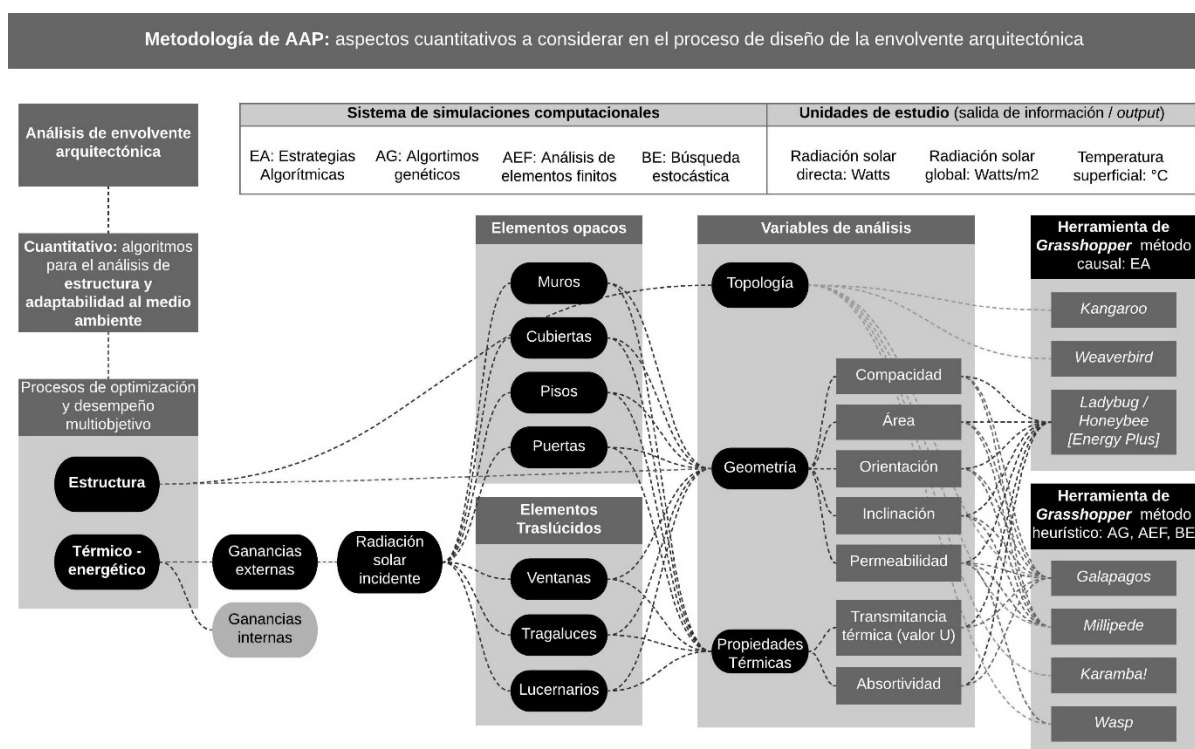


Figura 113. Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema metodológico de análisis cuantitativo: análisis termo-energético y la estructura. Elaboración propia (2020).

Las principales variables por considerar para efectuar el análisis termo-energético y estructural son: (1) *los aspectos geométricos*, (2) *los aspectos topológicos de la forma* y (3)

las *propiedades térmicas de la envolvente*. Desde el punto de vista *estructural* los *aspectos geométricos* y *topológicos* son los principales parámetros que influyen en su diseño y generación.

En cuestión de análisis *termo-energético* el enfoque de estudio de la envolvente arquitectónica es con relación a las *ganancias de calor externas* obtenidas por la *radiación solar*. En este sentido, *la geometría* y las consecuentes variables de (a) compacidad (relación superficie / volumen), (b) área, (c) orientación, (d) inclinación y (e) permeabilidad son los principales parámetros para evaluar en términos de optimización y desempeño. También, verifica las *propiedades térmicas* a partir de los parámetros de transmitancia térmica (coeficiente de transferencia de calor U) y absortividad, como referentes importantes a considerar.

Los elementos opacos de análisis para efectuar la evaluación de las propiedades térmicas y geométricas de una edificación son: (a) muros, (b) cubiertas, (c) pisos, (d) puertas. Los elementos translúcidos para tomar en cuenta son: (e) ventanas, (f) tragaluces y (g) lucernarios.

Desde el punto de vista fisiológico arquitectónico los parámetros *topológicos* pueden ser articulados con las herramientas algorítmicas de *Grasshopper: Kangaroo* y *Weaverbird* como recursos algorítmicos bajo el método causal. Desde la perspectiva de los métodos heurísticos las utilidades pueden relacionarse con: *Galapagos*, *Millipede*, *Karamba* y *Wasp*.

Los parámetros *geométricos* pueden ser articulados con las herramientas: *Ladybug-Honeybee [Energy Plus]* como utilidad de la lógica causal. *Galapagos* y *Millipede* como recursos algorítmicos que se encuentran bajo los métodos de optimización heurística digital.

Finalmente, desde la perspectiva del análisis de las *propiedades térmicas* las utilidades de la lógica causal con las que se pueden relacionar son: *Ladybug-Honeybee [Energy Plus]*, y *Galapagos* como herramienta perteneciente al método heurístico.

Ladybug-Honeybee [Energy Plus] es una aplicación algorítmica para el análisis, optimización y desempeño termo-energético. *Millipede* es una herramienta para simulaciones de análisis de elementos finitos y desarrollo de *iso surfaces* (iso superficies) a través de *meshes* (mallas) computacionales, con aplicación en optimización y desempeño de aspectos topológicos y estructurales. Por último, *Karamba!* es una utilidad para simulaciones de optimización y desempeño estructural mediante una combinación de algoritmos genéticos y análisis de elementos finitos.

3.3.3. Metodología de diseño arquitectónico AP

El desarrollo previamente descrito, en su línea de análisis cualitativo y cuantitativo, consolida una estructura amplia metodológica que tiene el objetivo de brindar rutas de análisis, evaluación, optimización y automatización en el proceso de diseño arquitectónico, a través de estrategias Algorítmicas Paramétricas (ver Figura 114).

Las estrategias, al estar enfocadas en la concepción del espacio a partir de la envolvente arquitectónica, tienen aplicación en todos los ámbitos de análisis del proceso de diseño. Es decir, pueden servir como referente en la manera de estructurar los procedimientos de recopilación de datos provenientes de un análisis de sitio o del usuario.

Por otro lado, la metodología no busca dirigirse exclusivamente al objeto arquitectónico; los parámetros de análisis que incluye también tienen implicación en enfoques arquitectónicos especializados como: la arquitectura del paisaje, el diseño de interiores, y algunas consideraciones acotadas del diseño urbano.

Todos los datos generados por el desarrollo algorítmico pueden ser exportados para aplicar fichas de diagnóstico o generar gráficas que ayuden a visualizar de manera abstracta y visual el compendio de resultados y sus aspectos a detalle.

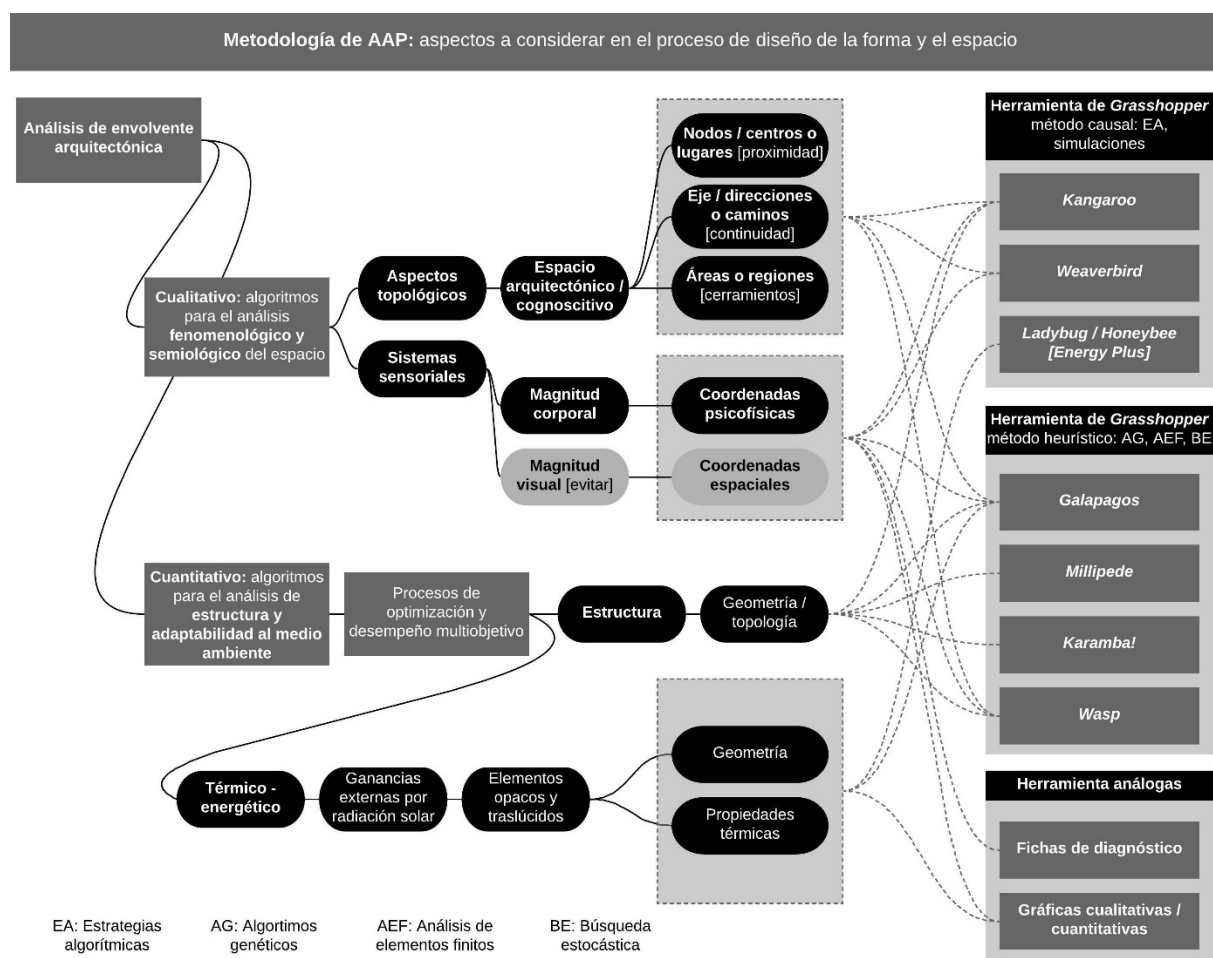


Figura 114. Propuesta metodológica de Arquitectura Algorítmica Paramétrica (AAP). Esquema global metodológico que incluye las rutas de análisis cualitativas y cuantitativas. Elaboración propia (2020).

3.4. Ejemplos de aplicación de la metodología de diseño arquitectónico AP

Esta sección ejemplifica algunas estrategias algorítmicas en *Grasshopper* con la metodología de diseño arquitectónico AP implementada. El objetivo es aumentar la capacidad de análisis y respuesta a las problemáticas que el proceso de diseño arquitectónico demande, de una manera integral y sinérgica.

Los ejemplos que se mostrarán complementan las estrategias algorítmicas básicas descritas en secciones previas, y aplicarán las herramientas algorítmicas de *Grasshopper* expuestas en la metodología, con el fin de encontrar soluciones de diseño a partir del manejo

de las variables descritas que funcionarán como parámetros implícitos en el algoritmo que se construirá.

3.4.1. Transformación topológica de una superficie con *Weaverbird*.

El objetivo de los algoritmos expuestos en esta sección es mostrar el funcionamiento de la herramienta algorítmica de *Grasshopper Weaverbird*. La aplicación sirve como un editor topológico de mallas (*Meshes*) con el fin de optimizarlas y transformarlas desde una perspectiva AP. *Weaverbird* es desarrollado por Giulio Piacentino como un implemento a la plataforma de programación visual *Grasshopper*.

A continuación, se muestran ejemplos algorítmicos de transformación topológica de una superficie. Se utilizará como recurso superficies generadas previamente en el apartado “Estrategias algorítmicas básicas en *Grasshopper* para aplicación en diseño arquitectónico” para posteriormente ser modificadas con *Weaverbird*. El objetivo es encontrar rutas de diseño cualitativo como aplicación de la metodología de diseño arquitectónico AP propuesto en el presente trabajo de investigación. (ver Figuras 15, 16 y Tabla 12).

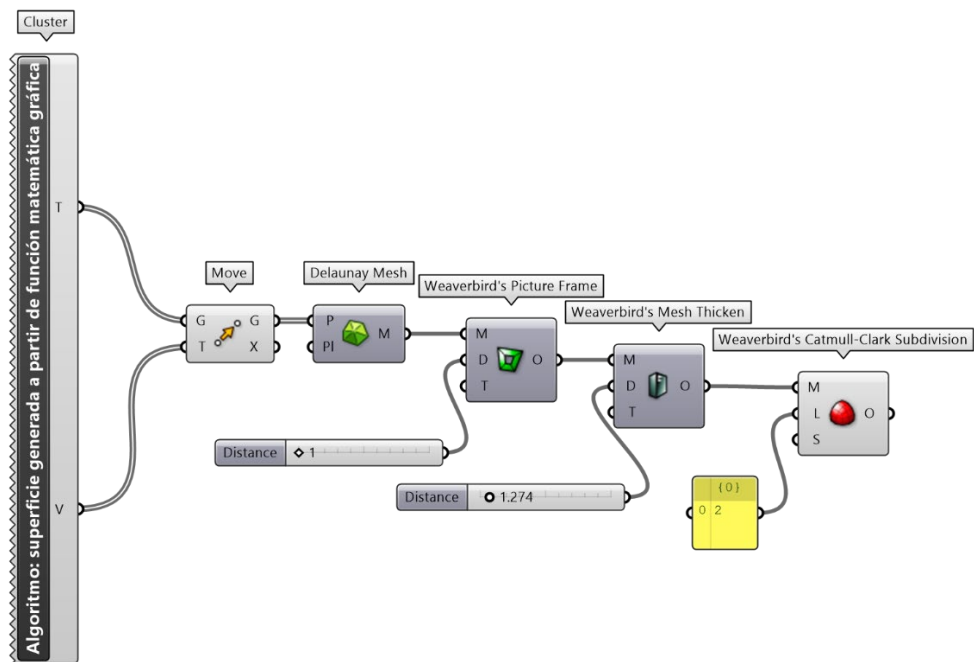


Figura 115. Algoritmo de *Grasshopper* “transformación topológica de una superficie con *Weaverbird*”. Variante A: manipulación de superficie producto de estrategia “superficie generada a partir de función matemática gráfica”. Elaboración propia (2020).

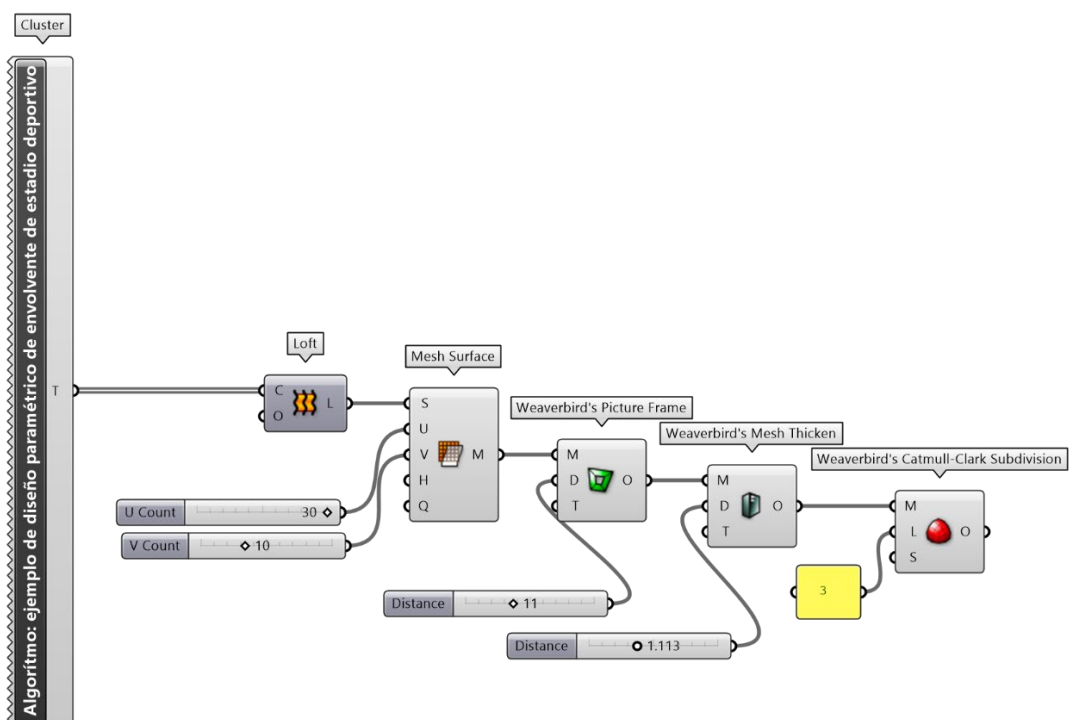


Figura 116. Algoritmo de *Grasshopper* “transformación topológica de una superficie con *Weaverbird*” Variante B: manipulación de superficie producto de estrategia “ejemplo de diseño paramétrico de envoltorio de estadio deportivo”. Elaboración propia (2020).

Tabla 12

Síntesis descriptiva de algoritmo: transformación topológica de una superficie con

Weaverbird.

Sección	Paso	Descripción
1. Preparar superficie para aplicar transformación topológica	1.1	El algoritmo comienza con un parámetro de superficie como insumo. Si la superficie tiene un origen NURBS, debe ser convertida al modelo <i>Mesh</i> .
	1.2	Para efectos de este ejemplo, el algoritmo retomará las estrategias algorítmicas básicas: (a) superficie generada a partir de función matemática gráfica, (b) ejemplo de diseño paramétrico de envolvente de estadio deportivo.
	1.3	Generar a partir del algoritmo previo una malla (<i>Mesh</i>): si se trabaja el ejemplo (a), utilizar un componente <i>Delaunay Mesh</i> para generar una malla a partir del traslado de puntos.
	1.4	Si se trabaja ejemplo (b) convertir la superficie NURBS producto de la función <i>Loft</i> en malla. Utilizar componente <i>Mesh Mesh Brep</i> o <i>Mesh Surface</i> para hacer la conversión.
2. Transformar topología de malla	2.1	A partir de que el algoritmo produce una malla como parámetro, es posible manipularla con las herramientas de transformación topológica de <i>Weaverbird</i> .
	2.2	<i>Weaverbird</i> incluye un conjunto diverso de algoritmos de transformación y subdivisión topológica
	2.3	Un ejemplo es generar aperturas en cada cara (<i>face</i>) de subdivisión de la malla. Utilizar componente <i>Wb Picture Frame</i> e implementar en el <i>input</i> correspondiente la distancia de apertura.
	2.4	Posterior a cualquier edición hecha con las herramientas de <i>Weaverbird</i> , es posible seguir transformándola de manera acumulativa. Por ejemplo, después de generar aberturas en cada cara, dar grosor a la malla con la función <i>Wb Mesh Thicken</i> . El segundo <i>input</i> del componente permite determinar el grosor de la malla.
	2.5	Existen diversas estrategias de subdivisión y suavizado de malla. Entre mayor densidad de geometría tenga la malla, mayor capacidad de suavizado puede tener.
	2.6	Los algoritmos de <i>Weaverbird</i> para subdividir la malla tienen diferentes métodos topológicos de subdivisión. <i>No se recomienda establecer más de tres pases de subdivisión en la malla, ya que puede provocar que la computadora se paralice.</i>
	2.7	Utilizar función de subdivisión <i>Wb Catmull-Clark Subdivision</i> y establecer cantidad de subdivisiones.

Nota. Elaboración propia (2020).

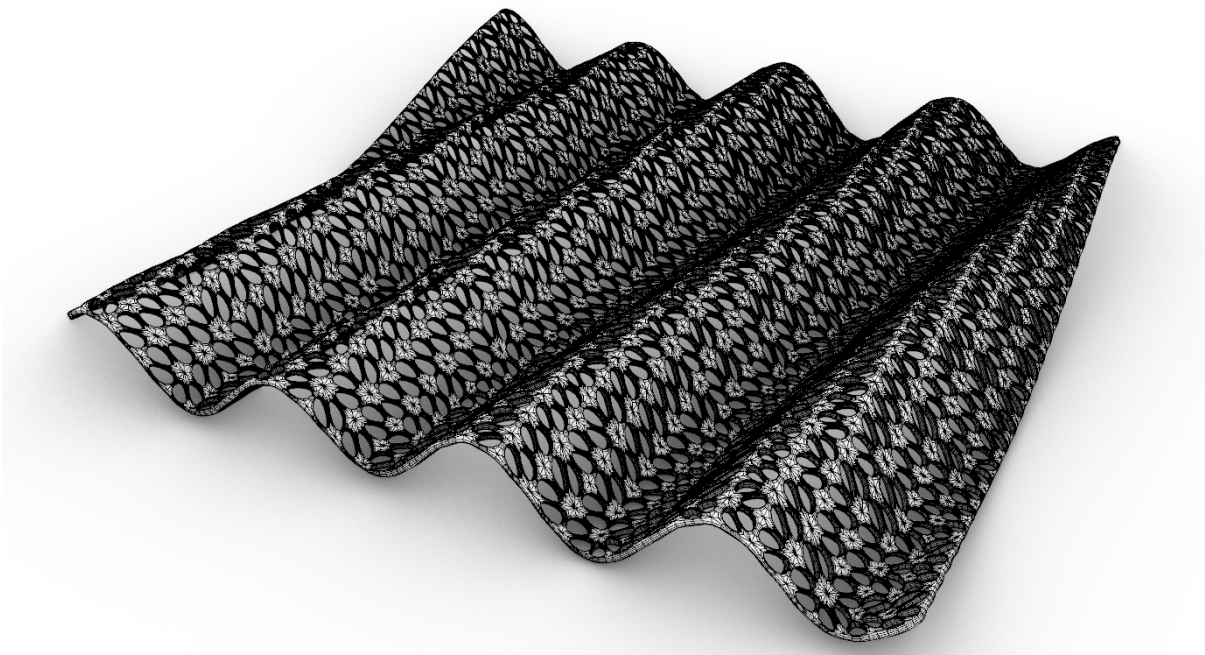


Figura 117. Resultado de algoritmo de *Grasshopper* “transformación topológica de una superficie con *Weaverbird*” Variante A: manipulación de superficie producto de estrategia “superficie generada a partir de función matemática gráfica”. Elaboración propia (2020).

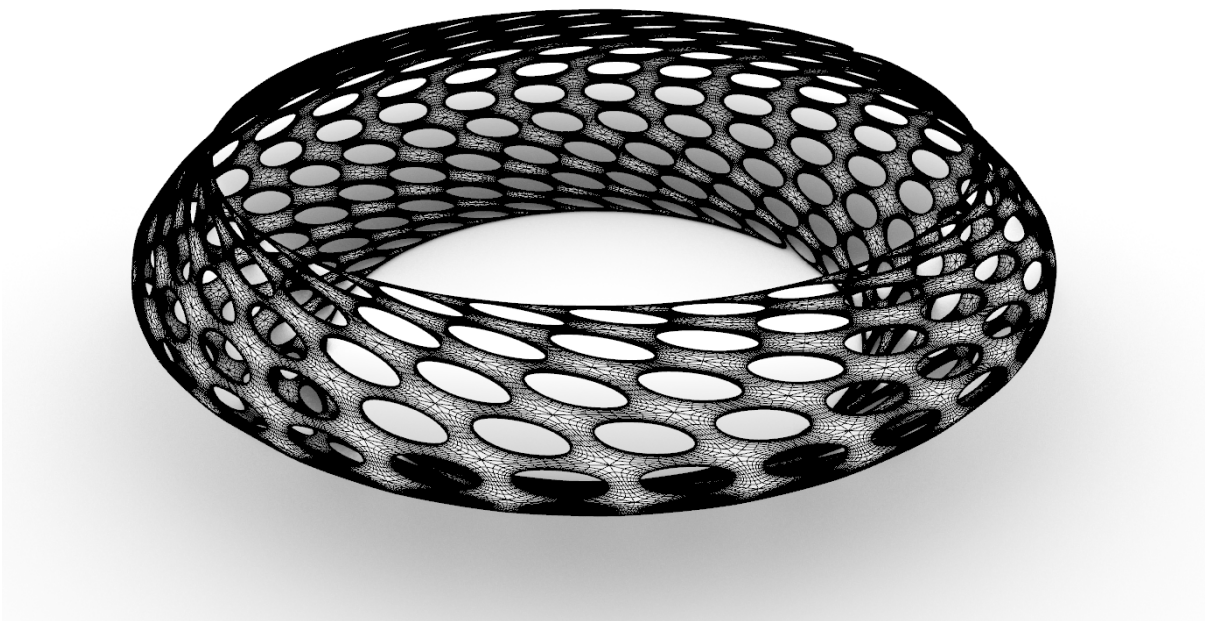


Figura 118. Resultado de algoritmo de *Grasshopper* “transformación topológica de una superficie con *Weaverbird*” Variante B: manipulación de superficie producto de estrategia “ejemplo de diseño paramétrico de envolvente de estadio deportivo”. Elaboración propia (2020).

3.4.2. Simulación de sistemas topológicos estructural con *Kangaroo*.

Como anteriormente se definió, *Kangaroo* es una herramienta algorítmica con aplicación en el análisis y diseño de aspectos topológicos para el análisis estructural (arquitectónico-constructivo). La utilidad permite generar simulaciones computacionales de sistemas de resorte de partículas; es decir, sistemas de cálculo y visualización de físicas.

Kangaroo se rige por la ley de Hooke de comportamiento elástico, por lo que los parámetros con los que se puede trabajar en la herramienta están relacionados con ese criterio (ver Tabla 13). Tedeschi (2014) enuncia la ley de Hooke de la siguiente forma: “ Los desplazamientos o el tamaño de deformación de un cuerpo (tratado como un resorte) es directamente proporcional a la fuerza o carga de deformación” (2014, p. 370).

Dentro de los estudios que se pueden realizar con la herramienta de *Kangaroo*, también se encuentran los fenómenos físicos que derivan de la simulación de la curva catenaria, la cual obedece a la deformación de un elemento flexible suspendido a partir sus extremos y afectado por una fuerza unitaria -la gravedad- (ver Tabla 13). De igual manera Tedeschi (2014) la define como “una curva que es formada por un perfectamente flexible, uniformemente denso e inextensible cable suspendido a partir de sus dos extremos” (2014, p. 375).

Tabla 13

Relación de las principales ecuaciones matemáticas y parámetros a considerar para trabajar con la herramienta algorítmica Kangaroo.

Ecuaciones matemáticas	
Ley de Hooke	
Ecuación	Descripción
$F=k*X$	<p>F: fuerza aplicada, comúnmente expresada en Newtons (N)</p> <p>k: es una constante positiva llamada rigidez, el valor de <i>k</i> depende de las propiedades geométricas transversales del material del cuerpo elástico. La constante <i>k</i> es comúnmente expresada en N/cm.</p> <p>X: es la variable de cambio de longitud o deformación del cuerpo (resorte), comúnmente expresado en cm.</p>

Ecuaciones matemáticas	
Curva catenaria	
Ecuación	Descripción
$y=a*\cosh(x/a)$	a: es la distancia desde el eje x hasta el punto de la curva con una línea tangente a la pendiente igual a cero.
Principales parámetros de la primera versión de <i>Kangaroo</i> basados en la ley de Hooke	
Connection	Conexiones: los resortes son conexiones lineales y elásticas. En <i>Kangaroo</i> el <i>input Connection</i> requiere de líneas, algún otro tipo de geometría introducida será marcada como nula.
Stiffness	Rigidez: de acuerdo con la ley de Hooke, el <i>input Stiffness</i> , establece la rigidez del resorte o el valor k . Mientras más alto es el valor de k , más baja es la deformación. La rigidez es determinada tanto por las propiedades del material, como por el área transversal del resorte.
Damping	Amortiguamiento: el <i>input Damping</i> influencia la velocidad de deformación, sin influenciar el cambio de longitud. El amortiguamiento preestablecido de <i>Kangaroo</i> es el valor diez.
Rest Lenght	Longitud de descanso: es la longitud que el resorte intenta alcanzar una vez que las cargas se quitan. El <i>input Rest-Length</i> es esencial para simular el comportamiento de diferentes materiales.

Nota. Información recuperada de AAD_ *Algorithmis-Aided Design: Parametric Strategies Using Grasshopper* (Tedeschi, 2014, pp. 371, 373, 375).

La herramienta *Kangaroo* ha pasado por dos versiones de desarrollo, los parámetros mostrados obedecen a la primera versión realizada. A diferencia de la edición más antigua, la segunda versión cambia la manera de manejar los parámetros e independiza las funciones a trabajar con el fin de ampliar el desempeño de la herramienta y hacerla más flexible. Cada función obedece a una estrategia elástica del objeto a simular y es medida con el parámetro de fuerza o *input Strength*.

A continuación, se ejemplifican dos estrategias afectación de objetos influenciados por las leyes de la elasticidad determinadas por Robert Hooke. El primer ejemplo simula una estructura extensible sencilla, en la que una superficie será afectada por fuerzas de tensión con el fin de encontrar posibilidades formales en un producto arquitectónico (ver Figura 119 y Tabla 13).

El segundo algoritmo simula y reinterpreta digitalmente uno de experimentos realizados por el ingeniero Sergio Musmeci para analizar la deformación de una superficie suspendida como una película de jabón (ver Figura 122 y Tabla 14).

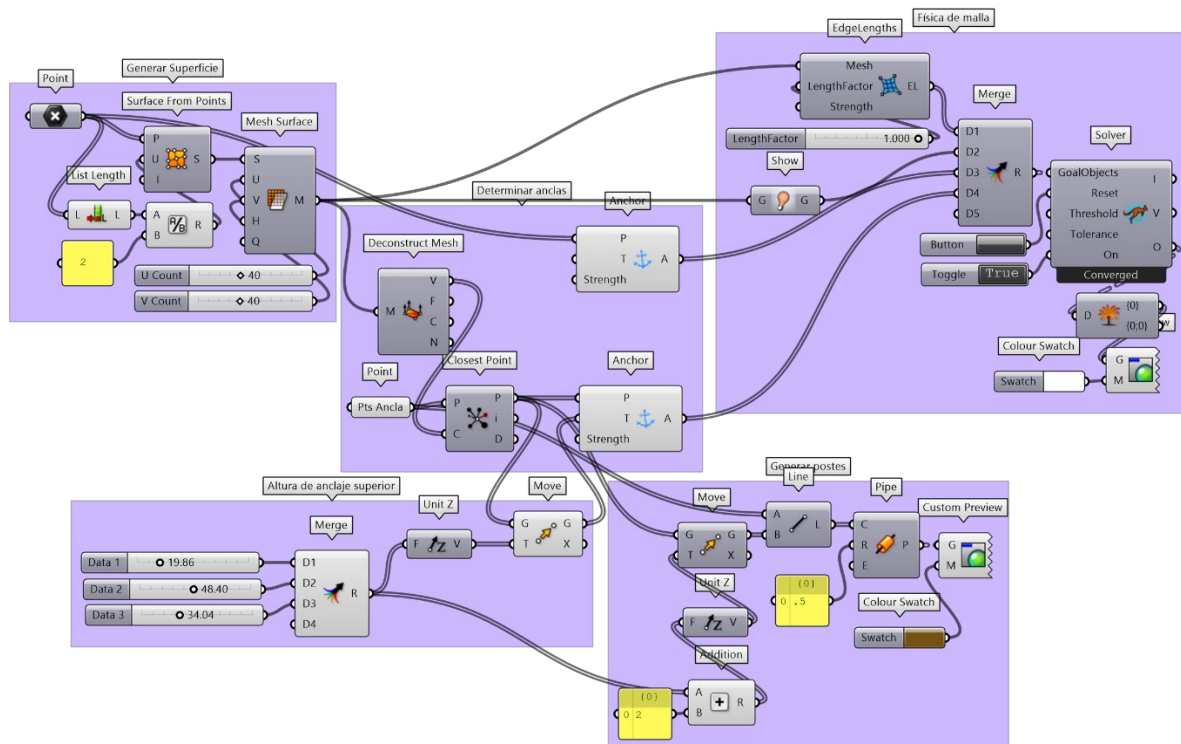


Figura 119. Algoritmo de Grasshopper “simulación de sistemas topológicos estructurales con Kangaroo”. Variante: estructura extensible. Elaboración propia (2020).

Tabla 13

Síntesis descriptiva de algoritmo: simulación de sistemas topológicos estructurales con Kangaroo. Variante: estructura extensible.

Sección	Paso	Descripción
1. Elaborar superficie inicial	1.1	Generar una superficie, puede ser dibujada directamente en <i>Rhinceros</i> ®, o desarrollada algorítmicamente en <i>Grasshopper</i> .
	1.2	Si la superficie fue generada como NURBS, convertirla a <i>Mesh</i> . Utilizar funciones <i>Mesh Brep</i> o <i>Mesh Surface</i> como componente de conversión.
2. Extraer puntos de la superficie que funcionaran como anclas	2.1	Extraer de la malla (<i>Mesh</i>) generada, los puntos necesarios para funcionar como anclas. La extracción puede suceder con las funciones <i>Deconstruct Mesh</i> o <i>Wb Vértices Component</i> .
	2.2	Extraer función <i>Anchor</i> de <i>Kangaroo</i> para reconocer los puntos extraídos como anclas. Conectar en <i>input P</i> los puntos que

Sección	Paso	Descripción
		funcionarán como anclas Es posible determinar la fuerza de anclaje en el <i>input Strength</i> del componente.
3. Determinar puntos adicionales y manipulables que funcionaran como anclas	3.1	Dibujar directamente en <i>Rhinoceros</i> ® puntos sobre la superficie y referenciarlos en un parámetro <i>Point</i> , o directamente generarlos en <i>Grasshopper</i> con la misma herramienta.
	3.2	Desarrollar estrategias de traslado de los puntos con el componente <i>Move</i> . Es importante que la estrategia generada tenga capacidad de moverlos en eje <i>Z</i> , para determinar la altura de empuje de la superficie.
	3.3	Extraer función <i>Anchor</i> de <i>Kangaroo</i> para reconocer los puntos generados a funcionar como anclas. Conectar en <i>input P</i> los puntos iniciales (antes del traslado) a referencias como anclas. Es posible determinar la fuerza de anclaje en el <i>input Strength</i> del componente.
	3.4	Implementar en el <i>input T</i> de la función <i>Anchor</i> el objetivo de empuje de los puntos a anclar. Es importante conectar en este <i>input</i> el traslado de los puntos.
4. Generar postes de empuje	4.1	Para efectos visuales, puede generarse los postes que representan el anclaje de empuje de la superficie. Relacionar los puntos ancla iniciales (antes del traslado) con los trasladados a través de una línea (función <i>Line</i>).
	4.2	Generar un tubo a partir de las líneas desarrolladas con el componente <i>Pipe</i> .
5. Generar la simulación de malla	5.1	Construir la simulación de transformación de superficie por fuerzas físicas y criterios de elasticidad.
	5.2	Conectar la malla (<i>Mesh</i>) resultante al componente de conexiones resorte de <i>Kangaroo</i> . La función <i>Edge Lenghts</i> se encargará de interpretar las relaciones de conexión de la malla para su funcionamiento como elementos resorte. Es posible determinar la fuerza de rigidez de la malla través de la relación de los <i>inputs LenghtFactor</i> y <i>Strength</i> del componente.
	5.3	Extraer función <i>Show</i> para visualizar en tiempo real la simulación de la malla. Conectar la malla inicial al <i>input G</i> del componente.
	5.4	Conectar todos los elementos generados en el simulador. Extraer la función <i>Solver</i> de <i>Kangaroo</i> . Conectar todas las funciones <i>Anchor</i> , <i>Edge Lenghts</i> y <i>Show</i> en el <i>input GoalObjects</i> del componente <i>Solver</i> .
	5.5	Establecer un parámetro dinámico <i>Button</i> en el <i>input Reset</i> de la función <i>Solver</i> para restablecer la simulación cada que se presione el botón. Implementar un parámetro <i>Boolean Toggle</i> en el <i>input On</i> del <i>Solver</i> para habilitar la simulación.

Nota. Elaboración propia (2020).

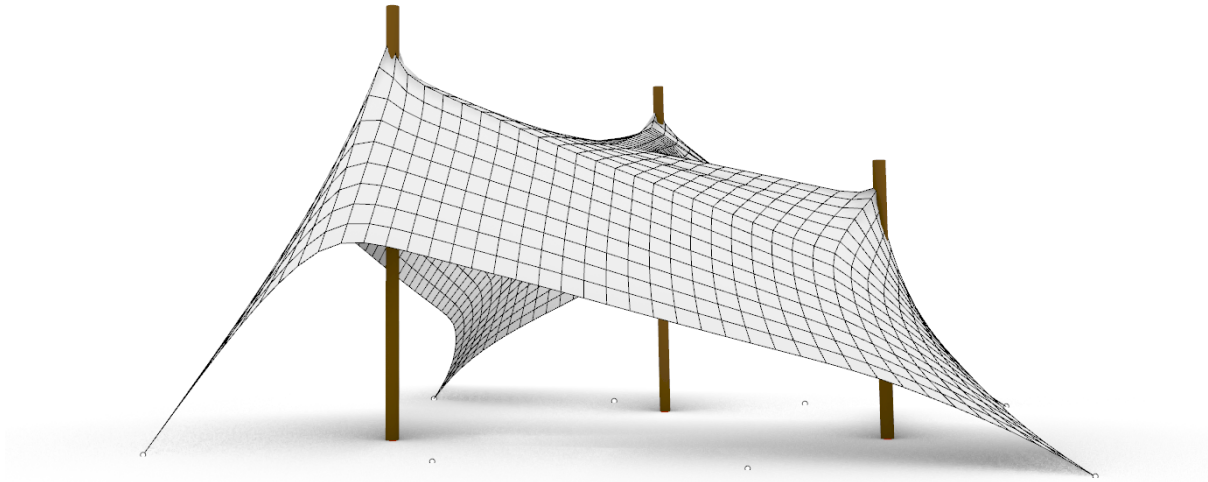


Figura 120. Resultado de algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*” Variante: estructura extensible. Elaboración propia (2020).

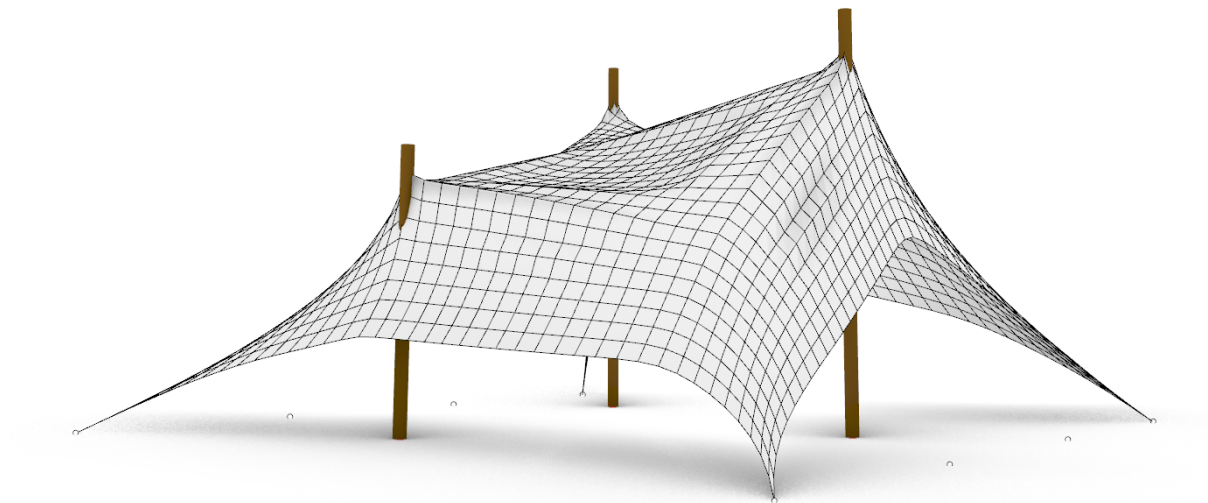


Figura 121. Resultado de algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*” Variante: estructura extensible. Elaboración propia (2020).

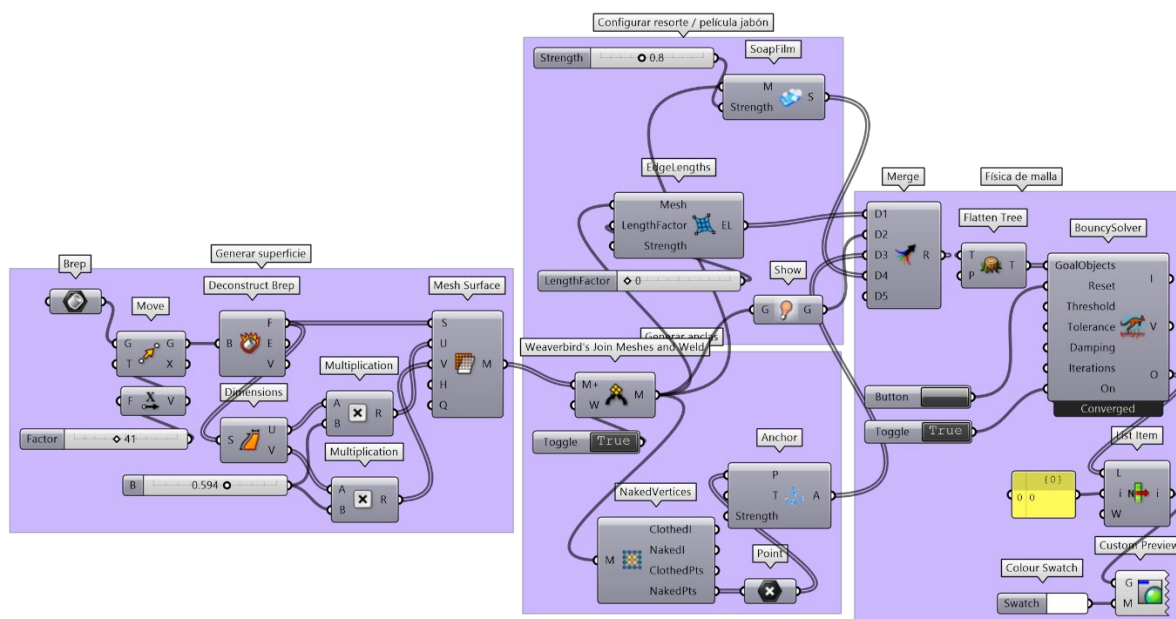


Figura 122. Algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*”. Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).

Tabla 14

Síntesis descriptiva de algoritmo: simulación de sistemas topológicos estructurales con *Kangaroo*. Variante: superficie suspendida “película de jabón”.

Sección	Paso	Descripción
1. Elaborar polisuperficie inicial	1.1	Generar una polisuperficie, puede ser dibujada directamente en <i>Rhino</i> ®, o desarrollada algorítmicamente en <i>Grasshopper</i> . Una manera rápida de hacerla es a través del comando de <i>Rhino</i> ® <i>Box</i> .
	1.2	Referenciar polisuperficie NURBS dibujada en <i>Rhino</i> ® con un parámetro <i>Brep</i> (<i>Boundary Representation</i>).
	1.3	Desplazar polisuperficie NURBS referenciada (<i>Brep</i>) con la función <i>Move</i> para separarla del producto final simulado.
	1.4	Deconstruir polisuperficie NURBS referenciada (<i>Brep</i>) para manipular cada una de las superficies que la componen y poder ser convertidas a malla (<i>Mesh</i>)
	1.5	Convertir polisuperficie NURBS a <i>Mesh</i> . Utilizar funciones <i>Mesh Surface</i> como componente de conversión.
	1.6	Garantizar que las subdivisiones de las caras de la malla resultante coincidan en su entramado de caras, bordes y vértices. Para lograr este objetivo, tomar como referencia las dimensiones de ancho y largo de cada superficie para multiplicarlas por un factor fraccionario. Tomar el resultado como parámetro de dimensión <i>U</i> y <i>V</i> de la malla en los <i>inputs</i> correspondientes del componente <i>Mesh Surface</i> .

Sección	Paso	Descripción
2. Extraer puntos de la superficie que funcionaran como anclas	2.1	Primeramente, soldar las mallas resultantes para que convierta en una sola antes de extraer los puntos que funcionarán como anclas. Utilizar el componente <i>Wb Join Meshes and Weld</i> y establecer el <i>input W</i> en verdadero (<i>True</i>) para activar la unión.
	2.2	La malla unida será el parámetro para las futuras acciones.
	2.3	Extraer todos los vértices de los bordes de la malla (todos ellos funcionarán como ancla). Utilizar la función <i>Naked Vértices</i> de <i>Kangaroo</i> para obtener estos puntos.
	2.4	Reconocer todos los puntos extraídos como anclas. Extraer función <i>Anchor</i> de <i>Kangaroo</i> , y conectar en el <i>input P</i> los puntos. Es posible determinar la fuerza de anclaje en el <i>input Strength</i> del componente.
3. Configurar malla como resortes de conexión	3.1	Construir la simulación de transformación de superficie por fuerzas físicas y criterios de elasticidad.
	3.2	Conectar la malla (<i>Mesh</i>) resultante al componente de conexiones resorte de <i>Kangaroo</i> . La función <i>Edge Lenghts</i> se encargará de interpretar las relaciones de conexión de la malla para su funcionamiento como elementos resorte. Es posible determinar la fuerza de rigidez de la malla través de la relación de los <i>inputs LenghtFactor</i> y <i>Strength</i> del componente. Establecer en el <i>input Strength</i> el valor cero.
	3.3	Implementar en la configuración, el comportamiento de la malla como película de jabón. Extraer función <i>Soapfilm</i> de <i>Kangaroo</i> y conectar la malla en el <i>input M</i> .
	3.4	Extraer función <i>Show</i> para visualizar en tiempo real la simulación de la malla. Conectar la malla inicial al <i>input G</i> del componente.
4. Generar la simulación de malla	4.1	Conectar todos los elementos generados en el simulador. Extraer la función <i>BouncySolver</i> de <i>Kangaroo</i> . Conectar todas las funciones <i>Anchor</i> , <i>Edge Lenghts</i> , <i>Soapfilm</i> y <i>Show</i> en el <i>input GoalObjects</i> del componente <i>BouncySolver</i> . Establecer un filtro <i>Flatten Tree</i> entre las conexiones y el simulador.
	4.2	Establecer un parámetro dinámico <i>Button</i> en el <i>input Reset</i> de la función <i>Solver</i> para restablecer la simulación cada que se presione el botón. Implementar un parámetro <i>Boolean Toggle</i> en el <i>input On</i> del <i>Solver</i> para habilitar la simulación.

Nota. Elaboración propia (2020).

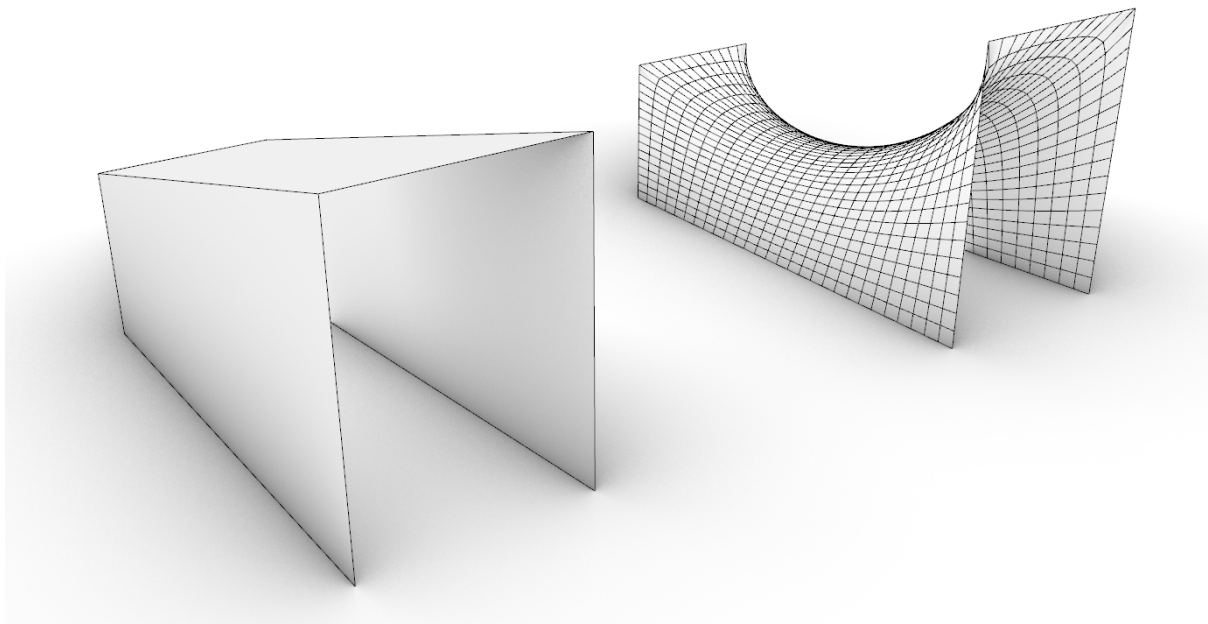


Figura 123. Resultado de algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).

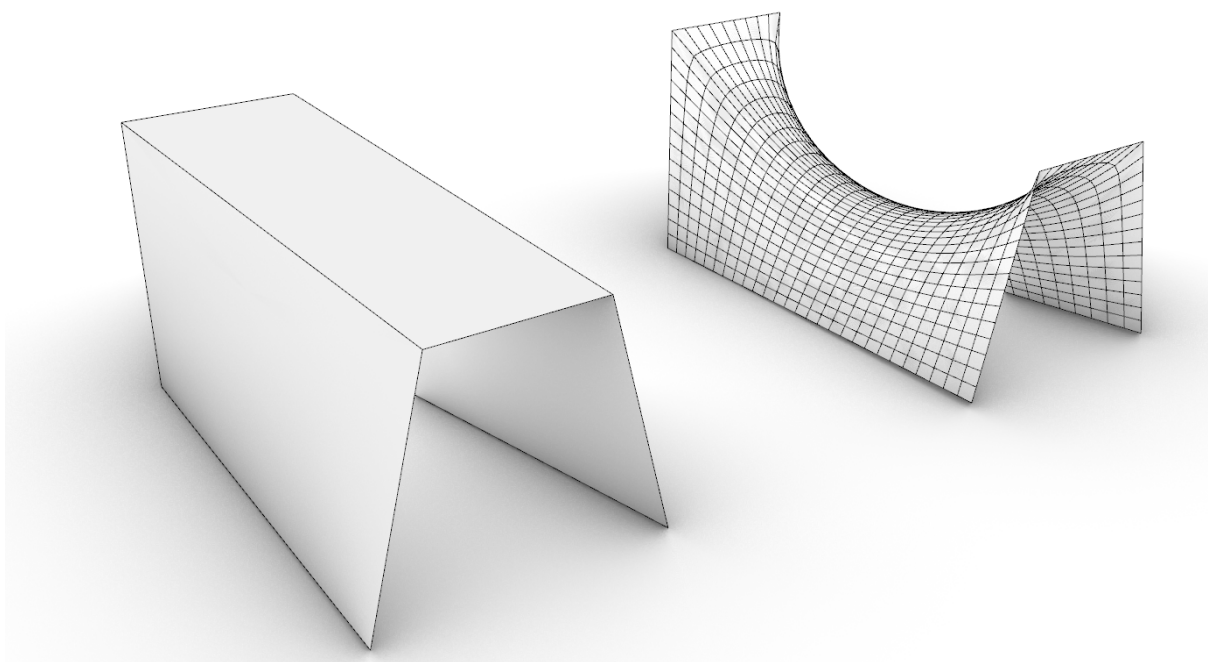


Figura 124. Resultado de algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).

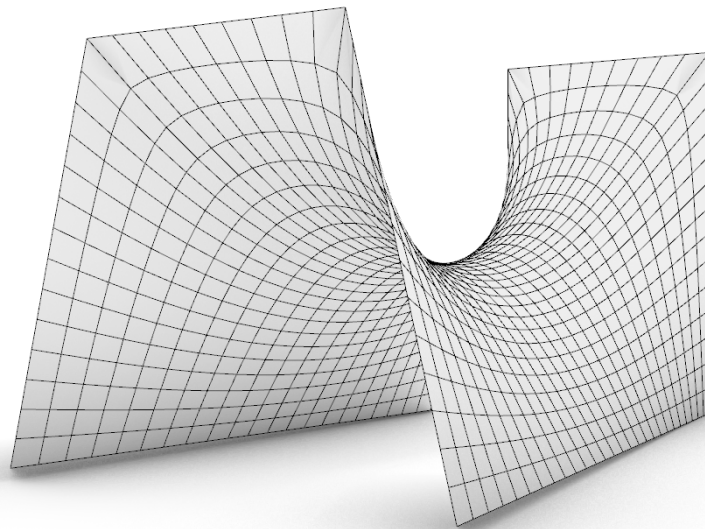


Figura 125. Resultado de algoritmo de *Grasshopper* “simulación de sistemas topológicos estructurales con *Kangaroo*” Variante: superficie suspendida “película de jabón”. Elaboración propia (2020).

3.4.3. Sistemas de agregación compleja con *Wasp*.

La presente estrategia está basada en la herramienta de *Grasshopper Wasp* para generar sistemas de agregación compleja a través de un mismo elemento discreto con propiedades topológicas y geométricas idénticas. *Wasp* trabaja con parámetros de *geometría*, *ubicación de conexiones* y *orientación*. Los sistemas topológicos generados por *Wasp* siguen estrategias de *agregación estocástica* y *agregación impulsada por campos (field-driven aggregation)*.

Wasp toma como recurso un elemento geométrico con propiedades de orientación y conexión establecidas por el usuario. En el sistema de agregación se establecen condiciones de acoplamiento para poder desarrollar la repetición seriada de manera agregada. Las reglas de acoplamiento pueden ser determinadas por la computadora o por el usuario.

El objetivo es desarrollar entornos espaciales arquitectónicos que promuevan el análisis de los aspectos cualitativos (fenomenológicos y semiológicos) que se analizan en la metodología de diseño arquitectónico AP del presente trabajo de investigación. El resultado topológico y geométrico obtenido es el punto de partida, para posteriormente, desarrollar análisis cuantitativos estructurales y de adaptabilidad ambiental (ver Figura 126, 127 y Tabla 15).

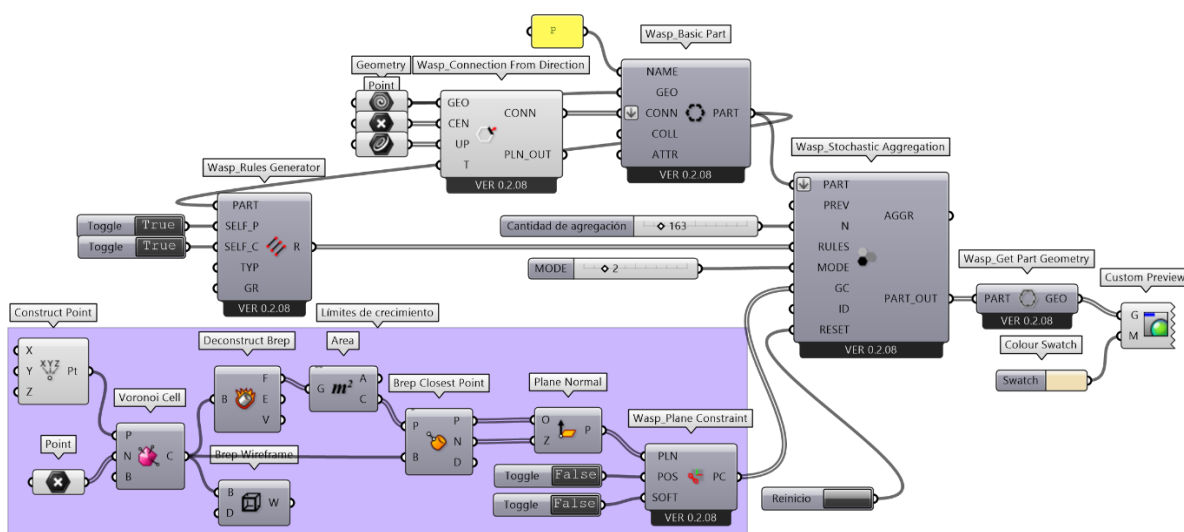


Figura 126. Algoritmo de *Grasshopper* “sistemas de agregación compleja con *Wasp*”. Variante A: reglas de agregación y conexión generadas por la computadora. Elaboración propia (2020).

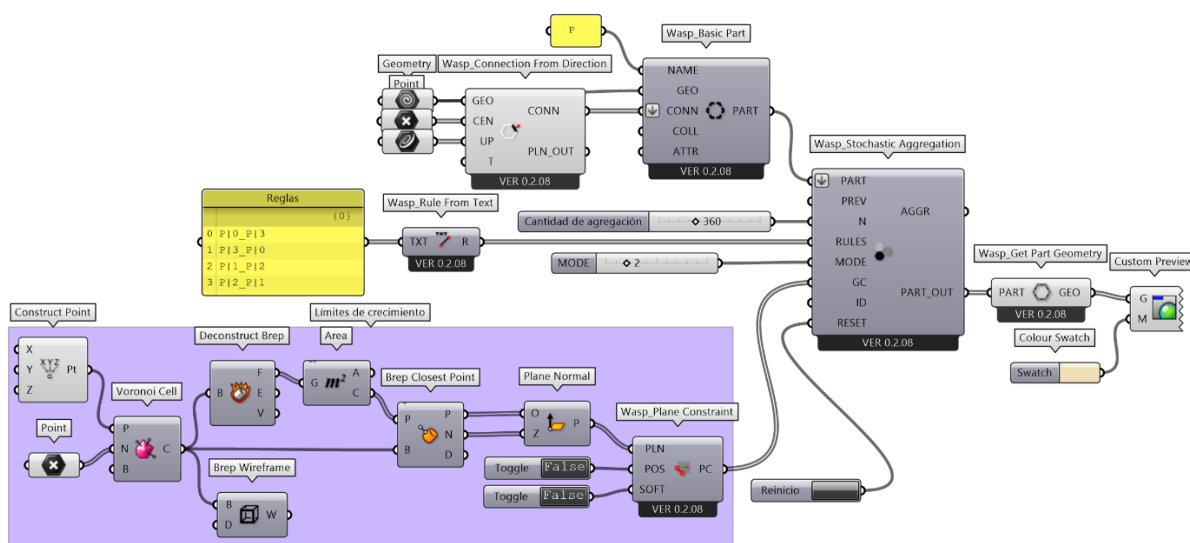


Figura 127. Algoritmo de *Grasshopper* “sistemas de agregación compleja con *Wasp*”. Variante B: reglas de agregación y conexión generadas por el usuario. Elaboración propia (2020).

Tabla 15

Síntesis descriptiva de algoritmo: sistemas de agregación compleja con Wasp.

Sección	Paso	Descripción
1. Establecer la geometría a multiplicar	1.1	Dibujar directamente en <i>Rhinoceros</i> ® (puede desarrollarse en <i>Grasshopper</i>) una geometría para multiplicar de forma agregada.
	1.2	Establecer en la geometría dibujada los puntos de conexión con los elementos multiplicados. Dibujar en <i>Rhinoceros</i> ® puntos en la ubicación deseada, hay que considerar que los puntos determinarán la conexión con otro elemento.
	1.3	Determinar la orientación que tendrá la conexión. Dibujar una línea con origen en los puntos previamente dibujados. Cada punto debe tener una línea y es importante dibujarla de acuerdo con el sentido de conexión que tendrá. El punto representa el origen y la línea el eje de referencia (x o y) de un plano de trabajo (<i>CPlane</i>).
	1.4	Referenciar cada geometría dibujada con los siguientes parámetros de <i>Grasshopper</i> : <i>Geometry</i> para referenciar la geometría a multiplicar, <i>Point</i> para los puntos de conexión (tener presente el orden de referencia), y finalmente <i>Line</i> para las líneas que representan el sentido del plano de trabajo. Las líneas deben seleccionarse en el mismo orden de los puntos.
	1.5	Extraer el componente <i>Wasp_Connection From Direction</i> para dar reconocimiento a la geometría referenciada como puntos de conexión de <i>Wasp</i> . Conectar el parámetro de geometría en el <i>input GEO</i> , el parámetro de puntos en el <i>input CEN</i> y el de curva en el <i>input UP</i> .
2. Dar reconocimiento de la geometría a multiplicar como una Parte- <i>Wasp</i> .	2.1	Extraer función <i>Wasp_Basic Part</i> para dar reconocimiento de la geometría dibujada como una parte de <i>Wasp</i> . Establecer un nombre de identificación para la geometría en el <i>input NAME</i> . Conectar el parámetro de <i>Geometry</i> en el <i>input GEO</i> . Conectar el <i>output CONN</i> de la función <i>Wasp_Connection From Direction</i> en el <i>input CONN</i> del componente <i>Wasp_Basic Part</i> .
3. Implementar el generador de reglas de <i>Wasp</i> (variante A)	3.1	Utilizar el componente <i>Wasp_Rules Generator</i> como la función para que la computadora establezca las reglas de conexión de los elementos a multiplicar. Conectar en el <i>input PART</i> , el <i>output PART</i> de la función <i>Wasp_Basic Part</i> .
	3.2	Establecer condiciones opcionales de conexión. El <i>input SELF_P</i> permite crear reglas automáticas entre las conexiones de una misma parte. El <i>input SELF_C</i> permite crear reglas automáticas entre las conexiones de elementos con una misma identidad.
4. Sustituir el generador automático de reglas por condiciones personalizadas por el usuario (variante B: paso opcional)	4.1	Si se desea personalizar las reglas de conexión, extraer el componente <i>Wasp_Rules From Text</i> . Implementar en el <i>input TXT</i> un parámetro dinámico de <i>Panel</i> . Escribir en el panel (desactivar propiedad <i>Multiline Data</i>) las condiciones de conexión.
	4.2	Cada renglón del panel es una regla de conexión. Para escribir la regla debe de definirse que parte se desea conectar con otra. Para determinar la parte de <i>Wasp</i> anteponer la identidad de la parte, seguido de un guión vertical y el número que representa el orden del elemento a seleccionar (ejemplo: P 0). Posteriormente se escribe guion bajo para relacionar con otra parte de <i>Wasp</i> (ejemplo: P 0_P 3).

Sección	Paso	Descripción
5. Generar de manera opcional límites espaciales de crecimiento	5.1	Es posible referenciar o desarrollar una geometría en Grasshopper para reconocerla como límite de crecimiento.
	5.2	Dibujar geometría <i>Brep</i> , descomponerla para extraer el plano de orientación (<i>CPlane</i>) de cada superficie que la constituye.
	5.3	Extraer la función <i>Wasp_Plane Constraint</i> para dar reconocimiento de la geometría como límite de crecimiento. Conectar los planos de orientación extraídos de la geometría límite en el <i>input PLN</i> .
6. Implementar el mecanismo de agregación estocástica	6.1	Utilizar el componente <i>Wasp_Stochastic Aggregation</i> como mecanismo de agregación estocástica.
	6.2	Conectar el <i>output PART</i> de la función <i>Wasp_Basic Part</i> en el <i>input PART</i> del componente <i>Wasp_Stochastic Aggregation</i> .
	6.3	Conectar un <i>Number Slider</i> o un <i>Panel</i> en el <i>input N</i> , con el objetivo de determinar la cantidad de elementos a agregar.
	6.4	Conectar el <i>output R</i> de la función <i>Wasp_Rules Generator</i> o <i>Wasp_Rules From Text</i> en el <i>input RULES</i> del componente <i>Wasp_Stochastic Aggregation</i> .
	6.5	Establecer el modo de funcionamiento de la geometría límite en el <i>input MODE</i> .
	6.6	Conectar el <i>output PC</i> de la función <i>Wasp_Plane Constraint</i> en el <i>input GC</i> del componente <i>Wasp_Stochastic Aggregation</i> .
	6.7	Conectar un botón para reiniciar el mecanismo de agregación estocástica en el <i>input RESET</i> .
7. Ver animación del resultado	6.7	Conectar una función <i>Wasp_Get Part Geometry</i> a partir del <i>output PART_OUT</i> del componente <i>Wasp_Stochastic Aggregation</i> para visualizar la geometría resultante.
	7.1	Ver animación del resultado de crecimiento en el siguiente vínculo web: https://youtu.be/d154dbPt-6Y

Nota. Elaboración propia (2020).

Reglas de crecimiento estocástico.

Agregación:
Aleatoria

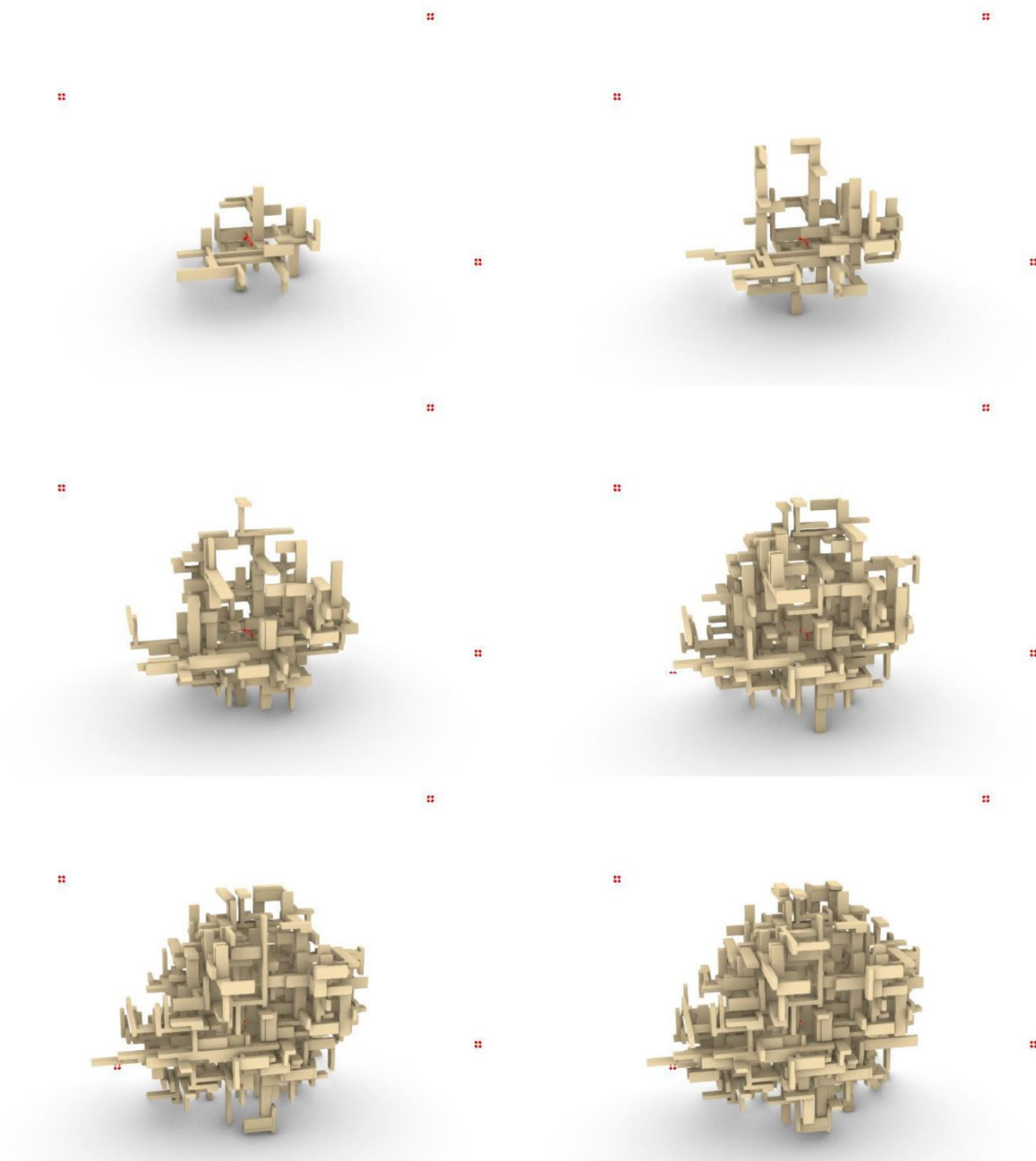


Figura 128. Resultado de algoritmo de *Grasshopper* “sistemas de agregación compleja con *Wasp*”.

Variante A: reglas de agregación y conexión generadas por la computadora. Ver animación en el siguiente vínculo: <https://youtu.be/d154dbPt-6Y>. Elaboración propia (2020).

Reglas de crecimiento estocástico.

Agregación:

P|0_P|3

P|3_P|0

P|1_P|2

P|2_P|1

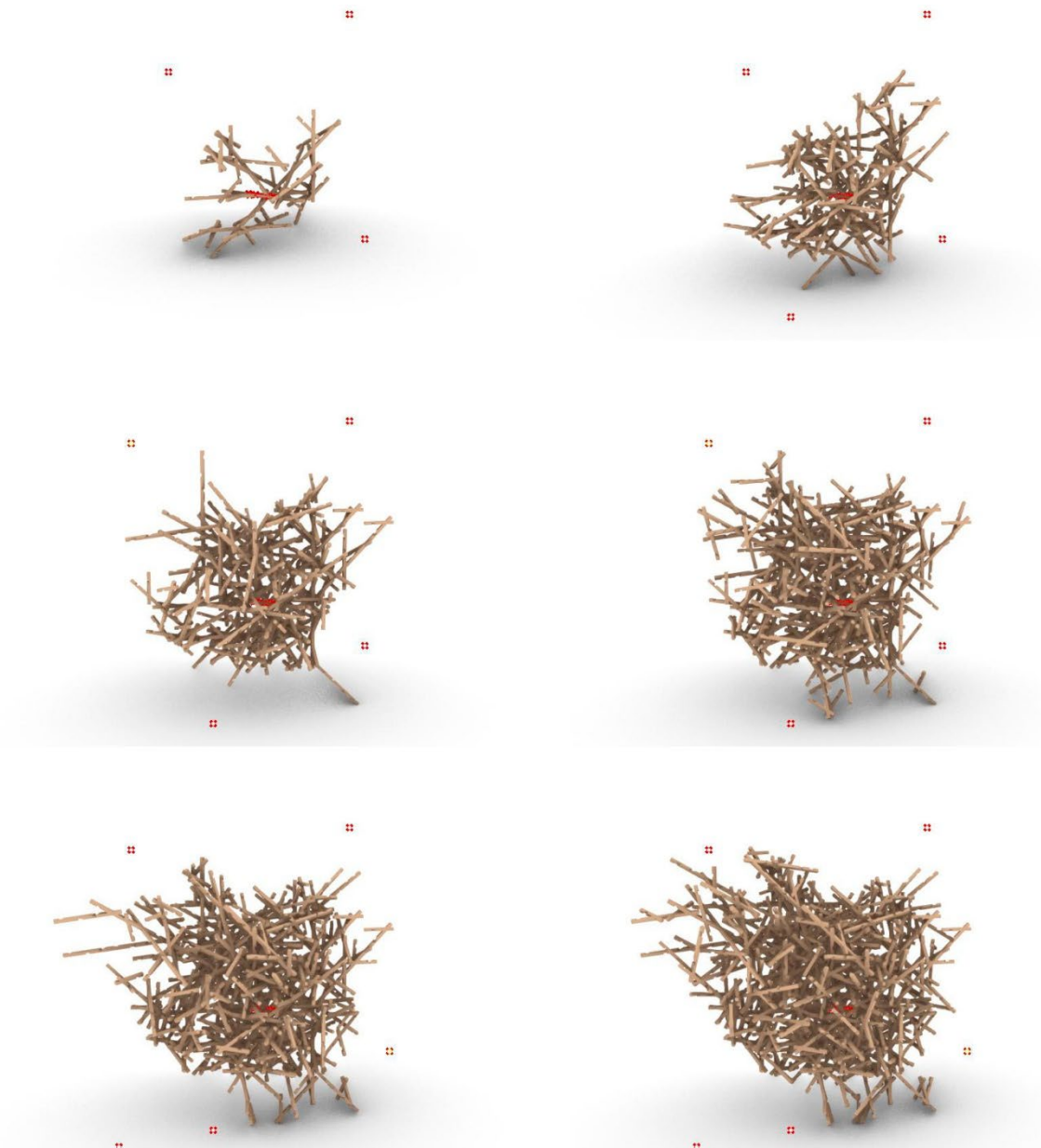


Figura 129. Resultado de algoritmo de *Grasshopper* “sistemas de agregación compleja con *Wasp*”.

Variante B: reglas de agregación y conexión generadas por el usuario. Ver animación en el siguiente

vínculo: <https://youtu.be/d154dbPt-6Y>. Elaboración propia (2020).

3.4.4. Simulación para análisis de radiación solar sobre una superficie con *Ladybug*.

Ladybug es parte de un conjunto de herramientas algorítmicas para el análisis solar y termo-energético. Las principales herramientas que conforman este grupo son: *Ladybug*, *Honeybee*, *Butterfly* y *Dragonfly*; cada una de estas, se encuentra enfocada en diferentes aspectos de análisis como la radiación solar, el análisis térmico-energético, el flujo de aire, así como las islas de calor.

Específicamente *Ladybug* es la herramienta algorítmica que se encarga del análisis y simulación de radiación solar sobre la envolvente de edificios y superficies de un contexto determinado. *Ladybug* recoge datos de los principales formatos de archivos climáticos: EPW, STAT y DDY, de los cuales extrae información como magnitud vectorial de radiación solar directa, difusa, global, humedad relativa, temperatura bulbo seco, velocidad y dirección de viento, presión atmosférica, radiación infrarroja, entre otros.

En esta sección se expondrán estrategias para efectuar simulación de radiación solar anual acumulada, así como el estudio de incidencia solar directa sobre una edificación residencial (ver Figuras 133, 136 y Tabla 17 y 18). Las unidades que arroja la simulación son Wh/m^2 para los resultados de radiación solar, y h (horas) para el análisis de incidencia solar directa. La construcción es realizada por el despacho de Arquitectura SPAU (Servicios Profesionales de Arquitectura y Urbanismo, S.C.), específicamente por los arquitectos Joaquín Rodríguez y Aarón Onchi.

El proyecto se llama Casa RV (2017), fue realizado y construido en la ciudad de Hermosillo, Sonora, México; sin embargo, para efectos de los presentes análisis y simulaciones se modificará la ubicación a la ciudad de Mexicali, Baja California, México. Las principales características de la residencia se describen a continuación (ver Figuras 130-132 y Tabla 16).

También se mostrará una simulación de radiación solar anual, con la cual, se recuperarán los resultados y se utilizarán como referencia para aberturas en una superficie. La abertura de la superficie será indirectamente proporcional a la cantidad de radiación solar anual acumulada en la superficie (ver Figura 139 y Tabla 19).



Figura 130. Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de <https://www.spaumx.com/casa-criv>.

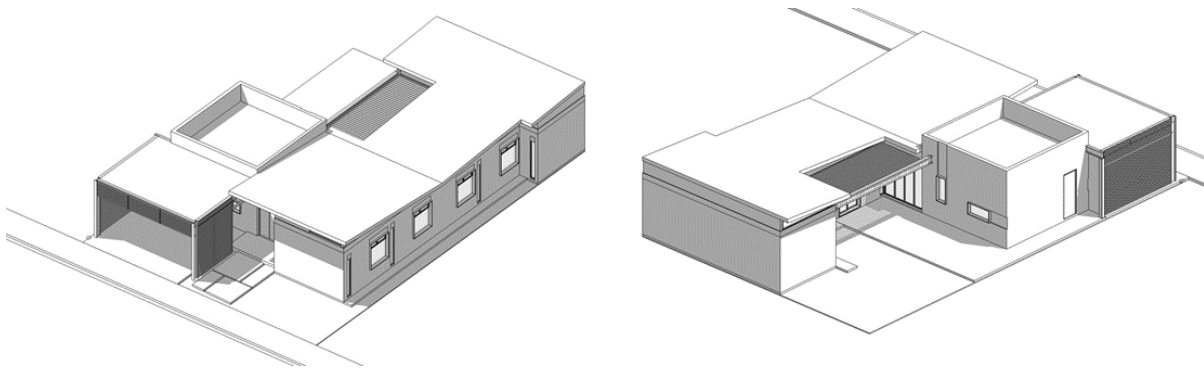


Figura 131. Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de <https://www.spaumx.com/casa-criv>.



Figura 132. Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C (SPAU). 2015-2017. Hermosillo, Sonora, México. SPAU (2017) [Imagen]. Recuperado de <https://www.spaumx.com/casa-crv>.

Tabla 16

Casa RV. Servicios Profesionales de Arquitectura y Urbanismo, S.C. (SPAU). Hermosillo, Sonora, México. 2015-2017. Relación de características de edificación residencial para análisis y simulación (reubicación por diseño a ciudad de Mexicali, B.C., México).

Proyecto			
Casa RV			
Tipo de proyecto			
Residencia. Vivienda Unifamiliar			
Cantidad de niveles			
1			
Superficie de construcción (m2)			
165			
Año			
Proyecto: 2015		Construcción: 2017	
Página web			
https://www.spaumx.com/casa-crv			
Ubicación			
País	Estado	Municipio	Ciudad
México	Baja California	Mexicali	Mexicali
Zona Climática (California Energy Comission)			

Proyecto	
Zona climática 15, CA, USA (latitud 32.8° Norte)	
Zona Climática (IECC ASHRAE Climate Zones)	
Zona climática 2B, Phoenix AZ, USA	
Altura sobre el nivel del mar	
1m	

Nota. Elaboración propia (2020).

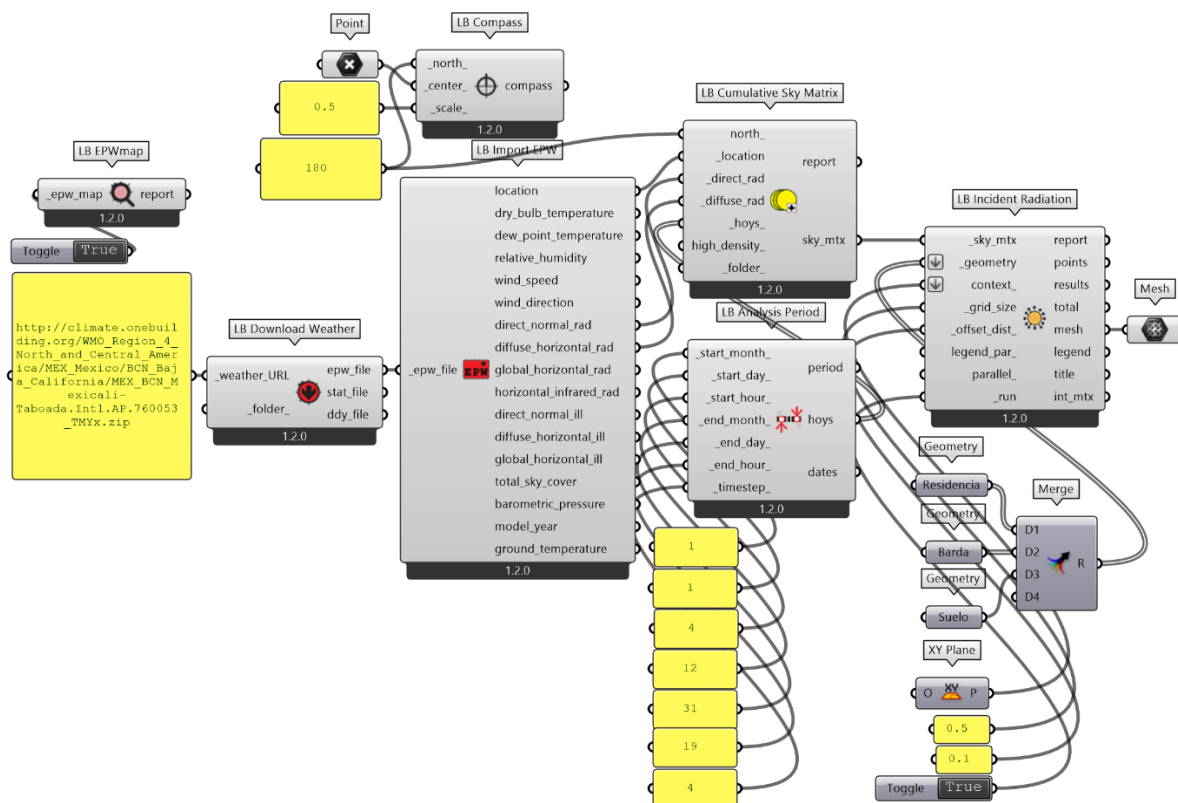


Figura 133. Algoritmo de Grasshopper “simulación para análisis de radiación solar sobre una superficie con Ladybug”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

Tabla 17

Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre una superficie con Ladybug. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.

Sección	Paso	Descripción
1. Determinar la geometría a analizar	1.1	Referenciar en <i>Grasshopper</i> la geometría a analizar. Utilizar parámetros <i>Geometry</i> para referenciar la geometría de las edificaciones y la superficie del suelo-contexto a analizar.
2. Importar archivo climático	2.1	Existen varias maneras para importar un archivo climático en el algoritmo de <i>Grasshopper</i> mediante <i>Ladybug</i> .
	2.2	Si se tiene el archivo, utilizar un parámetro <i>File Path</i> para importar el archivo a <i>Grasshopper</i> .
	2.3	Si no se tiene el archivo, se puede descargar directamente en <i>Grasshopper</i> con el componente de <i>Ladybug</i> llamado <i>LB Download Weather</i> . Introducir con un panel la dirección URL del archivo climático a descargar, y conectar el panel en el <i>input _weather_URL</i> . El tipo de archivos climáticos que ofrece la función de descarga son: EPW, STAT y DDY.
3. Importar datos de archivo climático	3.1	El archivo climático más habitual para usar es tipo EPW. Utilizar función <i>LB Import EPW</i> para extraer los datos que contiene el archivo.
	3.2	En este algoritmo se utilizarán los <i>outputs location</i> (ubicación geográfica), <i>direct_normal_rad</i> (radiación directa), <i>diffuse_horizontal_rad</i> (radiación difusa).
4. Calcular la radiación acumulada de un periodo determinado de tiempo	4.1	Extraer el componente <i>LB Cumulative Sky Matrix</i> .
	4.2	Implementar el norte (<i>input north_</i>) a través de un valor numérico. En dirección contra reloj, el valor 0 es norte, 90 oeste, 180 sur y 270 este.
	4.3	Establecer la ubicación, conectar <i>output location</i> del componente <i>LB Import EPW</i> en el <i>input _location</i> de la función <i>LB Cumulative Sky Matrix</i> .
	4.4	Establecer radiación directa, conectar <i>output direct_normal_rad</i> del componente <i>LB Import EPW</i> en el <i>input _direct_rad</i> de la función <i>LB Cumulative Sky Matrix</i> .
	4.5	Establecer radiación difusa, conectar <i>output diffuse_horizontal_rad</i> del componente <i>LB Import EPW</i> en el <i>input _diffuse_rad</i> de la función <i>LB Cumulative Sky Matrix</i> .
	4.6	Determinar el periodo de tiempo de análisis. Extraer función <i>LB Analysis Period</i> , determinar con valores numéricos el periodo y conectar <i>output hoys</i> en el <i>input _hoys_</i> del componente <i>LB Cumulative Sky Matrix</i> .
5. Generar simulación de radiación solar	5.1	Extraer componente <i>LB Incident Radiation</i> para generar simulación.
	5.2	Conectar <i>output sky_mtx</i> del componente <i>LB Cumulative Sky Matrix</i> en el <i>input _sky_mtx</i> de la función <i>LB Incident Radiation</i> .
	5.3	Conectar los parámetros de geometría de las edificaciones y suelo a analizar en el <i>input _geometry</i> de la función <i>LB Incident Radiation</i> .
	5.4	Conectar una función <i>XY Plane</i> en el <i>input context_</i> del componente <i>LB Incident Radiation</i> .
	5.5	Determinar la densidad de malla a analizar. Con un valor numérico establecer el tamaño de la retícula en el <i>input _grid_size</i> del componente <i>LB Incident Radiation</i> .
	5.6	Establecer en el <i>input _offset_dist_</i> el valor 0.1 (10cm)
	5.7	Conectar un parámetro dinámico <i>Boolean Toggle</i> en el <i>input _run</i> del componente <i>LB Incident Radiation</i> , para activar la simulación.

Nota. Elaboración propia (2020).

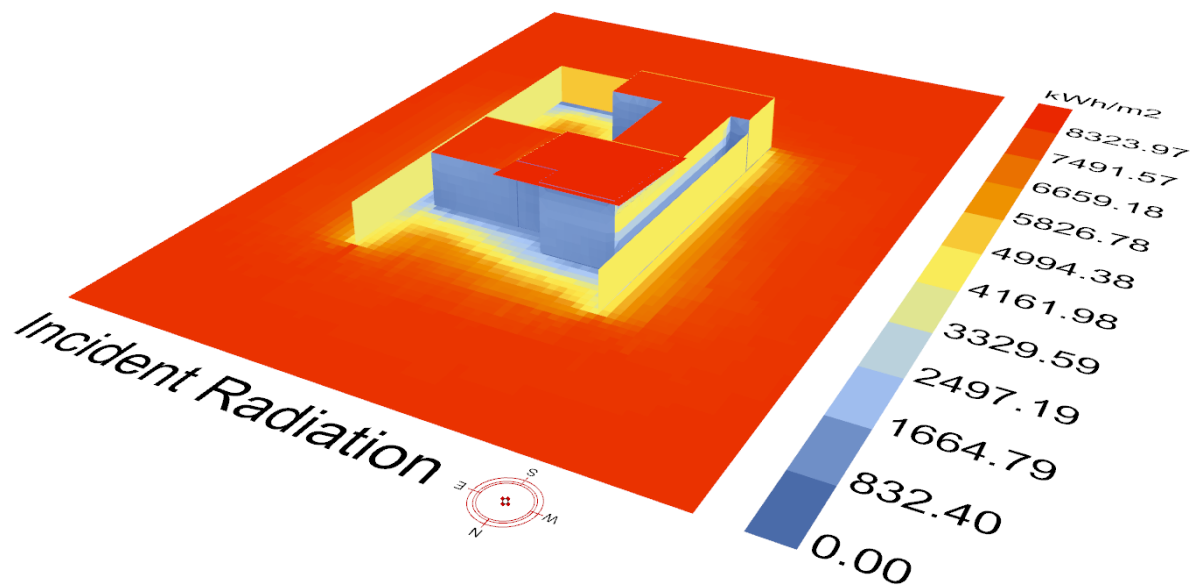


Figura 134. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

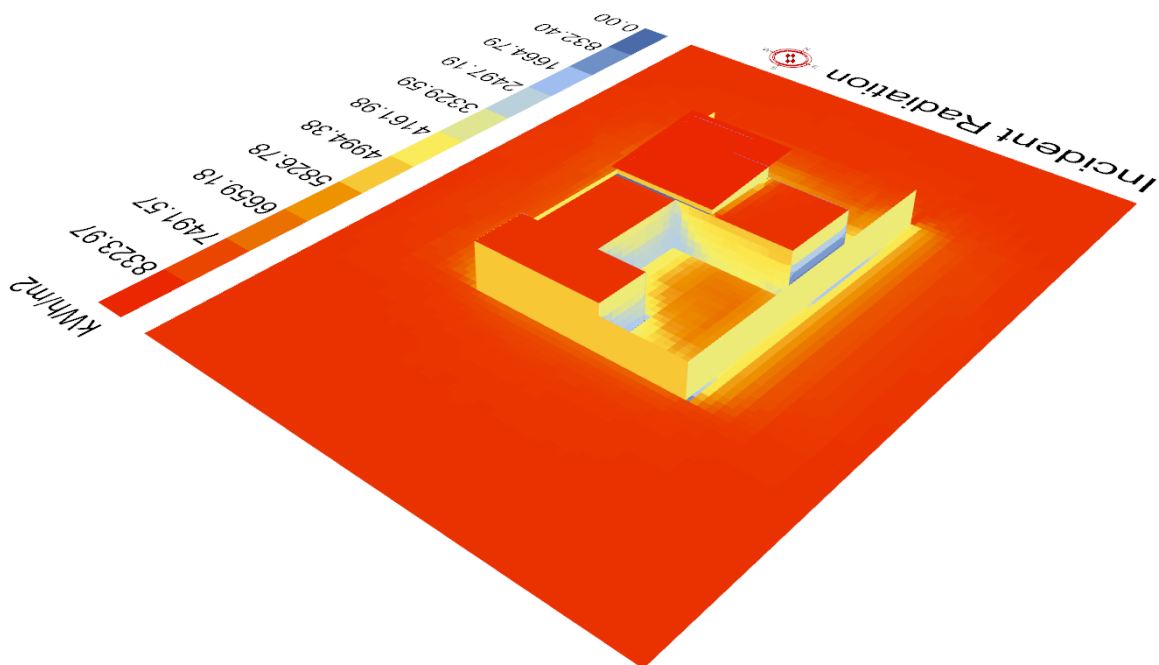


Figura 135. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante A: simulación de radiación solar anual acumulada de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

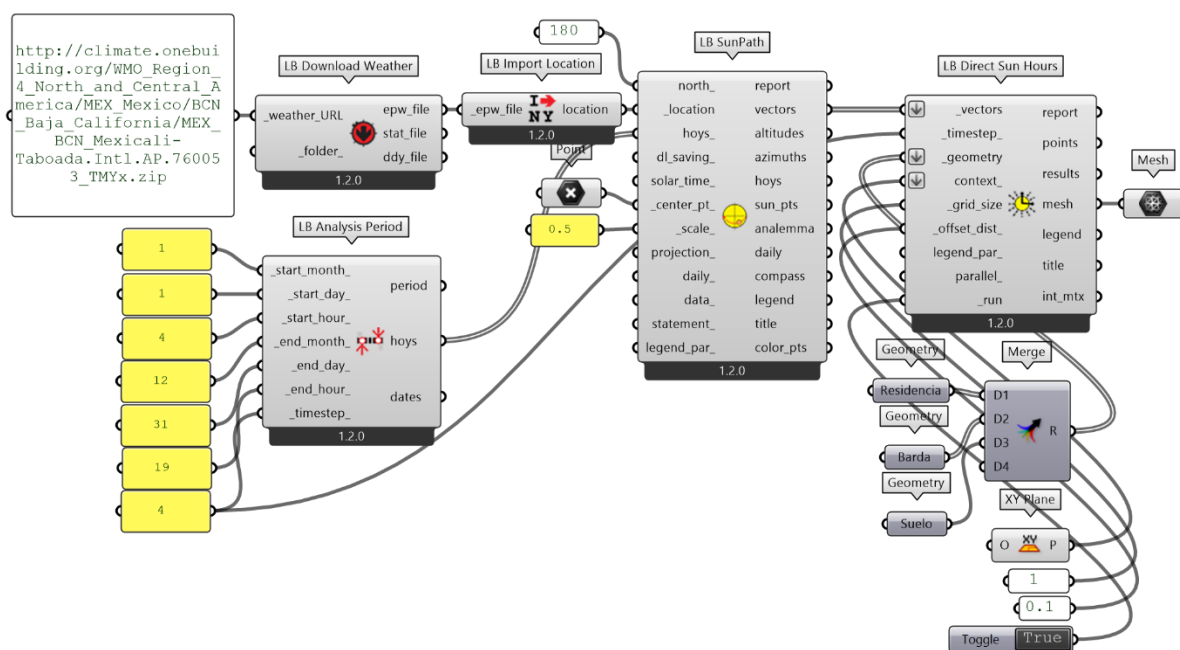


Figura 136. Algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

Tabla 18

Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre una superficie con Ladybug. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.

Sección	Paso	Descripción
1. Determinar la geometría a analizar	1.1	Referenciar en <i>Grasshopper</i> la geometría a analizar. Utilizar parámetros <i>Geometry</i> para referenciar la geometría de las edificaciones y la superficie del suelo-contexto a analizar.
2. Importar archivo climático	2.1	Existen varias maneras para importar un archivo climático en el algoritmo de <i>Grasshopper</i> mediante <i>Ladybug</i> .
	2.2	Si se tiene el archivo, utilizar un parámetro <i>File Path</i> para importar el archivo a <i>Grasshopper</i> .
	2.3	Si no se tiene el archivo, se puede descargar directamente en <i>Grasshopper</i> con el componente de <i>Ladybug</i> llamado <i>LB Download Weather</i> . Introducir con un panel la dirección URL del archivo climático a descargar, y conectar el panel en el <i>input _weather_URL</i> . El tipo de archivos climáticos que ofrece la función de descarga son: EPW, STAT y DDY.

Sección	Paso	Descripción
3. Importar datos de archivo climático	3.1	El archivo climático más habitual para usar es tipo EPW. Utilizar función <i>LB Import Location</i> para extraer los datos de ubicación geográfica que el archivo contiene.
4. Calcular el recorrido solar y los vectores de dirección solar	4.1	Extraer el componente <i>LB SunPath</i> .
	4.2	Implementar el norte (<i>input north_</i>) a través de un valor numérico. En dirección contra reloj, el valor 0 es norte, 90 oeste, 180 sur y 270 este.
	4.3	Establecer la ubicación, conectar <i>output location</i> del componente <i>LB Import Location</i> en el <i>input_location</i> de la función <i>LB SunPath</i> .
	4.4	Determinar el periodo de tiempo de análisis. Extraer función <i>LB Analysis Period</i> , determinar con valores numéricos el periodo y conectar <i>output hoys</i> en el <i>input hoys_</i> del componente <i>LB SunPath</i> .
5. Generar simulación de incidencia solar directa	5.1	Extraer componente <i>LB Direct Sun Hours</i> para generar simulación.
	5.2	Conectar el <i>output vectors</i> de la función <i>LB SunPath</i> en el <i>input_vectors</i> del componente <i>LB Direct Sun Hours</i> .
	5.3	Determinar con un valor numérico el lapso por hora para la realización de los cálculos de la simulación en el <i>input_timestep_</i> .
	5.4	Conectar los parámetros de geometría de las edificaciones y suelo a analizar en el <i>input_geometry</i> de la función <i>LB Direct Sun Hours</i> .
	5.5	Conectar una función <i>XY Plane</i> en el <i>input context_</i> del componente <i>LB Direct Sun Hours</i> .
	5.6	Determinar la densidad de malla a analizar. Con un valor numérico establecer el tamaño de la retícula en el <i>input_grid_size</i> del componente <i>LB Direct Sun Hours</i> .
	5.7	Establecer en el <i>input_offset_dist_</i> el valor 0.1 (10cm).
	5.8	Conectar un parámetro dinámico <i>Boolean Toggle</i> en el <i>input_run</i> del componente <i>LB Direct Sun Hours</i> , para activar la simulación.

Nota. Elaboración propia (2020).

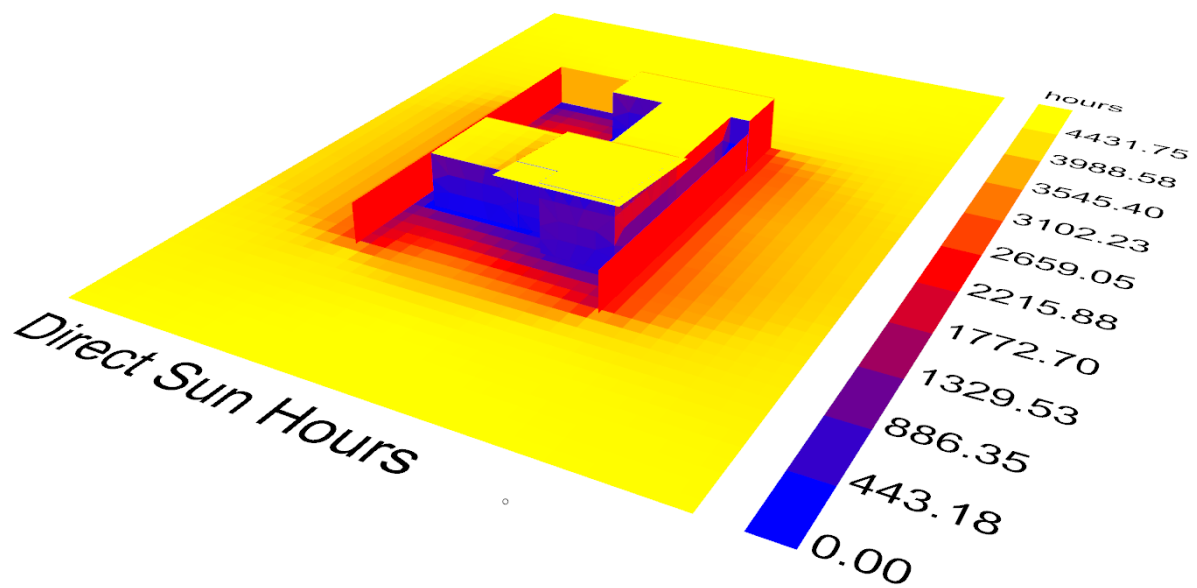


Figura 137. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

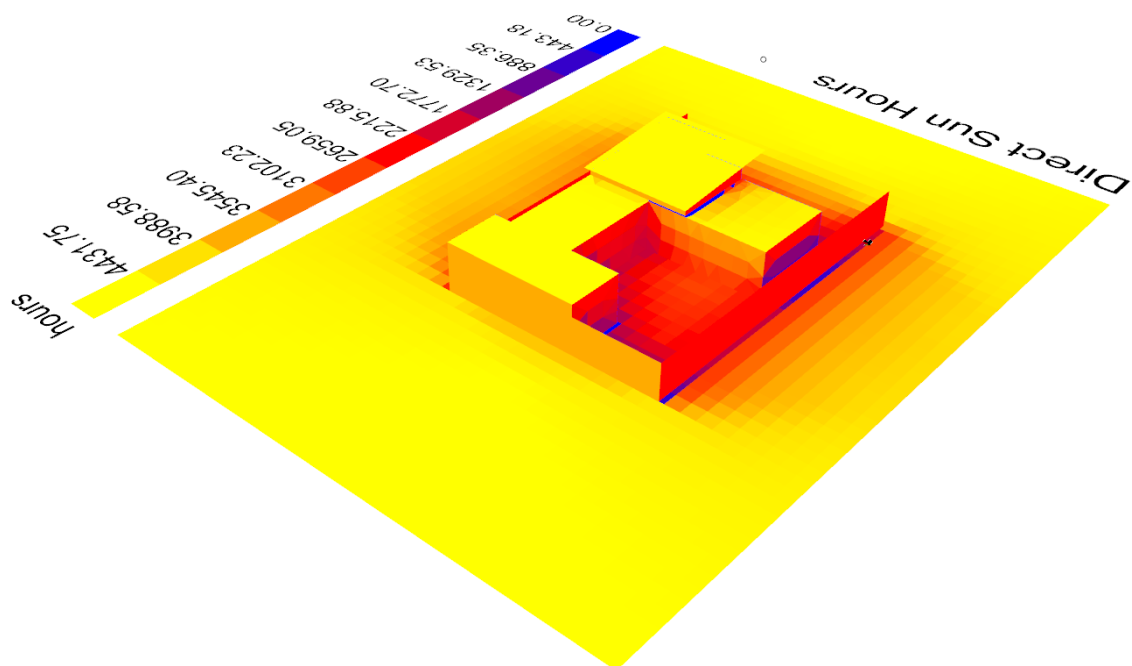


Figura 138. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

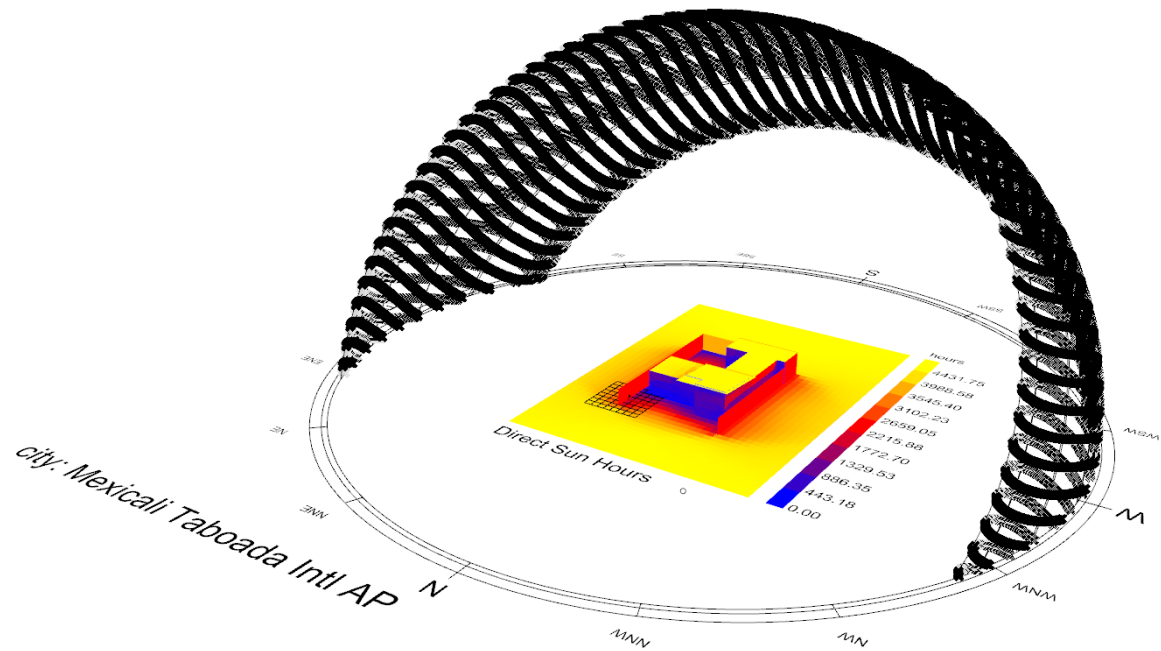


Figura 139. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante B: estudio de incidencia solar anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

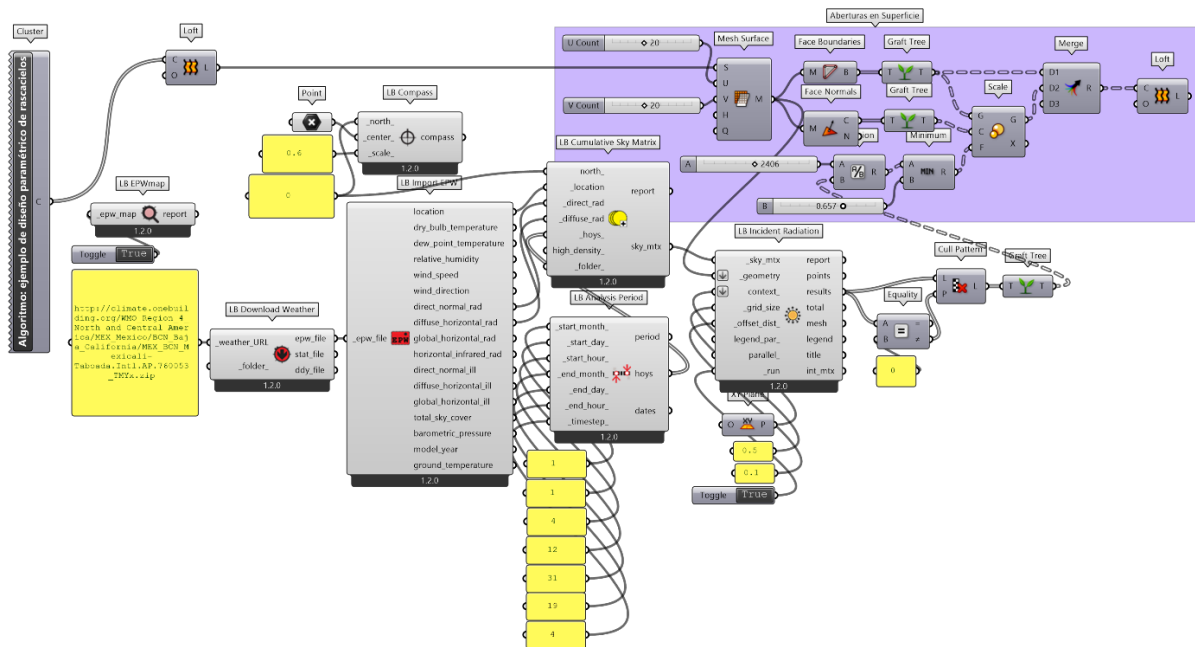


Figura 140. Algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante C: simulación de radiación solar anual de una superficie producto

de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).

Tabla 19

Síntesis descriptiva de algoritmo: simulación para análisis de radiación solar sobre una superficie con Ladybug. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”.

Sección	Paso	Descripción
1. Establecer superficie a analizar	1.1	Establecer superficie a analizar. Este algoritmo recuperará la superficie generada en la estrategia “ <u>ejemplo de diseño paramétrico de rascacielos</u> ”.
	1.2	Eliminar punto atractor de la estrategia algorítmica “ <u>ejemplo de diseño paramétrico de rascacielos</u> ”.
2. Implementar algoritmo de análisis de radiación solar anual acumulada	2.1	Implementar algoritmo “simulación de radiación solar anual acumulada”.
	2.2	Sustituir geometría referenciada y conectada inicialmente por la superficie del rascacielos.
	2.3	Conectar el <i>output M</i> de la función <i>Mesh Surface</i> de la estrategia “ <u>ejemplo de diseño paramétrico de rascacielos</u> ” en el <i>input geometry</i> del componente <i>LB Incident Radiation</i> .
3. Determinar la abertura de envolvente del rascacielos	3.1	Utilizar los datos resultantes de la simulación, para implementarlos en el factor de escala de abertura de la envolvente del rascacielos.
	3.2	Filtrar datos nulos o con valor 0 de la lista de información. Extraer función <i>Cull Pattern</i> y conectar el <i>output results</i> del componente <i>LB Incident Radiation</i> . Establecer el patrón de omisión, utilizar componente <i>Equality</i> y conectar el <i>output ≠</i> en el <i>input P</i> de la función <i>Cull Pattern</i> .
	3.3	Implementar un filtro para agrupar la información en ramas (diversas listas) con el componente <i>Graft</i> .
	3.4	Conectar la información resultante de la simulación (<i>output T</i> de <i>Graft</i>) en el <i>input B</i> de la función <i>Division</i> del remanente de componentes del ejercicio atractor de la estrategia “ <u>ejemplo de diseño paramétrico de rascacielos</u> ”. Se sustituye las distancias del elemento atractor eliminado por la información resultante de la simulación.

Nota. Elaboración propia (2020).

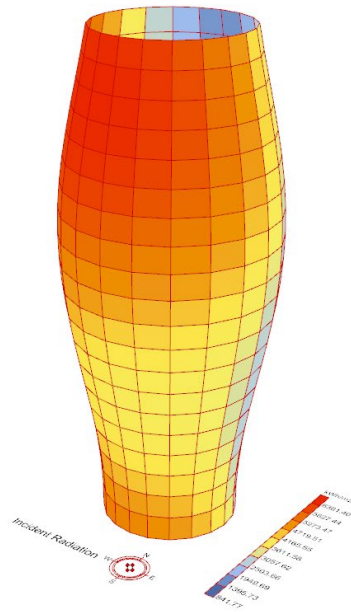


Figura 141. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).

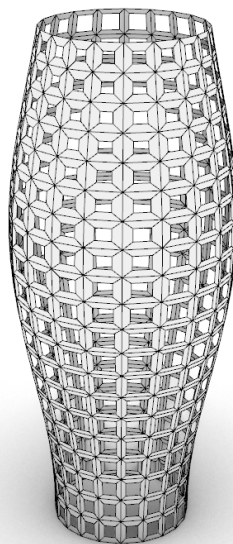


Figura 142. Resultado de algoritmo de *Grasshopper* “simulación para análisis de radiación solar sobre una superficie con *Ladybug*”. Variante C: simulación de radiación solar anual de una superficie producto de la estrategia algorítmica “ejemplo de diseño paramétrico de rascacielos”. Elaboración propia (2020).

3.4.5. Simulación para análisis de iluminación natural interior con *Honeybee*.

Honeybee es una herramienta algorítmica complementaria a *Ladybug* para hacer análisis de comportamiento de flujo de calor, confort térmico e iluminación interior. Es un simulador paramétrico que utiliza motores complementarios de programas de análisis térmico-energético como: *OpenStudio* que incluye *Energy Plus*, *Radiance*, *THERM* y *UrbanOpt*.

El análisis que efectúa *Honeybee* en sus simulaciones es a mediante dos estrategias de evaluación cuando se trata de cálculos de iluminación natural (solar): (a) factor de iluminación natural interior, y (b) análisis de iluminación natural anual, ambos expresados en porcentaje de iluminación (%) o en luxes.

Los algoritmos que se describirán a continuación ejemplifican estas dos estrategias expresadas en porcentaje de iluminación. El objeto de estudio a analizar será la Casa RV, la cual es descrita a detalle en la previa sección llamada “simulación para análisis de radiación solar sobre una superficie con *Ladybug*” (ver Figuras 143, 145 y Tablas 21, 22). Para el manejo de información en las estrategias algorítmicas mostradas a continuación, es importante tener en cuenta los parámetros y ecuaciones matemáticas expresadas en la siguiente tabla (ver Tabla 20).

Tabla 20

Relación de las principales ecuaciones matemáticas y parámetros a considerar para trabajar simulaciones de iluminación interior con la herramienta algorítmica Honeybee.

Ecuaciones matemáticas	
Flujo de calor por elementos translúcidos (cálculo de ganancias y pérdidas de calor por ventanas y ventanales)	
Ecuación	Descripción
$Q_{tr}=(A*CS*I_t)+UA(T_{amb}-T_d)$	Q_{tr} : ganancia de calor por vidrios (W) A : área (m ²)

Ecuaciones matemáticas	
<p>CS: coeficiente de sombreado del vidrio I_t: radiación solar, según orientación, más radiación reflejada y ganada por superficies exteriores y pisos U: coeficiente global de transferencia de calor de ventana T_{amb}: temperatura exterior (ambiente) T_d: temperatura de diseño (termostato)</p>	
Parámetros	
CS	<p>Coficiente de sombreado del vidrio: es la relación entre la ganancia solar que permite pasar el elemento translúcido, con respecto a la que permite pasar un vidrio claro de 3mm.</p> <p>Valor entre 0 y 1 Valor mínimo para asignar 0.2</p>
CS por tipo de vidrio	<p>Vidrio claro: 1.00 Vidrio doble: 0.88 Vidrio triple: 0.70</p>

Nota. Elaboración propia (2020).

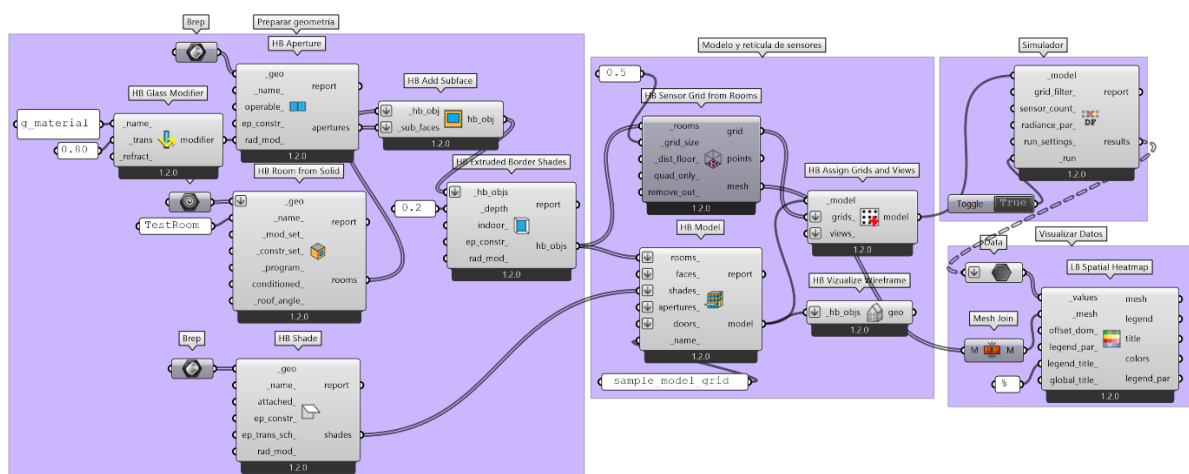


Figura 143. Algoritmo de *Grasshopper* “simulación para análisis de iluminación natural interior con *Honeybee*”. Variante A: factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

Tabla 21

Síntesis descriptiva de algoritmo: simulación para análisis de iluminación natural interior con Honeybee. Variante A: simulación de factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.

Sección	Paso	Descripción
1. Preparar geometría para analizar.	1.1	Preparar geometría para analizar. Se requieren polisuperficies (<i>Breps</i>) básicos para poder efectuar la simulación. El algoritmo permite trabajar con volúmenes y superficies.
	1.2	Preparar superficies opacas. Dibujar a manera de volumen básico cada espacio de la edificación a analizar.
	1.3	Preparar superficies translucidas. Dibujar aberturas como superficies sobre los elementos opacos (geometría empalmada).
	1.4	Preparar geometría del contexto que tiene efecto de sombreado sobre la edificación. Dibujar volúmenes o superficies incidentes.
	1.5	Referenciar en <i>Grasshopper</i> cada grupo de elementos en un parámetro <i>Geometry</i> .
	1.6	Dar reconocimiento a través de <i>Honeybee</i> a las aberturas. Extraer función <i>HB Aperture</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de superficies translucidas.
	1.7	Asignar propiedades de material del vidrio de las ventanas que son parte de los elementos translúcidos. Extraer función <i>HB Glass Modifier</i> , asignar un nombre en el <i>input_name_e</i> e implementar el coeficiente de sombreado (<i>CS</i>) del vidrio con un valor numérico conectado en el <i>input_trans</i> . Conectar <i>output modifier</i> del componente <i>HB Glass Modifier</i> al <i>input rad_mod_</i> de la función <i>HB Aperture</i> .
	1.8	Dar reconocimiento a través de <i>Honeybee</i> a los elementos opacos. Extraer función <i>HB Room from Solid</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de superficies opacas. Asignar un nombre en el <i>input_name_</i> con un parámetro <i>Panel</i> .
	1.9	Dar reconocimiento a través de <i>Honeybee</i> a los elementos incidentes que afectarán con sombreado en el análisis. Extraer función <i>HB Shade</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de los elementos del contexto.
	1.10	Relacionar las aberturas en los objetos opacos. Extraer función <i>HB Add Subface</i> y conectar el <i>output rooms</i> del componente <i>HB Room from Solid</i> en el <i>input_hb_obj</i> de la función <i>HB Add Subface</i> . También, conectar <i>output apertures</i> del componente <i>HB Aperture</i> en el <i>input_sub_faces</i> de la función <i>HB Add Subface</i> .
	1.11	De manera opcional, agregar marco parte-soles en las aberturas como estrategia de sombreado. Extraer componente <i>HB Extruded Border Shades</i> y conectar el <i>output</i> de la función <i>HB Add Subface</i> en el <i>input_hb_objs</i> . Establecer en el <i>input_depth</i> la profundidad del parte-sol.
2. Declarar modelo de análisis y generar retícula de sensores para evaluación.	2.1	Extraer el componente <i>HB Model</i> para declarar el modelo de análisis. Conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input_rooms_</i> . Conectar el <i>output shades</i> de la función <i>HB Shade</i> en el <i>input_shades_</i> del componente <i>HB Model</i> .

Sección	Paso	Descripción
	2.2	Generar retícula de sensores para evaluación de iluminación interior. Extraer componente <i>HB Sensor Grid from Rooms</i> y conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input rooms_</i> . Establecer la densidad de análisis y determinar el tamaño de la retícula de sensores en el <i>input _grid_size</i> del componente <i>HB Sensor Grid from Rooms</i> .
	2.3	Asignar la retícula de sensores al modelo de análisis. Extraer función <i>HB Assign Grids and Views</i> . Conectar el <i>ouput model</i> de la función <i>HB Model</i> en el <i>input _model</i> y el <i>output grid</i> de la función <i>HB Sensor Grid from Rooms</i> en el <i>input grids_</i> del componente <i>HB Assign Grids and Views</i> .
	2.4	Utilizar el componente <i>HB Vizualize Wireframe</i> para ver únicamente el contorno de los volúmenes y superficies que se analizan. Conectar el <i>output model</i> de la función <i>HB Model</i> en el <i>input _hb_objs</i> .
3. Implementar el simulador	3.1	Extraer el componente <i>HB Daylight Factor</i> para habilitar el mecanismo de análisis. Conectar <i>output model</i> de la función <i>HB Assign Grids and Views</i> en el <i>input _model</i> del simulador. Conectar un parámetro <i>Boolean Toggle</i> en el <i>input _run</i> para activar la simulación.
4. Visualizar datos	4.1	Extraer la función <i>LB Spatial Heatmap</i> y conectar los resultados del simulador en el <i>input _values</i> (se recomienda implementar una función de <i>Flatten Tree</i> para desagrupar los datos). También conectar el <i>output mesh</i> del componente <i>HB Sensor Grid from Rooms</i> en el <i>input _mesh</i> de la función <i>LB Spatial Heatmap</i> .

Nota. Elaboración propia (2020).

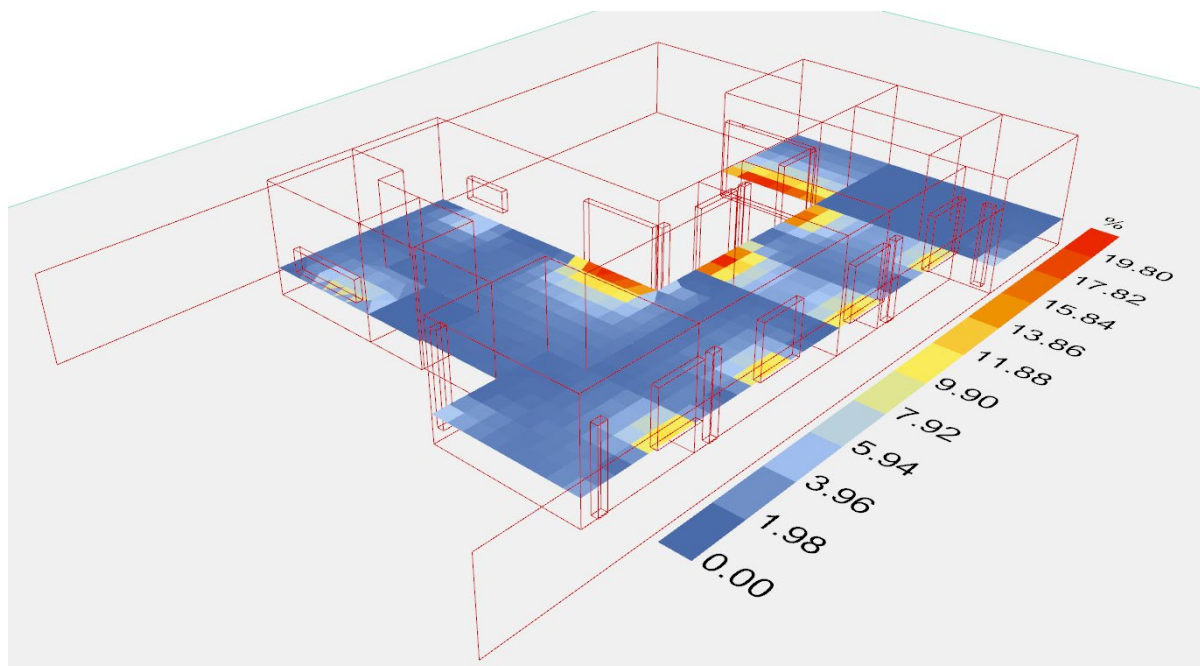


Figura 144. Resultado de algoritmo de *Grasshopper* “simulación para análisis de iluminación natural interior con *Honeybee*”. Variante A: factor de iluminación natural interior de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

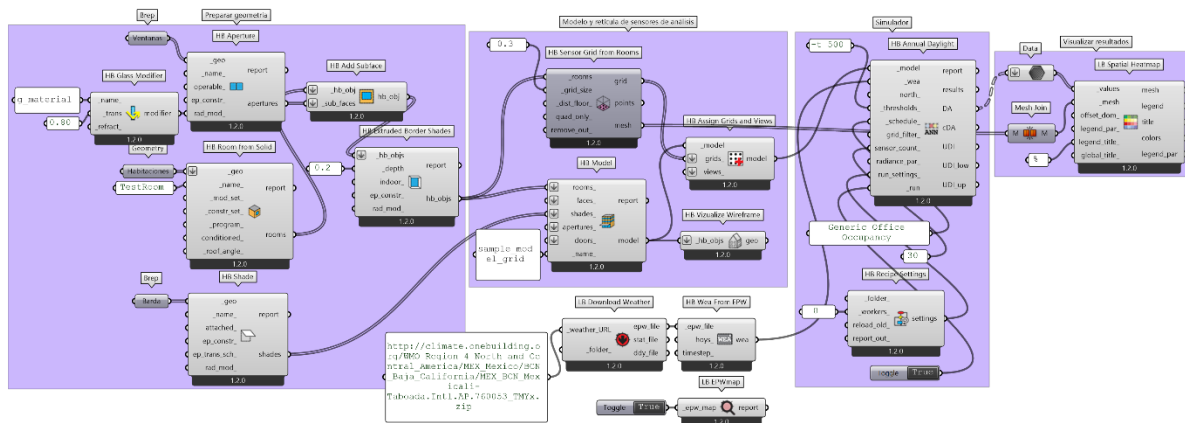


Figura 145. Algoritmo de *Grasshopper* “simulación para análisis de iluminación natural interior con *Honeybee*”. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

Tabla 22

Síntesis descriptiva de algoritmo: simulación para análisis de iluminación natural interior con Honeybee. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C.

Sección	Paso	Descripción
1. Preparar geometría para analizar.	1.1	Preparar geometría para analizar. Se requieren polisuperficies (<i>Breps</i>) básicos para poder efectuar la simulación. El algoritmo permite trabajar con volúmenes y superficies.
	1.2	Preparar superficies opacas. Dibujar a manera de volumen básico cada espacio de la edificación a analizar.
	1.3	Preparar superficies translucidas. Dibujar aberturas como superficies sobre los elementos opacos (geometría empalmada).
	1.4	Preparar geometría del contexto que tiene efecto de sombreado sobre la edificación. Dibujar volúmenes o superficies incidentes.
	1.5	Referenciar en <i>Grasshopper</i> cada grupo de elementos en un parámetro <i>Geometry</i> .
	1.6	Dar reconocimiento a través de <i>Honeybee</i> a las aberturas. Extraer función <i>HB Aperture</i> y conectar en el <i>input _geo</i> el parámetro <i>Geometry</i> de superficies translucidas.
	1.7	Asignar propiedades de material del vidrio de las ventanas que son parte de los elementos translúcidos. Extraer función <i>HB Glass Modifier</i> , asignar un nombre en el <i>input _name_</i> e implementar el coeficiente de sombreado (<i>CS</i>) del vidrio con un valor numérico conectado en el <i>input _trans</i> . Conectar <i>output modifier</i> del componente <i>HB Glass Modifier</i> al <i>input rad_mod_</i> de la función <i>HB Aperture</i> .
	1.8	Dar reconocimiento a través de <i>Honeybee</i> a los elementos opacos. Extraer función <i>HB Room from Solid</i> y conectar en el <i>input _geo</i> el parámetro <i>Geometry</i> de superficies opacas. Asignar un nombre en el <i>input _name_</i> con un parámetro <i>Panel</i> .
	1.9	Dar reconocimiento a través de <i>Honeybee</i> a los elementos incidentes que afectarán con sombreado en el análisis. Extraer función <i>HB Shade</i> y conectar en el <i>input _geo</i> el parámetro <i>Geometry</i> de los elementos del contexto.
	1.10	Relacionar las aberturas en los objetos opacos. Extraer función <i>HB Add Subface</i> y conectar el <i>output rooms</i> del componente <i>HB Room from Solid</i> en el <i>input _hb_obj</i> de la función <i>HB Add Subface</i> . También, conectar <i>output apertures</i> del componente <i>HB Aperture</i> en el <i>input _sub_faces</i> de la función <i>HB Add Subface</i> .
	1.11	De manera opcional, agregar marco parte-soles en las aberturas como estrategia de sombreado. Extraer componente <i>HB Extruded Border Shades</i> y conectar el <i>output</i> de la función <i>HB Add Subface</i> en el <i>input _hb_objs</i> . Establecer en el <i>input _depth</i> la profundidad del parte-sol.
2. Declarar modelo de análisis y generar retícula de sensores para evaluación.	2.1	Extraer el componente <i>HB Model</i> para declarar el modelo de análisis. Conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input rooms_</i> . Conectar el <i>output shades</i> de la función <i>HB Shade</i> en el <i>input shades_</i> del componente <i>HB Model</i> .

Sección	Paso	Descripción
	2.2	Generar retícula de sensores para evaluación de iluminación interior. Extraer componente <i>HB Sensor Grid from Rooms</i> y conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input rooms_</i> . Establecer la densidad de análisis y determinar el tamaño de la retícula de sensores en el <i>input _grid_size</i> del componente <i>HB Sensor Grid from Rooms</i> .
	2.3	Asignar la retícula de sensores al modelo de análisis. Extraer función <i>HB Assign Grids and Views</i> . Conectar el <i>ouput model</i> de la función <i>HB Model</i> en el <i>input _model</i> y el <i>output grid</i> de la función <i>HB Sensor Grid from Rooms</i> en el <i>input grids_</i> del componente <i>HB Assign Grids and Views</i> .
	2.4	Utilizar el componente <i>HB Vizualize Wireframe</i> para ver únicamente el contorno de los volúmenes y superficies que se analizan. Conectar el <i>output model</i> de la función <i>HB Model</i> en el <i>input _hb_objs</i> .
3. Importar y extraer datos climáticos	3.1	Existen varias maneras para importar un archivo climático en el algoritmo de <i>Grasshopper</i> mediante <i>Ladybug</i> .
	3.2	Si se tiene el archivo, utilizar un parámetro <i>File Path</i> para importar el archivo a <i>Grasshopper</i> .
	3.3	Si no se tiene el archivo, se puede descargar directamente en <i>Grasshopper</i> con el componente de <i>Ladybug</i> llamado <i>LB Download Weather</i> . Introducir con un panel la dirección URL del archivo climático a descargar, y conectar el panel en el <i>input _weather_URL</i> . El tipo de archivos climáticos que ofrece la función de descarga son: EPW, STAT y DDY.
	3.4	El archivo climático más habitual para usar es tipo EPW. Utilizar función <i>HB Wea From EPW</i> para extraer los datos que contiene el archivo.
4. Implementar el simulador	4.1	Extraer el componente <i>HB Annual Daylight</i> para habilitar el mecanismo de análisis. Conectar <i>output model</i> de la función <i>HB Assign Grids and Views</i> en el <i>input _model</i> del simulador.
	4.2	Conectar el <i>output wea</i> de la función <i>HB Wea From EPW</i> en el <i>input _wea</i> del simulador.
	4.3	Establecer el umbral de luxes de diagnóstico (<i>input _thresholds</i>). En el ejemplo se implementa un umbral de 500 luxes autónomos de iluminación natural.
	4.4	Establecer una bitácora de ocupación en el <i>input _schedule_</i>
	4.5	Determinar la cantidad de sensores activos paralelamente por cada ejecución del simulador.
	4.6	Establecer configuración de la simulación. Extraer componente <i>HB Recipe Settings</i> y establecer condiciones de simulación. En el ejemplo se implementa una cantidad de 8 CPUs para la simulación (no exceder cantidad núcleos de procesamiento de la computadora).
	4.7	Conectar un parámetro <i>Boolean Toggle</i> en el <i>input _run</i> para activar la simulación.
5. Visualizar datos	5.1	Extraer la función <i>LB Spatial Heatmap</i> y conectar los resultados del simulador en el <i>input _values</i> (se recomienda implementar una función de <i>Flatten Tree</i> para desagrupar los datos). También conectar el <i>output mesh</i> del componente <i>HB Sensor Grid from Rooms</i> en el <i>input _mesh</i> de la función <i>LB Spatial Heatmap</i> .

Nota. Elaboración propia (2020).

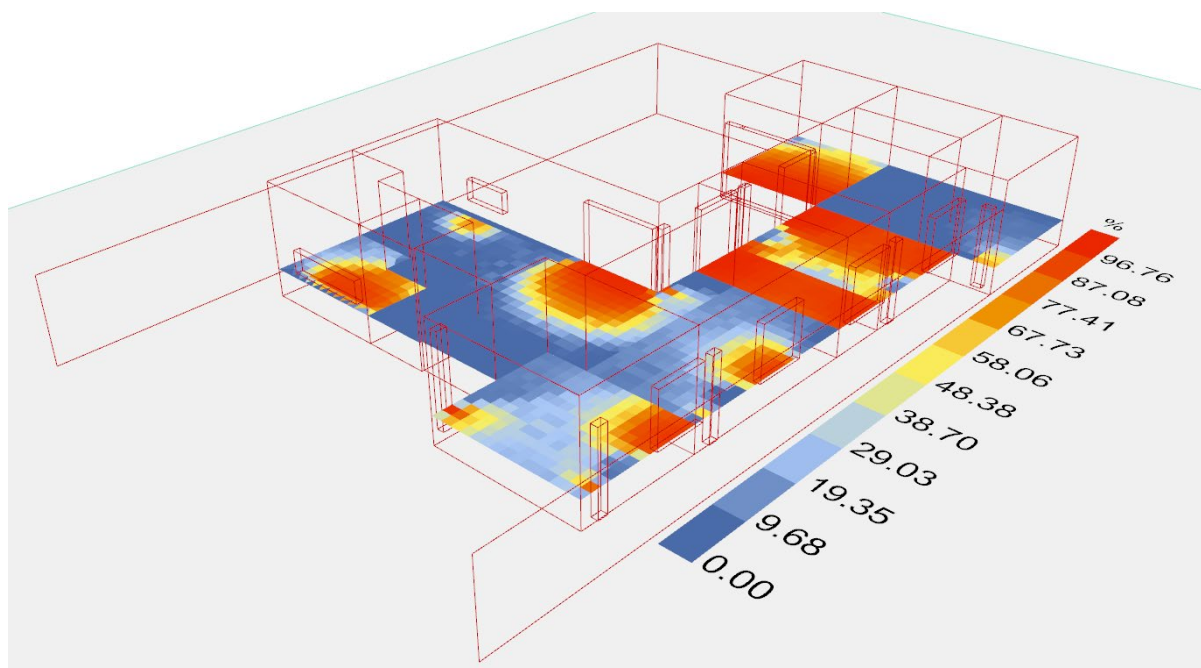


Figura 146. Resultado de algoritmo de *Grasshopper* “simulación para análisis de iluminación natural interior con *Honeybee*”. Variante B: simulación de iluminación interior anual de Casa RV, SPAU (2017). Aplicación: Mexicali, B.C. Elaboración propia (2020).

3.4.6. Simulación para análisis de confort térmico espacial interior con *Honeybee*.

Como parte del conjunto de simulaciones que la herramienta *Honeybee* puede realizar, se encuentra el cálculo de flujo de calor, balance y zonas de confort térmico de una edificación. Los cálculos térmico-energéticos se desarrollan en el marco de las leyes de la termodinámica.

Para obtener información relacionada con el comportamiento de transferencia de calor que fluye por una edificación, es importante identificar las pérdidas o ganancias de calor que se producen entre el espacio interior y exterior de una edificación a través de los fenómenos de conducción, convección, radiación y evaporación.

La transferencia de calor es el principal mecanismo a través del cual los efectos ambientales se manifiestan en las edificaciones. En este sentido la conducción de calor

esencialmente se efectúa a través de las propiedades de la envolvente de una edificación y la transmisión de radiación solar sucede mediante las aberturas o ventanas (Kreider et al., 2002).

El estudio de la envolvente arquitectónica y sus características es fundamental en el desarrollo de un proyecto arquitectónico. La búsqueda de adaptabilidad de una edificación con su entorno, por parte del diseñador, es un compromiso importante para el cuidado del medio ambiente natural, la sustentabilidad y objetivo general de supervivencia.

El mecanismo de transferencia de calor de una edificación, por tanto, se define como la suma de ganancias y pérdidas de energía térmica del espacio interior y el exterior a través de diversos factores físicos que son parte del fenómeno. Para el cálculo y simulación de este mecanismo, es importante primeramente estar familiarizado con los siguientes conceptos (ver Tabla 23).

Tabla 23

Relación de las principales ecuaciones matemáticas y parámetros a considerar para trabajar simulaciones de flujo de calor con la herramienta algorítmica Honeybee.

Ecuaciones matemáticas	
Flujo de calor por ganancias o pérdidas externas	
Muros, techos y puertas	
Ecuación	Descripción
$Q=UA(T_{as}-T_d)$	Q: ganancia de calor (W) A: área (m ²) T_{as}: Temperatura aire sol T_d: Temperatura de diseño (termostato)
Pisos	
Ecuación	Descripción
$Q_p=F_2 * P(T_d-T_{amb})$	Q_p: pérdida de calor por perímetro del piso (m) F₂: coeficiente de pérdida de calor por metro lineal del perímetro de piso, según tipo de muro (W/m°C) P: Perímetro de piso (m) T_{amb}: Temperatura ambiente T_d: Temperatura de diseño (termostato)
Infiltración	
Ecuación	Descripción
$Q_{inf}=1.20Q(h_{e_{ext}}-h_{e_{int}})$	Q_{inf}: ganancia o pérdida de calor por infiltración (W)

Ecuaciones matemáticas	
	<p>I_{20}: coeficiente de densidad del aire a 25°C, sobre el nivel del mar (kg/m³)</p> <p>Q: cambios de aire por hora (depende del volumen del espacio)</p> <p>he_{ext}: entalpia del aire exterior (kj/kg)</p> <p>he_{int}: entalpia del aire interior (kj/kg)</p>
Elementos translúcidos	
Ecuación	Descripción
$Q_{tv}=(A*CS*I_t)+UA(T_{amb}-T_d)$	<p>Q_{tv}: ganancia de calor por vidrios (W)</p> <p>A: área (m²)</p> <p>CS: coeficiente de sombreado del vidrio</p> <p>I_t: radiación solar, según orientación, más radiación reflejada y ganada por superficies exteriores y pisos</p> <p>U: coeficiente global de transferencia de calor de ventana</p> <p>T_{amb}: temperatura exterior (ambiente)</p> <p>T_d: temperatura de diseño (termostato)</p>
Parámetros	
Conductividad térmica (k)	Propiedad fundamental para que se dé la transferencia de calor en los materiales. La conducción se da por contacto entre moléculas o partículas de los cuerpos. El calor se propaga en todas direcciones hacia los puntos de menor temperatura.
Retraso térmico	Tiempo que tarda en pasar el calor a través de una capa de material. Es el tiempo transcurrido entre los momentos en que se dan las temperaturas máximas en cada una de las superficies de material.
Resistencia térmica	Es la capacidad de un material para oponerse al flujo de calor. Es la razón entre el espesor del material y la conductividad térmica que posee.
Convección	<p>Si dos cuerpos entran en contacto, y uno de ellos es un fluido, se da la transferencia de calor por convección en dos fases:</p> <ol style="list-style-type: none"> 1. El calor se transmite por contacto o por conducción 2. El cambio experimentado por la temperatura del fluido modifica su densidad, por lo que adquiere un movimiento convectivo ascendente o descendente, según sea el cambio de temperatura.
Coficiente global de transferencia de calor (valor U)	Es el fenómeno inverso a la resistencia equivalente y representa la velocidad de transferencia de energía que se da desde el aire exterior al interior a través del sistema constructivo por unidad de área y diferencia de temperatura unitaria, se expresa en W/m ² °C

Nota. Elaboración propia (2020).

Para efectos de ejemplificar los análisis térmico-energéticos que *Honeybee* puede desarrollar, en esta sección se expondrá la estrategia algorítmica de simulación para el análisis de confort térmico de la Casa RV, la cual es descrita a detalle en la previa sección llamada

“simulación para análisis de radiación solar sobre una superficie con *Ladybug*” (ver Figura 147 y Tabla 24).

Las propiedades del sistema constructivo y la zona climática con la que se hará la simulación se encuentran configuradas a través de un esquema preestablecido que ofrece *Honeybee* basado en ASHRAE 90.1 2013 | IEEC 2015 para hacer simulaciones rápidas, sin embargo, es posible personalizar a detalle las características del sistema constructivo.

Tabla 24

Síntesis descriptiva de algoritmo: simulación para análisis de confort térmico espacial

interior con Honeybee. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C.

Sección	Paso	Descripción
1. Preparar geometría para analizar.	1.1	Preparar geometría para analizar. Se requieren polisuperficies (<i>Breps</i>) básicos para poder efectuar la simulación. El algoritmo permite trabajar con volúmenes y superficies.
	1.2	Preparar superficies opacas. Dibujar a manera de volumen básico cada espacio de la edificación a analizar.
	1.3	Preparar superficies translucidas. Dibujar aberturas como superficies sobre los elementos opacos (geometría empalmada).
	1.4	Preparar geometría del contexto que tiene efecto de sombreado sobre la edificación. Dibujar volúmenes o superficies incidentes.
	1.5	Referenciar en <i>Grasshopper</i> cada grupo de elementos en un parámetro <i>Geometry</i> .
	1.6	Referenciar en <i>Grasshopper</i> cada grupo de elementos en un parámetro <i>Geometry</i> .
	1.7	Dar reconocimiento a través de <i>Honeybee</i> a las aberturas. Extraer función <i>HB Aperture</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de superficies translucidas.
	1.8	Dar reconocimiento a través de <i>Honeybee</i> a los elementos opacos. Extraer función <i>HB Room from Solid</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de superficies opacas. Asignar un nombre en el <i>input_name</i> con un parámetro <i>Panel</i> .
	1.9	Configurar el sistema constructivo y zona climática de la edificación. Extraer componente <i>HB Construction Set by Climate</i> . Establecer configuración preestablecida según ASHRAE 90.1 2013 IECC 2015 de zona climática y tipo de construcción. Utilizar funciones <i>HB ClimateZones</i> y <i>HB ConstructionTypes</i> y conectarlos en el <i>input</i> correspondiente para determinar la configuración.
	1.10	Conectar output del componente <i>HB Construction Set by Climate</i> en el <i>input_constr_set</i> de la función <i>HB Room from Solid</i> .
	1.11	Dar reconocimiento a través de <i>Honeybee</i> a los elementos incidentes que afectarán con sombreado en el análisis. Extraer función <i>HB Shade</i> y conectar en el <i>input_geo</i> el parámetro <i>Geometry</i> de los elementos del contexto.
	1.12	Relacionar las aberturas en los objetos opacos. Extraer función <i>HB Add Subface</i> y conectar el <i>output_rooms</i> del componente <i>HB Room from Solid</i> en el <i>input_hb_obj</i> de la función <i>HB Add Subface</i> . También, conectar <i>output_apertures</i> del componente <i>HB Aperture</i> en el <i>input_sub_faces</i> de la función <i>HB Add Subface</i> .
	1.13	De manera opcional, agregar marco parte-soles en las aberturas como estrategia de sombreado. Extraer componente <i>HB Extruded Border Shades</i> y conectar el <i>output</i> de la función <i>HB</i>

Sección	Paso	Descripción
		<i>Add Subface</i> en el <i>input_hb_objs</i> . Establecer en el <i>input_depth</i> la profundidad del parte-sol.
2. Declarar modelo de análisis y generar retícula de sensores para evaluación.	2.1	Extraer el componente <i>HB Model</i> para declarar el modelo de análisis. Conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input_rooms_</i> . Conectar el <i>output_shades</i> de la función <i>HB Shade</i> en el <i>input_shades_</i> del componente <i>HB Model</i> .
	2.2	Generar retícula de sensores para evaluación de iluminación interior. Extraer componente <i>HB Sensor Grid from Rooms</i> y conectar el <i>output</i> de la función <i>HB Extruded Border Shades</i> en el <i>input_rooms_</i> . Establecer la densidad de análisis y determinar el tamaño de la retícula de sensores en el <i>input_grid_size</i> del componente <i>HB Sensor Grid from Rooms</i> .
	2.3	Asignar la retícula de sensores al modelo de análisis. Extraer función <i>HB Assign Grids and Views</i> . Conectar el <i>output_model</i> de la función <i>HB Model</i> en el <i>input_model</i> y el <i>output_grid</i> de la función <i>HB Sensor Grid from Rooms</i> en el <i>input_grids_</i> del componente <i>HB Assign Grids and Views</i> .
	2.4	Utilizar el componente <i>HB Vizualize Wireframe</i> para ver únicamente el contorno de los volúmenes y superficies que se analizan. Conectar el <i>output_model</i> de la función <i>HB Model</i> en el <i>input_hb_objs</i> .
3. Importar y extraer datos climáticos	3.1	Existen varias maneras para importar un archivo climático en el algoritmo de <i>Grasshopper</i> mediante <i>Ladybug</i> .
	3.2	Si se tiene el archivo, utilizar un parámetro <i>File Path</i> para importar el archivo a <i>Grasshopper</i> .
	3.3	Si no se tiene el archivo, se puede descargar directamente en <i>Grasshopper</i> con el componente de <i>Ladybug</i> llamado <i>LB Download Weather</i> . Introducir con un panel la dirección URL del archivo climático a descargar, y conectar el panel en el <i>input_weather_URL</i> . El tipo de archivos climáticos que ofrece la función de descarga son: EPW, STAT y DDY.
	3.4	El archivo climático más habitual para usar es tipo EPW, sin embargo, en este algoritmo se requerirá datos de los archivos EPW y DDY.
4. Implementar el simulador	4.1	Extraer el componente <i>HB PMV Comfort Map</i> para habilitar el mecanismo de análisis. Conectar <i>output_model</i> de la función <i>HB Assign Grids and Views</i> en el <i>input_model</i> del simulador.
	4.2	Conectar el <i>epw_file</i> y <i>ddy_file</i> de la función <i>LB Download Weather</i> en los <i>inputs</i> correspondientes del simulador.
	4.3	Determinar la cantidad de sensores activos paralelamente por cada ejecución del simulador.
	4.4	Establecer una referencia de vestimenta de usuarios para alinear los datos de confort con relación a la temperatura. Valor 0.7 es vestimenta camisa manga larga y pantalón.
	4.5	Establecer configuración de simulación. En el ejemplo se dispuso "--workers 20"
	4.6	Conectar un parámetro <i>Boolean Toggle</i> en el <i>input_run</i> para activar la simulación.
5. Visualizar mapa de confort	5.1	Extraer la función <i>LB Spatial Heatmap</i> y conectar los resultados de sensación de calor del simulador. Conectar <i>output_HSP</i> en el <i>input_values</i> (se recomienda implementar una función de <i>Flatten Tree</i> para desagrupar los datos).

Sección	Paso	Descripción
	5.2	De manera opcional extraer otro componente <i>LB Spatial Heatmap</i> y conectar los resultados de sensación de frío del simulador. Conectar <i>output CSP</i> en el <i>input _values</i> (se recomienda implementar una función de <i>Flatten Tree</i> para desagrupar los datos).
	5.3	También conectar el <i>output mesh</i> del componente <i>HB Sensor Grid from Rooms</i> en el <i>input _mesh</i> de la función <i>LB Spatial Heatmap</i> .
	5.4	Configurar en cada componente <i>LB Spatial Heatmap</i> los parámetros de leyendas y color del mapa. Utilizar función <i>LB Legend Parameters</i> para ese objetivo.
6. Visualizar datos de temperatura interior	6.1	Extraer componente <i>HB Read Thermal Matrix</i> . Conectar <i>output temperature</i> del simulador en el <i>input _comf_result</i> .
	6.2	Conectar un parámetro <i>Boolean Toggle</i> en el <i>input _load</i> para obtener la información.
	6.3	Implementar componente <i>HB Visualize Thermal Map</i> . Conectar el <i>output comf_mtx</i> de la función <i>HB Read Thermal Matrix</i> en el <i>input _comf_mtx</i> .
	6.4	También conectar el <i>output mesh</i> del componente <i>HB Sensor Grid from Rooms</i> en el <i>input _mesh</i> de la función <i>HB Read Thermal Matrix</i> (se recomienda implementar una función de <i>Flatten Tree</i> para desagrupar los datos).

Nota. Elaboración propia (2020).

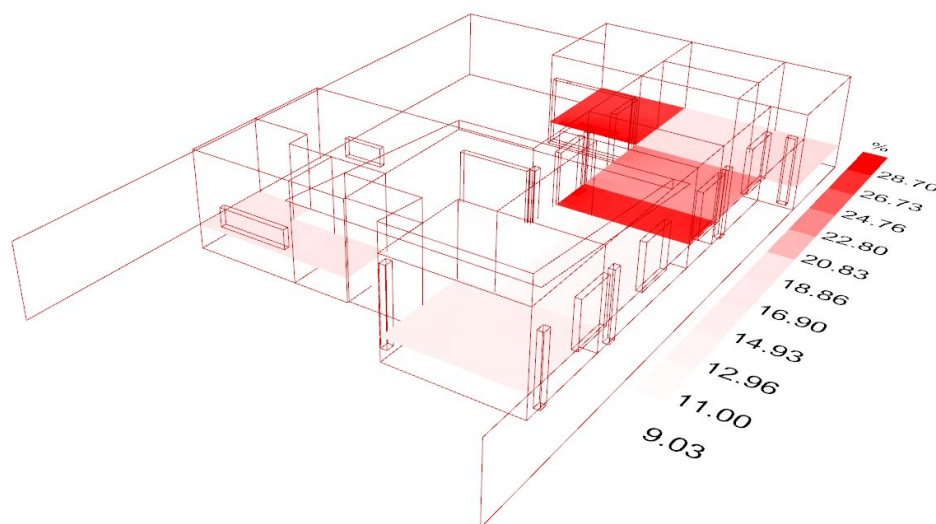


Figura 148. Resultado de algoritmo de *Grasshopper* “simulación para análisis de confort térmico espacial interior con *Honeybee*”. La imagen muestra las zonas interiores de confort térmico de acuerdo con la sensación de calor. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).

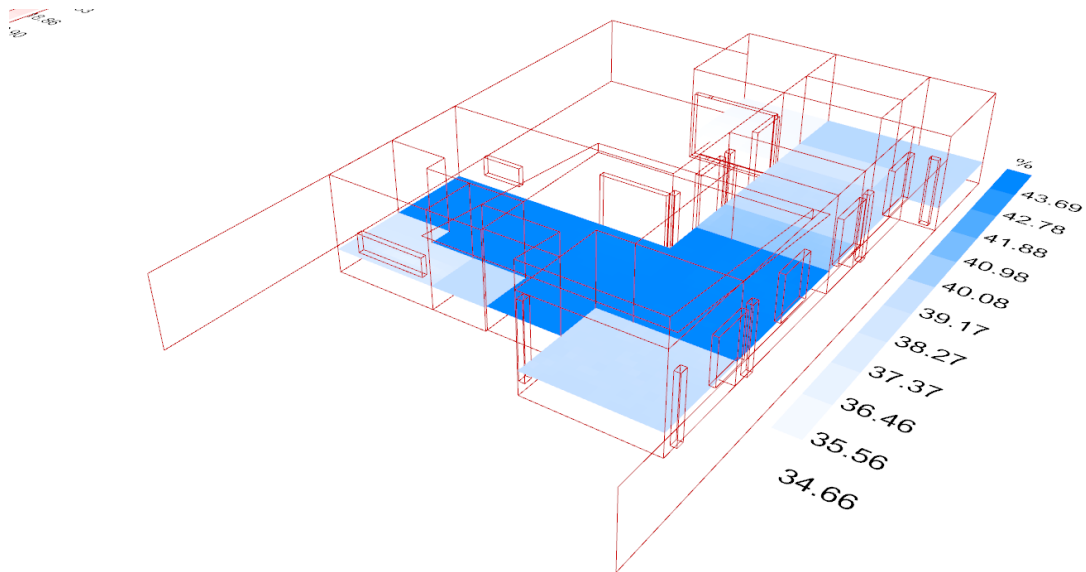


Figura 149. Resultado de algoritmo de *Grasshopper* “simulación para análisis de confort térmico espacial interior con *Honeybee*”. La imagen muestra las zonas interiores de confort térmico de acuerdo con la sensación de frío. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).

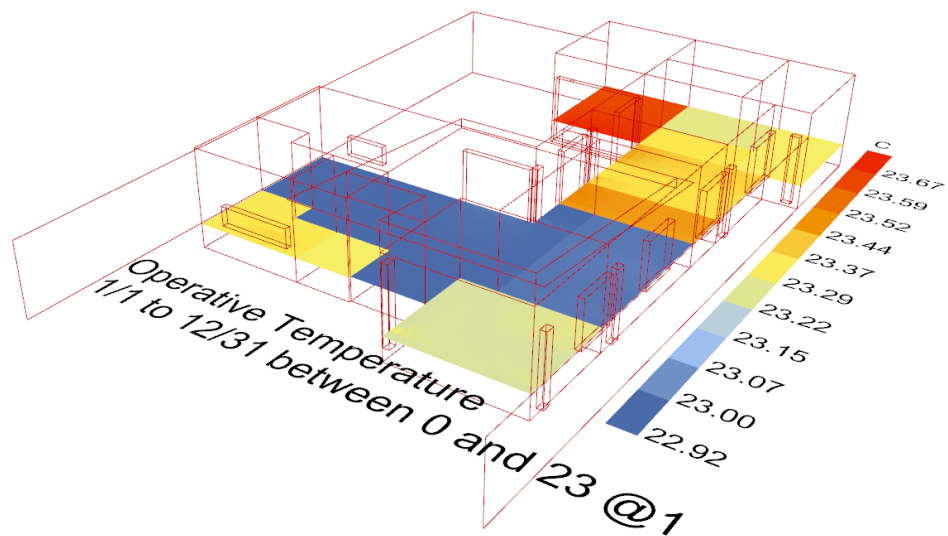


Figura 150. Resultado de algoritmo de *Grasshopper* “simulación para análisis de confort térmico espacial interior con *Honeybee*”. La imagen muestra la temperatura operativa interior anual. Aplicación: Casa RV, SPAU (2017). Mexicali, B.C. Elaboración propia (2020).

3.4.7. Optimización evolutiva con Algoritmos Genéticos (AG) mediante *Galápagos*.

El presente algoritmo recupera la estrategia “simulación para análisis de radiación solar sobre una superficie con *Ladybug*” e implementa una metodología heurística de optimización evolutiva a través de Algoritmos Genéticos (AG).

La finalidad del algoritmo aquí presentado es transformar los parámetros que determinan la morfología de una superficie, con el fin de maximizar o minimizar la incidencia de radiación solar anual; y de esta manera, optimizar la forma para un objetivo de diseño determinado (ver Figura 151 y Tabla).

Tabla 25

Síntesis descriptiva de algoritmo: optimización evolutiva con Algoritmos Genéticos (AG).

Variante: minimizar incidencia de radiación solar anual sobre superficie.

Sección	Paso	Descripción
1. Generar superficie paramétrica	1.1	Desarrollar en <i>Grasshopper</i> una superficie paramétrica. Es importante que la superficie se origine de un conjunto de datos que determinen su forma.
2. Implementar parámetros genéticos	2.1	Implementar parámetro <i>Gene Pool</i> , el cual, funciona como un grupo de datos que modificarán la superficie. Cada dato del <i>Gene Pool</i> es un gen evolutivamente transformable.
	2.2	Debe implementarse en toda función, que transforme la morfología de la superficie, un grupo de datos <i>Gene Pool</i> .
3. Importar archivo climático	3.1	Existen varias maneras para importar un archivo climático en el algoritmo de <i>Grasshopper</i> mediante <i>Ladybug</i> .
	3.2	Si se tiene el archivo, utilizar un parámetro <i>File Path</i> para importar el archivo a <i>Grasshopper</i> .
	3.3	Si no se tiene el archivo, se puede descargar directamente en <i>Grasshopper</i> con el componente de <i>Ladybug</i> llamado <i>LB Download Weather</i> . Introducir con un panel la dirección URL del archivo climático a descargar, y conectar el panel en el <i>input _weather_URL</i> . El tipo de archivos climáticos que ofrece la función de descarga son: EPW, STAT y DDY.
4. Importar datos de archivo climático	4.1	El archivo climático más habitual para usar es tipo EPW. Utilizar función <i>LB Import EPW</i> para extraer los datos que contiene el archivo.
	4.2	En este algoritmo se utilizarán los <i>outputs location</i> (ubicación geográfica), <i>direct_normal_rad</i> (radiación directa), <i>diffuse_horizontal_rad</i> (radiación difusa).
5. Calcular la radiación acumulada de un periodo determinado de tiempo	5.1	Extraer el componente <i>LB Cumulative Sky Matrix</i> .
	5.2	Implementar el norte (<i>input north_</i>) a través de un valor numérico. En dirección contra reloj, el valor 0 es norte, 90 oeste, 180 sur y 270 este.
	5.3	Establecer la ubicación, conectar <i>output location</i> del componente <i>LB Import EPW</i> en el <i>input _location</i> de la función <i>LB Cumulative Sky Matrix</i> .
	5.4	Establecer radiación directa, conectar <i>output direct_normal_rad</i> del componente <i>LB Import EPW</i> en el <i>input _direct_rad</i> de la función <i>LB Cumulative Sky Matrix</i> .
	5.5	Establecer radiación difusa, conectar <i>output diffuse_horizontal_rad</i> del componente <i>LB Import EPW</i> en el <i>input _diffuse_rad</i> de la función <i>LB Cumulative Sky Matrix</i> .
	5.6	Determinar el periodo de tiempo de análisis. Extraer función <i>LB Analysis Period</i> , determinar con valores numéricos el periodo y conectar <i>output hoys</i> en el <i>input _hoys_</i> del componente <i>LB Cumulative Sky Matrix</i> .
	6.1	Extraer componente <i>LB Incident Radiation</i> para generar simulación.

Sección	Paso	Descripción
6. Generar simulación de radiación solar	6.2	Conectar <i>ouput sky_mtx</i> del componente <i>LB Cumulative Sky Matrix</i> en el <i>input _sky_mtx</i> de la función <i>LB Incident Radiation</i> .
	6.3	Conectar los parámetros de geometría de las edificaciones y suelo a analizar en el <i>input _geometry</i> de la función <i>LB Incident Radiation</i> .
	6.4	Conectar una función <i>XY Plane</i> en el <i>input context_</i> del componente <i>LB Incident Radiation</i> .
	6.5	Determinar la densidad de malla a analizar. Con un valor numérico establecer el tamaño de la retícula en el <i>input _grid_size</i> del componente <i>LB Incident Radiation</i> .
	6.6	Establecer en el <i>input _offset_dist_</i> el valor 0.1 (10cm)
	6.7	Conectar un parámetro dinámico <i>Boolean Toggle</i> en el <i>input _run</i> del componente <i>LB Incident Radiation</i> , para activar la simulación.
	7. Implementar Algoritmos Genéticos	7.1
7.2		Extraer componente <i>Galapagos</i> y conectar el <i>input Fitness</i> al parámetro numérico extraído.
7.3		Conectar el <i>input Genome</i> de <i>Galapagos</i> directamente con todos los parámetros de <i>Gene Pool</i> existentes.
7.4		Dar doble <i>click</i> del <i>mouse</i> en <i>Galapagos</i> , determinar si se desea maximizar o minimizar el parámetro de forma en las opciones <i>Generic</i> .
7.5		Ir a la pestaña <i>Solvers</i> , activar la opción “visualizar todas las geometrías producidas” y finalmente activar la optimización evolutiva al presionar <i>Start Solver</i> .
8. Ver animación del resultado	8.1	Ver animación de transformación evolutiva en el siguiente vínculo web: https://youtu.be/eoK3zD0A4WY .

Nota. Elaboración propia (2020).

Optimización de forma evolutiva con Algoritmos Genéticos para minimizar la incidencia de radiación solar anual. Herramienta de Grasshopper: Ladybug y Galapagos.

Optimización evolutiva: minimizar incidencia

Ejemplares: 6

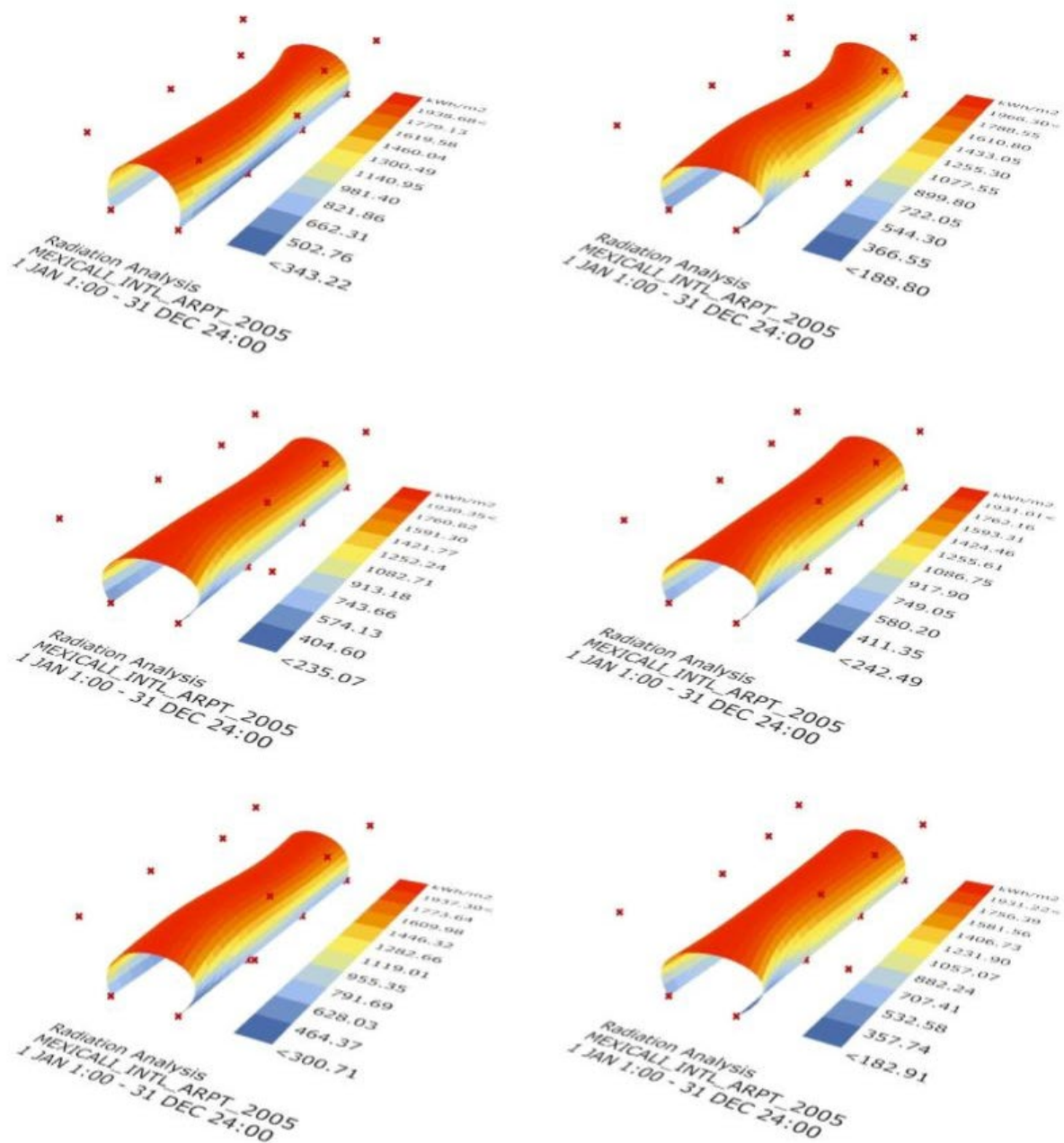


Figura 152. Resultado de algoritmo de *Grasshopper* “optimización evolutiva con Algoritmos Genéticos (AG) mediante *Galapagos*”. Variante: minimizar incidencia de radiación solar anual sobre superficie. Ver animación en el siguiente vínculo: <https://youtu.be/eoK3zD0A4WY>. Elaboración propia (2020).

IV. APLICACIÓN

El desarrollo metodológico propuesto anteriormente en torno a la Arquitectura Algorítmica Paramétrica (AAP) es una guía de referencia para su implementación en los procesos de diseño arquitectónico. Por tanto, *identificar condiciones de pertinencia para la aplicación de la metodología de diseño arquitectónico AP a través de instrumentos de comprobación*, obedece a la resolución del objetivo particular número tres del presente trabajo de investigación.

La aplicación de la metodología de diseño arquitectónico AP será analizada en esta sección y considerará como muestra de estudio al trabajo realizado por los estudiantes del taller de *Diseño Arquitectónico I (DAI)* y la asignatura optativa *Nociones de Arquitectura y Urbanismo Paramétrico (NAUP)*, de la Facultad de Arquitectura y Diseño, de la Universidad Autónoma de Baja California campus Mexicali. El objetivo es realizar un análisis comparativo entre una metodología de diseño tradicional aplicada en el taller de DAI con la metodología de diseño arquitectónico AP aplicada en el curso optativo NAUP.

En la aplicación de la metodología tradicional, las herramientas a utilizar para desarrollar el proyecto serán principalmente analógicas, a diferencia de las implementadas en la metodología de diseño arquitectónico AP que serán esencialmente digitales.

También es importante tomar en cuenta que, con fines experimentales y de análisis comparativo, se introdujo deliberadamente a metodologías con aspectos en común a considerar de diseño arquitectónico. DAI es una asignatura obligatoria de tercer semestre y los estudiantes por primera vez desarrollan un proyecto arquitectónico, bajo un esquema metodológico de diseño. NAUP es una clase optativa que puede ser elegida a partir de quinto semestre. La brecha formativa existente entre grupo de quinto semestre (en adelante) y el de

tercero, se reduce con el hecho de que el grupo más avanzado se enfrenta por primera ocasión a diseñar bajo un esquema de programación digital.

Para poder medir una metodología frente a otra, es necesario aplicarlas en el desarrollo de un mismo proyecto arquitectónico y periodo de tiempo. También se requerirá implementar un instrumento de evaluación, el cual consistirá en aplicar una ficha de diagnóstico que califique las características cualitativas y cuantitativas que tenga cada proyecto realizado con metodologías diferentes.

A través de este análisis es posible identificar brechas de pertinencia para la implementación de la metodología de diseño AP en los procesos de diseño arquitectónico profesional, académico y estudiantil. La finalidad es encontrar rutas de innovación en las estructuras epistemológicas y formativas de la disciplina.

Las características del diseño del análisis comparativo de metodologías se muestran en la siguiente matriz de congruencia metodológica (ver Tabla 26). La clasificación de características de análisis metodológico presentes en la matriz, fueron retomadas de referentes en el desarrollo de metodologías de investigación con relación a la disciplina de Arquitectura, los cuales son: Groat y Wang (2002), Hernández Sampieri et al. (2014), Barrantes Echavarría (1999) y Martínez Zarate (1991).

El objetivo particular de investigación se evalúa a detalle en la matriz a partir de su (a) enfoque, (b) sujetos u objetos de estudio, (c) variables que utilizará, (e) métodos que aplicará y (f) técnicas que utilizará.

Tabla 26

Matriz de congruencia metodológica del tercer objetivo particular de investigación.

Objetivo de investigación		
Identificar condiciones de pertinencia para la aplicación de la metodología de diseño arquitectónico algorítmico a través de instrumentos de comprobación, se considera a estudiantes del taller de Diseño Arquitectónico 1 y la clase optativa Nociones de Arquitectura y Urbanismo Paramétrico como sujetos de estudio.		
Enfoque	Variables	Técnicas

Objetivo de investigación		
1. Experimental y cuasi-experimental: -Hipotético -Tecnológico digital-virtual -Tecnológico performativo	1. Inter 2. Dicotómica	1. Instrumentos de aplicación cualitativa / cuantitativa 2. Instrumentos de evaluación cualitativa / cuantitativa 3. Gráficos de información cualitativa y cuantitativa 4. Tablas de información cualitativa / cuantitativa 5. Gráficos descriptivos 6. Instrumentos algorítmicos 7. Modelos virtuales 8. Simulaciones virtuales
Sujeto	Metodología	
1. Estudiantes de Arquitectura del taller Diseño Arquitectónico 1 de la FAD, UABC. 2. Estudiantes de Arquitectura del curso optativo semestral Nociones de Arquitectura y Urbanismo Paramétrico de la FAD, UABC.	Estrategias combinadas: 1. Simulación y modelaje 2. Cualitativa 3. Cuantitativa 4. Argumentación lógica	

Nota. Elaboración propia (2019). Clasificación de las características de análisis metodológico retomadas de Groat y Wang (2002), Hernández Sampieri et al. (2014), Barrantes Echavarría (1999) y Martínez Zarate (1991).

Tiene un enfoque experimental y cuasi-experimental de tipo hipotético, tecnológico digital-virtual y tecnológico performativo. Los sujetos de estudio serán los (1) estudiantes de arquitectura del taller de DA1, y (2) los estudiantes de arquitectura del curso optativo semestral NAUP.

Las variables para utilizar serán del tipo: (1) inter, por que ejercerá comparaciones entre miembros de una misma clase; y (2) dicotómica, porque contrastará resultados entre miembros de dos clases distintas.

Los métodos para aplicar serán combinados: (1) simulación y modelado en tercera dimensión, (2) análisis de aspectos cualitativos, (3) análisis de aspectos cuantitativos y (4) argumentaciones lógicas que demanda la elaboración de algoritmos.

Se utilizarán técnicas que implican (1) instrumentos de aplicación y (2) evaluación cualitativa y cuantitativa, (3) gráficos y (4) tablas de información cualitativa y cuantitativa, (5) gráficos descriptivos, (6) instrumentos algorítmicos, (7) modelos virtuales, (8) simulaciones virtuales.

4.1. Aplicación metodológica: descripción de proyecto arquitectónico

Para poder obtener datos para el análisis comparativo entre una metodología de diseño arquitectónico tradicional y una AP, se asignó un mismo proyecto a dos grupos diferentes.

Se trata de hacer una casa estudio localizada en la ciudad de Mexicali, Baja California, México. El tema fue rehabilitar un galerón en desuso con el fin de desarrollar una casa-habitación unifamiliar a manera de *loft*. Uno de los requisitos importantes para la realización del proyecto fue conservar la estructura existente de la edificación abandonada, y utilizarla como referencia para el desarrollo de una nueva envolvente que de refugio a todas las actividades de la casa.

Dentro de la estructura del galerón se consideraron tanto áreas interiores, como exteriores. El objetivo era generar entornos arquitectónicos dentro de la estructura y fomentar a través de ésta, una amplia comunicación entre espacios interiores y patios. El proyecto fue destinado a una pareja profesionista de mediana edad, con un rango de edad de 40 a 45 años.

4.1.2. Descripción del tema.

Una pareja de profesionistas de mediana edad heredó de su familia un galerón ubicado en la zona centro-norte de la ciudad. La pareja busca reformar el almacén y construir un *loft*, al aprovechar la estructura y la envolvente del edificio existente.

El almacén se encuentra en avenida República de Uruguay No. 85 Col. Cuauhtémoc Sur a solo media cuadra al este de la Calzada Justo Sierra (ver Tabla 27 y Figuras 153-155). La pareja se encuentra conformada por un periodista y promotor cultural con estudios de teatro y una oceanóloga con una vida agitada como activista proambiental y especialista en el Golfo de California.

Tabla 27

Características de ubicación del sitio a desarrollar proyecto: casa estudio loft.

Proyecto			
Tipo de proyecto			
Residencia. Vivienda Unifamiliar			
Cantidad de niveles			
2			
Superficie para construir (m²)			
110			
Año			
2020			
Ubicación			
País	Estado	Municipio	Ciudad
México	Baja California	Mexicali	Mexicali
Zona Climática (California Energy Comission)			
Zona climática 15, CA, USA (latitud 32.8° Norte)			
Zona Climática (IECC ASHRAE Climate Zones)			
Zona climática 2B, Phoenix AZ, USA			
Altura sobre el nivel del mar			
1m			

Nota. Elaboración propia (2020).



Figura 153. Imagen frontal de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.



Figura 154. Imagen panorámica de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.



Figura 155. Imagen interior de galerón a rehabilitar como casa estudio. Mexicali, Baja California, México.

4.1.3. Programa arquitectónico y condiciones.

El *lotf* deberá adecuarse para habilitar el programa arquitectónico de una casa-estudio. Debe considerar las características físicas, estructurales y espaciales del galerón existente, al igual que los espacios exteriores. Debido a que la superficie total de la construcción será significativamente menor a la superficie del almacén, *se considerarán patios interiores dentro de la envolvente*.

La propuesta de materiales para la construcción será: concreto para piso, tabique de barro para muros, entrepisos y cubierta de madera. En los exteriores se buscará generar un microclima confortable a través de vegetación endémica o adaptada al clima de Mexicali, que requiera poco mantenimiento. También es posible implementar estanques o espejos de agua y cubiertas a base de pérgolas, mallas o lonas.

El galerón mide aproximadamente 10.20m por 20.40m., con una altura de 8.40m. Se solicita no exceder de 110m² de construcción de espacios interiores. El programa arquitectónico contempla espacios enlistados a continuación (ver Tabla 28).

Tabla 28

Relación de espacios a diseñar para proyecto casa estudio loft en la ciudad de Mexicali.

Espacio	Superficie (m²)	Características
Estancia-comedor	36	Lugar para descanso en privado o con visitas, deberá invitar a la relajación y contemplación, escuchar música y leer. Tendrá un comedor de seis plazas y una sala de estar, totalmente comunicadas y libres entre sí. También puede incluir juego de sillas, hamacas y otras opciones de mobiliario. Contemplar chimenea para invierno.
Cocina	12	Área para preparar alimentos. Contará con estufa, refrigerador y tarja de dimensiones pequeñas, gabinetes para almacenar utensilios, además de una barra (tipo isla) para la preparación de los alimentos, con una capacidad de tres plazas.
Medio baño	3	Accesible desde el área social interior y desde el área exterior. Debe contener inodoro y lavabo.
Lavandería	6	Lugar para labores domésticas.
Estudio-biblioteca	24	Área para el trabajo y la lectura. Deberá ubicarse entre el área social y el área privada. Se debe considerar librerías, dos escritorios con espacio para computadora, sillas, mesa para impresora y archiveros.
Dormitorio y baño	24	El espacio con mayor nivel de privacidad. Contará con cama tamaño <i>King</i> , tocador, mesas de noche y closet. El baño tendrá lavabo, regadera e inodoro.
Espacios exteriores	-	Consistirán en patios (interiores a envolvente de galerón), terrazas-miradores, balcones, jardines y estanques. Se contemplará estacionamiento para dos automóviles como elemento externo al galerón.

Nota. Elaboración propia (2020).

4.2. Aplicación de metodología de diseño arquitectónico tradicional: Taller de Diseño

Arquitectónico 1

El taller de Diseño Arquitectónico 1 (DA1) consiste en una asignatura obligatoria dentro del programa de licenciatura de la Facultad de Arquitectura y Diseño (FAD) de la Universidad Autónoma de Baja California (UABC). Los talleres de Diseño Arquitectónico (DA) son la columna vertebral de la carrera de Arquitectura, ya que son las asignaturas donde se desenvuelve el proceso de diseño espacial-constructivo y del cual derivan el resto de los talleres y cursos que oferta la FAD.

La FAD ofrece tres programas de licenciatura: Arquitectura, Diseño Gráfico y Diseño Industrial. El programa de estudios de Arquitectura se desarrolla en seis semestres, más dos semestres iniciales que obedecen al tronco común de las tres carreras ofertadas (ocho semestres en total). Por tanto, es en el tercer semestre de la FAD donde se cursa la asignatura de DA1, y donde se abre paso al aprendizaje de todas las competencias relacionadas con la disciplina de Arquitectura.

Es en DA1 donde se da la introducción a conocer la metodología de diseño arquitectónico. El primer proyecto para resolver consiste en el diseño de una edificación uni-espacial; posteriormente, en el segundo ejercicio se desarrolla un proyecto de vivienda unifamiliar de planta abierta con la posibilidad de incluir un espacio de trabajo. Finalmente, se diseña el tercer proyecto que consiste en una vivienda de tipo vacacional.

El segundo proyecto, descrito en el esquema de trabajo anterior, es el que tomó como referencia para hacer el estudio comparativo entre metodologías de diseño. Específicamente, el ejercicio obedeció al ciclo escolar de la primera mitad del año 2020 (ciclo 2020-1). El grupo estuvo conformado por trece estudiantes con un rango de edad de 19 a 20 años, seis de ellos son mujeres y siete hombres. Cada uno de ellos desarrolló el proyecto asignado de manera individual, en un periodo de ocho semanas.

Para poder ejercer una comparación entre las dos metodologías, fue importante asegurar la implementación de elementos metodológicos en común en el contenido de cada uno de los dos cursos que fueron sujeto de estudio. Se entregó al inicio del curso un encuadre que estructura y muestra el contenido de temas completo, las competencias, los objetivos y alcances a cumplir en su desarrollo (ver Tabla 29).

Tabla 29

Encuadre del taller de Diseño Arquitectónico 1. Tercer semestre, ciclo escolar 2020-1.

Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California.

1 Unidad 2: diseño de una vivienda unifamiliar de planta abierta (casa-estudio)	
Competencia: Diseñar objetos arquitectónicos de planta abierta, preferentemente no mayor de 80m ² , con programas arquitectónicos que impliquen pocas unidades espaciales como, por ejemplo: el taller de un artista, un estudio, apartamento para estudiante, etc. Dando una respuesta congruente con el medio físico y las necesidades del usuario.	
Duración: ocho semanas.	
Semana 1	
1.1	Introducción al diseño arquitectónico. Presentación de clase, entrega y explicación de encuadre.
1.2	Exposiciones por parte del alumno: casos análogos, proyectos de intervención sobre construcciones existentes.
1.3	Presentación de proyecto y programa arquitectónico por parte del maestro.
1.4	Visita al sitio
Semana 2	
1.4	Investigación y exposición por parte de alumno del tema: Elementos de diseño. <ul style="list-style-type: none"> ▪ Equipo 1. Dimensiones de una casa ▪ Equipo 2. Armado y tipo de cubiertas ▪ Equipo 3. Armado y tipo de muros ▪ Equipo 4. Jardines, pérgolas y celosías
Semanas 3 y 4	
1.5	Metodología de diseño arquitectónico
1.5.1	Análisis del sitio
1.5.2	Concepto arquitectónico
1.5.3	<i>Arquitectura como un sistema de comunicaciones</i> <ul style="list-style-type: none"> ▪ <i>Principios funcionales y formales: organización, articulación, fenomenología, semiología, utilidad y belleza.</i>
1.5.4	Análisis de función espacial <ul style="list-style-type: none"> ▪ Diagramas y matrices de funcionamiento y relación de espacios
1.5.5	Zonificación espacial
1.5.6	Diseño volumétrico y emplazamiento
1.5.7	Desarrollo de plantas arquitectónicas
1.5.8	Desarrollo de alzados y cortes arquitectónicos
1.5.9	Criterio estructural
1.5.10	<i>Adaptabilidad ambiental: estrategias bioclimáticas, sustentables y de accesibilidad</i> <ul style="list-style-type: none"> ▪ <i>Optimización y desempeño: geometría del espacio, envolvente arquitectónica, aberturas, materiales, iluminación y ventilación natural.</i>
1.5.11	Visualización arquitectónica: perspectivas, isométricos, axonométricos, renders, collages.
1.6	Entrega de primera etapa de la metodología de diseño arquitectónico: <ul style="list-style-type: none"> ▪ Punto 1.5.1 a 1.5.6 de la metodología enlistada anteriormente.
Semana 5	
1.7	Primeras propuestas de diseño, plantas arquitectónicas a escala 1:50 amuebladas.
Semana 6	
1.8	Propuesta de planta arquitectónica
Semana 7	
1.9	Propuesta de alzados y cortes
Semana 8	
1.10	Maquetas de estudio, revisiones y consolidación de proyecto
Semana 9 y 10	
1.11	Entrega final de proyecto

Nota. Elaboración propia (2020).

Los aspectos metodológicos compartidos con el curso optativo NAUP son: análisis de los principios funcionales y formales; relacionados con la organización, la articulación, la fenomenología, la semiología, utilidad y belleza, desde una perspectiva cualitativa, según la metodología AP, que tiene como referencia la teoría y estructura metodológica racionalizada por Schumacher (2009, 2011, 2012).

También, otro punto compartido es el análisis de adaptabilidad del proyecto con el entorno, para considerar en el desarrollo del proyecto estrategias bioclimáticas, sustentables y de accesibilidad, a través del estudio del desempeño la envolvente arquitectónica, las aberturas, los materiales, la iluminación y ventilación natural.

Por la naturaleza de las competencias del tercer semestre en la FAD, los estudiantes se introducen a una metodología de diseño arquitectónico, y lo hacen a través de herramientas analógicas bajo un esquema tradicional. El proceso de diseño tuvo una resolución intuitiva y experimental, apoyados con prototipos a escala, bosquejos, diagramas y planos arquitectónicos hechos a mano (ver Figura 156).

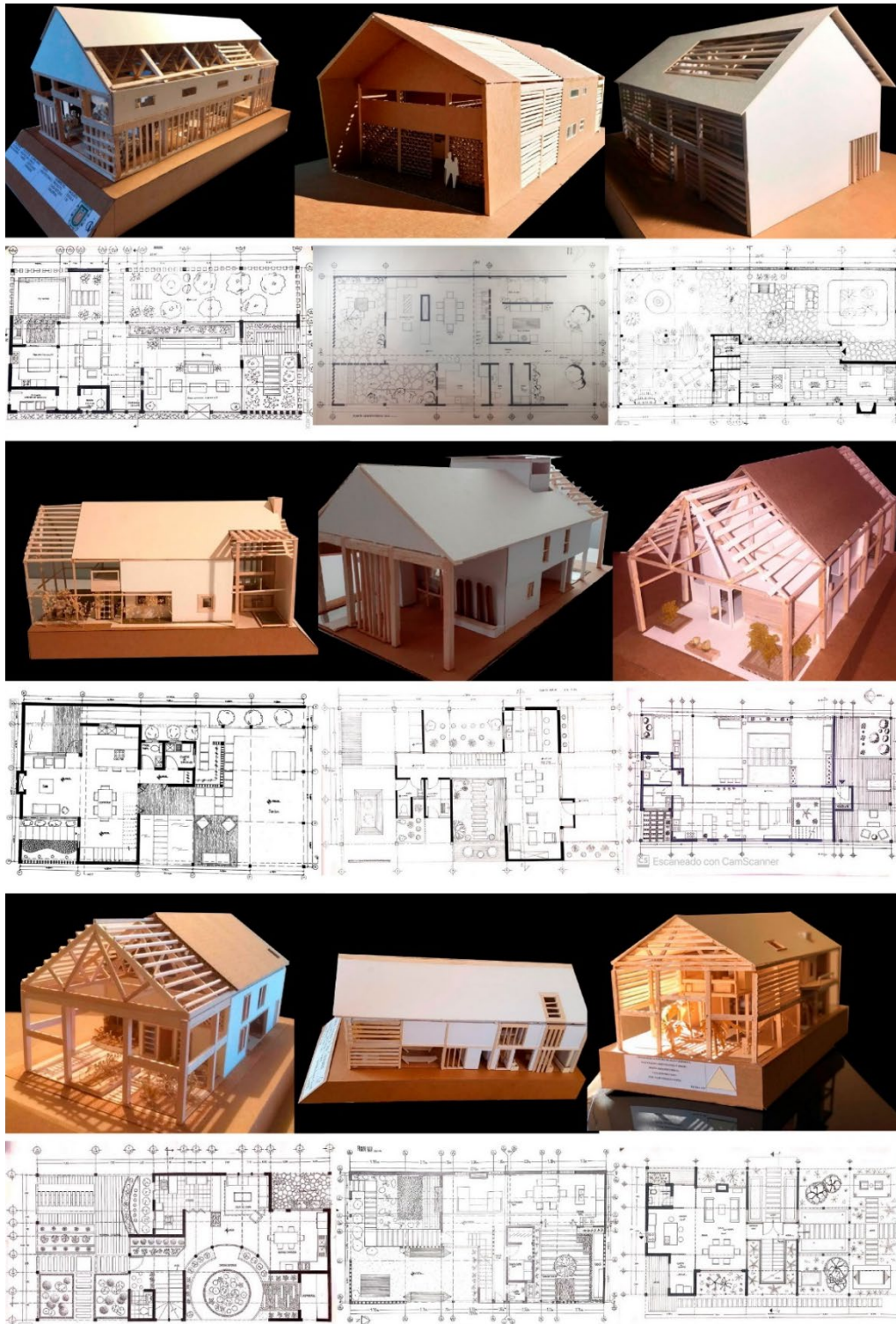


Figura 156. Metodología de diseño arquitectónico tradicional: imagen de resultados del taller de Diseño Arquitectónico 1. Tercer semestre, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California. Elaboración propia (2020).

4.3. Aplicación de metodología de diseño AAP: Nociones de Arquitectura y Urbanismo Paramétrico.

Nociones de Arquitectura y Urbanismo Paramétrico (NAUP) es una clase optativa que se oferta a estudiantes de licenciatura de Arquitectura de la FAD con interés en la tecnología, las herramientas computacionales y la vanguardia arquitectónica. El curso se implementó por primera vez en el programa de licenciatura en el año 2012, y tiene disponibilidad de elección para estudiantes de quinto semestre en adelante.

En el curso se introduce a la teoría y la práctica de la programación digital dirigida al diseño de objetos con la construcción de algoritmos visuales a través de *Grasshopper*. El curso se encuentra conformado por cuatro etapas: (1) introducción teórica, (2) introducción a *Rhinoceros* como programa CAD, (3) introducción a la programación visual con *Grasshopper*, y (4) aplicación de una metodología de Arquitectura AP (ver Tabla 30).

Tabla 30

Encuadre del curso optativo Nociones de Arquitectura y Urbanismo Paramétrico. Quinto semestre en adelante, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California.

Competencia general	
Diseñar definiciones paramétricas a través de la composición de algoritmos visuales para la aplicación al modelado en dos y tres dimensiones de objetos geométricos, mediante el uso de programas computacionales especializados de representación digital, para crear, transformar y manipular elementos arquitectónicos con apego a condiciones reales, con creatividad, orden e innovación.	
1	Unidad 1: introducción teórica. (2 semanas)
Competencia: Identificar los conceptos teóricos del modelado paramétrico, a través de la revisión bibliográfica y electrónica, para comprender su evolución, contextos del modelado y procesos de diseño paramétrico-algorítmico.	
1.1	Introducción a la arquitectura algorítmica-paramétrica.
1.2	Ejemplos de arquitectura algorítmica-paramétrica.
1.3	Posición histórica de la arquitectura algorítmica-paramétrica
1.4	Contexto disciplinar de la arquitectura algorítmica-paramétrica
1.5	Morfogénesis en el proceso de diseño
	1.5.1 Sistemas de organización y forma: Geometría y Topología
	1.5.2 Morfogénesis digital, modelado a partir de la información
1.6	Actividad: investigación y presentación sobre la Arquitectura Algorítmica-Paramétrica

Competencia general**2 Unidad 2: introducción al modelado 3D a través de *Rhinceros*®. (3 semanas)**

Competencia:

Modelar elementos en dos y tres dimensiones, a través del manejo de herramientas CAD, para su implementación en la metodología de diseño arquitectónico.

- 2.1 Introducción al uso de *Rhinceros*® como programa CAD
- 2.2 Introducción a modelos matemáticos para la generación y representación gráfica computacional de formas:
 - Superficies y su clasificación
 - Mallas (*Mesh*): vértices, bordes y caras.
 - NURBS: curvas, superficies y sus componentes
- 2.3 Ejercicios:
 - Creación, edición y análisis de curvas, superficies, sólidos NURBS
 - Prácticas de *Rhinceros*® para el modelado de objetos.
- 2.4 Actividad: prácticas diversas de modelado con *Rhinceros*®

3 Unidad 3: introducción al diseño paramétrico a través de *Grasshopper*. (4 semanas)

Competencia:

Desarrollar objetos bidimensionales y tridimensionales, mediante el uso de herramientas de programación algorítmica visual, para su implementación en la metodología de diseño arquitectónico.

- 3.1 ¿Qué es y cómo funciona un algoritmo?
- 3.2 Inducción al uso de *Grasshopper* como plataforma paramétrica
 - 3.2.1 Interfaz de *Grasshopper*
 - 3.2.2 Parámetros y funciones (componentes de *Grasshopper*)
 - *Inputs / outputs*
 - Flujo de información
 - 3.2.3 Listas y árbol de información
- 3.3 Geometría múltiple en superficies subdivididas
- 3.4 Elementos atractores
- 3.5 Superficie NURBS / *Mesh* a partir de puntos aleatorios
- 3.6 Superficie NURBS / *Mesh* a partir de función matemática gráfica
- 3.7 Elementos atractores en superficies subdivididas
- 3.8 Ejercicios algorítmicos-paramétricos de mayor complejidad
- 3.9 Actividad: practicas diversas de diseño con *Grasshopper*

4 Unidad 4: Algoritmos y arquitectura: una aportación al diseño. (6 semanas)

Competencia:

Aplicar metodologías de modelado paramétrico, mediante la composición de los algoritmos visuales, para diseñar, simular y analizar objetos industriales y urbano-arquitectónicos, con orden, proactividad y creatividad.

- 4.1 Metodologías de la arquitectura algorítmica-paramétrica
 - 4.1.1 Estructura metodológica del Parametricismo de Schumacher
 - 4.1.2 Estructura metodológica de la Arquitectura Algorítmica de Terzidis
 - 4.1.3 Metodología de investigación y producción de la Arquitectura Performativa y del Discretismo
- 4.2 *Introducción a una metodología de arquitectura algorítmica-paramétrica como instrumento de aplicación en el proceso de diseño arquitectónico.*
 - 4.2.1 *Metodología para el análisis cualitativo de la forma y el espacio:*
 - *Arquitectura como un sistema de comunicaciones: principios causales-heurísticos para el análisis de la función y la forma (organización, articulación, fenomenología, semiología, utilidad y belleza).*
 - 4.2.2 *Metodología para el análisis cuantitativo de la forma y el espacio:*
 - *Optimización y desempeño a través de principios causales-heurísticos para el análisis de la estructura y la adaptabilidad al medio ambiente.*
- 4.3 *Aplicación de una metodología de arquitectura algorítmica-paramétrica*

Competencia general

4.3.1 Proyecto arquitectónico

Nota. Elaboración propia (2020).

En la cuarta etapa los estudiantes aplicaron la metodología AP (propuesta en este trabajo de investigación) a través de la realización del mismo proyecto de diseño arquitectónico asignado para la clase de DA1 en un periodo de seis semanas para realizarlo. El grupo estuvo conformado por veintitrés estudiantes, de los cuales, once fueron hombres y doce mujeres. El rango de edad de los estudiantes fue de veinte a veintidós años (ver Figura 157). Posteriormente, se hizo un filtro y se seleccionó a dieciséis trabajos con mayor desempeño en el resultado.

Se entregó, primeramente, un encuadre con el contenido de las competencias a desarrollar, los objetivos y los alcances. La estructura metodológica de diseño arquitectónico AP, comparte el análisis de los principios funcionales y formales cualitativos, así como los aspectos cuantitativos de optimización y desempeño para el diseño del proyecto arquitectónico.

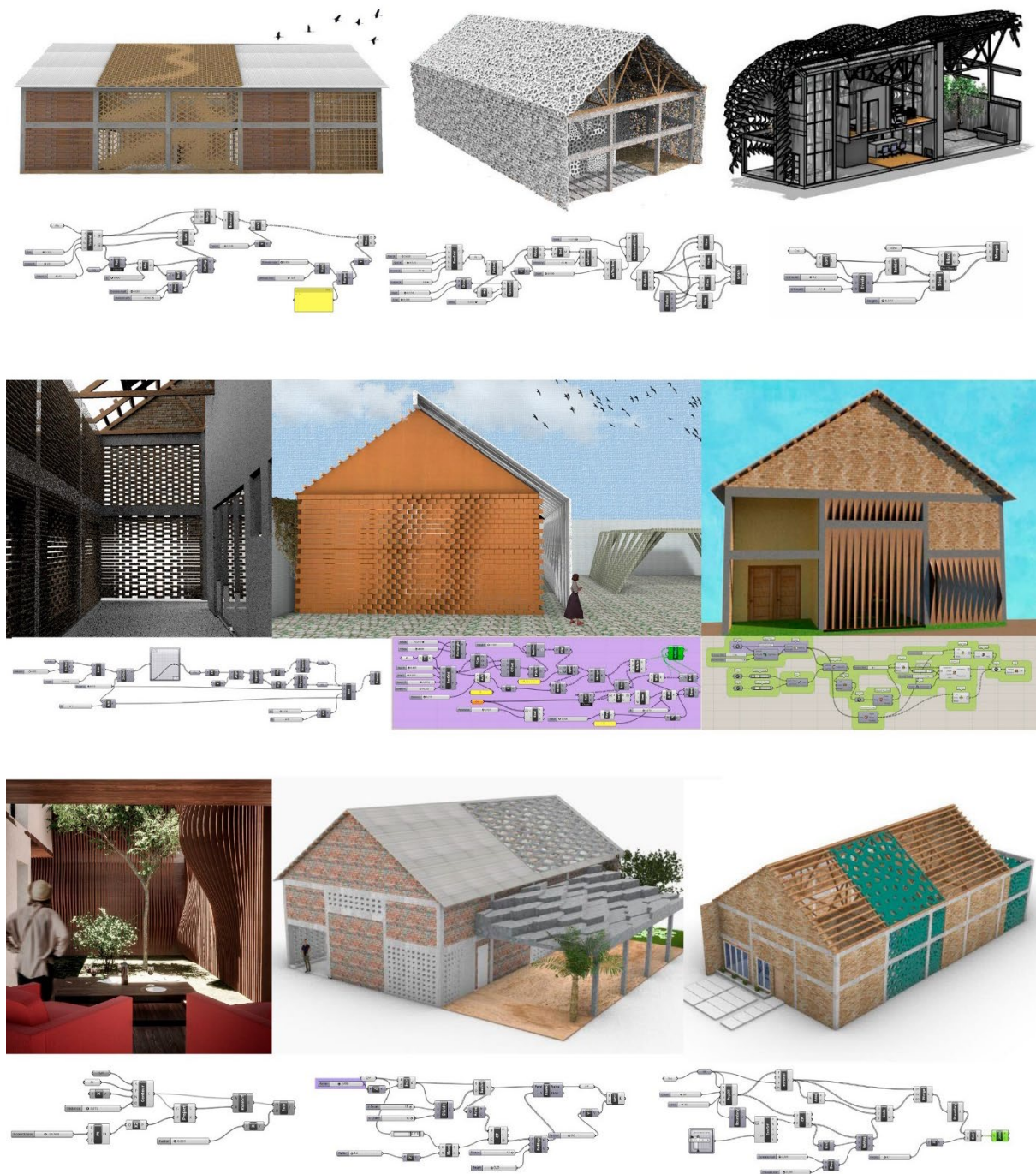


Figura 157. Metodología de diseño arquitectónico Algorítmica Paramétrica (AP): imagen de resultados del curso optativo Nociones de Arquitectura y Urbanismo Paramétrico. Quinto semestre en adelante, ciclo escolar 2020-1. Facultad de Arquitectura y Diseño, Universidad Autónoma de Baja California. Elaboración propia (2020).

4.4. Instrumentos de análisis y evaluación metodológica

La revisión de proyectos arquitectónicos realizados tanto por el taller de DA1 y por la clase optativa NAUP, fueron contrastados a través de una ficha de *análisis y evaluación metodológica*. El objetivo de la ficha es analizar los aspectos metodológicos que compartieron los procesos de diseño de los dos grupos que son sujeto de estudio.

El diseño de este instrumento de evaluación se dividió en dos partes, por tanto, contempló dos aplicaciones con diferente nivel de análisis de variables. Primeramente, las variables del instrumento tienen un carácter dicotómico, ya que contrastaron el desempeño metodológico de una metodología de diseño arquitectónico tradicional, con respecto a la metodología de diseño AP.

Posteriormente, el instrumento analizó las variables desde un enfoque inter debido a que, en esta segunda parte, el objetivo es hacer una comparación y evaluación de la habilidad obtenida por los estudiantes de la clase NAUP con el uso de las herramientas algorítmicas, guiados por la metodología AP instruida en el curso.

4.4.1. Primera parte del instrumento de análisis y evaluación metodológica.

La primera parte del instrumento metodológico se encuentra constituida por parámetros que son parte del desarrollo metodológico AP propuesto (ver Figura 158). Las variables para analizar son (a) la metodología que el estudiante utilizó: (1) tradicional, (2) algorítmica paramétrica; (b) la herramienta que implicó la aplicación de la metodología: (1) analógica, (2) digital.

También se analiza el enfoque que cada proyecto realizado tuvo, según el marco teórico de la Arquitectura AP, el cual puede ser (a) morfológico, (b) tecnológico o (c) político. Por otro lado, da seguimiento al carácter que la propuesta arquitectónica presenta a partir de las líneas de análisis a través de la cual la metodología AP se desenvuelve.

La primera línea de análisis es la cualitativa, que se basa en los principios formales y funcionales propuestos por Schumacher (2009, 2011, 2012) en la teoría y metodología del Parametricismo. Esta línea, verifica las relaciones cognitivas de articulación a partir del carácter (a) fenomenológico o (b) semiológico que presente cada uno de los proyectos resultantes. El criterio teórico a través del cual se analiza estas variables se basan en los aspectos topológicos cognitivos de Norberg Schulz (1999; 1971) y los estímulos sensoriales producidos por la envolvente corporal y las coordenadas psicofísicas según Bloomer y Moore (1983).

También, esta línea hace una evaluación formal según las relaciones físicas de organización, mediante características dicotómicas como: (1) equilibrio o inestabilidad, (2) simetría o asimetría, (3) regularidad o irregularidad, (4) simplicidad o complejidad, (5) unidad o fragmentación, (6) economía o profusión. Finalmente, analiza los códigos de utilidad – función o belleza – forma (Schumacher, 2009, 2011, 2012).

Por otro lado, la segunda línea de análisis tiene un carácter cuantitativo y está relacionado con las aptitudes de optimización y desempeño de cada proyecto realizado. Verifica si la estructura tiene un carácter (a) geométrico o (b) topológico, evalúa si la propuesta presenta criterios de adaptabilidad con el ambiente y analiza si fueron con estrategias de (a) masa térmica, (b) protección solar o (c) ventilación, a través de implementar (1) materialidad, (2) permeabilidad, (3) celosías, (3) pérgolas, (4) parte soles o (5) vegetación.

Algoritmos y arquitectura: una aportación al diseño	
Ficha de análisis y evaluación metodológica	
Maestría en Arquitectura, Urbanismo y Diseño Facultad de Arquitectura y Diseño Universidad Autónoma de Baja California	
Matrícula	Nombre de alumno
Semestre	Curso
Metodología	
Tradicional	<input type="radio"/> Algorítmica paramétrica <input checked="" type="radio"/>
Herramienta	
Análogica	<input type="radio"/> Digital <input checked="" type="radio"/>
Enfoque	
Morfológico	<input type="radio"/> Tecnológico <input checked="" type="radio"/> Político <input type="radio"/>
Análisis de aspectos cualitativos (principios formales y funcionales)	
a) Relaciones cognitivas (articulación)	
Fenomenológico	<input checked="" type="radio"/> Semiológico <input type="radio"/>
b) Relaciones físicas (organización)	
Equilibrio	<input checked="" type="radio"/> Simetría <input type="radio"/> Regularidad <input checked="" type="radio"/> Simplicidad <input checked="" type="radio"/> Unidad <input checked="" type="radio"/> Economía <input checked="" type="radio"/>
Inestabilidad	<input type="radio"/> Asimetría <input type="radio"/> Irregularidad <input type="radio"/> Complejidad <input type="radio"/> Fragmentación <input type="radio"/> Profusión <input type="radio"/>
c) Código	
Utilidad / Función	<input checked="" type="checkbox"/> Belleza / Forma <input type="checkbox"/> No aplica <input type="checkbox"/>
Análisis de aspectos cuantitativos (optimización y desempeño)	
a) Estructura	
Geométrica	<input checked="" type="radio"/> Topológica <input type="radio"/>
b) Adaptabilidad ambiental	
Si	<input checked="" type="radio"/> No <input type="radio"/>
c) Estrategia	
Masa térmica	<input checked="" type="checkbox"/> Protección solar <input type="checkbox"/> Ventilación <input type="checkbox"/>
d) Implemento	
Materialidad	<input type="checkbox"/> Permeabilidad <input checked="" type="checkbox"/> Celosías <input checked="" type="checkbox"/> Pérgolas <input checked="" type="checkbox"/> Partesoles <input type="checkbox"/> Vegetación <input type="checkbox"/>



Figura 158. Ejemplar de ficha de análisis y evaluación metodológica, primera parte (análisis dicotómico). Elaboración propia (2021).

4.4.2. Segunda parte del instrumento de análisis y evaluación metodológica.

La segunda parte del instrumento metodológico, también se encuentra constituida por parámetros que son parte del desarrollo metodológico AP propuesto (ver Figura 159). Las variables para analizar son el enfoque que cada proyecto realizado tuvo, según el marco teórico de la Arquitectura AP, el cual puede ser (a) morfológico, (b) tecnológico o (c) político.

También, a partir de ese enfoque se analiza la tendencia con la que el proyecto de cada estudiante se identificó, las cuales son (según el marco teórico): (a) escultórico, (b) parametricista, (c) algorítmico, (d) performativo o (d) discretismo.

Finalmente se identifica, a partir del instrumento, la estrategia algorítmica que utilizó: (1) geometría en superficie subdividida, (2) elementos atractores, (3) retícula diagonal (*Diagrid*) a partir de una superficie, (4) superficie generada a partir de una función matemática, (5) superficie generada a partir de puntos aleatorios, (6) superficie generada por una función matemática, (7) transformación topológica de mallas a través de *Weaverbird*, (8) simulación de sistemas topológicos estructurales con *Kangaroo*, (9) agregación compleja con *Wasp*, simulaciones para análisis térmico energético con (10) *Ladybug* o (11) *Honeybee*, y (12) Algoritmos Genéticos (AG) con *Galapagos*.

Algoritmos y arquitectura: una aportación al diseño			
Ficha de análisis y evaluación metodológica			
Maestría en Arquitectura, Urbanismo y Diseño Facultad de Arquitectura y Diseño Universidad Autónoma de Baja California			
Matrícula	Nombre de alumno		
Semestre	Curso		
	Nociones de Arquitectura y Urbanismo Paramétrico		
Enfoque			
Morfológico <input type="radio"/>	Tecnológico <input checked="" type="radio"/>		
	Político <input type="radio"/>		
Tendencia			
Escultórico <input type="radio"/>	Algorítmico <input type="radio"/>		
Parametricista <input type="radio"/>	Performativo <input type="radio"/>		
	Discretismo <input checked="" type="radio"/>		
Estrategia algorítmica			
1. Geometría en superficie subdividida	<input type="checkbox"/>	7. <i>Weaverbird</i>	<input type="checkbox"/>
2. Elementos atractores	<input type="checkbox"/>	8. <i>Kangaroo</i>	<input type="checkbox"/>
3. Retícula diagonal (<i>Diagrid</i>)	<input type="checkbox"/>	9. <i>Wasp</i>	<input type="checkbox"/>
4. Superficie - función matemática	<input type="checkbox"/>	10. <i>Ladybug</i>	<input type="checkbox"/>
5. Superficie - aleatoria	<input type="checkbox"/>	11. <i>Honeybee</i>	<input type="checkbox"/>
6. Superficie - función matemática gráfica	<input type="checkbox"/>	12. <i>Galapagos (Algoritmos Genéticos)</i>	<input type="checkbox"/>



Figura 159. Ejemplar de ficha de análisis y evaluación metodológica, segunda parte (análisis inter).

Elaboración propia (2021).

4.5. Resultados

Los instrumentos de análisis y evaluación de las metodologías de diseño arquitectónico fueron utilizados a partir de la entrega final de cada proyecto realizado por los integrantes de los grupos que son sujeto de estudio para la presente investigación. Las dos partes del instrumento ayudaron a efectuar las mediciones de tipo dicotómica e inter necesarias para analizar el comportamiento de cada una de las metodologías contrastadas.

Todos los proyectos asignados fueron entregados, sin embargo, no todos cumplieron con los requerimientos de entrega, por lo que fueron descartados para efectos de realizar la medición. Cabe mencionar, que las respectivas calificaciones que los estudiantes obtuvieron en cada curso no influyeron de ninguna forma con la medición aquí establecida. También, los criterios de evaluación fueron totalmente distintos, por tanto, las rúbricas contenían diferentes aspectos para análisis de los proyectos y estaban orientadas a diferentes propósitos, contraste y medición.

También, de los veintitrés participantes de la clase optativa NAUP del ciclo 2020-1, fueron filtrados una cantidad de siete trabajos. Es así como, solo dieciséis trabajos fueron seleccionados para realizar el análisis y fueron aquellos que cumplieron de mejor manera los objetivos, competencias y presentación del curso. Todos los estudiantes del taller DA1 (trece participantes) aplicaron una metodología de diseño tradicional, con herramientas de trabajo analógicas. También, todos los estudiantes de la clase optativa NAUP aplicaron una metodología de diseño arquitectónico AP.

Según el criterio de análisis y evaluación del instrumento propuesto en este trabajo de investigación, el primer aspecto a contrastar, de una metodología con respecto a otra, es el enfoque de diseño al cual se inclinaron cada uno de los estudiantes que desarrollaron el proyecto en los diferentes grupos.

Los enfoques para contrastar fueron: (a) el morfológico, (b) el tecnológico y (c) el político. Aunque estos enfoques derivan del estado del arte de la Arquitectura Algorítmica Paramétrica, funcionaron como elementos de referencia para verificar si los estudiantes se aproximaban más a resolver la forma, los aspectos funcionales y tecnológicos o si los orientaban a un sentido crítico de las cosas. Los resultados de contrastar estos aspectos se muestran a continuación (ver Figura 160).

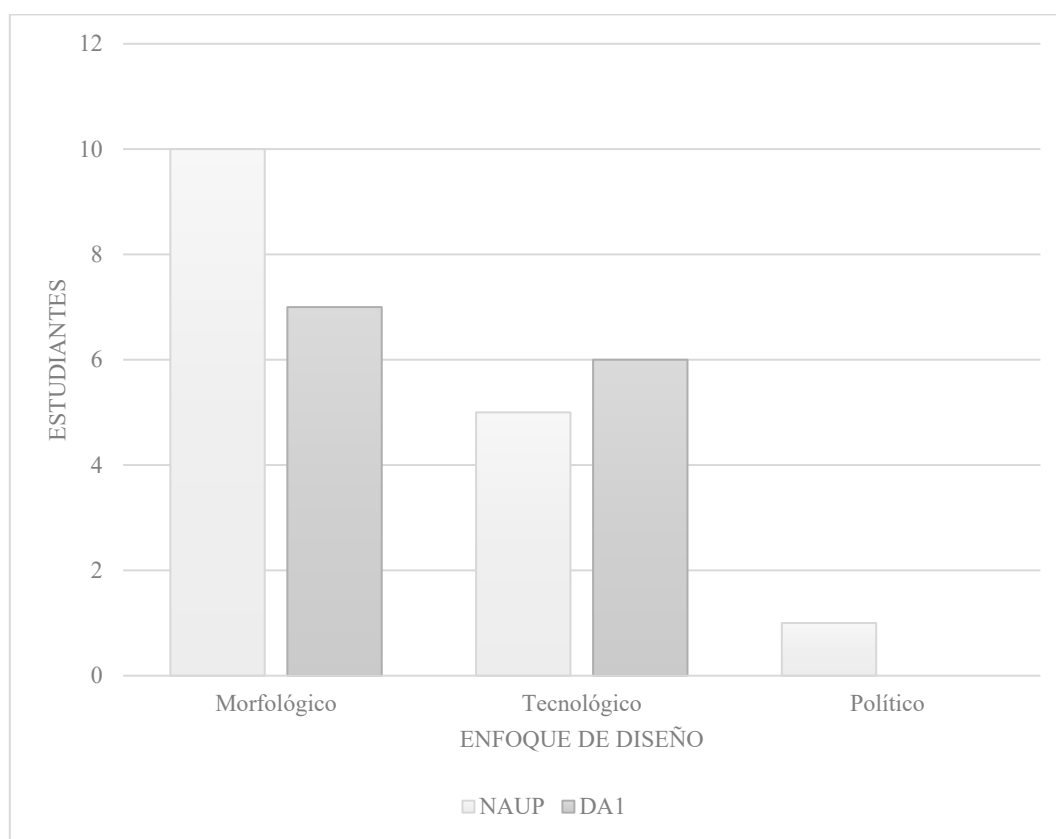


Figura 160. Relación de enfoques de diseño manifestados en la realización del proyecto “Casa Estudio Loft” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

La gráfica muestra una mayor cantidad de estudiantes de la clase optativa NAUP, que manifestaron un enfoque de diseño morfológico. A diferencia de los estudiantes de DA1 que tuvieron mayor interés en descifrar aspectos funcionales y tecnológicos en el diseño del

proyecto. Solo un estudiante de la clase NAUP, expreso un enfoque crítico y político en la resolución del proyecto.

A partir del previo análisis, las evaluaciones que siguieron se desarrollaron bajo dos líneas de análisis según la estructura del instrumento. La primera línea corresponde al (a) análisis de los aspectos cualitativos de la forma y el espacio en el proceso de diseño. La segunda línea, analiza los aspectos cuantitativos de optimización y desempeño fisiológico del proyecto arquitectónico.

4.5.1. Análisis y evaluación de los aspectos cualitativos de la forma y el espacio.

El análisis contrasta los aspectos cualitativos de la forma y el espacio a partir de las relaciones cognitivas de articulación, las relaciones físicas de organización y los códigos de utilidad – función o belleza – forma, según Schumacher (2009, 2011, 2012). A continuación, se muestran los resultados obtenidos de haber aplicado el instrumento de evaluación que este trabajo propone.

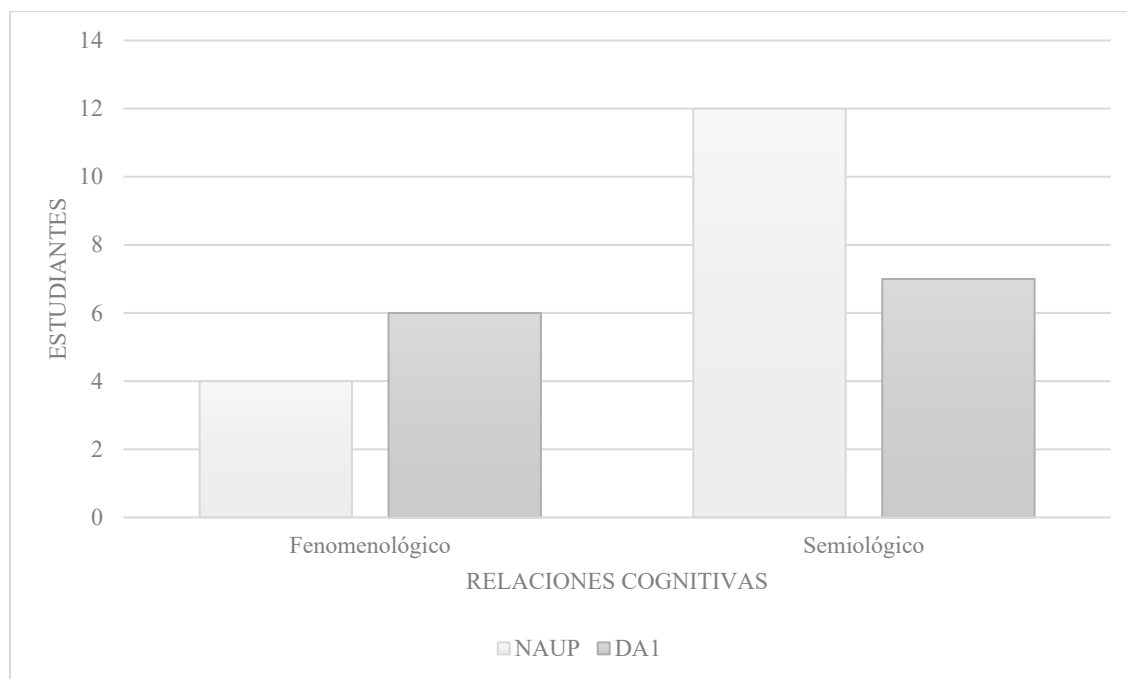


Figura 161. Relaciones cognitivas fenomenológicas y semiológicas manifestados en la realización del proyecto “Casa Estudio Loft” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

La gráfica (ver Figura 161) muestra que, una cantidad mayor de estudiantes del grupo de DA1 profundizó más en los aspectos fenomenológicos de la percepción del espacio, mientras que el curso optativo NAUP se apegó a la exploración conceptual y formal de los proyectos. Sin embargo, los aspectos semiológicos no fueron descartados por el taller de DA1, ya que puede apreciarse que la mayoría de los estudiantes exploraron estos aspectos.

Con respecto a las relaciones físicas de organización, se hizo un análisis a partir de las propiedades que manifestó cada proyecto en su forma. Cada una de esas propiedades son evaluadas de manera dicotómica, de acuerdo con la aptitud formal que el objeto arquitectónico expresa (ver Figura 162).

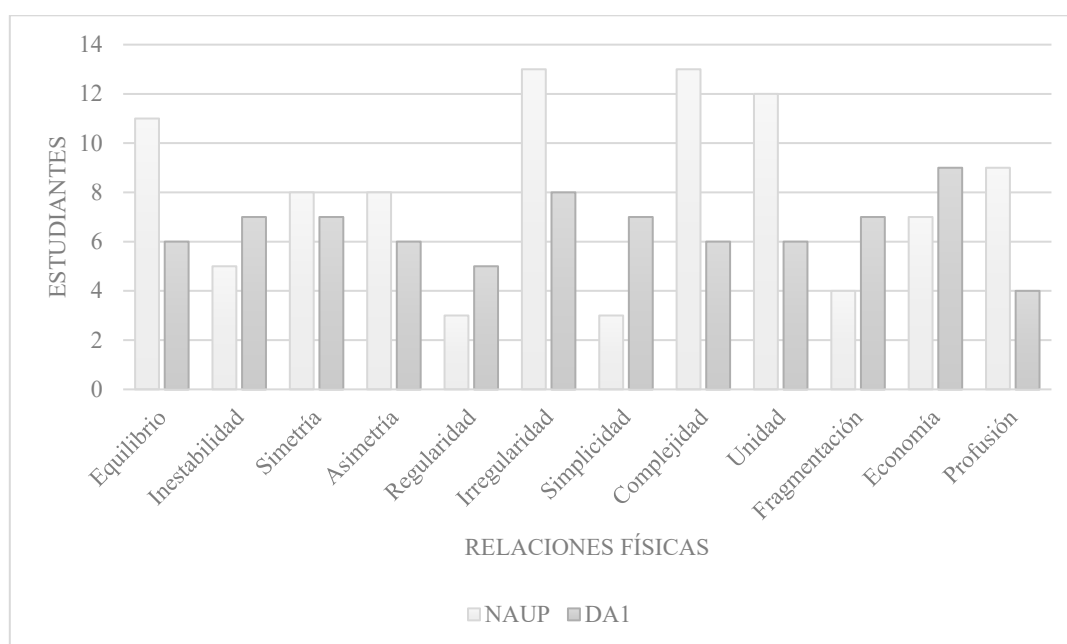


Figura 162. Relaciones físicas manifestadas en la realización del proyecto “Casa Estudio Loft” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

Aunque los integrantes del curso optativo NAUP, mostraron una mayor cantidad de propuestas con un carácter físico de organización equilibrada, es también evidente que

significativamente exploraron alternativas de diseño con relaciones de organización irregular, compleja y características que expresan mayor profusión.

A pesar de ello, la mayoría de los estudiantes del curso, desarrollaron esa complejidad bajo el yugo expresivo de unidad formal. En este análisis es posible discernir que el uso de los recursos computacionales permite hacer exploraciones formales complejas en un periodo menor de tiempo, a diferencia de las herramientas análogas que requieren mayor esfuerzo y tiempo de ejecución.

Finalmente, esta línea de análisis evaluó cuál fue la aproximación de diseño de los proyectos con relación a los códigos de utilidad-función y belleza-forma, según Schumacher (2009, 2011, 2012). Estos aspectos de análisis funcionaron como referencia para constatar si las relaciones formales organizacionales evaluadas anteriormente, son determinantes en el enfoque de solución que los participantes de los grupos mostraron.

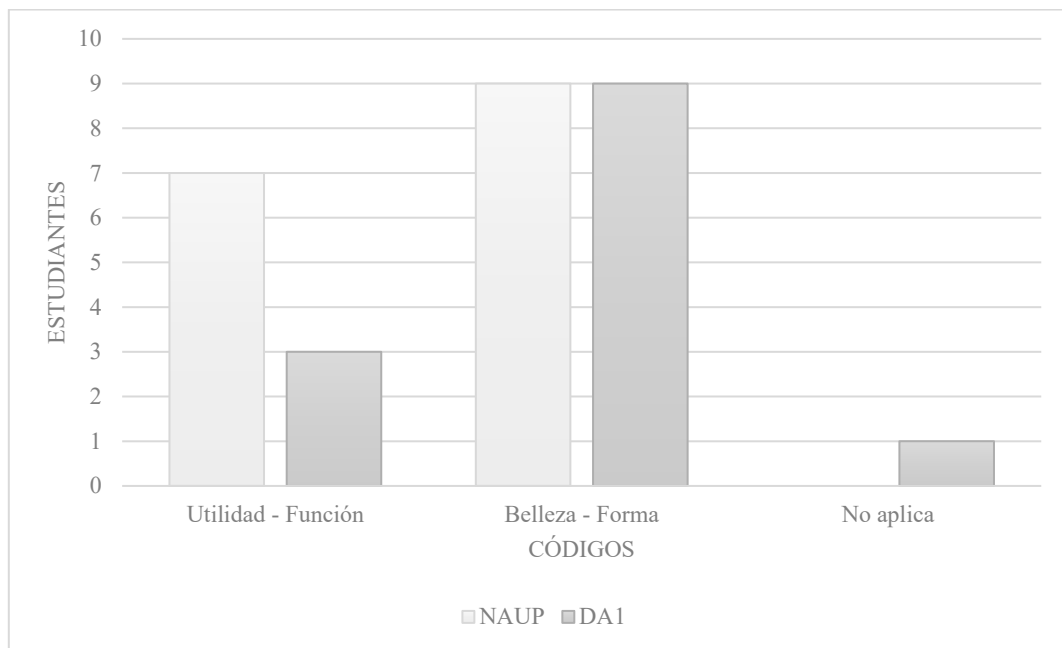


Figura 163. Relación de códigos de utilidad-función, belleza-forma manifestados en la realización del proyecto “Casa Estudio *Loft*” para la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

A pesar de que los integrantes del grupo NAUP mostraron mayor atención a las relaciones cognitivas semiológicas, así como a las relaciones físicas de organización irregular y compleja, con relación a los códigos de solución de proyecto el enfoque fue principalmente apegado a los aspectos de funcionalidad y utilidad del espacio en comparación con el taller de DA1. Sin embargo, tanto el taller de DA1 como el grupo NAUP, de manera independiente, tuvo una inquietud mayor por explorar los códigos de belleza y la forma.

4.5.2. Análisis y evaluación de los aspectos cuantitativos de la forma y el espacio.

La presente línea de análisis evalúa los aspectos cuantitativos de la forma y el espacio, que incluye los procesos de optimización y el desempeño de los elementos fisiológicos de cada proyecto arquitectónico. Los elementos fisiológicos para analizar son: el criterio estructural y las estrategias de adaptabilidad ambiental.

A partir del instrumento de evaluación, es posible verificar si la estructura tiene un carácter principalmente geométrico o topológico (ver Figura 164). También evalúa si se consideraron estrategias de adaptabilidad ambiental, así como implementos específicos para aplicar cada estrategia. Se consideró como estrategias la masa térmica, la protección solar y la ventilación natural, a través de implementar características especiales de materialidad, permeabilidad, disposición de dispositivos como celosías, pérgolas o elementos de vegetación (ver Figuras 165-166).



Figura 164. Relación de carácter estructural propuesto para el proyecto “Casa Estudio *Loft*” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

Los integrantes del curso NAUP, propusieron criterios estructurales con un carácter mayormente topológico, a diferencia del taller DA1 que dimensionó la propuesta de su sistema estructural desde un punto de vista geométrico. El criterio topológico explorado por NAUP, permitió desarrollar esquemas espaciales y características de la envolvente arquitectónica con mayor complejidad. El uso de estrategias algorítmicas digitales aportó a analizar el espacio y la forma desde una perspectiva topológica.

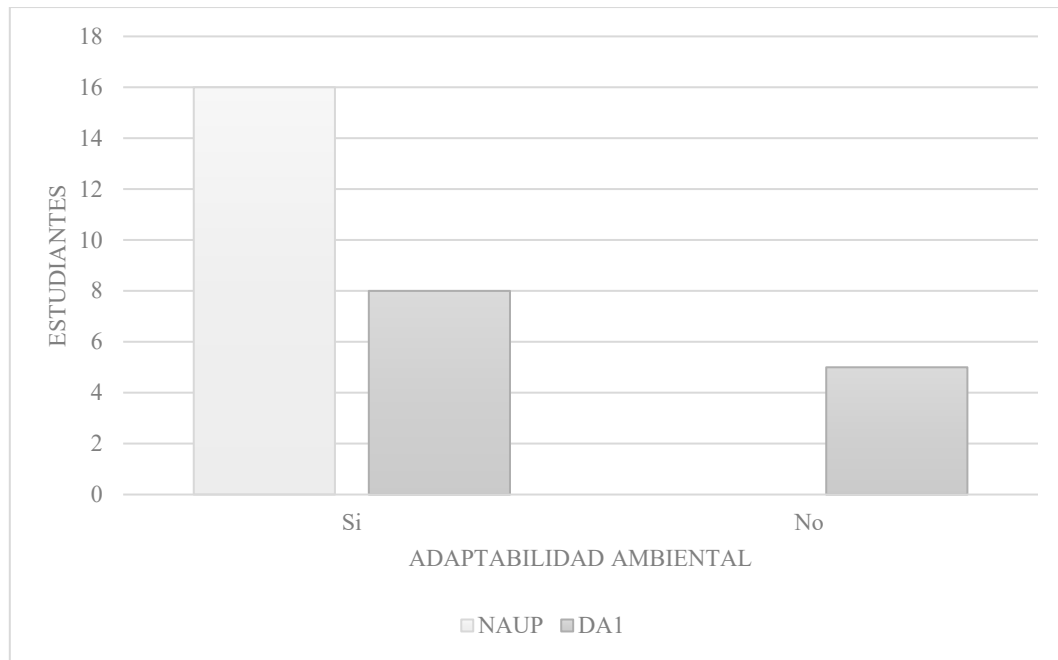


Figura 165. Adaptabilidad ambiental para el proyecto “Casa Estudio Loft” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

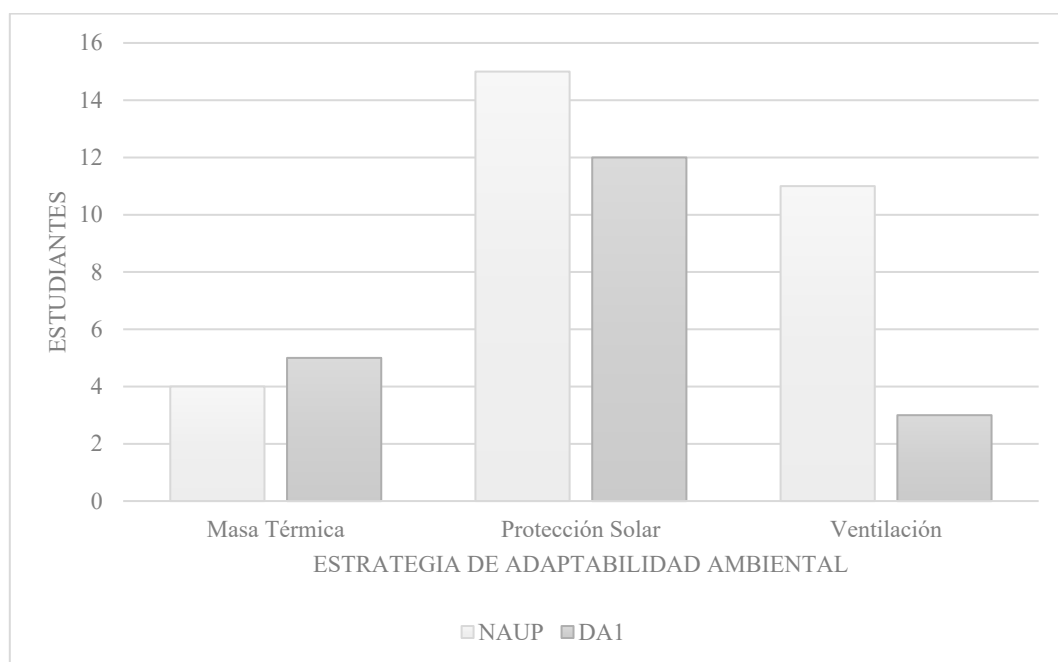


Figura 166. Relación de estrategias de adaptabilidad ambiental para el proyecto “Casa Estudio Loft” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

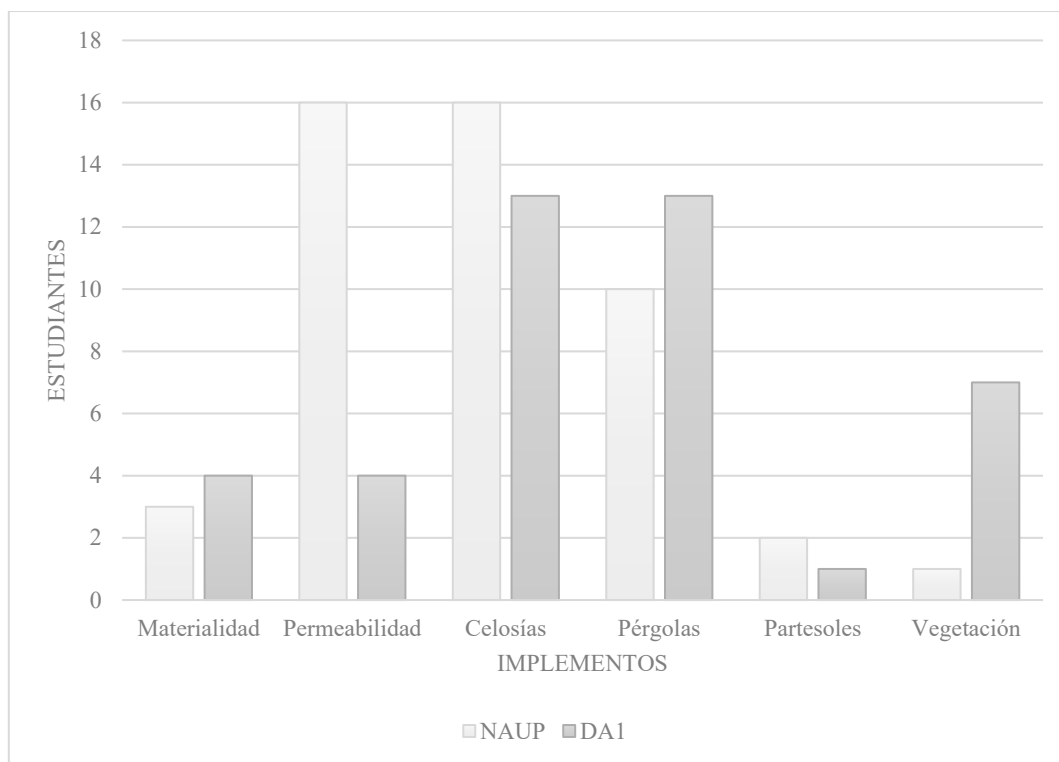


Figura 167. Relación de implementos para efectuar estrategias de adaptabilidad ambiental para el proyecto “Casa Estudio *Loft*” en la ciudad de Mexicali, Baja California, por los estudiantes del taller de DA1 y la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

Todos los estudiantes del curso NAUP consideraron estrategias de adaptabilidad ambiental a nivel conceptual y diseño básico. Algunos estudiantes del taller de DA1, no consideraron estrategias con profundidad y solución satisfactoria. La mayoría de las estrategias de adaptabilidad ambiental se enfocaron en la protección solar a través del diseño de celosía y pérgolas como elementos arquitectónicos para ofrecer permeabilidad en los proyectos.

A pesar de que las gráficas muestran una amplia mayoría de estudiantes del curso NAUP que implementaron estrategias de adaptación con el ambiente, a diferencia del taller DA1, no alcanzaron a desarrollar una distribución adecuada a las necesidades del usuario que la descripción del proyecto demanda.

Las estrategias algorítmicas y paramétricas utilizadas por los estudiantes de NAUP profundizaron en elementos puntuales, como celosías, pérgolas, cubiertas, toldos, barandales y mobiliario, pero no en el desarrollo integral de todas las variables cualitativas y cuantitativas que constituye un proyecto arquitectónico. Es necesario ampliar las horas de aprendizaje, la investigación y la práctica de las metodologías de diseño Algorítmico Paramétrico en los programas de licenciatura de la FAD, UABC.

4.5.3. Análisis descriptivo de las tendencias y estrategias algorítmicas aplicadas.

La segunda parte del instrumento de análisis y evaluación verifica las características y comportamiento interno de aplicación de la metodología de diseño arquitectónico AP del grupo que cursó la clase optativa Nociones de Arquitectura y Urbanismo Paramétrico (NAUP). Hace un registro del acercamiento de las propuestas a las principales tendencias de la Arquitectura Algorítmica Paramétrica (AAP) que se identificaron en el marco teórico de este trabajo de investigación. También, analiza las estrategias AP utilizadas en el proceso de diseño del proyecto arquitectónico asignado.

Las tendencias que se registran son: (a) escultórico, (b) parametricista, (c) algorítmico, (d) performativo y (e) discretismo. Las estrategias algorítmicas analizadas son: (1) geometría múltiple en superficies subdivididas, (2) elementos atractores, (3) retícula diagonal (*Diagrid*), (4) superficie generada a través de una función matemática, (5) superficie generada a través de puntos aleatorios, (6) superficie generada a través de una función matemática gráfica, (8) transformación topológica con *Weaverbird*, (9) simulación topológica estructural con *Kangaroo*, (10) agregación compleja con *Wasp*, (11) simulación térmico energético con *Ladybug* y (12) *Honeybee*, (13) Algoritmos Genéticos con *Galapagos*.

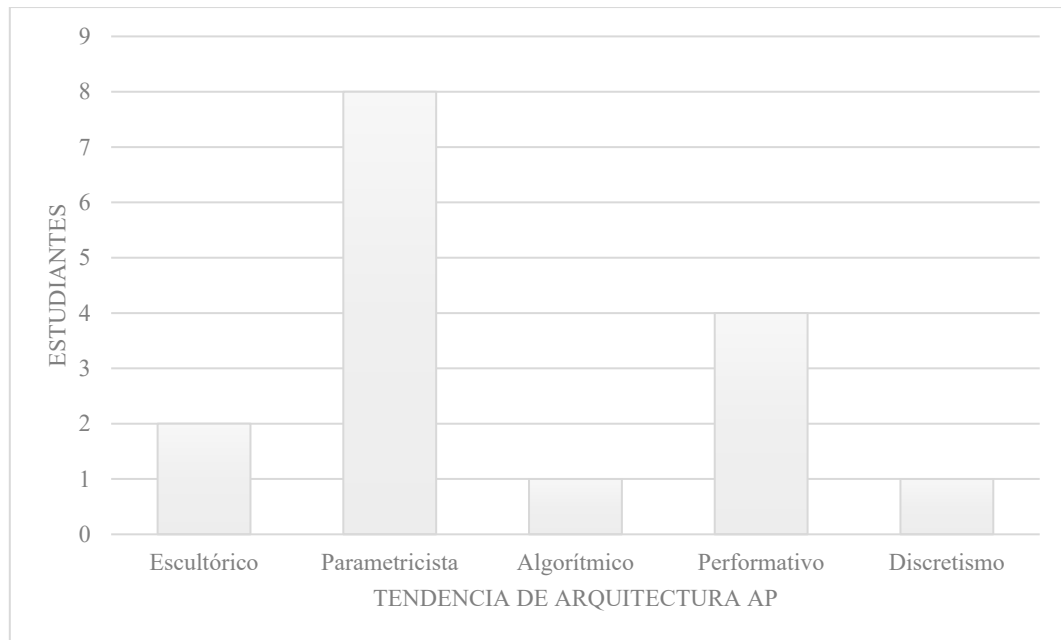


Figura 168. Aproximación a tendencias de la Arquitectura AP de los estudiantes que cursaron la clase optativa NAUP (2020-1) para el desarrollo del proyecto “Casa Estudio *Loft*” en la ciudad de Mexicali, Baja California. Elaboración propia (2021).

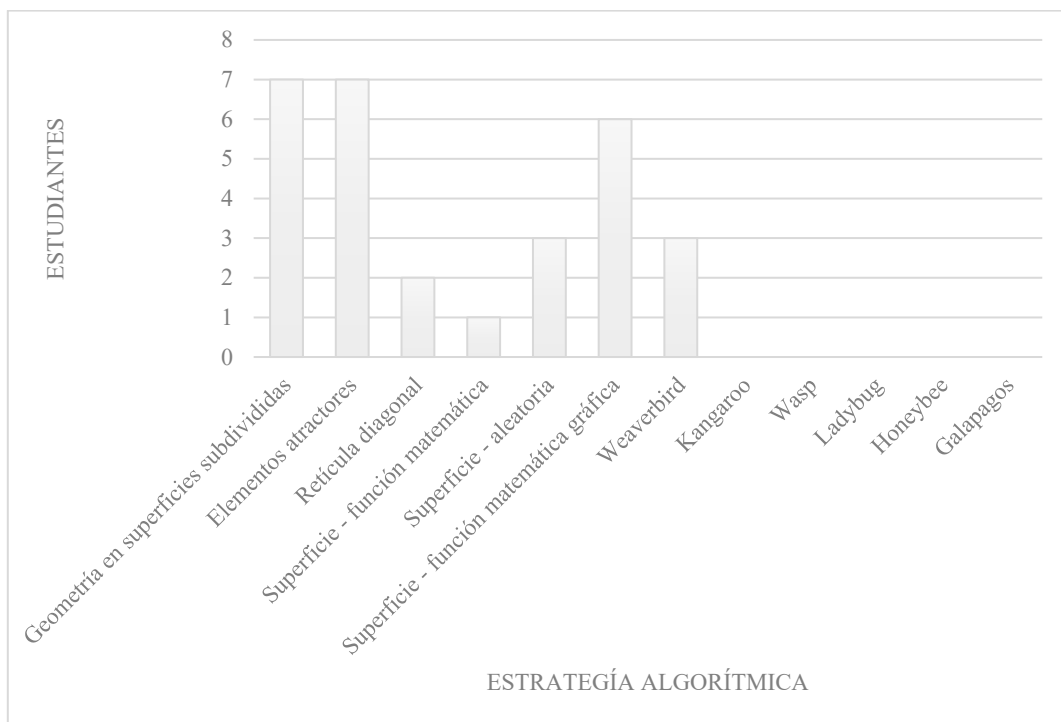


Figura 169. Relación de estrategias algorítmicas para efectuar el proyecto “Casa Estudio *Loft*” en la ciudad de Mexicali, Baja California, por los estudiantes la clase optativa NAUP, ciclo 2020-1. Elaboración propia (2021).

La mayoría de los proyectos resultantes manifestaron un acercamiento a la tendencia parametricista, con características de complejidad formal a través de geometría curvilínea y fluida, con un apego mayor a los resultados formales a través de estrategias algorítmicas paramétricas. Otra aproximación que hicieron los estudiantes fue a la tendencia performativa, donde el resultado formal es menos importante, y el objetivo principal es desarrollar estrategias algorítmicas que aporten a la optimización del proyecto arquitectónico.

V. DISCUSIÓN

El análisis del estado del arte de la Arquitectura Algorítmica Paramétrica (AAP) permitió identificar los principales conceptos, teorías y metodologías consolidadas que fueron referencia importante para la propuesta de una metodología de diseño que vincule los aspectos cualitativos y cuantitativos que conforman el espacio y la forma arquitectónica.

Desde el punto de vista cuantitativo, las estrategias AP se adaptan de mejor manera a las condiciones del desarrollo de un proyecto arquitectónico, ya que, al trabajar un algoritmo con parámetros y funciones computacionales, ofrece oportunidades de automatizar el trabajo y permitir analizar una gran cantidad de información compleja con menor tiempo y esfuerzo.

Sin embargo, la falta de investigación relacionada con la interpretación de variables desde la perspectiva cualitativa ha fomentado críticas y falta de confianza al uso de herramientas computacionales para los procesos y metodologías de diseño arquitectónico. El escepticismo se orienta principalmente al desapego humano con el quehacer arquitectónico y la falta de sensibilización con el espacio y las atmosferas (Zumthor, 2011, 2016) que se generan a partir de manipular variables cognitivas y existenciales relacionadas con la organización, la articulación y la materialidad espacial.

Las principales críticas en torno a la Arquitectura computacional tienen un origen más amplio y complejo del que pudiera discernirse a simple vista. El debate proviene de una tradición conflictiva entre la Filosofía materialista e idealista, del predominio de la razón o de

los sentidos en el desarrollo creativo, de la fricción existente entre los aspectos fenomenológicos y semiológicos en el desarrollo arquitectónico.

El escepticismo hacia el uso de la computadora en la Arquitectura, gira en torno a la preocupación por la pérdida de control sobre el diseño como dice Terzidis (2006), pero también por una inquietud relacionada con el desapego a la realidad, una patología que deriva de la alienación espacial producida por la realidad virtual, que promueve el excesivo uso del sentido de la vista y la lejanía del sistema sensorial háptico.

Tras un largo período, el escepticismo de Christopher Alexander era típico de muchos arquitectos. Muchos sentían que los ordenadores mecanizarían el proceso de diseño, dejando poco espacio para la intuición y el talento personal, elementos considerados como ingredientes básicos de una buena Arquitectura. (Krauel, 2011, p. 119).

En el orden teórico previamente descrito, con relación a los aspectos fenomenológicos de la arquitectura, Pallasmaa (2014) muestra una seria preocupación por el uso de la computadora, considera que promueve una manipulación visual pasiva y un viaje retiniano. De la misma manera Montaner (2015), advierte que la radicalización del uso de la computadora en el proceso de diseño, ha propiciado formas y espacios arbitrarios y poco relacionados con su contexto:

La radicalización de este cambio (de lo analógico a lo virtual) ha potenciado la eclosión de una “Arquitectura digital” ... en la que las geometrías complejas y sinuosas, surgidas del mundo virtual en la pantalla, plantean una pretendida liberación de las formas y espacios mediante una arquitectura de redes y corrientes, fluidas y transparentes, líquidas y dinámicas, a veces arbitrarias y generalmente poco relacionadas con su contexto (Montaner, 2015, p. 22).

Por otro lado, el arquitecto Peter Zumthor (2011, 2016) no manifiesta inquietudes enfocadas en el uso de la tecnología en sí (en este caso digital), sino que su preocupación

principalmente se apoya en la aplicación de metodologías basadas en teorías, filosofías y aspectos semiológicos, como son los conceptos y el excesivo uso de diagramas que pudieran fomentar un desapego con la realidad y su inherente capacidad de generar estímulos sensoriales a partir de su cualidad material y espacial.

Es importante analizar el contexto en el que se ha desarrollado la Arquitectura computacional. Primeramente, la computadora es usada intensivamente como una herramienta representacional, en definitiva, esta manera promueve una patología de desapego y alienación espacial.

Sin embargo, el avance tecnológico y la accesibilidad pedagógica actual con relación a la computadora, ha traído como consecuencia otro enfoque de uso, en el cual deja de ser comprendida como una herramienta pasiva de dibujo y proyección virtual, para convertirse en un aliado activo en la concepción de la forma y espacio, gracias al aprendizaje de la programación digital como un medio de comunicación amplia con una inteligencia artificial, cada vez más avanzada.

Actualmente la inteligencia artificial, alimentada por los sistemas de intercomunicación global gracias al Internet y el *Big Data* (Carpo, 2011, 2017, 2019), se encuentra en un proceso de transformación de los modelos, no solo informáticos, sino de los métodos de investigación científica, que pasan de tener un carácter principalmente causal a otro de tipo heurístico.

Es decir, los recursos digitales actualmente generan posibilidades espaciales y arquitectónicas en toda la extensión de la palabra. Sin embargo, este hecho no deja de lado al hombre en un papel pasivo de participación en el diseño, no son un sustituto sino un complemento, ya que es responsabilidad del diseñador tomar decisiones en términos de designar el valor inherente que le da a los datos y parámetros que entran en juego en la programación digital de un proyecto.

La racionalización de los datos por parte del diseñador juega un papel activo y relevante en el proceso de diseño con el manejo de sistemas computacionales.

Para utilizar el ordenador como un instrumento que resuelve problemas en el proceso de diseño, se debe responder a la pregunta de Christopher Alexander sobre el uso del ejército estúpido. La naturaleza es proclive a los ejércitos de estúpidos, las hormigas son un ejemplo ... se debe usar como una herramienta generativa que crea una lógica interna, la cual se puede desarrollar para producir una amplia gama de posibilidades (Krauel, 2011, p. 119).

Por otro lado, con relación al debate entre la razón y los sentidos, así como el consecuente desapego con la realidad, es oportuno considerar esta problemática desde una perspectiva Deleuziana o rizomática, en lugar de dicotómica. Puede tomarse como ejemplo el símbolo de la mitología nórdica de la serpiente que se come su propia cola: el Uróboro (ver Figura 170), como referencia de un sistema recursivo. Es decir, es más interesante considerar la razón como un aspecto recurrente de los sentidos o viceversa, que contrapuestos.



Figura 170. El Uróboros, la serpiente que come su propia cola (símbolo de la mitología nórdica), como referencia a la recursividad. Fuente: “Uróboros” (2020) [Imágen]. Recuperado de <https://es.wikipedia.org/w/index.php?title=Ur%C3%B3boros&oldid=126403838>. Dominio público (2020).

Finalmente, cualquier estímulo sensorial tiene que pasar por una evaluación mental, así como cualquier razonamiento o idea, no tiene significado si no es parte de un esquema cognoscitivo derivado de la interacción de los sistemas sensoriales y la envolvente corporal con los objetos físicos.

De la misma manera, en términos del uso computacional como inteligencia artificial; cualquier algoritmo generado, necesariamente tiene que pasar por el juicio, supervisión y creación del operador y del valor que éste da a cada parámetro y función asignada (ver Figura 171).

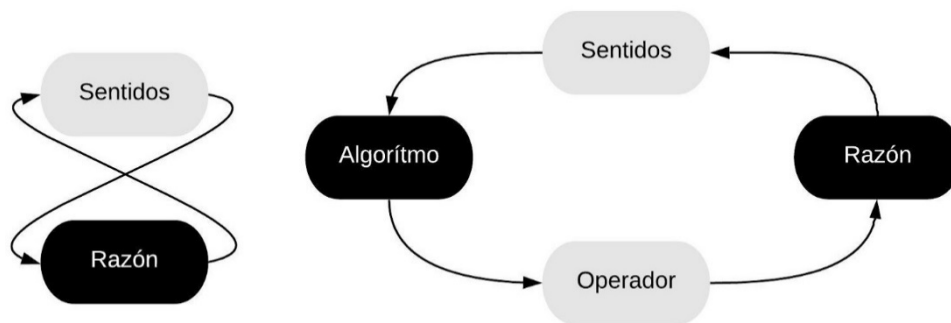


Figura 171. El Uróboros: recursividad entre los sentidos y la razón, entre el algoritmo y el operador. Elaboración propia (2020).

Los análisis y evaluaciones presentadas en este trabajo de investigación muestran que a pesar de que, en un corto periodo de tiempo, los arquitectos y estudiantes de arquitectura que se introducen a la programación digital comienzan a generar algoritmos que permiten abordar mayores niveles de complejidad de información, el espacio y la forma; no alcanzan a dar una respuesta con profundidad a las variables cualitativas y cuantitativas que un proyecto arquitectónico demanda.

Es necesario ampliar el tiempo efectivo dedicado al aprendizaje de la programación digital orientada al diseño de objetos en los programas de licenciatura y posgrado de las Facultades de Arquitectura de Latinoamérica. Con el fin de ampliar el desarrollo

arquitectónico, abrir nuevas rutas de investigación, y generar nuevos procesos de ejecución, producción y difusión de una Arquitectura más adaptada a la vanguardia tecnológica, a los problemas socioculturales, económicos, naturales y de supervivencia.

VI. CONCLUSIONES

El presente trabajo de investigación aborda la temática relacionada con la Arquitectura Algorítmica Paramétrica. El objetivo fue crear un marco teórico nutrido que evidencie los avances en teoría, metodología y producción de las manifestaciones arquitectónicas más avanzadas y con vanguardia en el uso de la tecnología.

Se identifica, que el objetivo de automatización buscado a través de la Inteligencia Artificial (IA) es optimizar los procesos de diseño, ejecución y producción de la Arquitectura, a fin de consumir el tiempo en procesos intelectuales y creativos, antes de invertirlos en trabajo mecánico. Estos procesos no buscan sustituir los procesos creativos y artesanales, sino que buscan ser un complemento al desarrollo de las actividades humanas.

También, fue posible concluir que las principales manifestaciones de arquitectura computacional fueron avanzando de acuerdo con el progreso tecnológico, y pasaron de tener un enfoque esculturalista y representacional a idear nuevos esquemas de producción e investigación.

Actualmente, el estudio y desarrollo de la Arquitectura digital -como parte de un campo disciplinar- se encuentra concentrado en las principales instituciones educativas de Norteamérica y Europa; eso puede obedecer a las condiciones económicas y culturales que privilegian a este contexto dentro del mundo.

Existe un latente rezago en el Sur global para acceder a la vanguardia tecnológica y arquitectónica. Es necesario romper esa realidad, a través del desarrollo y transmisión de conocimiento en los contextos donde exista rezago relacionado con el acceso y el desarrollo de productos arquitectónicos realizados a través de metodologías y medios computacionales.

Por otro lado, dentro del ámbito digital y virtual del desarrollo arquitectónico, se encuentra que los procedimientos algorítmicos y paramétricos son un medio importante para ampliar capacidades de respuesta a las problemáticas que la disciplina demanda.

Las teorías y metodologías más consolidadas, en el marco de esta temática, son las desarrolladas por Schumacher y Terzidis. Este trabajo presenta una descripción de esos planteamientos, a fin de encontrar rutas para realizar propuestas metodológicas que implementen aspectos cualitativos y cuantitativos de un proyecto arquitectónico en su fase de creación.

Primeramente, presenta un análisis de los aspectos cualitativos a través del reconocimiento de variables fenomenológicas y semiológicas que puedan ser efectivas en la implementación de la Arquitectura computacional. Este análisis verifica y evidencia los principales problemas y potencialidades relacionadas con el uso de los medios digitales en el proceso de diseño arquitectónico.

Aborda las principales críticas existentes en torno al cada vez más necesario uso de las computadoras como herramienta o medio de producción en la Arquitectura, las cuales manifiestan una preocupación por promover una patología de alienación de la realidad en el entendimiento del espacio desde la perspectiva fenomenológica, a costa de un dominio latente de un enfoque semiológico, idealista o virtual.

Analiza con referentes especializados que una respuesta a esta patología es a través del modo de comprender y usar la computadora. Primeramente, ejemplifica con la Arquitectura de Kengo Kuma (2010) y el enfoque de Arquitectura computacional llamado Discretismo, que existen manifestaciones arquitectónicas que son equilibradas en considerar los aspectos fenomenológicos y semiológicos de manera sinérgica y recursiva.

Posteriormente, muestra que la computadora puede ser más participativa en el desarrollo de la forma y el espacio a través de la programación digital, y los mecanismos

algorítmicos y paramétricos consecuentes de este proceso. Verifica con las aportaciones teóricas de Terzidis y Schumacher que existen metodologías relacionadas con la Arquitectura computacional que abordan los aspectos fenomenológicos y semiológicos como parámetros a analizar en la metodología de diseño.

Por otro lado, hace un recuento de las principales variables y herramientas algorítmicas que pueden ser importantes en el análisis de los aspectos cuantitativos. El objetivo es lograr procesos de optimización y desempeño en la fisiología de la edificación o el espacio a diseñar. Desde esa perspectiva el presente trabajo propone un esquema metodológico para implementar parámetros importantes para la evaluación del proyecto a través de las principales herramientas algorítmicas visuales existentes.

También, propone un esquema metodológico algorítmico y paramétrico que analiza e incorpora aspectos cualitativos con una participación sinérgica y activa en el diseño arquitectónico, a través de herramientas algorítmicas que pueden ser utilizadas en la plataforma de programación visual llamada *Grasshopper*.

El esquema metodológico propuesto, funciona como un algoritmo que tiene el objetivo de incentivar el análisis de elementos que interactúan como parámetros de evaluación en un proyecto, en torno a los aspectos fenomenológicos y semiológicos de la arquitectura.

El algoritmo incorpora las consideraciones teóricas existenciales, relacionadas con el análisis topológico del espacio de Norberg-Schulz (1971), así como la evaluación de los parámetros que estimulan sistemas sensoriales y la envolvente corporal, según Boomer y Moore (1983) y Peter Zumthor (2011, 2016).

Estas consideraciones son una referencia eficiente para conducir el análisis del espacio arquitectónico a un proceso de diseño Algorítmico-Paramétrico; y así dirigir al diseñador, a verificar herramientas algorítmicas útiles que, con ayuda de estrategias

computacionales e inteligencia artificial, considere las propiedades cualitativas, existenciales y psicológicas del espacio a diseñar.

Finalmente, el trabajo ejerce una aplicación de la propuesta metodológica en una asignatura de licenciatura de la Facultad de Arquitectura y Diseño (UABC) orientada al aprendizaje de procesos paramétricos y programación visual orientada al diseño de objetos, llamada Nociones de Arquitectura y Urbanismo Paramétrico. Posteriormente, se compara el resultado de la aplicación de la metodología AP, frente a una tradicional de diseño arquitectónico trabajada de forma analógica.

Se identifica que los proyectos realizados con la propuesta metodológica de diseño arquitectónico AP, presentan mayor nivel de complejidad y profusión en el desarrollo formal y espacial. Sin embargo, los resultados evidenciaron que los estudiantes que aplicaron esta propuesta metodológica lograron dar respuesta avanzada a elementos arquitectónicos puntuales, pero no a resolver con el mismo alcance todos los aspectos espaciales que un proyecto demanda.

En este sentido, con respuestas menos complejas, los estudiantes de la metodología tradicional alcanzaron a solucionar problemáticas elementales en el diseño del espacio, pero se vieron limitados en el análisis y la comprobación de sus propuestas, ya que no les fue posible hacer más prototipos que sirvan para hacer comparaciones.

Las estrategias algorítmicas resultantes del desarrollo de esta metodología de diseño arquitectónico AP, aportan al desarrollo de rutas de aprendizaje, investigación y desarrollo tecnológico, en la búsqueda de eficiencia, automatización y la evaluación. Es así como el ejercicio comparativo, expone que los procesos de aprendizaje relacionados con la programación digital requieren de más tiempo dedicado en los programas de licenciatura de Arquitectura, para aprovechar así, todo el alcance que pueden tener estos recursos.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



FACULTAD DE ARQUITECTURA Y DISEÑO
FACULTAD DE INGENIERÍA ARQUITECTURA Y DISEÑO
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y TECNOLOGÍA

**MAESTRÍA Y DOCTORADO
EN ARQUITECTURA, URBANISMO Y DISEÑO**

ALGORITMOS Y ARQUITECTURA: UNA APORTACIÓN AL DISEÑO

TESIS PARA OBTENER EL GRADO
DE MAESTRO EN ARQUITECTURA, URBANISMO Y DISEÑO

PRESENTA
AARÓN TADEO ONCHI RASCÓN

Correo electrónico UABC: onchia@uabc.edu.mx

Correo personal: at.onchi@gmail.com

Identificador ORCID: [0000-0003-2167-6758](https://orcid.org/0000-0003-2167-6758)

Director de tesis: Dr. Alejandro José Peimbert Duarte

Identificador ORCID: [0000-0003-4209-0203](https://orcid.org/0000-0003-4209-0203)

Comité tutorial:

M. Arq. Cuauhtémoc Robles Cairo
Identificador ORCID: [0000-0003-2452-4960](https://orcid.org/0000-0003-2452-4960)

Dra. Aurora García García de León
Identificador ORCID: [0000-0003-0583-4998](https://orcid.org/0000-0003-0583-4998)

REFERENCIAS

- Abalos, I. (2017). *La Buena vida: Visita guiada a las casas de la modernidad*. Gustavo Gili.
- Aksamija, A. (2016). *Integrating innovation in architecture: Design, methods and technology for progressive practice and research*. Wiley.
- Alexander, C. (2002). *Notes on the synthesis of form* (17. printing). Harvard Univ. Press.
- Anton, I., y Tănase, D. (2016). Informed Geometries. Parametric Modelling and Energy Analysis in Early Stages of Design. *Energy Procedia*, 85, 9-16.
<https://doi.org/10.1016/j.egypro.2015.12.269>
- Aranda, B., y Lasch, C. (2006). *Tooling*. Princeton Architectural Press.
- Aranda, B., y Lasch, C. (2008, 2013). - *Work—The Morning Line*. Aranda\Lasch.
<http://arandalasch.com/works/the-morning-line/>
- Barrantes Echavarría, R. (1999). *Investigación: Un camino al conocimiento un enfoque cualitativo y cuantitativo*.
- Bloomer, K. C., y Moore, C. W. (1983). *Cuerpo, memoria y arquitectura: Introducción al diseño arquitectónico* (M. T. Muñoz, Trad.). Hermann Blume.
- Burphy, M. (2016). Antoni Gaudí and Frei Otto: Essential Precursors to the Parametricism Manifesto. *Architectural Design*, 86(2), 30-35. <https://doi.org/10.1002/ad.2021>
- Caetano, I., Santos, L., y Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, S2095263520300029. <https://doi.org/10.1016/j.foar.2019.12.008>
- Cantrell, B., y Mekies, A. (2018). *Codify: Parametric and computational design in landscape architecture*.
<http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1769869>
- Carpó, M. (2011). *The alphabet and the algorithm*. MIT Press.

Carpó, M. (2017). *The second digital turn: Design beyond intelligence*. The MIT Press.

Carpó, M. (2019). Particled: Computational Discretism, or The Rise of the Digital Discrete.

Architectural Design, 89(2), 86-93. <https://doi.org/10.1002/ad.2416>

Deleuze, G., y Guattari, F. (2002). *Mil mesetas: Capitalismo y esquizofrenia* (5. ed). Pre-Textos.

Deleuze, G., y Guattari, F. (2017). *El anti-edipo: Capitalismo y esquizofrenia* (F. Monge, Trad.). Paidós.

Dunn, N. (2012). *Digital fabrication in architecture*. King.

Eisenman, P. (2011). *Diez edificios canónicos 1950-2000* (A. Lourie, Ed.; M. Puente, Trad.). Gustavo Gili.

Fornes, M. (2017). *Astana_TheVeryMany_©NAARO-42.jpg (1500×1137)*.

THEVERYMANY. <https://theverymany.com/world-expo-17-astana-kz-1>

Foster, N. (2012). *Queen Alia International Airport*. Home | Foster + Partners.

<https://www.fosterandpartners.com/projects/queen-alia-international-airport/>

Frampton, K. (2014). *Historia crítica de la arquitectura moderna* (J. Sainz, Trad.; Cuarta edición). Gustavo Gili.

Groat, L. N., y Wang, D. (2002). *Architectural research methods*. J. Wiley.

Hadid, Z. (2012). *Heydar Aliyev Centre – Zaha Hadid Architects*. Zaha Hadid Architects.

<https://www.zaha-hadid.com/architecture/heydar-aliyev-centre/>

Hensel, M., Sunguroglu, D., y Menges, A. (2008). Material Performance. *Architectural Design*, 78(2), 34-41. <https://doi.org/10.1002/ad.639>

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., Méndez Valencia, S., y Mendoza Torres, C. P. (2014). *Metodología de la investigación*. McGraw-Hill Education.

Hirschberg, U., Hovestadt, L., y Fritz, O. (Eds.). (2020). *Atlas of digital architecture:*

Terminology, concepts, methods, tools, examples, phenomena. Birkhauser.

Ito, T. (2002). *Serpentine Gallery Pavilion 2002.* Toyo Ito & Associates, Architects.

http://www.toyo-ito.co.jp/WWW/Project_Descript/2000-/2000-p_08/2000-p_08_en.html

Jabi, W. (2013). *Parametric design for architecture.* Laurence King.

Krauel, J. (2011). *Arquitectura digital: Innovación y diseño.* Links.

Kreider, J. F., Curtiss, P. S., y Rabl, A. (2002). *Heating and cooling of buildings: Design for efficiency.* McGraw-Hill.

Kuma, K. (2015a). *Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_11.jpg* (936×624). https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_11.jpg

Kuma, K. (2015b). *Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_31.jpg* (936×624). https://static.dezeen.com/uploads/2015/10/Yure_Kengo-Kuma_The-Galerie-Philippe-Gravier_dezeen_936_31.jpg

Kuma, K. (2019a). *Kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-02.jpg* (818×1027). <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-02.jpg>

Kuma, K. (2019b). *Kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-05.jpg* (818×1226). <https://static.designboom.com/wp-content/uploads/2020/11/kengo-kuma-CLT-park-harumi-tokyo-japan-designboom-05.jpg>

Kuma, K., y Watanabe, H. (2010). *Anti-object: The dissolution and disintegration of architecture* (Repr). Architectural Association Publ.

- Linder, M., Claypool, M., Jimenez Garcia, M., Retsin, G., Soler, V., y Institut für Internationale Architektur-Dokumentation. (2019). *Robotic building: Architecture in the age of automation*.
- Lynn, G. (1994). *GREG LYNN FORM – Cardiff Bay Opera House*. GREG LYNN FORM.
<http://glform.com/buildings/cardiff-bay-opera-house/>
- Lynn, G. (1998). *Folds, bodies & blobs: Collected essays*. La Lettre volée.
- Martinez Zarate, R. (1991). *Investigación aplicada al diseño arquitectónico: Un enfoque metodológico*. Trillas.
- Menges, A. (2008). Manufacturing Performance. *Architectural Design*, 78(2), 42-47.
<https://doi.org/10.1002/ad.640>
- Menges, A. (2011). *ICD/ITKE Research Pavilion 2011* | *achimmenges.net*.
Achimmenges.Net. <http://www.achimmenges.net/?p=5123>
- Menges, A. (2015). About the Guest Editor: About the Guest Editor. *Architectural Design*, 85(5), 6-7. <https://doi.org/10.1002/ad.1946>
- Menges, A. (2019). *BUGA Wood Pavilion 2019* | *achimmenges.net*. Achimmenges.Net.
<http://www.achimmenges.net/?p=20987>
- Menges, A., y Knippers, J. (2020). *Architecture research building: ICD/ITKE 2010/20* (1.^a ed.). Birkhauser.
- Montaner, J. M. (2015). *La condición contemporánea de la arquitectura*. GG.
- Morel, P. (2019). The Origins of Discretism: Thinking Unthinkable Architecture. *Architectural Design*, 89(2), 14-21. <https://doi.org/10.1002/ad.2407>
- Norberg-Schulz, C., y Bonanno, A. (1999). *Arquitectura occidental: La arquitectura como historia de formas significativas*. Gustavo Gili.
- Norberg-Schulz, C., y Schulz, C. N.-. (1971). *Existence, space & architecture*. Studio Vista.

Oxman, N. (2014). *Gemini | by Neri Oxman*. Neri Oxman.

<http://neri.media.mit.edu/projects/details/gemini>

Oxman, N. (2015). Templating Design for Biology and Biology for Design: Templating

Design for Biology and Biology for Design. *Architectural Design*, 85(5), 100-107.

<https://doi.org/10.1002/ad.1961>

Pallasmaa, J. (2014). *Los ojos de la piel: La arquitectura y los sentidos* (2. ed.). Gustavo Gili.

Payne, A., y Issa, R. (2009). *Grasshopper Primer: For version 0.6.0007 - Spanish Edition* (F.

Calvo y K. Caceres, Trads.; Segunda). Robert McNeel & Associates.

www.tectonicasdigitales.com

Reas, C., y McWilliams, C. (2010). *Form+code in design, art, and architecture* (1st ed).

Princeton Architectural Press.

Retsin, G. (2017). *Tallinn Architecture Biennale Pavilion—Www.retsin.org*. gilles retsin.

<https://www.retsin.org/Tallinn-Architecture-Biennale-Pavilion>

Retsin, G. (2018). *Nuremberg Concert Hall—Www.retsin.org*. gilles retsin.

<https://www.retsin.org/Nuremberg-Concert-Hall>

Retsin, G. (2019). Discrete Architecture in the Age of Automation. *Architectural Design*,

89(2), 6-13. <https://doi.org/10.1002/ad.2406>

Rodríguez, J. R., Onchi, A., y Obeso, F. (2017). *Casa RV*. spuarquitectura.

<https://www.spaumx.com/casa-crv>

Sanchez, J. (2019a). *Combinatorial Nest for Tallin Pavilion Program, Tallin Architecture*

Biennale 2019. Plethora Project. <https://www.plethora-project.com/combinatorial-nest>

Sanchez, J. (2019b). Architecture for the Commons: Participatory Systems in the Age of

Platforms. *Architectural Design*, 89(2), 22-29. <https://doi.org/10.1002/ad.2408>

Sanchez, J. (2020). *Architecture for the commons*. Routledge.

- Sanchez, J., y Andrasek, A. (2012). *Bloom*. Plethora Project. <https://www.plethora-project.com/bloom>
- Santamouris, M. (Ed.). (2007). *Advances in building energy research*. Earthscan.
- Schumacher, P. (2009). Parametricism: A New Global Style for Architecture and Urban Design. *Architectural Design*, 79(4), 14-23. <https://doi.org/10.1002/ad.912>
- Schumacher, P. (2011). *The autopoiesis of architecture. Vol. 1: A new framework for architecture*. Wiley.
- Schumacher, P. (2012). *The autopoiesis of architecture. Vol. 2: A new agenda for architecture*. Wiley.
- Scott, G. (1970). *La arquitectura del humanismo: Un estudio sobre la historia del gusto*. Barral. <https://books.google.com.mx/books?id=KxvPXwAACAAJ>
- Shiffman, D. (2015). *Learning processing: A beginner's guide to programming images, animation, and interaction* (Second edition). Elsevier/Morgan Kaufmann.
- Stadler, M. M. (2002). ¿Qué es la topología? *Sigma: revista de matemáticas = matematika aldizkaria*, 20, 63-77. <https://dialnet.unirioja.es/servlet/articulo?codigo=803774>
- Tedeschi, A. (2014). *AAD - Algorithms-Aided Design: Parametric strategies using Grasshopper* (First edition). Le Penseur publisher.
- Terzidis, K. (2006). *Algorithmic architecture* (1st ed). Architectural Press.
- Uróboros. (2020). En *Wikipedia, la enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=Ur%C3%B3boros&oldid=126403838>
- van Berkel, B. (2006). *Mercedes-Benz Museum*. UNStudio. <http://www.unstudio.com/en/page/12482/mercedes-benz-museum>
- Wujec, T. (2017). *The Future of Making: Understanding the forces shaping how and what we create*. Melcher Media. <https://www.autodesk.com/future-of-making-things/hardcopy-book>

Yuan, P. (2016). Parametric Regionalism. *Architectural Design*, 86(2), 92-99.

<https://doi.org/10.1002/ad.2029>

Zumthor, P. (2000). Pabellón Swiss Sound—Ficha, Fotos y Planos. *WikiArquitectura*.

<https://es.wikiarquitectura.com/edificio/pabellon-swiss-sound/>

Zumthor, P. (2011). *Atmosferas: Entornos arquitectónicos : las cosas a mi alrededor*.

Gustavo Gili.

Zumthor, P. (2016). *Pensar la arquitectura*. Gustavo Gili.

Schumacher, P. (2012). *The autopoiesis of architecture. Vol. 2: A new agenda for architecture*. Wiley.

Shiffman, D. (2015). *Learning processing: A beginner's guide to programming images, animation, and interaction* (Second edition). Elsevier/Morgan Kaufmann.

Tedeschi, A. (2014). *AAD - Algorithms-Aided Design: Parametric strategies using Grasshopper* (First edition). Le Penseur publisher.

Terzidis, K. (2006). *Algorithmic architecture* (1st ed). Architectural Press.

Wujec, T. (2017). *The Future of Making: Understanding the forces shaping how and what we create*. Melcher Media. <https://www.autodesk.com/future-of-making-things/hardcopy-book>