



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Universidad Politécnica de Valencia

---

# Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios

---

Trabajo de fin de Máster

Autor: Diego Florindo Rabadán

Tutor Fernando Cos Gayón

Cotutor Begoña Giner Fuentes

---



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CURSO 2020-2021  
Máster universitario en edificación  
ETS de Ingeniería de Edificación  
Universitat Politècnica de València



ESCUELA TÉCNICA SUPERIOR  
INGENIERÍA DE  
EDIFICACIÓN



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Diego Florindo Rabadán  
Trabajo de Fin de Máster



ESCUELA TÉCNICA SUPERIOR  
INGENIERÍA DE  
EDIFICACIÓN

Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios



## AGRADECIMIENTOS

En primer lugar, agradecer a mis tutores y en especial a Dña. Begoña Fuentes Giner, por apoyarme desde el inicio con la idea que quería desarrollar, por ayudarme a darle forma y sobre todo por ser un gran apoyo moral a lo largo del proceso.

Por otra parte, agradecer a D. Pedro Salinas Martínez, por atenderme cuando hacía falta, por su comprensión y por ayudarme en la parte administrativa relativa a la defensa de este trabajo. También agradecer a todos los profesores del *Máster en Edificación* de la UPV, por transmitir a los alumnos su pasión y su conocimiento sobre nuestra profesión.

Finalmente, me gustaría agradecer a mi familia, a mis amigos y a mis compañeros de trabajo que son las personas que me motivan a mejorar como persona y como profesional. Es gracias a ellos que he podido afrontar este trabajo y culminarlo con éxito.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Diego Florindo Rabadán  
Trabajo de Fin de Máster



ESCUELA TÉCNICA SUPERIOR  
INGENIERÍA DE  
EDIFICACIÓN

Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios



## RESUMEN

Este trabajo de investigación tiene como objetivo principal demostrar la optimización en tiempo y recursos que la implementación de herramientas paramétricas puede suponer en el actual flujo de trabajo del diseño y la ejecución de proyectos de edificación.

A nivel práctico, se han desarrollado dos códigos de programación visual mediante el software *Rhino* y el plugin *Grasshopper*, que permiten realizar tareas comunes del proceso de diseño y ejecución de edificios. El primer código permite sectorizar o panelizar un sistema de envolvente en la multitud de paneles que lo conforman. El segundo código permite elevar el nivel de detalle de una serie de componentes modelados a nivel LOD200 a un nivel LOD400.

Los resultados obtenidos mediante los dos códigos de *Grasshopper* demuestran que el modelado paramétrico permite considerar más opciones a la hora de proyectar y por ende hacer un mejor diseño. También permiten obtener modelados a nivel LOD400 de forma más rápida y de mayor calidad.

**PALABRAS CLAVE:** *Modelado 3d, Modelado Paramétrico, Diseño Paramétrico, Diseño Computacional, Programación Visual, Grasshopper, Rhino.*



## ABSTRACT

The main aim of this work is to demonstrate the optimization in time and resources that parametric design tools can offer to the actual workflow of building design.

At a practical level, two visual programming codes have been developed through the software *Rhino* and its plugin *Grasshopper* to allow us to achieve common tasks of design and building execution process. The first code allows us to sectorize and panelise a buildings envelope. The second code allows us to raise the *Level of Detail* (LOD) of a series of components from LOD200 to LOD400.

The geometry obtained through programming codes scripted in Grasshopper, demonstrate that parametric design allows us to consider more options when designing a building system and therefore chose a better solution. It also allows us to obtain higher quality LOD400 models in less time.

**KEY WORDS:** 3D Model, Parametric design, Computational design, Visual programming, Grasshopper, Rhino.



## RESUM

Aquest treball de recerca té com a objectiu principal demostrar l'optimització en temps i recursos en la implementació d'eines paramètriques que pot suposar en l'actual flux de treball del disseny i l'execució de projectes d'edificació.

A nivell pràctic, s'han desenvolupat dos codis de programació visual mitjançant el programa *Rhino* i el plugin *Grasshopper*, que permeten fer tasques comunes del procés de disseny i execució d'edificis. El primer codi permet sectoritzar o panelitzar un sistema envoltent en la multitud de panells que el conformen. El segon codi permet elevar el nivell de detall d'una sèrie de components modelats a nivell LOD200 al nivell LOD400.

Els resultats obtinguts mitjançant els dos codis de *Grasshopper* demostren que el modelatge paramètric permet considerar més opcions a l'hora de projectar i per tant fer un millor disseny. També permeten obtindre modelatges a nivell LOD400 de forma més ràpida i de major qualitat.

**PARAULES CLAU:** *Modelatge 3d, Modelatge Paramètric, Disseny Paramètric, Disseny Computacional, Programació Visual, Grasshopper, Rhino.*



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Diego Florindo Rabadán  
Trabajo de Fin de Máster



ESCUELA TÉCNICA SUPERIOR  
INGENIERÍA DE  
EDIFICACIÓN





## CONTENIDOS

1. INTRODUCCIÓN:.....	13
2. HIPÓTESIS DE PARTIDA .....	16
3. OBJETIVOS .....	16
4. METODOLOGIA.....	17
5. ESTADO DEL ARTE .....	18
6. CONCEPTOS GENERALES .....	23
7. APLICACIONES .....	29
7.1. EJEMPLO 1: PANELIZACIÓN FACHADAS .....	29
7.1.1. Introducción .....	29
7.1.2. Desarrollo del código.....	33
7.1.3. Resultados .....	37
7.1.4. Análisis de resultados .....	46
7.2. EJEMPLO 2: MODELADO EN SERIE .....	49
7.2.1. Introducción .....	49
7.2.2. Desarrollo del código.....	56
7.2.3. Resultados .....	63



7.2.4.	Análisis de resultados .....	65
8.	ALCANCE DEL CODIGO .....	67
8.1.	MODIFICACIONES DE SUPERFICIE BASE (CODIGO 1 y 2): .....	67
8.2.	MODIFICACIONES DE PANELIZACIONES (CODIGO 2): .....	72
9.	CONCLUSIONES .....	76
10.	ANEXOS .....	82
10.1.	CODIGO 1.....	82
10.2.	CODIGO 2.....	83
11.	REFERENCIAS .....	84



## INDICE DE ILUSTRACIONES

Ilustración 1: GRAND THEATRE DE RABAT – Zaha Hadid .....	19
Ilustración 2: CENTRAL BANK OF IRAK – Zaha Hadid .....	20
Ilustración 3: FLORENCE NEW STATION – Arata Isozaki .....	24
Ilustración 4: MAQUETA FUNICULAR DE LA CRIPTA DE LA COLONIA GÜELL - Antoni Gaudi .....	25
Ilustración 5: CENTRO HEYDAR ALIYEV - Zaha Hadid .....	30
Ilustración 6: 30 ST MARY AXE “The Gherkin” - Fosters and Partners .....	32
Ilustración 7: KAZAKHSTAN PAVILION; Expo 2017 Astana - Adrian Smith and Gordan Gill.....	32
Ilustración 8: VISTA GENERAL - CODIGO 1 .....	33
Ilustración 9: SUPERFICIE INICIAL – CODIGO 1 .....	34
Ilustración 10: GRUPO 3 – CODIGO 1 .....	35
Ilustración 11: GRUPO 4 y 5 – CODIGO 1.....	35
Ilustración 12: GRUPO 4 y 5 – CODIGO 1.....	35
Ilustración 13: GRUPO 6 y 7 – CODIGO 1.....	36
Ilustración 14: GRUPO 8 – CODIGO 1 .....	37
Ilustración 15: PANELIZACIÓN OBTENIDA CONSTRUCTOR Nº1 .....	41
Ilustración 16: PANELIZACIÓN OBTENIDA CONSTRUCTOR Nº1 .....	41
Ilustración 17: PANELIZACIÓN OBTENIDA CONSTRUCTOR Nº2.....	42
Ilustración 18: PANELIZACIÓN OBTENIDA CONSTRUCTOR Nº2.....	42
Ilustración 19: PANEL ORIENTADO HORIZONTALMENTE.....	43
Ilustración 20: PANEL ORIENTADO HORIZONTALMENTE.....	43



Ilustración 21: PANEL ORIENTADO VERTICALMENTE.....	44
Ilustración 22: PANEL ORIENTADO VERTICALMENTE.....	44
Ilustración 23: NIVELES DE DETALLE (LOD).....	50
Ilustración 24: RIYADH WESTERN METRO - Zaha Hadid.....	53
Ilustración 25: PANELIZACIÓN DE RIYADH WESTERN METRO.....	53
Ilustración 26: DETALLE DE PANELIZACIÓN DE RIYADH WESTERN METRO.....	54
Ilustración 27: VISTA GENERAL - CODIGO 2.....	56
Ilustración 28: BLOQUE 1 - CODIGO 2.....	58
Ilustración 29: BLOQUE 2 - CODIGO 2.....	58
Ilustración 30: BLOQUE 3 - CODIGO 2.....	58
Ilustración 31: BLOQUE 1.....	59
Ilustración 32: BLOQUE 2.....	59
Ilustración 33: BLOQUE 3.....	59
Ilustración 34: VISTA DETALLE – RESULTADO UNION SOLIDA DE BLOQUES 1, 2 Y 3.....	59
Ilustración 35: UNION SOLIDA DE BLOQUES 1, 2 Y 3.....	60
Ilustración 36: BLOQUE 4.....	61
Ilustración 37: RESULTADO BLOQUE 4.....	61
Ilustración 38: VISTA DETALLE BLOQUE 4.....	61
Ilustración 39: BLOQUE FINAL DE EXPORTACIÓN DE GEOMETRIA A RHINO.....	62
Ilustración 40: SUPERFICIE PANELIZADA NIVEL LOD100.....	63
Ilustración 41: RESULTADO SUPERFICIE PANELIZADA POR CODIGO LOD400.....	64
Ilustración 42: SUPERFICIE BASE.....	67



Ilustración 43: SUPERFICIE PANELIZADA – CODIGO 1.....	68
Ilustración 44: SUPERFICIE PENALIZADA DETALLADA - CODIGO 2.....	68
Ilustración 45: DETALLE DE SUPERFICIE PANELIZADA LOD400 – CODIGO 2.....	69
Ilustración 46: SUPERFICIE BASE .....	70
Ilustración 47: SUPERFICIE PANELIZADA – CODIGO 1.....	70
Ilustración 48: SUPERFICIE PENALIZADA DETALLADA - CODIGO 2.....	71
Ilustración 49: PANELIZACIÓN HEXAGONAL.....	72
Ilustración 50: PANELIZACIÓN HEXAGONAL DETALLADA - CODIGO 2.....	73
Ilustración 51: DETALLE DE CARPINTERIA.....	73
Ilustración 52: PANELIZACIÓN TRIANGULAR .....	74
Ilustración 53: PANELIZACIÓN TRIANGULAR DETALLADA - CODIGO 2.....	74



## 1. INTRODUCCIÓN:

La evolución del sector de la arquitectura y la construcción a partir de los años 80 ha venido determinada por las posibilidades ofrecidas por el uso de la computación, que ha permitido primero esbozar simples geometrías o realizar cálculos estructurales y hoy en día permiten realizar replicas virtuales de los edificios que se pretenden construir.

Los modelos 3D, son la herramienta más fiable a la hora de entender la documentación gráfica de un proyecto. Permiten a cualquier persona proyectarse dentro de un espacio virtual y tener una mejor idea del diseño en cuestión. También permiten a profesionales del sector identificar las interferencias de un edificio con mayor facilidad y detectar e identificar con antelación cualquier problema que pueda darse en obra. La implementación de herramientas como los modelos 3D en la arquitectura nace de la necesidad de reducir los plazos de ejecución de proyectos, y de hacer un uso más responsable de los materiales en la ejecución de proyectos del ámbito de la construcción.

Existen dos formas de modelar en 3D. El modelado normal se hace exactamente de la misma forma que dibujaríamos en 2D. Esencialmente, consiste en definir gráficamente un modelo en torno a 3 ejes utilizando puntos, líneas y superficies.

La segunda forma de modelar en 3D es más novedosa y está menos presente actualmente en el sector de la construcción. El modelado paramétrico permite modelar en tres dimensiones mediante programación, es decir, permite definir las reglas geométricas del sólido que se pretende modelar en vez de modelar el objeto manualmente. Para modelar una superficie de forma paramétrica el usuario



define la cantidad de puntos que le resulte necesaria para definir esa superficie. Cada punto utilizado no está definido gráficamente sino matemáticamente con unas coordenadas precisas. Las coordenadas se consideran en este caso los parámetros de este simple modelo.

Uno de los lenguajes más utilizados dentro de la programación visual es Grasshopper (GH). GH es un plugin de *Rhino* que permite programar códigos para modelar geometría dentro del espacio tridimensional de *Rhino*.

Las expresiones para generar geometría con GH son infinitas, y en ningún caso el uso de códigos limita las posibilidades de dibujar geometría en 3 dimensiones. Al contrario, esta nueva metodología, sugiere un abanico de nuevas posibilidades que permiten a arquitectos y diseñadores encontrar una nueva forma de diseñar edificios.

El diseño paramétrico consiste en diseñar en base a parámetros, lo que permite que el concepto global de un diseño se modifique en todo momento sin que el "detalle" se vea afectado; el conjunto puede cambiar, pero las partes que lo componen mantienen una tipología que se ha definido mediante programación.

Este trabajo pretende estudiar formas coherentes de implementar herramientas paramétricas como GH, que permitan optimizar los actuales flujos de trabajo en el diseño y la ejecución de edificios. Lo anterior se pretende llevar a cabo estudiando en un primer tiempo, la optimización en el campo de la sectorización y panelización de sistemas de envolvente (CODIGO 1), y en un segundo tiempo, la



optimización en tiempo en las tareas de modelado en serie mediante herramientas de parametrización (CODIGO 2).

Veremos en este estudio como la utilización de GH, por un lado, sirve para considerar más posibilidades a la hora de proyectar y tener una visión más completa de las opciones de las que disponemos para resolver un diseño y, por otro, como la utilización de herramientas paramétricas permite obtener maquetas a nivel LOD400 de forma más rápida y con mayor calidad.





## 2. HIPÓTESIS DE PARTIDA

El uso de software de diseño paramétrico permite optimizar tiempo en el desarrollo de proyectos, además de permitir contemplar más posibilidades y por tanto obtener soluciones más adecuadas en la realización de proyectos de edificación.

## 3. OBJETIVOS

Mediante este TFM se pretenden investigar las formas de implementar herramientas de diseño paramétrico en el proceso de diseño de sistemas de envolvente y evaluar el potencial de mejora que puede suponer el uso de dichas herramientas en la industria AEC (*Arquitectura, Ingeniería y Construcción*).

Dicho objetivo se pretende alcanzar demostrando: Primero, la optimización en tiempo y recursos en el diseño de sistemas de envolvente mediante el uso de herramientas paramétricas. Segundo, la optimización en tiempo de las tareas de modelado de componentes mediante el uso de herramientas paramétricas. Finalmente, el potencial de mejora en flujos de trabajo en el ámbito del diseño y la ejecución de edificios mediante el uso de herramientas de modelado paramétrico, en base a los resultados obtenidos.



#### 4. METODOLOGIA

La metodología para cumplir los objetivos se centra en el análisis de ejemplos de tareas de modelado 3D del ámbito arquitectónico e ingenieril, donde se puedan incorporar herramientas de software paramétrico. Para ello se tomarán tareas de modelado, y se compara el tiempo invertido para realizar la tarea manualmente y mediante software paramétrico. Adicionalmente, se tratará de identificar todas las ventajas y desventajas que ofrece esta manera de trabajar.

En primer lugar, se va a desarrollar un código que permita optimizar tareas de panelización de sistemas de envolvente, analizar los resultados obtenidos con metodología de modelado paramétrico, y comparar esta metodología con metodologías de modelado normal.

En segundo lugar, se pretende desarrollar un código que permita optimizar tareas de modelado LOD400 de multitud de componentes de misma tipología. Estas laboriosas tareas son muy comunes en el proceso de elevar el nivel de un modelo 3D de LOD100 a LOD400.

Mediante estos dos ejemplos concretos, centrados en el diseño de sistemas de envolventes, se evaluará si realmente estas tecnologías suponen una mejora para el diseño de edificios.



## 5. ESTADO DEL ARTE

En 2008 Patrik Schumacher, el principal arquitecto del despacho de arquitectura de Zaha Hadid explica que ha nacido un nuevo movimiento en arquitectura: el *Parametricismo*.

Schumacher explica que, como estilo, el parametricismo está marcado por sus objetivos, ambiciones, principios metodológicos, y criterio de evaluación, además de su característico repertorio formal - “*As a style, parametricism is marked by its aims, ambitions, methodological principles, and evaluative criteria, as well as by its characteristic formal repertoire*” (Schumacher, 2010)<sup>1</sup>. Tras “Parametricism manifiesto”, Schumacher presenta varios ensayos más, y sobre todo el libro “*The Autopoiesis of Architecture*”, donde introduce algunos principios del diseño paramétrico (*Dogmas*) que tienen que seguirse para hacer paramétrico un diseño, y otros principios que tienen que ser evitados (Tabús) (Al-Azzawi, 2021)<sup>2</sup>.

Dentro de los “*dogmas*” que define Schumacher, podemos recoger principalmente el uso de generación de geometría mediante *script* (código), de componentes generativos, de *NURBS* (sólidos a partir de nubes de puntos) y *Splines* y, sobre todo, que todos los elementos de un modelo sean paramétricos. También podemos recalcar que Schumacher utiliza los siguientes adjetivos para calificar las formas que van a componer este nuevo estilo: *Interarticulate*, *Hyperdize*, *Morph* and *Deform* (Schumacher, 2011)<sup>3</sup>.

El concepto global descrito por Patrick Schumacher consiste en definir los edificios de una forma paramétrica. En otras palabras, el edificio no se diseña en un orden lineal donde las decisiones que se



toman son las definitivas, sino que se diseña en base a una cantidad de parámetros que se consideraran variables, hasta el final del diseño.

Patrik Schumacher es el arquitecto líder del despacho de arquitectura de Zaha Hadid. Encontramos en la arquitectura de Hadid, un parecido a través de toda su extensa obra. El recurso a la doble curvatura que usa el despacho con gran maestría se repite a través de sus distintos diseños (*Ilustración 1 y 2*).



*Ilustración 1: GRAND THEATRE DE RABAT – Zaha Hadid – Fuente: Arqui9, (2022), Render [ONLINE]. Available at: <https://www.arqui9.com/>*



*Ilustración 2: CENTRAL BANK OF IRAK – Zaha Hadid - Fuente: Arqui9, (2022), Render [ONLINE]. Available at: <https://www.arqui9.com/>*

A pesar de no definir con reglas exactas este nuevo estilo, la contribución de Schumacher a este nuevo movimiento es trascendental, debido a que hasta entonces la investigación teórica sobre esta nueva forma de definir formas es muy escasa. Su investigación de cierto modo da pie a la discusión sobre diseño computacional, y sobre la necesidad de dejar en el pasado la “obsesión de lo formal” (Oktan, 2017)<sup>4</sup>.

A medida que las herramientas tecnológicas han ido ganando presencia en la industria de la construcción, las metodologías de trabajo han cambiado constantemente. La irrupción de los sistemas CAD (Computer-Aided Design), se puede resumir en tres fases según Yehuda Kalay. En un primer tiempo estas herramientas permitieron solventar problemas que hasta entonces se resolvían de forma intuitiva como por ejemplo el cálculo de estructuras complejas. En una segunda fase, alrededor del final de los años 70, permitieron avanzar en el campo de la generación de documentación gráfica de ámbito arquitectónico e ingenieril. Hasta entonces los planos se realizaban



a mano, y de hacer falta varias copias, estas se copiaban a mano. La producción de dibujos mejoró considerablemente mediante este tipo de herramientas. Una de las herramientas más populares universalmente hoy en día, AutoCAD, fue publicada por Autodesk en 1982. Estas herramientas han seguido evolucionando hasta llegar a los modelos 3D, que hoy en día se han convertido en una necesidad en grandes proyectos. La tercera fase que considera Kalay, más reciente, engloba los sistemas de generación BIM. Esta nueva metodología permite incluir mucha más información, no necesariamente geométrica, mediante bases de datos (Kalay, 2004)<sup>5</sup>.

Sin embargo, uno de los desarrollos más importantes que ha visto la industria de la construcción (AEC), ha sido el diseño en torno a algoritmos. El diseño paramétrico es un paradigma disruptivo, donde en vez de crear un modelo del diseño que pretendemos realizar, se crea un programa que genera el modelo que pretendemos realizar. Diseñando de esta forma, introducimos un campo de exploración de arquitectura completamente nuevo (Kolarevic, 2003)<sup>6</sup>.

Diseñar o modelar paramétricamente, exige a los diseñadores y técnicos abstraerse de la actividad de diseñar directamente para concentrarse en la lógica que une un diseño y las instrucciones textuales que describen dicho diseño (Woodbury, 2010)<sup>7</sup>.

La utilización de códigos en la práctica de la arquitectura tiene dos grandes ventajas según Mark Burry: La productividad y la exploración de diseños. Por un lado, permite la exploración de nuevos diseños experimentales. Por el otro lado, la utilización de herramientas paramétricas permite la automatización de tareas repetitivas (Burry, 2011)<sup>8</sup>.



Basados en las dos claras ventajas que destaca Mark Burry sobre la utilización de herramientas paramétricas, evaluaremos primero la capacidad de mejora de un diseño arquitectónico sencillo evaluando diferentes tipos de paneles sobre una fachada curva mediante Scripting, y después la capacidad de mejora en tareas repetitivas de modelado 3D.





## 6. CONCEPTOS GENERALES

El avance de las herramientas tecnológicas ha cambiado los métodos de trabajo en la industria de arquitectura y construcción. Tareas que tradicionalmente se hacían a mano en el sector de la construcción, como serían el cálculo de estructuras o ejercicios de composición, se realizan más eficientemente desde hace años con la ayuda de ordenadores.

Una de las tecnologías que ha irrumpido en la profesión de arquitectos e ingenieros son las herramientas de modelado 3D. La industria ha adaptado dichas herramientas y hoy en día, se pueden considerar estas prácticamente indispensables al ejercicio de nuestra profesión. Los softwares de modelado 3D son variados, pero por lo general tienen la gran ventaja de permitir entender la documentación relativa a un proyecto con mayor facilidad. Además de permitir una mejor visualización, el modelado 3D ha aportado un avance significativo a la hora de detectar posibles fallos que de otra forma únicamente se detectarían durante la ejecución de la obra.

Uno de los softwares más avanzados en la modelización 3D es *Rhino* que, a pesar de haber tenido poca presencia en la industria de la construcción, cada vez es más utilizado, en parte por el modelado libre mediante puntos de control que permite.

*Rhino* no es el primer programa de modelización 3D, pero sí uno de los primeros en haber integrado un lenguaje de parametrización tan completo como el de *Grasshopper*. Lógicamente, una herramienta tan bien realizada como esta ha hecho que muchos arquitectos e ingenieros sientan la necesidad de





integrarlo en su flujo de trabajo, a pesar de que la herramienta no esté optimizada para la industria de la construcción en sus inicios.

Uno de los principales usos que permite GH es la exploración del diseño de nuevas formas mediante algoritmos generativos. De allí se haya utilizado principalmente en industrias del diseño y de la joyería. Es extremadamente útil en este sentido, permitiendo escribir algoritmos que asimismo generan soluciones únicas, que no podrían darse sin el uso del lenguaje de programación.



*Ilustración 3: FLORENCE NEW STATION – Arata Isozaki – Fuente: Wikipedia, (2022) [ONLINE]. Available at: [https://es.wikipedia.org/wiki/Arata\\_Isozaki](https://es.wikipedia.org/wiki/Arata_Isozaki)*



En cierto modo, mediante programación se pueden obtener resultados sorprendentes dentro del campo del diseño, particularmente en la búsqueda de formas óptimas, diferentes y atractivas (*Ilustración 3*). La manera de proceder en la búsqueda de formas en arquitectura se puede asemejar al proceso que seguía la arquitectura de Antoni Gaudí con sus modelos colgantes. Gaudí llevó los teoremas de Robert Hooke sobre las catenarias al extremo, realizando maquetas colgantes de sus edificios para aproximar la curvatura óptima de los arcos de sus edificios (*Ilustración 4*).



*Ilustración 4: MAQUETA FUNICULAR DE LA CRIPTA DE LA COLONIA GÜELL - Antoni Gaudí– Fuente: Wikipedia, (2022) [ONLINE].  
Available at: [https://es.wikipedia.org/wiki/Maqueta\\_funicular\\_de\\_la\\_cripta\\_de\\_la\\_colonia\\_G%C3%BCell](https://es.wikipedia.org/wiki/Maqueta_funicular_de_la_cripta_de_la_colonia_G%C3%BCell)*



Otros arquitectos e ingenieros del siglo pasado utilizaron esta misma metodología. Cabe destacar como ejemplos, las estructuras en tensión de Frei Otto, o las cubiertas de hormigón de Heinz Isler.

El caso es que esta metodología resultó ser muy atractiva para las tres personalidades mencionadas. Dicha metodología permite obtener formas complejas que resultan de un diseño estructural extremadamente eficiente. La forma obtenida resulta en menos trabajo interno estructural y alcanza la estabilidad con la mínima cantidad de material.

La metodología desarrollada por estos tres arquitectos e ingenieros es un gran ejemplo de parametrización en arquitectura: En vez de definir un diseño y laboriosamente dibujar este diseño en 3D, se obtiene la forma más adecuada estructuralmente mediante parámetros. En este caso, el diseño no es fruto de la imaginación de un arquitecto o ingeniero, sino de un razonamiento lógico y matemático, que se asemeja al de un ordenador. Los parámetros o reglas de diseño, como podrían ser en este caso las cargas, los puntos de amarre o de soporte, se podrían definir mediante código o gráficamente, pero lo demás puede ser realizado por un ordenador.

Sin embargo, la arquitectura de hoy en día no busca ser un reflejo de las formas óptimas. De ser así, todos los edificios incorporarían arcos parabólicos, y todas las cubiertas adoptarían la forma de paraboloides hiperbólicos (Rocker, 2006)<sup>9</sup>. A pesar de presentar una clara eficiencia en términos



estructurales, estas formas tienen inconvenientes. Las complicadas formas que se obtienen por los métodos explicados tienen como consecuencia superficies difíciles de explotar.

Por otra parte, se exige hoy en día la modelización de edificios a un nivel muy detallado, integrando toda la información de un proyecto en un solo fichero que se comparte entre distintos actores de la cadena de valor. La generación de modelos BIM se ha convertido en una tarea indispensable en grandes proyectos. Las grandes cantidades de información son más fácilmente manejables a partir de un único modelo, al que todos los agentes intervinientes acceden de forma remota e independiente.

Los modelos se realizan por lo general por etapas: De LOD 100 a LOD 400. Los niveles de detalle requeridos en modelos LOD 400 corresponden al nivel de detalle que incorpore toda la información gráfica de cada componente del edificio. La definición de los contenidos mínimos de un elemento o modelo LOD400, según *The American Institute of Architects (AIA)*, es la siguiente:

**LOD 400:** The Model Element is graphically represented within the Model as a specific system, object, or assembly in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element. (AIA, 2013)<sup>10</sup>

Por lo tanto, la modelización de modelos LOD400 es tan compleja que resulta en un esfuerzo gigantesco por parte de técnicos, que no necesariamente tienen un conocimiento especializado en la modelización de maquetas 3D.



Además de ser una tarea laboriosa e ineficiente en tiempo, la modelización 3D, cuando es realizada por personas, puede resultar en errores de modelado. Por lo general, arrastrar este tipo de errores tiene consecuencias en la compatibilidad entre programas de estos modelos, lo cual hoy en día ha sido uno de los mayores frenos a que la industria de la construcción adapte este tipo de tecnologías.

Las herramientas de parametrización pueden servir para realizar una modelización 3D más eficiente en tiempo, y sobre todo de más alta calidad. Mediante parametrización, el paso de un diseño arquitectónico genérico (LOD200) a proyecto detallado por todas sus características geométricas (LOD400) no solo permite ahorrar tiempo, sino que permite considerar más opciones y por ende encontrar mejores soluciones.



## 7. APLICACIONES

### 7.1. EJEMPLO 1: PANELIZACIÓN FACHADAS

#### 7.1.1. Introducción

En el presente apartado veremos a través de un ejemplo, como las herramientas de parametrización pueden servir para optimizar el material destinado a una obra, valorando diferentes posibilidades de segmentación o "*panelización*" de un sistema de envolvente. Llamaremos panelización a la tarea de segmentar una superficie de un sistema en componentes que sean realísticamente utilizables en una obra.

Los edificios icónicos que se construyen hoy en día se caracterizan por tener geometrías complejas, difíciles de entender estructuralmente (*Ilustración 5*). Por muchos motivos, este tipo de proyectos son extremadamente complicados de llevar a cabo y por ello, la optimización de tiempo y de recursos en la ejecución de este tipo de proyectos es primordial.





*Ilustración 5: CENTRO HEYDAR ALIYEV - Zaha Hadid – Fuente: Wikipedia, (2022) [ONLINE]. Available at: [https://es.wikipedia.org/wiki/Zaha\\_Hadid](https://es.wikipedia.org/wiki/Zaha_Hadid)*

La prefabricación de elementos constructivos es una de las formas más adecuadas, si no la única forma de llevar a cabo este tipo de construcciones. La prefabricación permite fabricar las piezas que van a conformar un edificio de forma mucho más controlada que se realizaría en obra. Estas se realizan en fábricas, bajo condiciones idénticas, y donde cada parámetro de calidad es debidamente comprobado. En el caso del hormigón, la prefabricación permite, una vez se vaya a ejecutar el proyecto, tener las piezas listas para utilizar, con su resistencia óptima en el momento de la ejecución, lo cual acorta considerablemente el tiempo y, por lo tanto, los costes de ejecución. La prefabricación sin embargo requiere anteriormente un enorme trabajo de diseño de cada pieza. Para que este método resulte realmente provechoso, es necesario invertir mucho tiempo en la fase de diseño.



Los vidrios que se usan hoy en día para este tipo de fachadas consisten en su gran mayoría en paneles de vidrio doble con cámara de aire. Por muchos motivos, esta solución es la única que presenta resultados notables, en particular en razón de sus características térmicas. Los vidrios dobles de los que disponemos hoy en día tienen que ser planos, por las limitaciones que impone su fabricación. Uno de los grandes esfuerzos dentro del proceso de diseño de fachada de vidrio curva por lo tanto es encontrar la combinación de paneles planos que mejor se asemeje al diseño inicial y que tenga el mayor número posible de paneles idénticos (*Ilustración 6 y 7*).

Estudiando diferentes posibilidades de panelización, y eligiendo la más óptima, los costes de manufactura del constructor se verán considerablemente reducidos. Este proceso de estudiar diferentes posibilidades de panelización se simplifica drásticamente mediante el uso de herramientas paramétricas como veremos a continuación.





*Ilustración 6: 30 ST MARY AXE "The Gherkin" - Fosters and Partners – Fuente: Wikipedia, (2022) [ONLINE]. Available at: [https://es.wikipedia.org/wiki/30\\_St\\_Mary\\_Axe](https://es.wikipedia.org/wiki/30_St_Mary_Axe)*



*Ilustración 7: KAZAKHSTAN PAVILION; Expo 2017 Astana - Adrian Smith and Gordon Gill – Fuente: Archello, (2022) [ONLINE]. Available at: <https://archello.com/project/kazakhstan-pavilion-and-science-museum-nur-alem>*

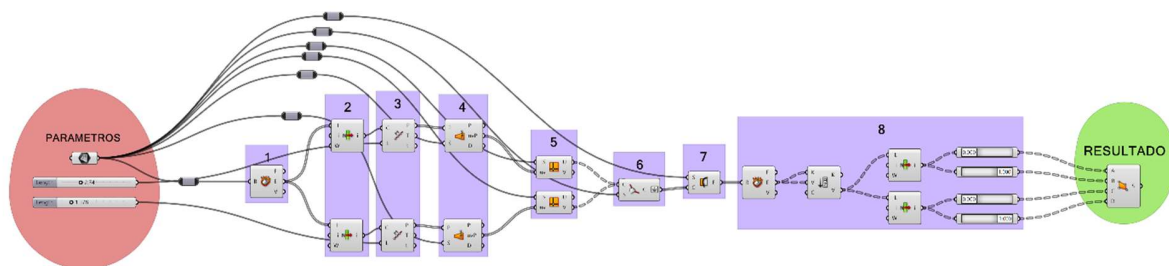
Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios

Mediante herramientas de programación como GH, se pueden llevar a cabo estudios sobre las mejores formas de panelizar una fachada. La mejor forma de panelizar una fachada siempre será la que alcanza el mayor porcentaje de paneles idénticos, a la vez que respeta de mejor forma el diseño inicial. A continuación, se presenta el código que se ha desarrollado para panelizar una fachada curva.

### 7.1.2. Desarrollo del código

El código desarrollado pretende permitir obtener la mejor solución para dividir una fachada curva en superficies planas más pequeñas. El código permite obtener resultados muy satisfactorios con fachadas con forma de paralelogramo con cualquier tipo de curvatura simple. De cierto modo, el código está preparado para ser ejecutado en superficies que tengan 4 aristas paralelas dos a dos. Como veremos, dependiendo de los casos, el código puede ser modificado en una dirección o en otra, dependiendo del resultado que se pretende.

Se presenta a continuación el funcionamiento del código base (*Ilustración 8*).

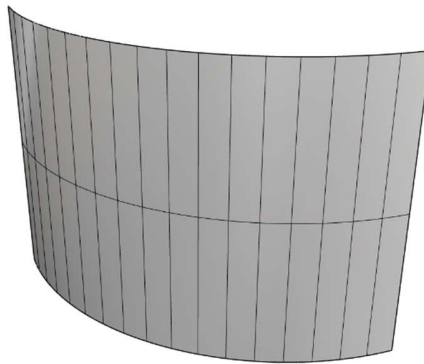


*Ilustración 8: VISTA GENERAL - CODIGO 1 – Fuente: Fuente propia*



Se describe a continuación las tareas que realizan cada grupo de comandos, con referencia a la numeración de la anterior imagen.

**PARAMETROS:** El usuario selecciona como datos de entrada la superficie que se pretende dividir en paneles o panelizar (*Ilustración 9*), y las dimensiones del panel principal que pretende utilizar. En nuestro código, los datos parametrizables, figuran en el círculo rojo.

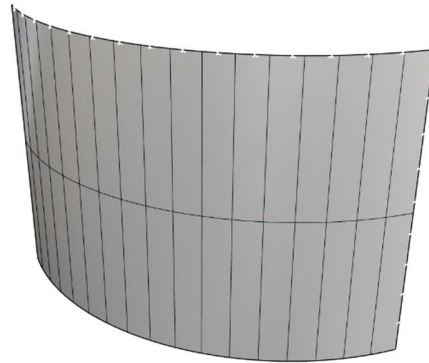


*Ilustración 9: SUPERFICIE INICIAL – CODIGO 1– Fuente: Fuente propia*

**GRUPO 1:** El código descompone la superficie para permitirnos utilizar las vertientes de la superficie por separado.

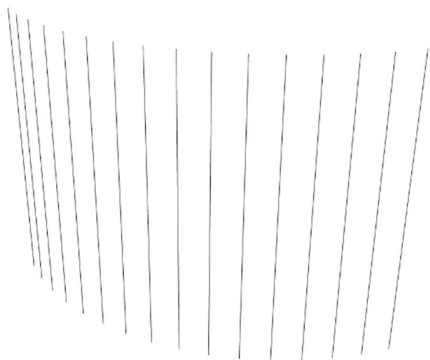
**GRUPO 2:** El código se divide en dos ramas que procederán a hacer las mismas tareas por un lado verticalmente y por otro horizontalmente. En el **GRUPO 2**, el código identifica los bordes correspondientes al eje vertical y al eje horizontal.

**GRUPO 3:** El código divide el borde longitudinal que haya elegido en segmentos de una longitud que será un parámetro definido por el usuario (*Ilustración 10*). Por ejemplo 100 cm.

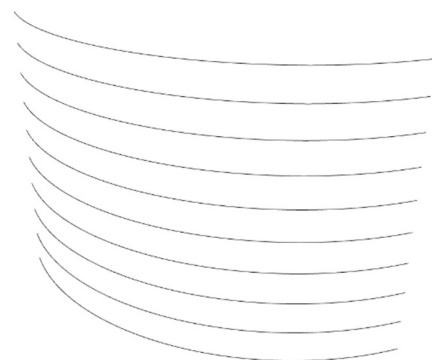


*Ilustración 10: GRUPO 3 – CODIGO 1 – Fuente: Fuente propia*

**GRUPO 4 y 5:** En cada punto de división de la curva, el código genera una iso-curva en dirección perpendicular al borde que haya dividido (*Ilustración 11 y 12*). Una iso-curva es una curva de construcción de la superficie.



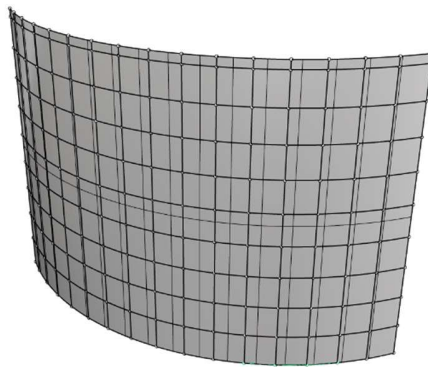
*Ilustración 11: GRUPO 4 y 5 – CODIGO 1 – Fuente: Fuente propia*



*Ilustración 12: GRUPO 4 y 5 – CODIGO 1 – Fuente: Fuente propia*

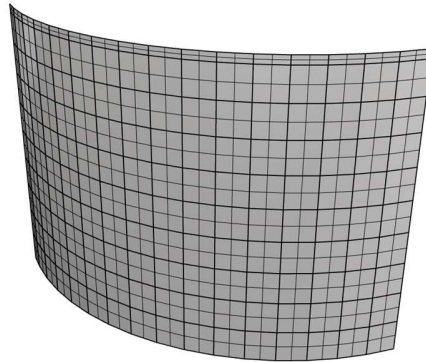
El código realiza los mismos pasos simultáneamente para el eje horizontal y el eje vertical.

**GRUPO 6 y 7:** Obtenemos un mallazo, vertical y horizontal con las distancias que hemos determinado como parámetro. El código divide la superficie inicial en varias superficies mediante el mallazo que ha generado (*Ilustración 13*).



*Ilustración 13: GRUPO 6 y 7 – CODIGO 1 – Fuente: Fuente propia*

**GRUPO 8:** El último paso consiste en planarizar todas las superficies del mallazo, que actualmente son superficies curvas. Lo anterior se soluciona añadiendo un paso más al código, que consiste en reconocer los cuatro vértices de cada superficie, y volver a generar una superficie nueva a partir de esos 4 vértices (*Ilustración 14*).



*Ilustración 14: GRUPO 8 – CODIGO 1– Fuente: Fuente propia*

### 7.1.3. Resultados

Para la superficie que se ha estudiado, probaremos 3 tipos de panelizaciones y se evaluará qué combinación es más económica en cuanto a precio material. Para ello se contactó con tres productores de paneles de vidrio para muros cortina. A cada uno se le preguntó cuáles eran las propiedades geométricas y físicas de su panel estándar. Adicionalmente se le preguntó cuál era el coste por panel estándar, el coste por m<sup>2</sup> de panel estándar, cuál sería el sobrecoste de no cubrir la fachada únicamente con paneles estándar y, finalmente, el precio de cubrir una fracción de la superficie con un panel específico. Como condición se dijo a todos los fabricantes que los vidrios tendrían que cumplir con la normativa vigente y ser de doble hoja de vidrio con cámara de aire.



- CONSTRUCTOR nº1<sup>1</sup>:

El primer constructor nos insistió que los paneles se construyen especialmente para cada cliente. Que no existe un panel estándar debido a que la producción de estos es adaptada a cada cliente. Sin embargo, puso como condición que cada panel debería tener una superficie menor a 2,5m<sup>2</sup>, y nunca ser más largo de 2m.

La carpintería del panel propuesto por este proveedor es de aluminio y de un grosor de aproximadamente 2-3mm. El tipo de vidrio propuesto es de doble con cámara de aire 5-8-5.

El precio por m<sup>2</sup> sería de 150\$USD. Lo cual correspondería a 129,36€/m<sup>2</sup>.<sup>2</sup>

- CONSTRUCTOR nº2:

El segundo constructor nos explicó que su panel estándar era de 1x1,2m, y que lógicamente sería más barato adaptar el diseño a este panel.

La carpintería del panel propuesto por este proveedor es de aluminio y de un grosor de aproximadamente 5mm. El tipo de vidrio es doble con cámara de aire 5-8-5.

---

<sup>1</sup> Por motivos de confidencialidad, en este trabajo no se comparte el nombre de ninguna de las compañías que fueron encuestadas sobre sus productos.

<sup>2</sup> En todos los ejemplos utilizados en este trabajo he utilizado un tipo de cambio de 1\$ USD = 0,862 €, según el tipo de cambio aproximado en septiembre de 2021.



El precio por  $m^2$  sería de 85\$USD. Lo cual correspondería a 73,30€/m<sup>2</sup>. En el caso de necesitar un segundo tipo de panel, estos costarían aproximadamente 175\$USD/m<sup>2</sup>.

- CONSTRUCTOR n°3:

El tercer constructor nos explicó que su panel estándar es de 1x2m. Sería más barato adaptar el diseño con este panel, pero únicamente se pagaría un poco más por los metros cuadrados de fachada que no se cubran con paneles estándar, siempre y cuando solo sean dos tipos de paneles para cubrir la fachada.

La carpintería del panel propuesto por este proveedor es de aluminio y de un grosor de aproximadamente 5mm. El tipo de vidrio es doble con cámara de aire 5-8-5.

El precio por  $m^2$  sería de 110\$USD. Lo cual correspondería a 94,86€. En el caso de los metros con un segundo tipo de panel costarían aproximadamente 125\$USD.





- RESUMEN:

Se resumen los datos de los tres constructores a continuación:

	<b>CONSTRUCTOR 1:</b>	<b>CONSTRUCTOR 2:</b>	<b>CONSTRUCTOR 3:</b>
Dim. del panel estándar	Max. 2,5m <sup>2</sup>	1m x 1,2m	2m x 1m
Precio del panel estándar	150\$USD/m <sup>2</sup>	85\$USD/m <sup>2</sup>	110\$USD/m <sup>2</sup>
Precio del segundo panel		175\$USD/m <sup>2</sup>	125\$USD/m <sup>2</sup>
Tipo de Carpintería	Aluminio (2-3mm)	Aluminio (5mm)	Aluminio (5mm)
Tipo de acristalamiento	5-8-5	5-8-5	5-8-5

A continuación, mediante el código que se procede a evaluar los precios de cada opción para diferentes tipos de fachada.

- SIMULACIÓN CONSTRUCTOR 1:

Para el primer constructor, se adapta el código con un cambio muy sencillo para que en vez de tener como parámetros el ancho y el alto de los paneles, divide la superficie de referencia a lo largo y a lo ancho, para obtener un panel estándar sobre toda la superficie. También se le pone un parámetro para que nos de la superficie de cada panel y que esta sea inferior a 2,5m<sup>2</sup>.



El código divide la superficie en 7 segmentos longitudinalmente, y en 9 transversalmente y nos da 63 paneles estándar de 1,74m de largo y 1,43m de alto. Cada panel tiene una superficie es de 2,49 m<sup>2</sup>, aprovechando al límite la superficie máxima impuesta por el fabricante de 2,5 m<sup>2</sup> (Ilustración 15 y 16).

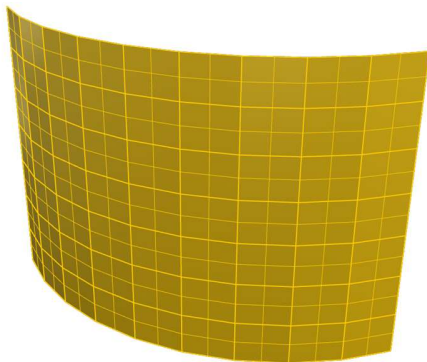


Ilustración 15: PANELIZACIÓN OBTENIDA CONSTRUCTOR N°1 -

Fuente: Fuente propia

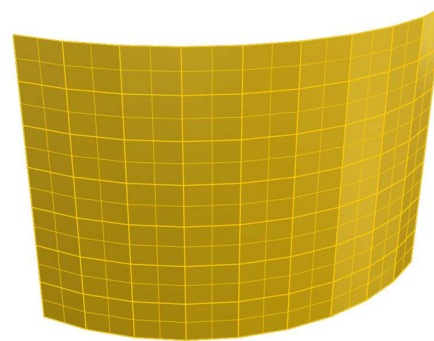


Ilustración 16: PANELIZACIÓN OBTENIDA CONSTRUCTOR N°1 -

Fuente: Fuente propia

Toda la fachada queda cubierta por paneles sin que se tenga que diseñar ningún otro panel que el descrito. Por lo que según la estimación del primer constructor de 150\$USD, el precio de la fachada acristalada sería de **23.562,00 \$USD**.

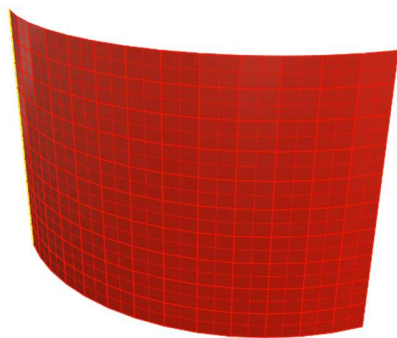
- SIMULACIÓN CONSTRUCTOR 2:

Para el segundo constructor, se vuelve al código inicial, realizando una pequeña modificación para que antes de cualquier otra tarea, nos indique "el resto" es decir la distancia que no va a poder panelizar con el panel estándar que se haya designado, horizontal y verticalmente. Esto sirve para que sepamos como orientar el panel vertical o horizontalmente.

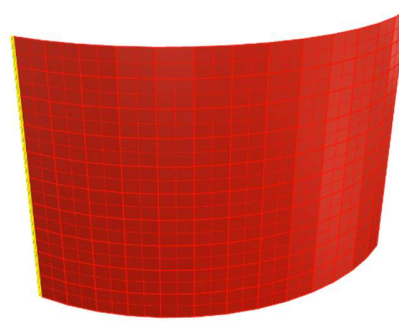


Vemos que en el caso de usar el panel verticalmente, es decir con el alto de 1,2 en la dirección transversal, obtenemos resto en ambas direcciones. Es decir, tanto longitudinalmente como transversalmente, no rellenamos la superficie y obtenemos 3 paneles distintos. Esto lógicamente se puede optimizar y probamos con la otra orientación. En el segundo caso, probamos el panel horizontalmente, y obtenemos resto en únicamente la dirección horizontal. Por lo tanto, ejecutamos el código de esa forma.

Realizando la panelización usando el panel horizontalmente, el código divide la superficie en 10 franjas longitudinales y en 13 transversalmente, con la decimocuarta franja siendo de diferentes dimensiones a las demás. Obtenemos 130 paneles estándares, y 10 paneles de dimensiones diferentes. Los 10 paneles diferentes representan menos del 1% (aproximadamente 0,68%) de la superficie inicial. Cada uno tiene unas dimensiones de 1x0,107m. Se representa a continuación los paneles estándar en rojo, y en amarillo la franja de paneles distintos (*Ilustración 17 y 18*).



*Ilustración 17: PANELIZACIÓN OBTENIDA CONSTRUCTOR N°2–  
Fuente: Fuente propia*



*Ilustración 18: PANELIZACIÓN OBTENIDA CONSTRUCTOR N°2–  
Fuente: Fuente propia*



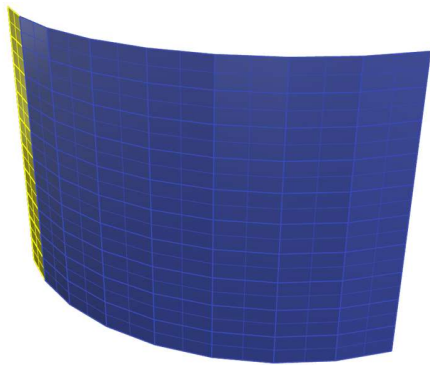
En total tenemos 156,00 m<sup>2</sup> de fachada panelizada con panel estándar de 1x1,2m, y 0,68 m<sup>2</sup> de fachada panelizada con un segundo tipo de panel. Sabiendo que el constructor nos cobra 85\$/m<sup>2</sup> por panel estándar, y 175\$/m<sup>2</sup> por segundo tipo de panel, obtenemos que el precio total es de:

**13.396,5\$USD**

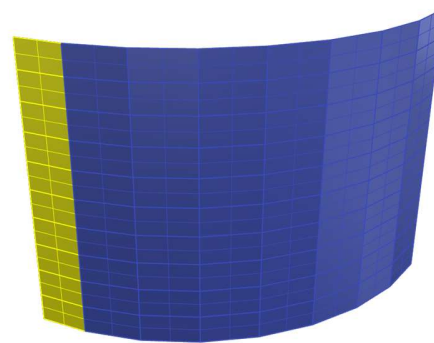
- SIMULACIÓN CONSTRUCTOR 3:

En el caso de la tercera simulación utilizamos exactamente el mismo código que en el segundo caso. En un primer tiempo probamos el resto que nos da el código en función de la orientación que le damos al panel.

Con el panel orientado horizontalmente, obtenemos un resto en la dirección longitudinal de 1,71m, y ningún resto en la dirección transversal (*Ilustración 19 y 20*).



*Ilustración 19: PANEL ORIENTADO HORIZONTALMENTE–  
Fuente: Fuente propia*

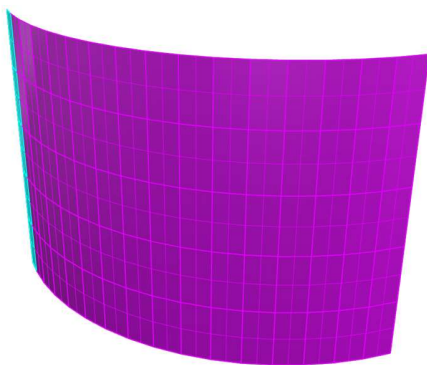


*Ilustración 20: PANEL ORIENTADO HORIZONTALMENTE–  
Fuente: Fuente propia*

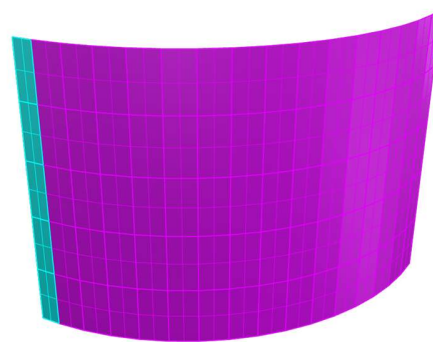


Con el panel orientado verticalmente, obtenemos un resto en la dirección longitudinal de 0,71m, y ningún resto en la dirección transversal.

Ejecutamos el código por lo tanto con la solución de panel orientado verticalmente. El código divide la superficie en 5 franjas longitudinales, y 16 franjas horizontales. Los 5 paneles diferentes representan 4,5% de la superficie inicial. Cada uno de los paneles tiene unas dimensiones de 2x0,71m (*Ilustración 21 y 22*).



*Ilustración 21: PANEL ORIENTADO VERTICALMENTE – Fuente: Fuente propia*



*Ilustración 22: PANEL ORIENTADO VERTICALMENTE – Fuente: Fuente propia*

En total tenemos 150,00 m<sup>2</sup> de fachada panelizada con panel estándar de 1x2m, y 7,08m<sup>2</sup> de fachada panelizada con un segundo tipo de panel. Sabiendo que el constructor nos cobra 110\$/m<sup>2</sup> por panel estándar, y 125\$/m<sup>2</sup> por segundo tipo de panel, obtenemos que el precio total es de: **17.385,00\$USD**



- TABLA COMPARATIVA:

	SIMULACIÓN 1		SIMULACIÓN 2		SIMULACIÓN 3	
	PANEL 1	PANEL 2	PANEL 1	PANEL 2	PANEL 1	PANEL 2
Dim (m)	1,743x1,429		1x1,2	1x0,108	2x1	2x0,708
S <sub>PANEL</sub> (m <sup>2</sup> )	2,49 m <sup>2</sup>		1,2 m <sup>2</sup>	0,108 m <sup>2</sup>	2 m <sup>2</sup>	1,461 m <sup>2</sup>
Unidades	63 piezas		130 piezas	10 piezas	75 piezas	5 piezas
S <sub>CUBIERTA</sub> (m <sup>2</sup> )	156,92 m <sup>2</sup>		156 m <sup>2</sup>	1,08 m <sup>2</sup>	150 m <sup>2</sup>	3,54 m <sup>2</sup>
S <sub>CUBIERTA</sub> (%)	100%		99,32%	0,68%	95,50%	4,50%
Precio/m <sup>2</sup>	150,00 \$/m <sup>2</sup>		85,00 \$/m <sup>2</sup>	175,00 \$/m <sup>2</sup>	110,00 \$/m <sup>2</sup>	125,00 \$/m <sup>2</sup>
Precio/u	373,61 \$/p		102,00 \$/p	18,90 \$/p	220,00 \$/p	177,00 \$/p
Precio (\$)	23.562,00 \$		13.260,00 \$	189,00 \$	16.500,00 \$	885,00 \$
TOTAL (\$)	23.562,00 \$		13.445,00 \$		17.385,00 \$	
TOTAL (€)	20.405,99 €		11.644,11 €		15.056,37 €	



#### 7.1.4. Análisis de resultados

Podemos decir que tenemos un fabricante que se destaca bastante de los dos otros. Como era de esperar, el más caro ha resultado ser el fabricante de vidrio doble que se comprometió a adaptar un tipo de panel para nuestra fachada. Eligiendo las dimensiones de un panel, podemos cubrir la totalidad de la fachada. A pesar de ser la solución más cara, esta tiene una clara ventaja sobre las demás y es que, a nivel estético, todos los paneles son idénticos.

La segunda y la tercera simulación dan resultados estéticamente similares debido a que en ambos casos únicamente tenemos un tipo de panel diferente. Dicho de otro modo, con los paneles propuestos rellenaríamos la fachada con dos tipos de panel, y en ambos casos únicamente quedaría una hilada vertical con un segundo tipo de panel.

En el segundo caso, el resultado obtenido es un caso particular, debido a que si panelizamos la fachada únicamente con el panel estándar, la superficie que no queda panelizada es muy escasa. En concreto, quedarían 10,7 cm lineales en el eje horizontal de la superficie a cubrir con paneles de vidrio. Debido a que dividimos la fachada en 13 franjas verticales, podríamos dividir dicho resto entre cada franja vertical, resultando en un hueco de menos de un centímetro entre cada franja, y rellenar el espacio con algún material elástico. De esta forma, el resultado quedaría más agradable visualmente, siendo todos los paneles igual, y abarataríamos aún más la solución.



En el tercer caso, el resultado es similar, pero la franja vertical del segundo tipo de paneles es notablemente más ancha y resulta en una opción menos estética que la primera pero significativamente más barata.

Podemos concluir que el segundo panel para este caso sería la mejor opción, pero requeriría algún tipo de modificación en el diseño. Las modificaciones podrían consistir en repartir el espacio sobrante entre las hiladas de paneles, y de esta forma prescindir del segundo tipo de panel, o modificar el diseño arquitectónico cambiando la superficie panelizada o bien acortándola y remplazando la hilada sobrante con otra solución, como podría ser una unión metálica iluminada.

Lo interesante de esta metodología es que, habiendo escrito el código para realizar las simulaciones que consideremos oportunas, podríamos seguir buscando tipos de paneles hasta encontrar un panel que nos permita cubrir toda la superficie que deseemos panelizar. Una vez el código queda redactado, llevar a cabo una simulación con un nuevo panel, es una tarea que prácticamente no requiere ni esfuerzo, ni tiempo, por lo que se podrían llevar a cabo cientos de simulaciones antes de decantarse por un diseño.

Además de lo anterior, en el caso de decidir modificar el diseño de un edificio y, por ende, la superficie que decidimos panelizar con vidrio, el código se adapta a la superficie y recalcula la disposición de los paneles. El código, como se ha especificado anteriormente, no requiere modificación mientras se respeten las reglas básicas que se han utilizado para programarlo, es decir





que se trate de una superficie con naturaleza de paralelogramo con cualquier tipo de curvatura simple.

Más adelante se proporciona información adicional sobre las tareas que puede solventar el código desarrollado, en la sección de *Alcances del Código*.



## 7.2. EJEMPLO 2: MODELADO EN SERIE

### 7.2.1. Introducción

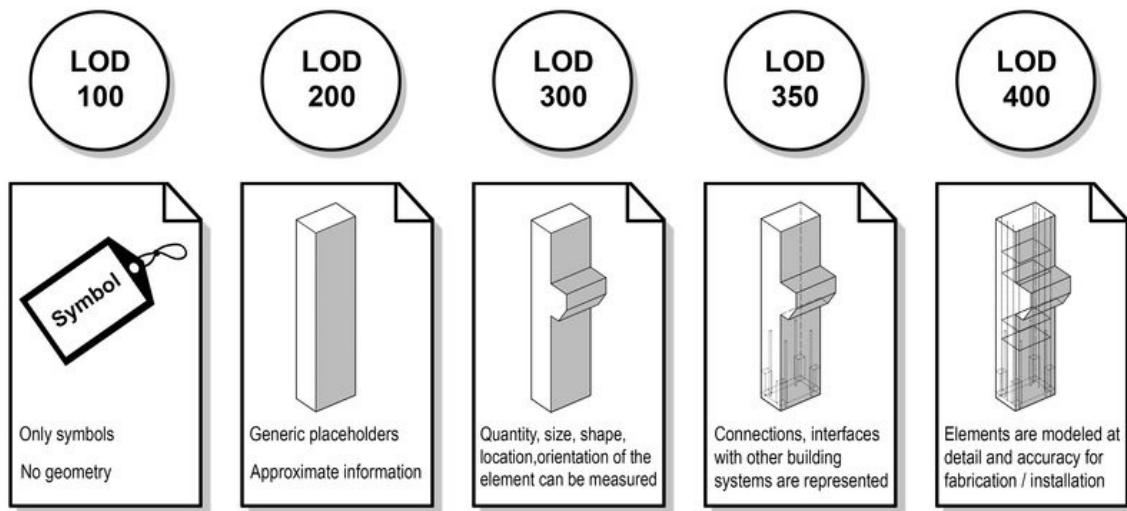
Habiendo visto, como se pueden realizar diferentes simulaciones de panelización para una fachada, veremos en un segundo tiempo otra aplicación de las herramientas de parametrización. La segunda aplicación que se pretende desarrollar consiste en la optimización en el tiempo de modelado 3D.

Las tareas de modelado son variadas, pero se podrían categorizar en dos grupos. Las que no pueden ser ejecutadas con código paramétrico y requieren que un técnico las ejecute, y las que pueden ser ejecutadas por código paramétrico. Algunos argumentarán que cualquier tarea de modelaje se puede ejecutar con código, y es verdad. Sin embargo, hay que saber combinar las dos formas para realmente poder optimizar el tiempo de modelado. Las tareas donde se repite un mismo patrón de comandos numerosas veces son las más susceptibles a permitir la implementación de códigos para ahorrar en tiempo.

El modelado 3D suele ser ejecutado por técnicos, que, en normas generales, no han tenido una formación particularmente exhaustiva de las metodologías de modelado. Por ello es habitual encontrar un modelado con errores y deficiencias.

La metodología de modelado mediante código es constante, y siempre ejecutará la tarea de la misma forma, por lo que si ejecuta una tarea correctamente siempre se ejecutará esa misma tarea con el mismo resultado. De cierto modo, se elimina por completo el factor de error humano.

El paso de llevar el nivel de detalle (*LOD*) de una maqueta LOD100 o LOD200 a un nivel LOD400 es un proceso ideal para incorporar metodologías paramétricas (*Ilustración 23*).



*Ilustración 23: NIVELES DE DETALLE (LOD) – Fuente: IMASGAL (2022) [ONLINE]. Available at: <https://imasgal.com/nivel-desarrollo-bim-lod/>*

Es ideal porque consiste en tareas repetitivas, que no requieren mucho conocimiento de un técnico para ser ejecutadas.

Como hemos comentado antes, la prefabricación de los materiales ha resultado ser una de las grandes innovaciones que han permitido realizar edificios tan ambiciosos como los ejemplos que hemos visto anteriormente. No obstante, la prefabricación de materiales únicamente es posible cuando ha habido un buen trabajo de diseño y modelado previo.



En el caso de ya haber realizado el proceso de panelización que se ha desarrollado en el primer ejemplo, y haber decidido sobre una forma de seccionar la fachada, el siguiente paso consiste en modelar cada panel, que actualmente consiste en una simple superficie, con su geometría "real". Cuando hablamos de geometría real, nos referimos a modelar los dos vidrios con sus propiedades geométricas y la carpintería que conforma cada panel con sus propiedades geométricas.

Imaginemos dos escenarios. Una vez obtenida la panelización, de la forma que se ha explicado en el primer ejemplo, existen dos formas de obtener un modelo LOD400. La primera es manualmente. Manualmente, la tarea consistiría en ir uno a uno por cada panel de la cubierta y de la fachada, modelando la superficie de referencia a su verdadera forma de fabricación. Podemos considerar que alguien que se le dé bien el modelado 3D y que sea bastante rápido, tardaría más o menos 5 minutos por panel siendo muy optimistas. Estimando el número de paneles a unos 140 para la mejor solución de panelización que hemos realizado, en el caso de la fachada que se ha estudiado en el primer ejemplo, estaríamos hablando de 11 horas y 40 minutos, en el mejor de los casos.

El ejemplo de fachada que hemos tomado en el ejemplo anterior es relativamente pequeño, al tratarse de una fachada de aproximadamente 160 m<sup>2</sup>.

En el caso de analizar edificios o conjuntos de edificios más complejos, estos plazos de modelación se dilatarían considerablemente. Si por ejemplo estudiamos el caso de la Western Metro Station, en Riad (Arabia Saudí), actualmente en construcción, se puede observar que tanto las fachadas como las



cubiertas, respetan una panelización con reglas geométricas que se mantienen a través de los paneles de hormigón que constituyen fachada y cubierta.

Dado que estimamos los paneles, de forma muy aproximada, a 20.000 paneles de hormigón, el modelado de la fachada de forma tradicional, estaríamos hablando de 1670 horas de modelado. Considerando una jornada normal de 8 horas, estaríamos hablando de más de 200 días de modelado para una persona. Considerando una jornada laboral normal, con festivos y vacaciones, podemos decir que sería el trabajo de un año para una persona.

Esto lógicamente, por muy aproximados que sean los cálculos, demuestra que no es viable realizar una tarea como ésta manualmente.

La segunda opción consiste en redactar un código mediante la herramienta de parametrización GrassHopper que nos permita obtener la geometría real de un panel de hormigón aplicando el código a una simple superficie.

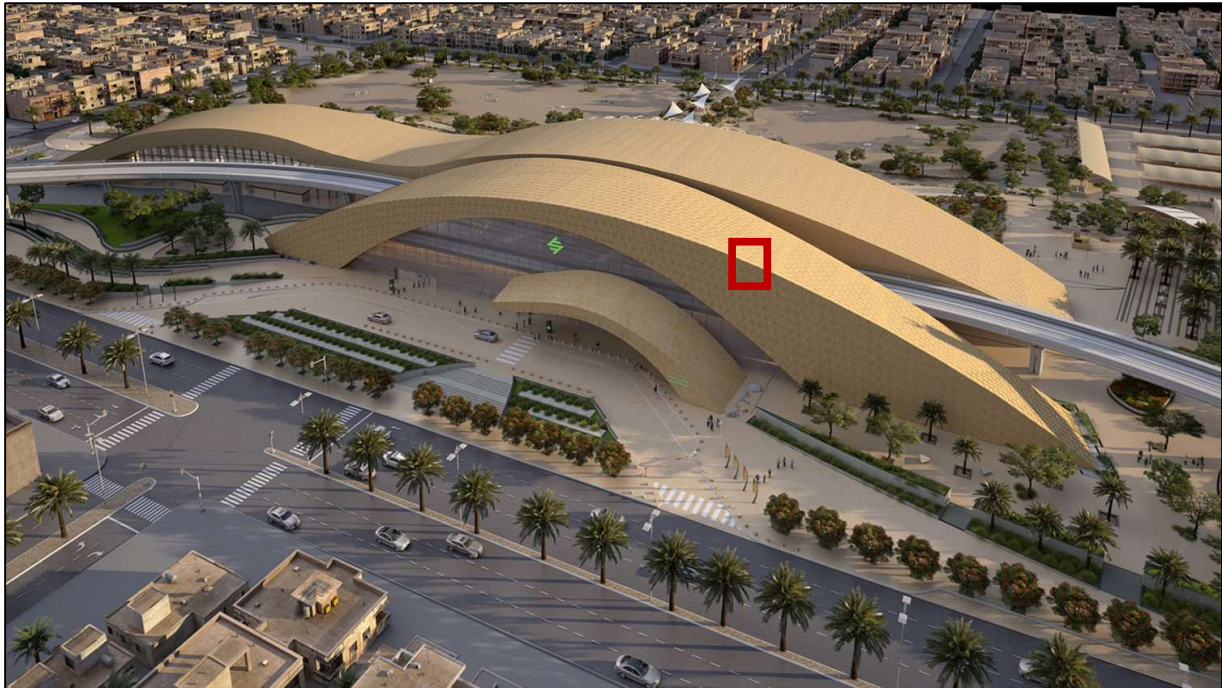


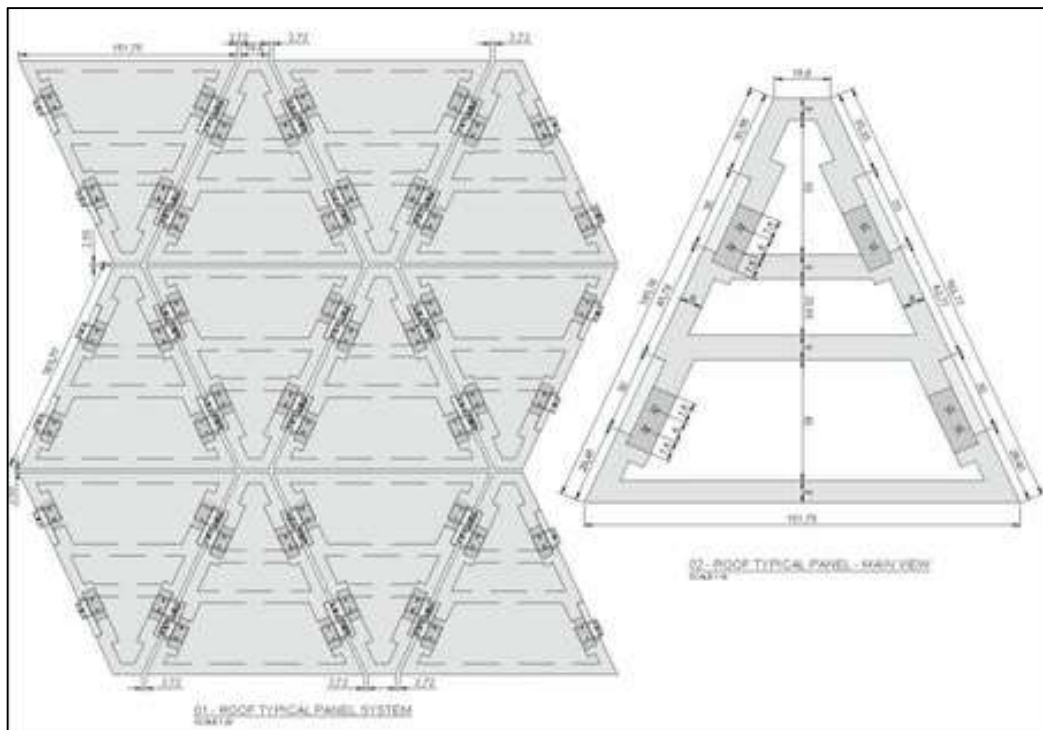
Ilustración 24: RIYADH WESTERN METRO - Zaha Hadid – Fuente: ONEWORKS (2022) [ONLINE]. Available at: <https://www.one-works.com/our-works/riyadh-metro-western-station>



Ilustración 25: PANELIZACIÓN DE RIYADH WESTERN METRO – Fuente: ONEWORKS (2022) [ONLINE]. Available at: <https://www.one-works.com/our-works/riyadh-metro-western-station>

Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios

En las fotografías anteriores podemos observar la Western Metro Station de Riyadh (*Ilustración 24 y 25*). Esta estación está actualmente en desarrollo de ejecución. La fachada consiste en paneles de hormigón UHPC, que se enganchan entre ellos al mismo tiempo que se enganchan a la estructura del edificio. Los paneles tienen una forma triangular, que quedan reforzados en sus bordes (*Ilustración 26*). Se pueden apreciar a continuación detalles de los paneles que conforman la fachada.



*Ilustración 26: DETALLE DE PANELIZACIÓN DE RIYADH WESTERN METRO – Fuente: NEWTECNIC (2022) [ONLINE]. Available at: <https://newtecnic.com/western-metro-station-riyadh-saudi-arabia>*

El desarrollo de un proyecto de la magnitud de la estación de metro Western Metro de Riyadh, requiere la modelización de una maqueta en 3D por dos motivos. Primero el modelo detallado a nivel



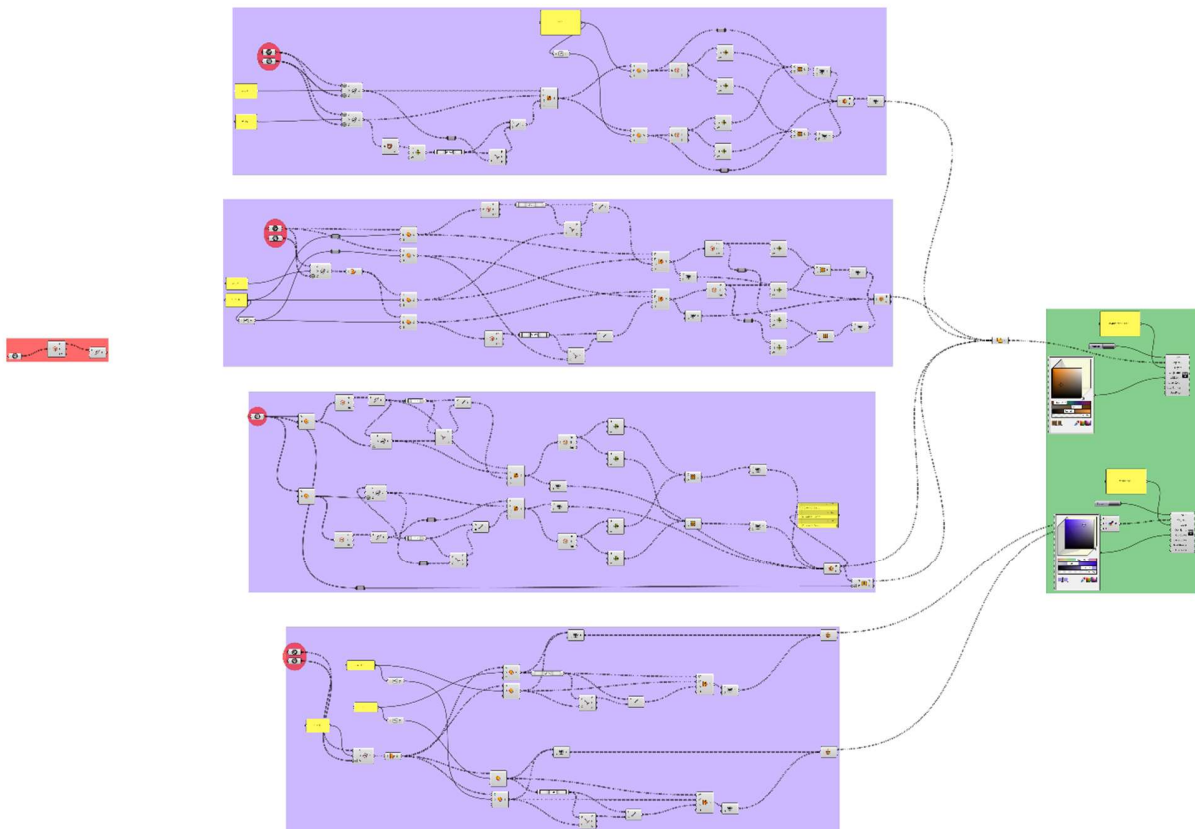
LOD400, permitirá encontrar cualquier tipo de interferencias entre los diferentes sistemas. Segundo, la modelización de cada panel es necesaria para su prefabricación. Una vez se haya modelado cada pieza, y se haya comprobado el diseño definitivo, cada panel queda definido para que un fabricante realice las piezas acordes con el diseño. De esta forma no puede haber desperdicio alguno.

Como se ha explicado anteriormente, el modelado 3D consiste en fases. Una vez alcanzado el nivel LOD100, se obtiene la geometría generica, y la disposición de cada panel. Una buena forma de obtener la disposición de los paneles sería mediante variantes de la metodología que hemos visto en el primer ejemplo. Sin embargo, los paneles todavía no tienen su geometría final. Los paneles en esta primera instancia se representarían como una superficie. Utilizando herramientas de parametrización, podemos llevar cada superficie a ser una representación tridimensional de la versión virtual del panel que se colocará en la fachada del edificio. Lógicamente, ésto consiste en modelización básica, y se tardaría seguramente menos en modelizar un panel por métodos tradicionales de modelización 3D. No obstante, la gran ventaja que supone hacerlo mediante software paramétrico es que una vez que se haya redactado el código que permita modelar la pieza a partir de la superficie, este mismo código puede ser reutilizado para cada pieza de la fachada. Además de ello, el código podrá generar múltiples paneles a la vez, dentro de una única secuencia. La cantidad de paneles que se podrán ejecutar a la vez es directamente proporcional a la capacidad y potencia del ordenador que tiene que ejecutar el código.



### 7.2.2. Desarrollo del código

A continuación, se presenta el código que se ha desarrollado para aumentar el nivel de detalle de la fachada que se había estudiado en el primer ejemplo (*Ilustración 27*). En un primer ejemplo hemos estudiado las diferentes posibilidades de panelización de una superficie curva. A partir de esa panelización a continuación, presentamos el código que permite generar los vidrios fijos de la fachada (LOD400) a partir de las superficies de referencia (LOD100).



*Ilustración 27: VISTA GENERAL - CODIGO 2– Fuente: Fuente propia*



Este segundo código es más extenso que el código que se ha estudiado anteriormente. El cuerpo del código consiste en 4 bloques, aquí en azul. Los tres bloques superiores sirven para generar la carpintería del panel. El último bloque sirve para generar los vidrios de del panel.

La geometría de entrada se encuentra a la izquierda, en el bloque de color rojo. En el nodo de entrada, se pueden asignar todas las superficies de una panelización, para que el código ejecute la serie de comandos sobre todas las superficies a la vez. En ese primer grupo, a partir de las superficies de entrada, el código genera una polilínea con las 4 vertientes de cada superficie. La superficie en sí, y la polilínea que limita la superficie, serán los dos elementos nativos que nos permiten dibujar toda la geometría adicional que nace del código. De hecho, para que el código quede más limpio visualmente, en cada grupo se han conectado estas dos propiedades geométricas a dos parámetros "espejo" en un círculo rojo que se encuentra al inicio de cada bloque del código. Las líneas que unen los parámetros espejo con los parámetros iniciales existen, pero por motivos de visualización se han hecho conexiones "transparentes" a cada uno de los grupos del cuerpo del código.

Como se ha indicado anteriormente, los tres primeros bloques consisten en el diseño de la carpintería (*Ilustración 28, 29 y 30*). Para no complicar excesivamente el código, se generan tres piezas que conforman la carpintería y que posteriormente se unen.

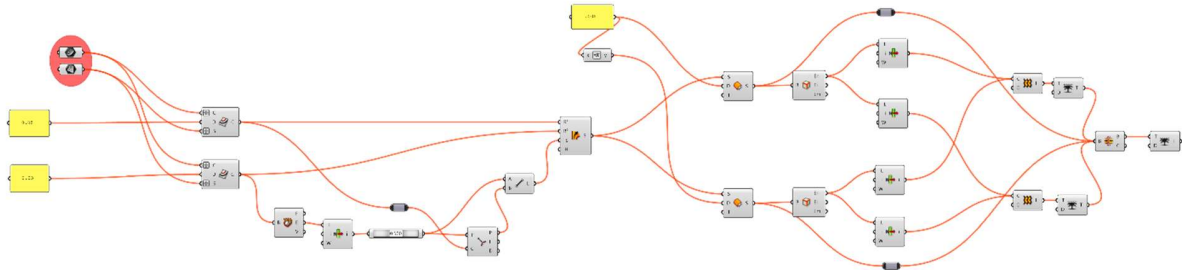


Ilustración 28: BLOQUE 1 - CODIGO 2 – Fuente: Fuente propia

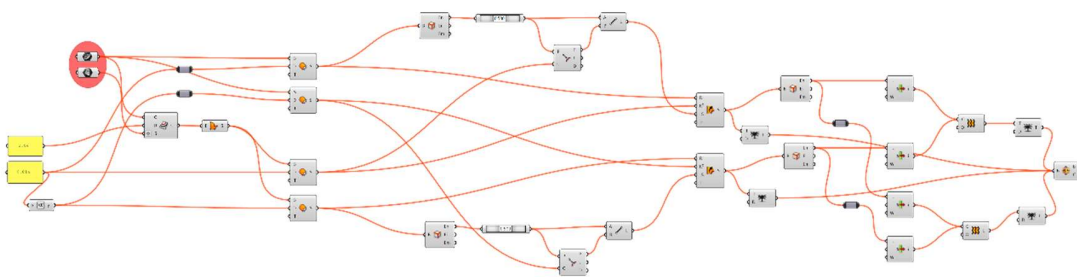


Ilustración 29: BLOQUE 2 - CODIGO 2– Fuente: Fuente propia

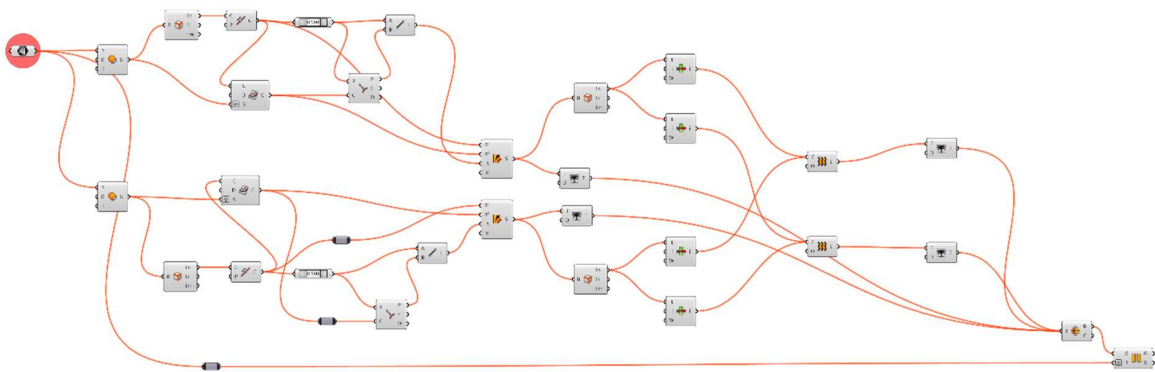


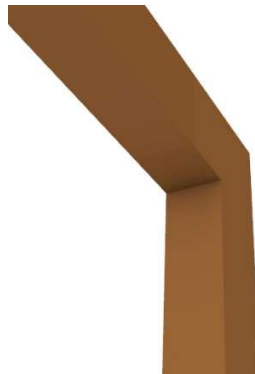
Ilustración 30: BLOQUE 3 - CODIGO 2– Fuente: Fuente propia



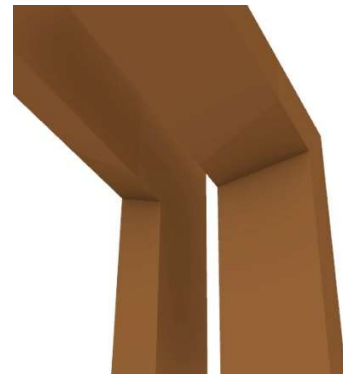
A continuación, se detallan los tres elementos que conforman la carpintería. Cada bloque (*Ilustración 31, 32 y 33*) genera uno de los elementos que después se unen mediante una unión sólida (*Ilustración 34 y 35*).



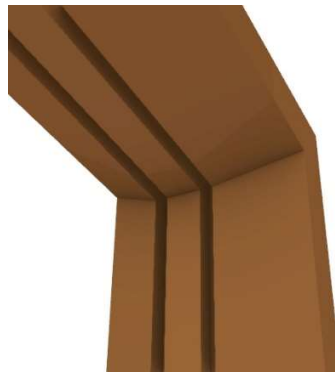
*Ilustración 31: BLOQUE 1 – Fuente:  
Fuente propia*



*Ilustración 32: BLOQUE 2 – Fuente:  
Fuente propia*



*Ilustración 33: BLOQUE 3 – Fuente:  
Fuente propia*



*Ilustración 34: VISTA DETALLE – RESULTADO UNION SOLIDA DE BLOQUES 1, 2 Y 3 – Fuente: Fuente propia*



*Ilustración 35: UNION SOLIDA DE BLOQUES 1, 2 Y 3– Fuente: Fuente propia*

El código simultáneamente modela los vidrios a partir de la superficie de la referencia. La parte del código que ejecuta los comandos para realizar los dos vidrios corresponde al cuarto bloque del código (*Ilustración 36*). El código ejecuta dos sólidos de 6mm de espesor, separados por una cámara de aire de 16mm (*Ilustración 37 y 38*). El bloque 4 es la parte del código responsable de generar los dos vidrios.

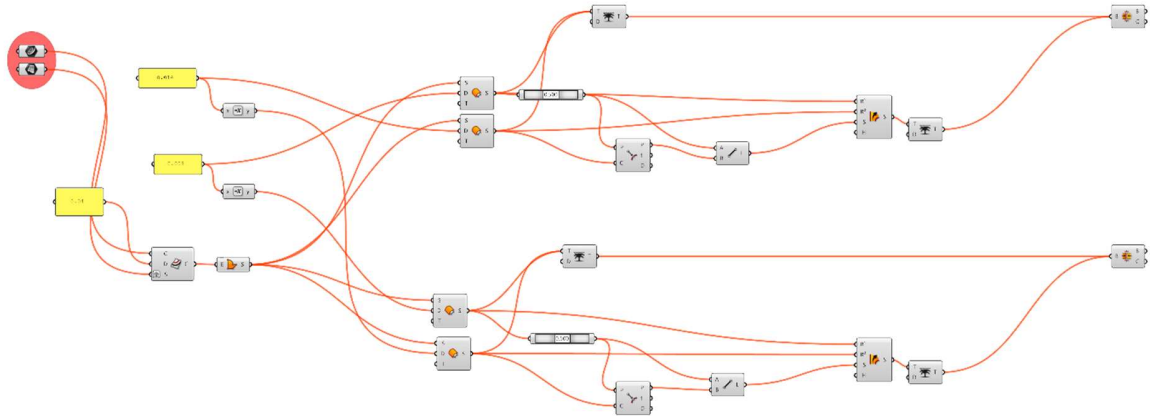


Ilustración 36: BLOQUE 4– Fuente: Fuente propia

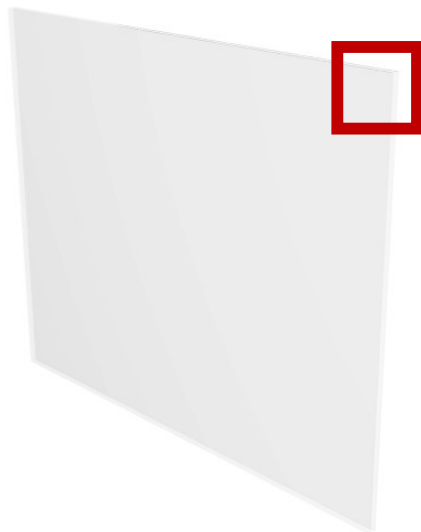


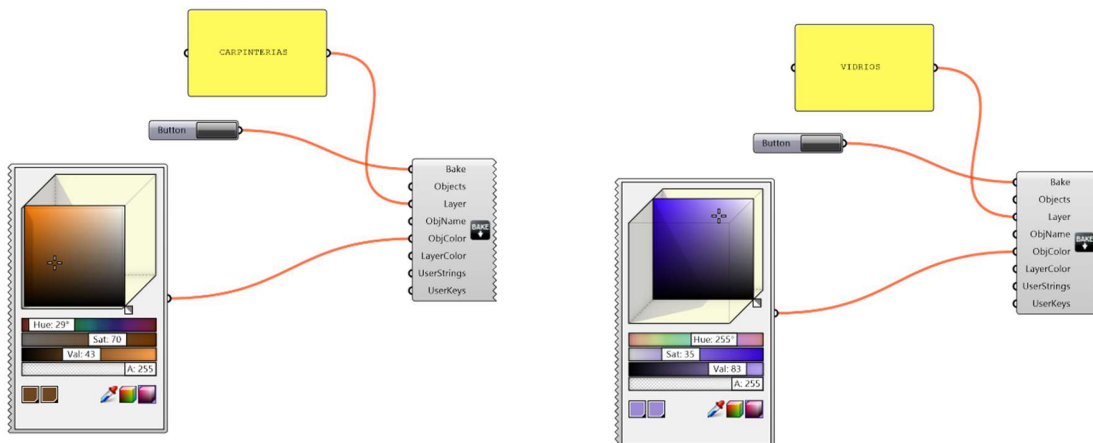
Ilustración 37: RESULTADO BLOQUE 4– Fuente: Fuente propia



Ilustración 38: VISTA DETALLE BLOQUE 4– Fuente: Fuente propia

En el bloque final, ubicado a la derecha del código en color verde (*Ilustración 39*) se ubican los comandos para pasar la geometría del código a geometría nativa del programa de modelado *Rhino*. En la parte superior se modela la pieza de carpintería, a la que se le asigna un color marrón, y la propiedad de capa "CARPINTERIA". Haciendo clic en el botón, ejecutamos el comando "BAKE" que permite generar la geometría detallada en *Rhino*.

Se procede de la misma forma para los vidrios a los que se le asigna la capa "VIDRIOS" y un color azul claro.

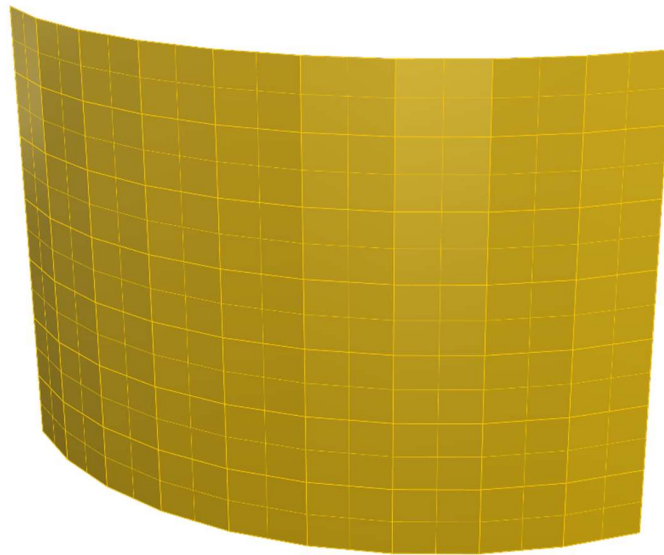


*Ilustración 39: BLOQUE FINAL DE EXPORTACIÓN DE GEOMETRIA A RHINO – Fuente: Fuente propia*



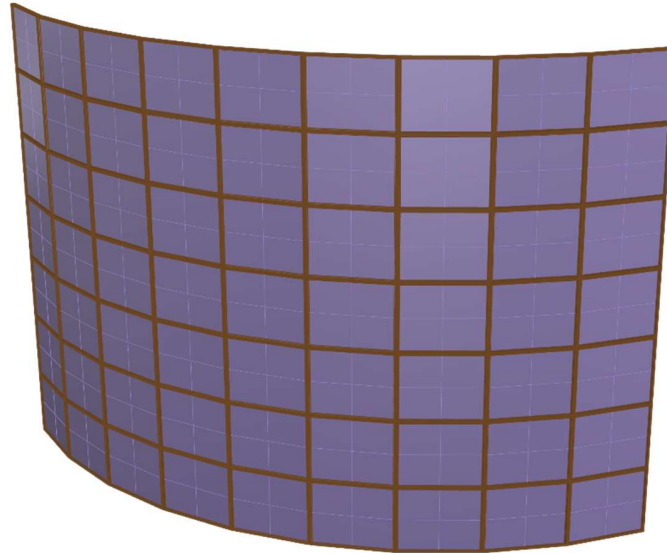
### 7.2.3. Resultados

El código, permite ejecutar todo lo que se ha explicado anteriormente sobre una multitud de paneles. En vez de seleccionar una única superficie como se ha hecho a título demostrativo, se seleccionan todas las superficies de la fachada que se pretenden panelizar en la parte de asignación de parámetros del código (*Ilustración 40*).



*Ilustración 40: SUPERFICIE PANELIZADA NIVEL LOD100 – Fuente: Fuente propia*





*Ilustración 41: RESULTADO SUPERFICIE PANELIZADA POR CODIGO LOD400 – Fuente: Fuente propia*

Los resultados obtenidos son muy satisfactorios por dos motivos: Primero, el código genera todos los componentes en menos de 15 segundos. En segundo lugar, los elementos no contienen ningún fallo de modelado. Las carpinterías generadas son de un solo elemento sólido cerrado por panel. El hecho de que sean sólidos cerrados (closed polysurfaces), nos permite primero decir que el elemento está bien generado y no contiene aristas abiertas, y también nos permite una fácil exportación a cualquier otro programa de modelado 3D. Los vidrios que se han generado también consisten en dos elementos sólidos cerrados de exactamente mismas características geométricas para cada panel (*Ilustración 41*).



#### 7.2.4. Análisis de resultados

Para generar todos los paneles de la superficie que se ha tomado como ejemplo, el código tarda aproximadamente 12 segundos en generar la geometría con un ordenador de gama media. Cuanto más potentes sean los procesadores del ordenador más corto será el tiempo de ejecución del código.

El tiempo de modelado de cada panel, con el nivel de detalle que se ha puesto como objetivo, se puede estimar de unos 5 minutos como mínimo para una persona altamente cualificada en modelado tridimensional. Teniendo en cuenta que hay 160 paneles, el tiempo estimado sería de 800 minutos, es decir más de 13 horas de modelado.

El tiempo que se dedica a escribir un código es mucho más difícil de estimar debido a que depende en gran parte de la experiencia y conocimientos del autor del código. De ser la capacidad de escribir código del autor relativamente alta, redactar un código como el que se ha presentado anteriormente, puede variar entre 2 a 10 horas. De estar menos familiarizado con la tecnología, los tiempos pueden dilatarse considerablemente.

La principal ventaja de modelar mediante tecnologías de parametrización es el ahorro en tiempo. De forma directa, el tiempo de modelado se ve reducido si consideramos el tiempo total invertido de las dos formas, tal y como se ha demostrado anteriormente. De forma indirecta, se ahorra un tiempo considerable en el caso de que se realicen modificaciones de proyecto. Se trate de modificaciones en la panelización de una fachada, como una variación en los tamaños de los paneles, o una variación en la geometría proyectada por panel, una simple modificación del código en el peor de los casos



permitiría ajustar el diseño lo que no repercutiría prácticamente en el tiempo de ejecución. De funcionar con métodos tradicionales de modelado, se tendría que volver a empezar la tarea desde el principio, desechando todo lo que se haya hecho anteriormente. También de forma indirecta, se ahorrará tiempo en las correcciones que se suelen tener que hacer a la hora de hacer un modelado manual. Durante el modelado manual, se cometen errores que no se dan en el modelado paramétrico. Finalmente, se puede decir que mientras que una geometría específica es imposible de reciclar, un código de modelado paramétrico sí lo es, y que de encontrar similitudes en proyectos se pueden realizar las oportunas modificaciones para reusar un código y de esa forma acortar aún más los tiempos de modelado.

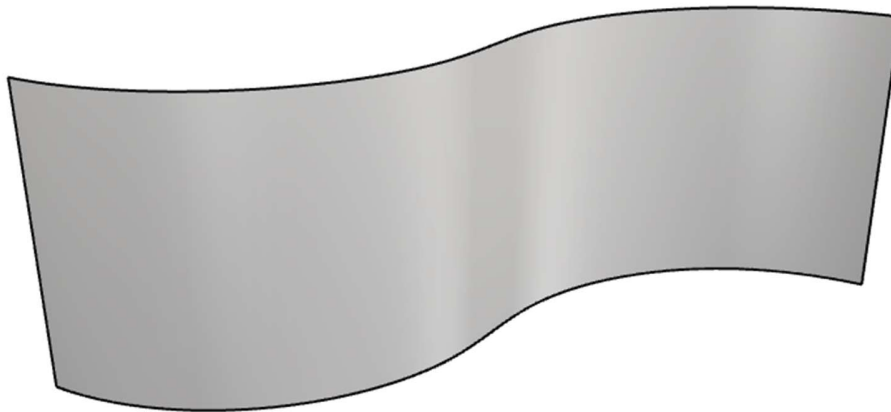


## 8. ALCANCE DEL CODIGO

En esta sección se pretende demostrar el alcance de los códigos que se han presentado en el *EJEMPLO 1 y 2*, y en particular, la capacidad de adaptarse a diferentes tipos de superficies. Para ello se verán dos facetas de la adaptabilidad de los dos códigos detallados en los anteriores ejemplos.

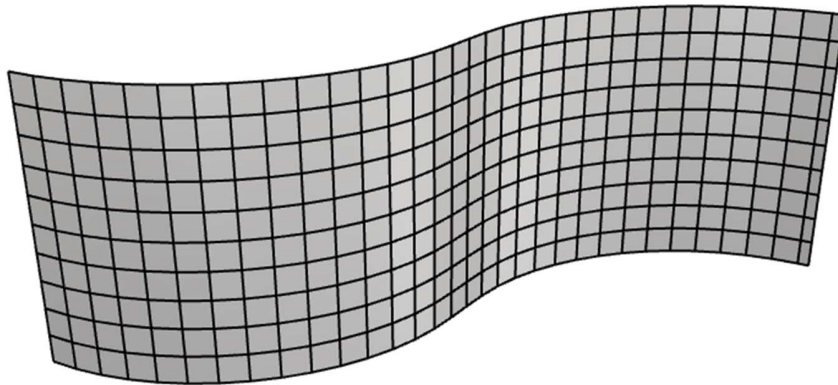
### 8.1. MODIFICACIONES DE SUPERFICIE BASE (CODIGO 1 y 2):

Como se ha avanzado en el primer código, la única limitación que existe para que se pueda ejecutar el *CÓDIGO 1* sería que la superficie base sea una forma de origen paralelogono, es decir que cada una de sus dos aristas opuestas sean paralelas. También es necesario, que la curvatura de tal superficie sea simple, y nunca doble. Es decir, que únicamente un par de aristas sean curvas paralelas y no líneas. Se podría adaptar el código para que responda ante superficies de doble curvatura, sin embargo, el resultado nunca sería el óptimo debido a las restricciones geométricas que estas superficies suponen.



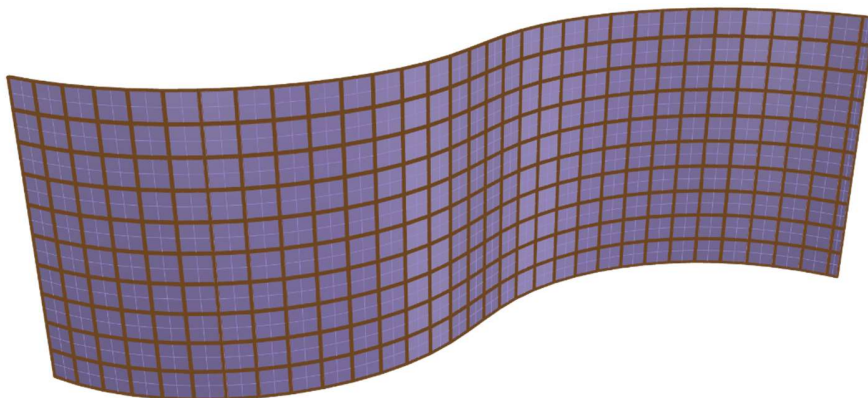
*Ilustración 42: SUPERFICIE BASE – Fuente: Fuente propia*

Se ejecuta el primer código sobre la superficie base con un panel cuadrado de 1m de ancho y de alto (*Ilustración 42*). El código responde positivamente, cubriendo toda la fachada menos la última fila que cubre con un panel de 0,28m (*Ilustración 43*).



*Ilustración 43: SUPERFICIE PANELIZADA – CODIGO 1 – Fuente: Fuente propia*

Procedemos a ejecutar el *CODIGO 2* sobre las superficies que hemos generado mediante el *CODIGO 1*, que otra vez responde sin problema a este cambio de superficie (*Ilustración 44 y 45*).



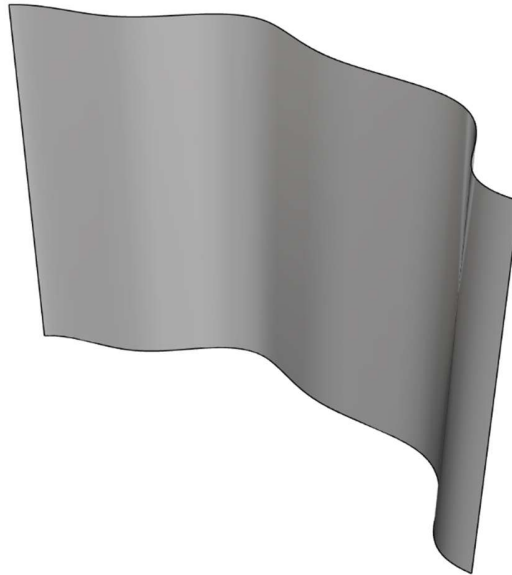
*Ilustración 44: SUPERFICIE PENALIZADA DETALLADA - CODIGO 2 – Fuente: Fuente propia*



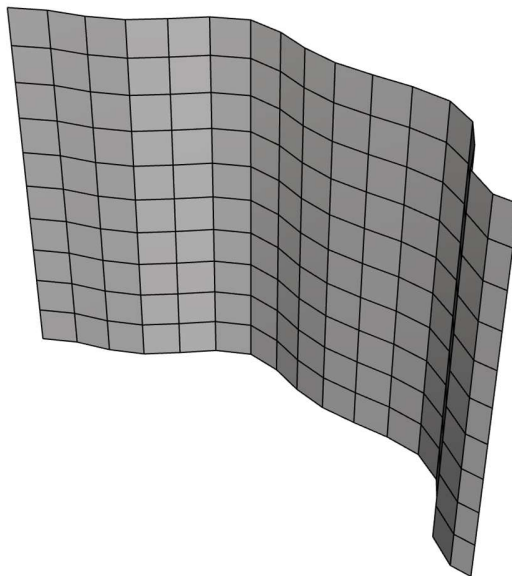
*Ilustración 45: DETALLE DE SUPERFICIE PANELIZADA LOD400 – CODIGO 2 – Fuente: Fuente propia*

Los dos códigos que se han probado responden perfectamente a este cambio de superficie por lo que procedemos a probar el código con una superficie más complicada. Para ello se ha tomado la superficie con la que se trabajó en ambos ejemplos, y mediante el control de puntos internos de la superficie, se ha distorsionado la forma aleatoriamente (*Ilustración 46*).

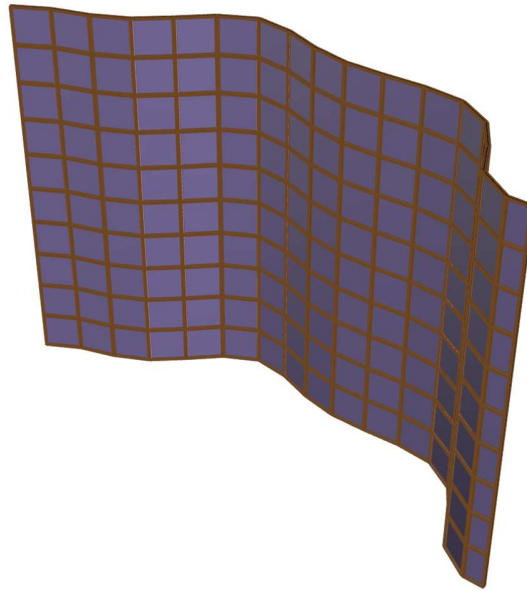
Como se puede apreciar, la curvatura de la superficie no respeta ninguna regla geométrica, y es difícil imaginar un diseño de fachada con una geometría parecida. Sin embargo, el *CODIGO 1* nos da un resultado bastante positivo, generando una fachada completamente panelizada, sin tener que realizar ninguna modificación al código, con una desviación máxima entre superficie inicial y proyectada de 7,44cm (*Ilustración 47*).



*Ilustración 46: SUPERFICIE BASE – Fuente: Fuente propia*



*Ilustración 47: SUPERFICIE PANELIZADA – CODIGO 1 – Fuente: Fuente propia*



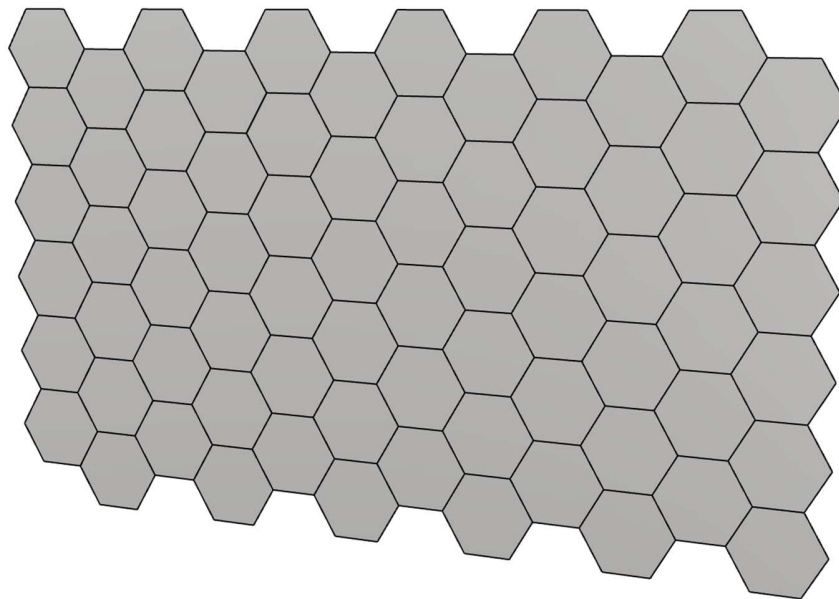
*Ilustración 48: SUPERFICIE PENALIZADA DETALLADA - CODIGO 2 – Fuente: Fuente propia*

El *CODIGO 2* también responde positivamente y, sin necesitar ningún ajuste, genera toda la carpintería y todo el doble vidrio de la fachada (*Ilustración 48*). Se puede concluir que el *CODIGO 1* se adapta perfectamente a cualquier superficie, siempre y cuando las aristas opuestas del cuadrilátero que se tome como superficie inicial sean curvas paralelas.

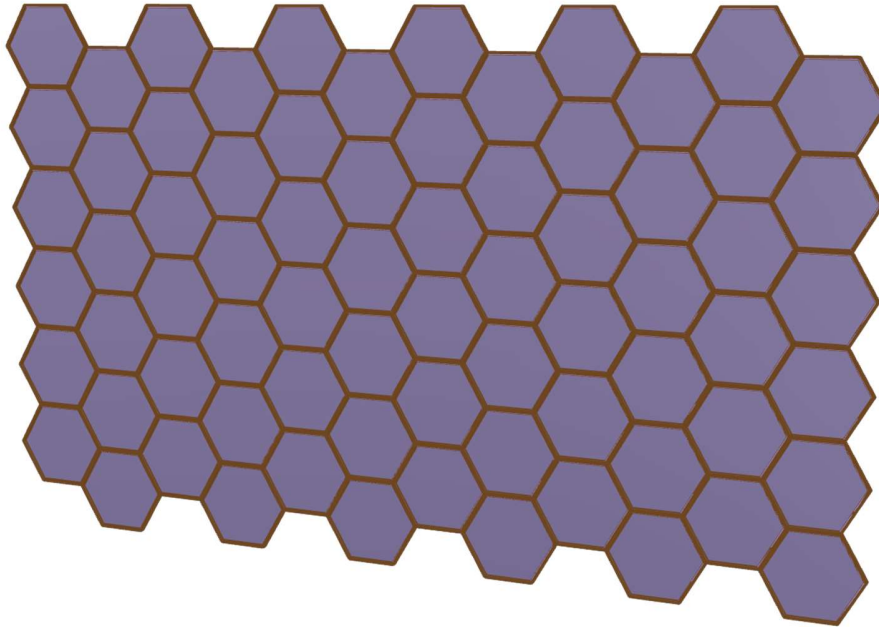


## 8.2. MODIFICACIONES DE PANELIZACIONES (CODIGO 2):

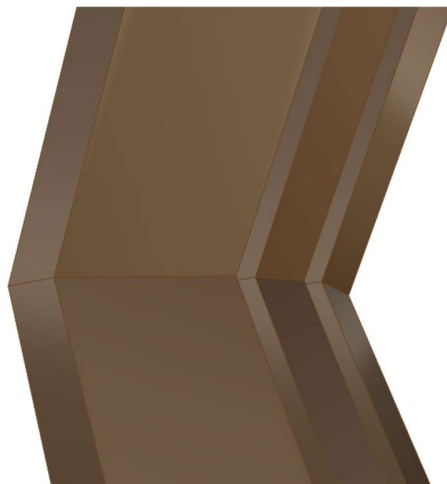
A continuación, veremos la capacidad del *CODIGO 2* para adaptarse a diferentes tipos de panelizaciones. Para ello, se propone estudiar panelizaciones que no sean rectangulares. Procederemos a probar si el código funciona, sin realizar ninguna modificación a éste, con una panelización hexagonal (*Ilustración 49*) y triangular (*Ilustración 52*). El *CÓDIGO 2* utiliza los bordes de la superficie de referencia para generar la carpintería del perfil y los vidrios, por lo que no tendría que suponer ningún problema.



*Ilustración 49: PANELIZACIÓN HEXAGONAL – Fuente: Fuente propia*



*Ilustración 50: PANELIZACIÓN HEXAGONAL DETALLADA - CODIGO 2 – Fuente: Fuente propia*



*Ilustración 51: DETALLE DE CARPINTERIA – Fuente: Fuente propia*

Como se puede apreciar, para fachadas con paneles hexagonales, el código se adapta perfectamente, generando carpinterías y vidrios, con el nivel de detalle para el que se ha programado el código (Ilustración 50 y 51).

A continuación, se pretende realizar las mismas comprobaciones con una panelización triangular.

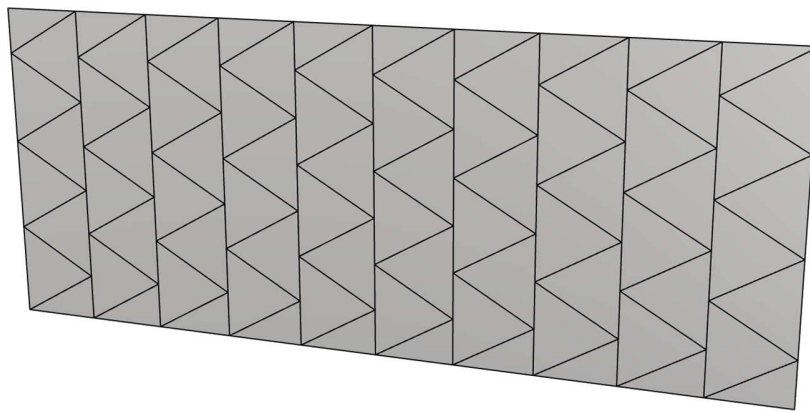


Ilustración 52: PANELIZACIÓN TRIANGULAR – Fuente: Fuente propia

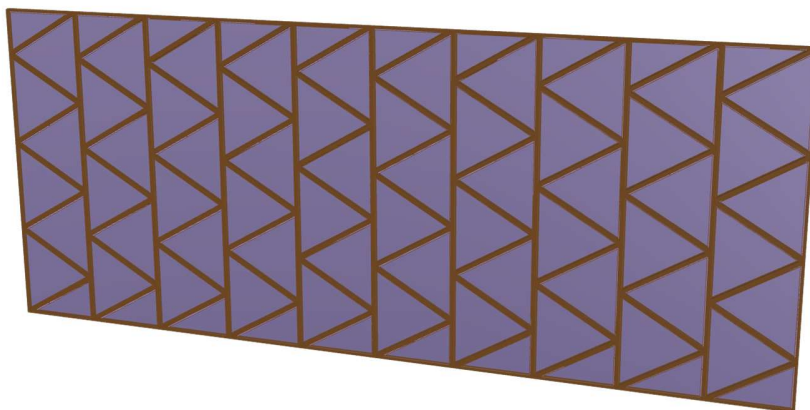


Ilustración 53: PANELIZACIÓN TRIANGULAR DETALLADA - CODIGO 2 – Fuente: Fuente propia



El CODIGO 2 también se adapta perfectamente a una tipología de panel triangular (*Ilustración 53*). Para generar la carpintería y los vidrios, el código utiliza las aristas (*naked edges*) del panel y realiza una serie de comandos tomando las aristas como única referencia, por lo que el código funciona perfectamente sin tener que realizar ninguna modificación, independientemente de la forma del panel que se pretende modificar.



## 9. CONCLUSIONES

Las herramientas paramétricas son una tecnología con relativamente poca presencia en la industria de la construcción. Sin embargo, tienen muchas aplicaciones que permiten optimizar flujos de trabajo y acelerar los tiempos en tareas de la fase de diseño de edificaciones. La utilización de estas tecnologías también puede repercutir en optimización de tiempos en fase de ejecución, ya que permiten realizar un modelo más completo de la edificación que se proyecta. En el marco de este estudio, hemos visto dos formas a través de ejemplos que permiten optimizar los flujos de trabajo del diseño y la proyección de edificios.

En la tarea de diseño de edificios, el actual flujo de trabajo tiene margen de optimización. En vez de diseñar, y después ajustar en base a los materiales de los que disponemos, resulta más eficiente diseñar teniendo una visión global, teniendo en cuenta todos los parámetros, incluidos los materiales y componentes de los que podemos disponer a la hora de ejecutar una obra.

En el diseño de fachadas, por ejemplo, la tendencia a construir con prefabricados requiere que cada pieza de un edificio se diseñe y se apruebe con anterioridad al inicio de la obra. Uno de los grandes esfuerzos para optimizar el proceso de prefabricación es el uso de piezas o componentes idénticos. Las mejores soluciones de diseño siempre suelen ser las que mayor porcentaje de piezas idénticas tienen.

El *CODIGO 1* que se ha desarrollado permite plantear diseños a partir de las piezas prefabricadas de las que disponemos. A partir de este código (*CODIGO 1*), podemos describir geoméricamente las



piezas que disponemos, en base a dos parámetros: el ancho y el alto, para que el código genere un sistema utilizando únicamente esa pieza. El código (*CODIGO 1*), permite plantear tantas opciones de diseño como tipos de componentes diferentes dispongamos.

Lo anterior permite obtener diseños de fachada en base a un componente. En vez de adaptar un diseño a las piezas de las que disponemos, podemos describir las piezas de las que disponemos para evaluar diferentes posibilidades de diseño y elegir la más conveniente. Por lo general, la mejor opción de panelización siempre será la que más alto porcentaje de piezas idénticas tenga. Por lo tanto, permite generar, con más rapidez, más opciones de diseño y de esta forma realizar un estudio más exhaustivo de las opciones de las que disponemos.

El código se ha escrito para el caso de una fachada de vidrio, sin embargo, este flujo de trabajo es extrapolable a cualquier optimización de disposición o sectorización. Se podría, siguiendo la misma metodología, plantear por ejemplo estudios de disposición de paneles fotovoltaicos en cubierta, de paneles de GRC en fachas ventiladas o incluso de azulejos en paramentos verticales y horizontales. todos estos ejemplos comparten la característica de formar un conjunto de piezas prefabricadas, que requiere un diseño de disposición.

Más allá del ahorro de tiempo directo que permite trabajar de esta forma, también es de destacar que estas metodologías se pueden adaptar a diferentes diseños, por lo que las modificaciones que se realicen a través del proyecto no repercuten en tiempos de modelado. El código, se adapta a las modificaciones que puedan darse sobre el diseño, y adapta la geometría que se haya proyectado



inicialmente a las modificaciones que se realicen en adelante. Esto permite que nunca se tenga que descartar y volver a modelar un estado proyectado en base a una modificación de diseño, por muy avanzado que esté el proyecto.

Las ventajas de realizar maquetas 3D a un nivel de detalle alto (LOD400) son notables, debido en particular a que permiten adelantarse a todos los problemas que pueden suceder en la ejecución de una obra. Por el contrario, este tipo de tareas serían altamente laboriosas de proceder mediante modelado “tradicional”, es decir pieza a pieza.

El *CODIGO 2* permite aplicar un diseño geométrico a una multitud de componentes a la vez. Dicho de otro modo, en vez de requerir modelar las piezas una a una, permite modelar a la vez una multitud de piezas de misma tipología a un nivel de detalle muy elevado. En nuestro caso, el código permite modelar paneles de vidrio doble con carpintería a partir de superficies básicas. Los paneles que se han considerado en el caso del muro cortina se repiten a través de una fachada.

Las herramientas de modelado paramétrico permiten actuar sobre una multitud de piezas a la vez, ahorrando un tiempo considerable dentro de esta tarea. Hemos visto, mediante el *CODIGO 2*, que el modelado paramétrico permite obtener un modelado de mayor calidad y detalle, de forma mucho más rápida que métodos de modelado tradicional.

Como en el caso del *CODIGO 1*, esta metodología se podría extrapolar a cualquier tarea de maquetación 3D donde sea necesario modelar varias veces un sólido de misma tipología, como



podrían ser por ejemplo cualquier elemento de construcción prefabricado. Además, se ha comprobado que se pueden realizar códigos, que se adapten a diferentes entornos, por lo que para un tipo de prefabricados que conserve las mismas características geométricas, se puede mantener el mismo código, ajustándolo si necesario y reutilizándolo todas las veces que sea necesario. Una ventaja adicional al ahorro en tiempo es que la geometría que se genera a partir de código no contiene fallos de modelado, debido a que se ejecuta por ordenador manteniendo siempre la misma combinación de comandos. De esta forma, se elimina por completo la posibilidad de fallos que nacen de un modelado hecho a mano.

En conclusión, podemos decir que ambos ejemplos demuestran la optimización en tiempo y recurso que supondría integrar dichas herramientas en el actual flujo de trabajo de diseño y proyección de sistemas de edificios. Por un lado, el primer código nos ha permitido explorar nuevos diseños experimentales que a su vez permiten tomar mejores decisiones a la hora de proyectar. Por otro lado, el segundo código nos permite automatizar tareas repetitivas.





## 10. FUTURAS LINEAS

Estas metodologías que hasta hace poco tenían muy escasa presencia en la industria de la construcción, cada vez son más susceptibles de irrumpir en la industria debido a los desarrollos que se están realizando para que el intercambio de información entre diferentes herramientas de modelado 3D se realice más fácilmente.

En agosto de 2019, McNeel, la compañía fundadora del software Rhino y de Grasshopper, publicaba la primera versión beta de *Rhino.inside.Revit*, que por primera vez permitía utilizar el software de Rhino y de Grasshopper, dentro de Revit. El desarrollo de este plug-in supone un desarrollo sin precedentes en la integración de estos dos programas que, a pesar de servir para objetivos similares, tienen muy poco en común a la hora de modelar.

En palabras de los fundadores del plug-in, "*Rhino.Inside.Revit* ofrece un conjunto de herramientas que acorta distancias entre el modelado de forma libre de Rhino y el mundo de BIM, abriendo el rico ecosistema de Grasshopper al entorno de Revit, al permitir nuevos flujos de trabajo de diseño y nuevas posibilidades." <sup>11</sup>

La integración de estos dos softwares supone un avance considerable en el modelado 3D de edificaciones. Rhino y Grasshopper permitían hasta ahora un modelado mucho más libre que Revit. Sin embargo, no permitían todas las posibilidades del entorno BIM que ofrece Revit, que hoy en día se han vuelto indispensables para la ejecución de proyectos de edificación. Hasta entonces, importar de un software a otro no era tarea fácil, y requería la utilización de otras herramientas como Dynamo,

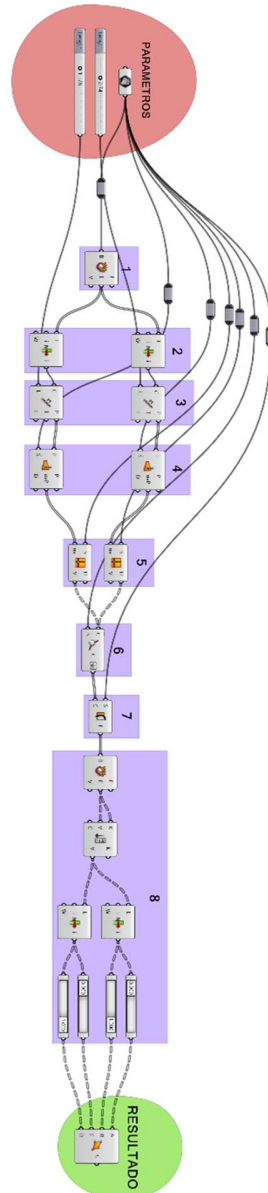


una herramienta de lenguaje paramétrico integrada en Revit, para importar un proyecto de Rhino, elemento a elemento, dentro de un modelo Revit.

Finalmente, podemos decir que, de forma general, los esfuerzos que se están realizando para incorporar tecnologías paramétricas a los flujos de trabajo actuales están sirviendo para sentar las bases de una nueva forma de desarrollar y ejecutar proyectos en el ámbito de la construcción. Esta nueva forma de trabajar se centra en la optimización de recursos, y sobre todo en el modelado de cada detalle de la edificación proyectada, antes del inicio de la construcción, para evitar cualquier contratiempo que se pueda dar en obra. Las herramientas paramétricas de modelado 3D como Grasshopper, permiten que este nuevo flujo de trabajo sea mucho más viable que antes, y su incorporación al software Revit, lo permite aún más.

## 11. ANEXOS

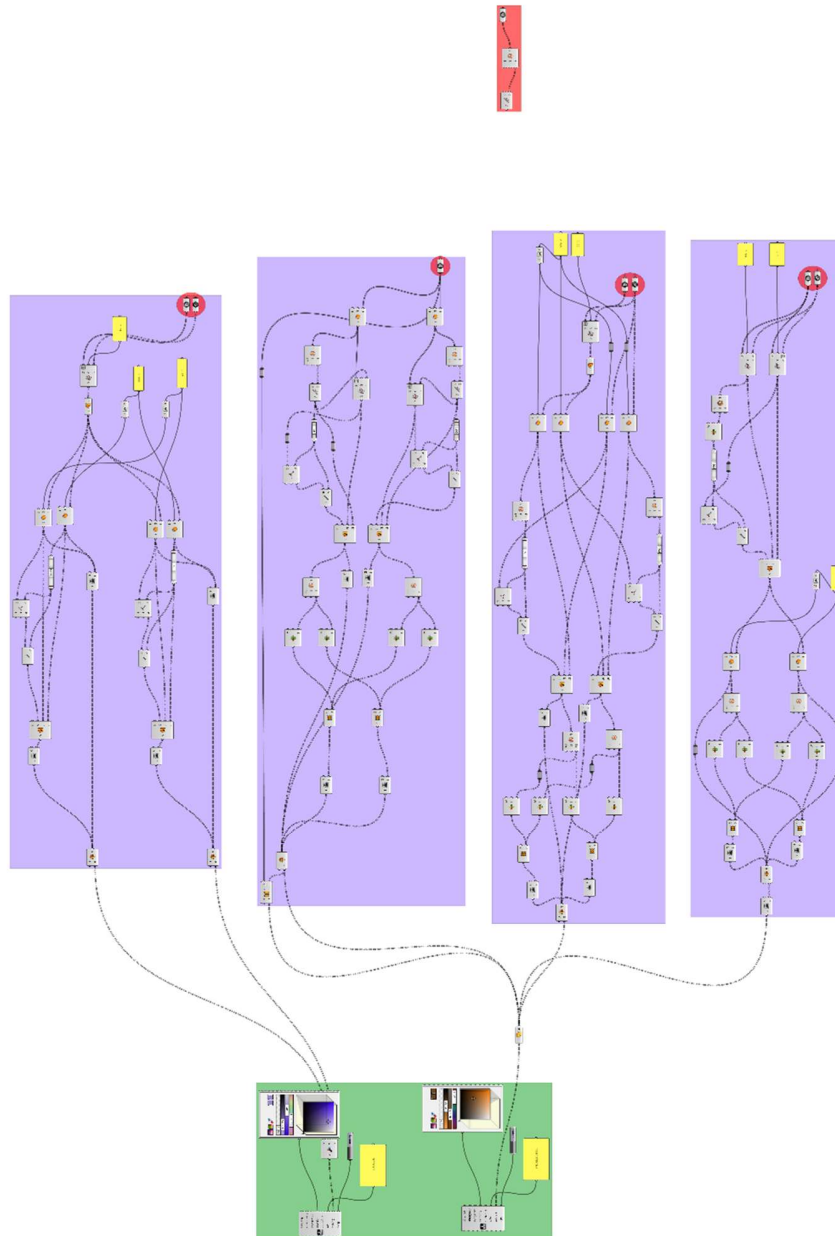
### 11.1. CODIGO 1



Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios



## 11.2. CODIGO 2



Implementación de herramientas de modelado paramétrico en el ámbito del diseño y ejecución de edificios



## 12. REFERENCIAS

1. Cilento, K. Parametricist Manifesto / Patrik Schumacher. *The Architect's Newspaper* (2010).
2. Al-Azzawi, T. & Al-Majidi, Z. Parametric architecture: the second international style. *IOP Conf. Ser. Mater. Sci. Eng.* **1067**, 012019 (2021).
3. Schumacher, P. *The Autopoiesis of Architecture, Volume I: A new Framework for Architecture.* (2011).
4. Oktan, S. & Serbulent, V. Parametricism: a style or a method? *Archtheo* 66–74 (2017).
5. Kalay, Y. E. *Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design.* MIT Press (2004).
6. Kolarevic, B. *Architecture in the Digital Age: Design and Manufacturing.* (2003).
7. Woodbury, R. *Elements of parametric design.* (Bentley Institute Press, 2010).
8. Burry, M. *Scripting Cultures : Architectural Design and Programming.* (2011).
9. Rocker, I. M. Calculus-based form: An interview with Greg Lynn. *Archit. Des.* **76**, 88–95 (2006).
10. AIA. *E203-2013 Building Information Modeling and Digital Data Exhibit.* (2013).
11. <https://www.rhino3d.com/features/rhino-inside-revit/>.