

UCUENCA

Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Ciencias de la Computación

**Inteligencia Artificial en Arquitectura: Automatización de la Planificación del
Espacio y Generación de Volumetría o Masa de un Edificio**

Trabajo de titulación previo a la
obtención del título de Ingeniero
en Ciencias de la Computación

Autor:

Wilson Renato Berrezueta Siguenza

Mateo Sebastian Flores Terreros

Director:

Jaime Eduardo Veintimilla Reyes

ORCID:  0000-0003-0409-9602

Cuenca, Ecuador

2024-08-27

Resumen

El avance de la inteligencia artificial ha transformado significativamente los métodos de construcción y diseño, aunque Cuenca, como patrimonio cultural de la humanidad, enfrenta normativas que limitan la adopción de estas tecnologías. Este trabajo propone desarrollar extensiones de código específicas para Autodesk Revit y su complemento DynamoBIM, con el fin de facilitar la integración de la inteligencia artificial mediante técnicas de diseño generativo en el proceso arquitectónico de la ciudad. Se realizó un análisis exhaustivo de la normativa local, lo que permitió crear algoritmos genéticos para generar estructuras y volumetrías optimizadas, considerando los parámetros topográficos y normativos. Estas extensiones de código brindan a los arquitectos la posibilidad de aprovechar las capacidades de la inteligencia artificial para producir soluciones óptimas en minutos, garantizando el cumplimiento normativo y mejorando la eficiencia en el diseño y la construcción. Los resultados indican que estas herramientas son no solo útiles y accesibles, sino que también permiten un ahorro de tiempo considerable. La percepción positiva de los arquitectos y su intención de usarlas en el futuro reflejan el gran potencial de estas soluciones en la industria. En consecuencia, la implementación de estas extensiones en Cuenca podría revolucionar los métodos de construcción y diseño, permitiendo la integración de tecnologías avanzadas dentro de un marco normativo restrictivo, lo que representa una oportunidad única para modernizar el sector arquitectónico en la ciudad.

Palabras clave del autor: inteligencia artificial, diseño generativo, algoritmos genéticos, bim



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: <https://dspace.ucuenca.edu.ec/>

Abstract

The advancement of artificial intelligence has profoundly influenced current construction and design methods. However, Cuenca, recognized as a UNESCO World Heritage site, encounters regulatory challenges that impede the adoption of these technologies. This thesis proposes developing specific code extensions for Autodesk Revit and its DynamoBIM add-in to facilitate the application of artificial intelligence through generative design techniques in Cuenca's architectural processes. A thorough analysis of local regulations was conducted, leading to the creation of genetic algorithms that generate optimized structures and volumetries based on the city's topographic and regulatory parameters. These code extensions empower architects to harness artificial intelligence's capabilities, enabling them to produce optimal solutions within minutes while ensuring compliance with regulations and enhancing efficiency in design and construction. The study's findings indicate that these tools are not only user-friendly but also provide significant time savings. The favorable perception among architects and their intent to use these solutions in the future highlight the potential for widespread adoption within the industry. Therefore, implementing these code extensions in Cuenca could revolutionize the city's construction and design methods, allowing for the integration of advanced technologies within a restrictive regulatory framework, ultimately modernizing the architectural landscape.

Author Keywords: artificial intelligence, generative design, genetic algorithms, bim



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice de contenido

1	Introducción	17
1.1	Contexto y Motivación	17
1.2	Planteamiento del problema	19
1.3	Solución propuesta	20
1.4	Objetivos	23
1.4.1	Objetivo general	23
1.4.2	Objetivos específicos	23
1.5	Estructura del trabajo	23
2	Marco Teórico	25
2.1	Building Information Modeling (BIM)	25
2.1.1	Autodesk Revit	27
2.1.2	Autodesk DynamoBIM	28
2.2	Inteligencia Artificial (IA) en Arquitectura y Construcción.	30
2.3	Diseño Generativo en Arquitectura.	31
2.3.1	Algoritmos Genéticos	33
2.3.2	Autodesk y el Diseño Generativo	36
2.4	Planificación de espacios	39
2.5	Normativas de construcción locales	40
3	Estado del Arte	43
3.1	Trabajos Relacionados	43
3.1.1	Planificación de espacios usando diseño generativo	43
3.1.2	Diseño de volumetrías usando diseño generativo	47
3.2	Limitaciones dentro de la Literatura Actual	48
3.2.1	Adaptación de soluciones a normativas específicas	49
3.2.2	Evaluación Empírica de los Resultados Generados por las Extensio- nes de Código	49
4	Metodología	50

4.1	Selección del enfoque para la solución del problema	50
4.2	Desarrollo de la extensión de código para la generación automática de volu- metrías de edificios	51
4.2.1	Recopilación de datos de diseños de edificios y normativa	52
4.2.2	Selección de las métricas relevantes para la evaluación de los diseños generados.	54
4.2.3	Desarrollo del generador de diseños de volúmetrías con DynamoBIM	54
4.2.4	Aplicación del Algoritmo Genético con Project Refinery	57
4.2.5	Aprobación de las soluciones de la extensión de código	58
4.2.6	Evaluación empírica de la extensión de código	68
4.3	Desarrollo de la extensión de código para la planificación automática de es- pacios	71
4.3.1	Recopilación de datos de planes de espacio	72
4.3.2	Selección de las métricas relevantes para la evaluación de los diseños generados.	75
4.3.3	Desarrollo del generador de diseños de planificación de espacios con DynamoBIM	76
4.3.4	Aplicación del Algoritmo Genético con Project Refinery	82
4.3.5	Aprobación de las soluciones de la extensión de código	84
4.3.6	Evaluación empírica de la extensión de código	87
5	Resultados	91
5.1	Extensión de código para la generación de volúmetrías	91
5.1.1	Aprobación de los resultados de la extensión de código	91
5.1.2	Evaluación empírica de la extensión de código	92
5.2	Extensión de código para la planificación automática de espacios	104
5.2.1	Aprobación de los resultados de la extensión de código	105
5.2.2	Evaluación empírica de la extensión de código	107
6	Discusión	118

7 Conclusiones	121
-----------------------	------------

Referencias	122
--------------------	------------

Índice de figuras

Figura 1: Representación del concepto BIM. Fuente: Adaptado de Salman et al. (2012).	25
Figura 2: Ciclo de vida de una edificación con BIM. Fuente: Adaptado de BIM (2015).	26
Figura 3: Interfaz del software Revit. Fuente: Adaptado de Sheets (2023).	28
Figura 4: Ejemplo de flujo de trabajo en DynamoBIM. Fuente: Elaboración propia.	29
Figura 5: Proceso de diseño generativo. Inicia con la idea del diseño a optimizar, el diseñador escribe el código fuente para el generador de soluciones, y estas son evaluadas por el mismo diseñador. Fuente: Adaptado de Bohnacker et al. (2009).	32
Figura 6: Esquema general de un algoritmo genético. Inicia con una población inicial de sujetos. Itera seleccionando los individuos más aptos, aplicando cruce y mutación para generar la siguiente generación, hasta que se cumplen los criterios de parada, obteniendo una solución mejorada. Fuente: Adaptado de Dastanpour et al. (2014).	34
Figura 7: Menú de configuración de un estudio de diseño generativo con Autodesk Project Refinery. Se necesita definir las entradas, objetivos de diseño, y configurar tamaño de población, generaciones y semilla. Fuente: Elaboración propia.	37
Figura 8: Estudio de diseño generativo que inicia con un generador que recibe los datos de entrada y variables. Este genera diferentes configuraciones, que son evaluadas por métricas y enviadas a un solucionador para obtener resultados optimizados de manera iterativa. Fuente: Elaboración propia.	39
Figura 9: Metodología para generar y evaluar una extensión de código automatizada para volumetrías de edificios. Se procesa información y normativas, se diseñan plantas y bloques, se usan algoritmos genéticos para mejorar resultados previos, culminando con evaluación empírica.	52

Figura 10: Flujo del generador de volumetrías: Se seleccionan y filtran normas de construcción, se genera una PolyCurva del terreno con retiros, se crea un array de dimensiones aleatorias, se diseñan pisos sólidos y se unen al edificio, calculando métricas como COS, CUS y área construida, garantizando normativas y diseño funcional.	57
Figura 11: Levantamiento inicial del terreno con retiros inicialmente propuestos. . .	59
Figura 12: Ubicación en Revit para ingresar al estudio de diseño generativo. . . .	60
Figura 13: Selección del generador de volumetrías en Revit para el estudio de diseño generativo, indicando la versión final utilizada en esta tesis. . . .	61
Figura 14: Menú de configuración del algoritmo generativo en Revit, mostrando las variables de entrada y opciones de configuración necesarias para optimizar la generación de volumetrías.	62
Figura 15: Selección del terreno en el menú de configuración del algoritmo generativo en Revit, mostrando la simplicidad del proceso.	63
Figura 16: Configuración de los objetivos del algoritmo generativo para la generación de volumetrías en el estudio.	64
Figura 17: Configuración del algoritmo generativo y establecimiento de restricciones para el estudio de generación de volumetrías.	65
Figura 18: Interfaz de visualización de generaciones del algoritmo generativo, mostrando las soluciones óptimas para la aprobación, con opción de ordenar por métricas de salida del modelo.	66
Figura 19: Interfaz de visualización de la solución óptima para el terreno seleccionado, con métricas y salidas del modelo detalladas, incluyendo número de pisos, COS, CUS y área total construida.	67
Figura 20: Volumetría final generada por los arquitectos, con métricas de COS, CUS y área total construida.	68

- Figura 21: El diagrama de flujo describe el proceso de generar y validar diseños de planificación de espacios usando DynamoBIM y Project Refinery en Revit. Inicia con la recopilación de datos, sigue con la generación de diseños por algoritmos de ubicación de espacios, los compara con los originales, y si son satisfactorios, realiza una evaluación empírica final para validar la solución. 72
- Figura 22: Representación simplificada de un plano de espacio. Cada espacio corresponde a un polígono independiente con sus etiquetas y sus relaciones. 73
- Figura 23: El diagrama de flujo resume el proceso de generación de diseños de distribución de espacios. Comienza con la extracción de datos del sitio y la generación de objetos de espacios y departamentos. Estos se procesan y organizan para alimentar el algoritmo de ubicación de espacios, que calcula nuevas posiciones. Por último, se evalúan las métricas definidas para el diseño resultante. 82
- Figura 24: Selección del estudio de diseño generativo realizado con la extensión de código utilizando Project Refinery. Se siguieron los pasos numerados. 84
- Figura 25: Interfaz del estudio. Se incluyen los estudios generados, los diseños del generador, los valores de variables, un diseño seleccionado y las métricas de cada diseño. 86
- Figura 26: Distribución de la experiencia en tareas de generación de volumetría entre los arquitectos evaluados. Predomina la experiencia media, aunque también hay arquitectos con experiencia alta y baja. 93
- Figura 27: Frecuencia con la que los arquitectos generan volumetrías de edificios según la encuesta. La planificación se realiza más frecuentemente una, dos o cuatro veces al mes, siendo menos común tres veces al mes. . . 94

- Figura 28: Tiempo ahorrado empleando la solución de planificación de espacios según la encuesta de evaluación empírica. El ahorro de tiempo más común al usar la extensión de código de generación de volumetría fue de 30 a 178 minutos (13 participantes). Además, se reportaron ahorros de 275 a 327 minutos (1 participante) e incluso de 425 a 475 minutos (2 participantes). 96
- Figura 29: Tiempo empleado para aprender a usar la extensión de código de generación de volumetría según la encuesta de evaluación empírica. Se observa que el tiempo de aprendizaje se concentra entre los 5 a 15 minutos. 97
- Figura 30: Percepción de la facilidad de uso de la extensión de código para la generación automática de volumetrías. En ella 14 participantes concuerdan que la extensión es intuitiva y fácil de utilizar, mientras que 2 de ellos no. 98
- Figura 31: Cumplimiento de las expectativas de la extensión de código para la generación automática de volumetrías según la encuesta de evaluación empírica. En su mayoría las expectativas fueron cumplidas aunque 2 arquitectos experimentados esperaron resultados más complejos que no pudieron ser cubiertos por la extensión de este trabajo. 99
- Figura 32: Cumplimiento de la normativa de construcción de Cuenca de la solución de generación de volumetrías según la encuesta de evaluación empírica. Los arquitectos encuestados consideran que dichas restricciones se cumplen en una alta medida. 100
- Figura 33: Satisfacción con la maximización de COS y CUS en la generación de volumetrías según la encuesta. En general, los arquitectos están totalmente de acuerdo con los resultados obtenidos. 101
- Figura 34: Satisfacción con la extensión de código en general según la encuesta de evaluación empírica. En general, gran parte de los arquitectos se encuentran satisfechos con la extensión de código. 102

- Figura 35: Cantidad de personas que recomendarían la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Todos los arquitectos concuerdan en recomendar la extensión de código a otros profesionales. 103
- Figura 36: Relación entre la experiencia de los arquitectos y el tiempo que ahorran empleando la extensión de código. Se destaca que los arquitectos con un nivel medio de experiencia ahorran un mayor tiempo pese a que sus problemas de diseño tienen menos restricciones que la de los arquitectos con mayor experiencia. 104
- Figura 37: Prueba realizada de la extensión de código. El diseño generado obtuvo un 44 % de porcentaje de área utilizada, un 75 % de porcentaje de espacios colocados y un tiempo de generación y selección de 54 minutos.105
- Figura 38: Prueba realizada de la extensión de código. El diseño generado obtuvo un 60 % de porcentaje de área utilizada, un 80 % de porcentaje de espacios colocados y un tiempo de generación y selección de 1 hora y 12 minutos. 106
- Figura 39: Prueba realizada de la extensión de código. El diseño generado obtuvo un 54 % de porcentaje de área utilizada, un 70 % de porcentaje de espacios colocados y un tiempo de generación y selección de 45 minutos.106
- Figura 40: Distribución de los niveles de experiencia en planificación de espacios de los arquitectos que participaron en la evaluación empírica. La mayoría tenía un nivel de experiencia alto, pero también hubo una presencia considerable de arquitectos con experiencia media y baja. 108
- Figura 41: Frecuencia con la que los arquitectos generan planificaciones de espacios en sus proyectos según la encuesta de evaluación empírica. Se indica que las planificaciones se realizan más frecuentemente una o dos veces al mes, mientras que una frecuencia mayor es menos común 109

- Figura 42: Tiempo ahorrado empleando la solución de planificación de espacios según la encuesta de evaluación empírica. El ahorro de tiempo más común al usar la extensión de código de planificación de espacios fue de 30 a 170 minutos (9 participantes). Además, se reportaron ahorros de 170 a 195 minutos (5 participantes) e incluso de 445 a 515 minutos (2 participantes). 110
- Figura 43: Tiempo empleado para aprender a usar la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Se observa que el tiempo de aprendizaje se concentra entre los 5 a 20 minutos. 111
- Figura 44: Percepción de la facilidad de uso de la extensión de código para la planificación de espacios. Todos los participantes concuerdan en que la extensión es intuitiva y fácil de utilizar. 112
- Figura 45: Cumplimiento de las expectativas de la extensión de código para la automatización de la planificación de espacios según la encuesta de evaluación empírica. En su mayoría las expectativas fueron cumplidas aunque 3 arquitectos experimentados esperaron resultados y métricas avanzadas que fueron recortadas de los alcances de la extensión debido a su complejidad. 113
- Figura 46: Cumplimiento de las restricciones y requerimientos establecidos en el documento de planificación por parte de las soluciones devueltas por la extensión de código para la planificación automática de espacios según la encuesta de evaluación empírica. Los arquitectos encuestados consideran que dichas restricciones se cumplen en una medida entre alta y media. 114
- Figura 47: Satisfacción con el área ocupada, porcentaje de espacios colocados y adyacencia de los diseños entregados por la extensión de código para la planificación de espacios según la encuesta de evaluación empírica. En general, los arquitectos mantienen una posición neutral sobre la satisfacción de los resultados obtenidos. 115

Figura 48: Satisfacción con la extensión de código de planificación de pisos según la encuesta de evaluación empírica. Gran parte de los arquitectos se encuentran satisfechos con la extensión de código, aunque se tienen posiciones neutrales por parte de una parte del estudio. 116

Figura 49: Cantidad de personas que recomendarían la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Todos los arquitectos concuerdan en recomendar la extensión de código a otros profesionales. 117

Índice de tablas

Tabla 1: Preguntas formuladas para la encuesta de evaluación empírica empleada a los miembros del estudio de arquitectura “A y D Estudio”.	71
Tabla 2: Configuración de las entradas del programa para el estudio generativo. .	85
Tabla 3: Configuración de los objetivos, los cuales se optimizan simultáneamente en cada iteración, y la generación del programa para el estudio generativo. .	85
Tabla 4: Preguntas formuladas para la encuesta de evaluación empírica empleada a los miembros del estudio de arquitectura “A y D Estudio”.	90
Tabla 5: Tabla final para realizar la comparación entre los resultados reales y los generados por la extensión de código.	92

Agradecimientos

Quiero expresar mi profundo agradecimiento a mis padres, María y César, por su apoyo incondicional a lo largo de toda mi vida estudiantil. A mis queridos hermanos, Christian y Joaquín, por ser una fuente constante de motivación en mi jornada estudiantil. También reconozco el respaldo fundamental de toda mi familia durante este recorrido. A mi tutor y profesor, Jaime Veintimilla, le agradezco por sus valiosas enseñanzas y orientación a lo largo de estos años. A mis compañeros y amigos, quienes han compartido conmigo esta experiencia y han sido un pilar en mi camino. Dedico este trabajo a todos ellos como muestra de mi más sincero agradecimiento. Sin su ayuda y apoyo, este logro no habría sido posible. Eternamente agradecido con todos.

Mateo Sebastian Flores Terreros

Agradecimientos

Con profunda gratitud, quiero agradecer a mis padres, Wilson y Judith, quienes han sido el pilar fundamental y la mayor inspiración de este proceso. A mis hermanos Steven y Matias, y a mi amada abuela Chochi, les doy las gracias por su apoyo incondicional y constante.

Mi más sincero agradecimiento a mi tutor y profesor, Jaime Veintimilla, cuya valiosa orientación ha sido clave a lo largo de mi carrera. Y a mis compañeros y amigos, con quienes he compartido esta maravillosa experiencia, superando juntos cada uno de los desafíos que se nos han presentado.

Dedico este logro a todos ustedes, como muestra de mi más sincero y profundo agradecimiento. Sin su ayuda y apoyo, este sueño no habría podido hacerse realidad.

Wilson Renato Berrezueta Siguenza

1. Introducción

1.1. Contexto y Motivación

La Inteligencia Artificial (IA) ha desempeñado un papel fundamental en la transformación de diversos sectores, entre ellos la arquitectura y la construcción. En los últimos años, estas áreas se han visto gratamente beneficiadas por la IA debido a la adaptación y uso de frameworks enfocados en recuperar la mayor cantidad de información posible dentro del proceso de diseño y edificación de las construcciones (Ismail Agha, 2019). Específicamente, la tendencia se ha visto marcada por el framework Building Information Model (BIM). Este framework proporciona una representación digital de las características físicas y funcionales de un edificio (Onur and Nouban, 2019).

En particular, la aplicación de la IA en el diseño de edificaciones utilizando el framework BIM ha sido destacada. Estas edificaciones han respondido a las necesidades específicas de los usuarios en cuestiones como la planificación y distribución de espacios de planta, la generación de la estructura de edificaciones, y la optimización de uso de materiales y energía eléctrica (Chaillou, 2020). Este fenómeno se ha materializado a través de la asistencia proporcionada a arquitectos en la generación y evaluación de opciones de diseño, considerando criterios cruciales como accesibilidad, estética y eficiencia energética (Dunaevskiy et al., 2022).

Es por esto que la exploración de técnicas de diseño generativo ha cobrado relevancia. A través de estos enfoques, los arquitectos encargados de las etapas de diseño de proyectos han establecido parámetros, restricciones y métodos de evaluación, que han permitido que los modelos generen soluciones óptimas que se evalúan y seleccionan basándose en el criterio del propio diseñador (Cudzik and Radziszewski, 2018). Para este caso se han diferenciado dos áreas necesarias, la división de espacios de planta de edificios y la generación de la volumetría de los mismos, teniendo en cuenta la superficie de construcción. Esta última hace referencia a las formas tridimensionales que conforman la estructura y el volumen total del edificio.

En estas áreas se han realizado investigaciones exhaustivas sobre las tendencias referentes a técnicas para el diseño generativo.

Específicamente, para la planificación de los espacios de planta en edificios, otros investigadores como Aalaei et al. (2023), Gan (2022) y Zhang et al. (2021) han desarrollado algoritmos y sistemas prototipo para la transformación de datos de modelos digitales y la generación automática de espacios de planta en edificaciones modulares y residenciales. Aunque los resultados de diseño conseguidos no han sido evaluados empíricamente para conocer su calidad y capacidad. En estas investigaciones, los datos de los edificios se han representado como grafos que son usados en modelos generativos y de optimización.

Estas investigaciones aportan con diferentes enfoques y herramientas para generar distribuciones de espacios automáticamente dentro de un área delimitada. Aunque, se cuenta con la limitación de la adaptabilidad al software Autodesk DynamoBIM el cual es indispensable para la conexión con software BIM como Revit. Frente a esta problemática, Das et al. (2016) presentan una metodología integral para la distribución automática de espacios mediante el uso del software DynamoBIM, partiendo de un archivo de texto que detalla las áreas a ser ubicadas. Esta metodología muestra capacidad para la generación automatizada de planos de espacio dentro del perímetro del sitio de construcción. Además, permite generar la volumetría de edificios a partir del espacio utilizado, y crear las correspondientes métricas de evaluación. Cabe mencionar que esta metodología no incluye la validación empírica de los resultados, ni la implementación de algoritmos de diseño generativo.

Dentro del diseño automático de volumetrías de edificios, el estudio de Zheng et al. (2021) propone una metodología basada en redes neuronales artificiales para la generación de diseños arquitectónicos y urbanos, centrándose específicamente en la generación de torres. Sin embargo, se sugiere que la metodología podría adaptarse para la creación de otros tipos de edificaciones y normas de construcción, aunque existen limitaciones inherentes respecto a la adaptabilidad de esta metodología al software DynamoBIM.

Además, Azadi et al. propone tres procedimientos principales: Planificación, Configuración y Modelamiento. La Planificación aborda las complejidades multiactores y multicriterio para especificar colaborativamente las relaciones y criterios de los espacios. La Configuración se

enfoca en generar una configuración de espacios que satisfaga lo establecido en la etapa previa, centrándose en las complejidades multidimensionales y multicriterio. Finalmente, el Modelamiento concretiza la geometría de la configuración topológica, tratando las complejidades multidimensionales y multivalor del diseño. Cada procedimiento utiliza formalizaciones matemáticas específicas que permiten aplicar una amplia gama de enfoques, desde optimización hasta juegos gamificados.

Una de las principales limitaciones del marco es que prioriza las decisiones abstractas sobre las concretas, lo cual dificulta llegar a soluciones de diseño específicas. Además, su enfoque matemático requiere que los criterios espaciales sean cuantificables, lo cual puede ser un desafío para integrar aspectos no cuantificables relacionados con factores humanos.

1.2. Planteamiento del problema

Tomando en cuenta que, a pesar de la disponibilidad de herramientas comerciales y extensiones de código que aplican las técnicas de diseño generativo, estas se limitan a reglas específicas de los diferentes países en las cuales se han realizado las investigaciones. Y como marca Chavarrea et al. las políticas locales han limitado los diseños arquitectónicos a las condiciones establecidas por las gobernaciones.

En el entorno arquitectónico de Cuenca, Ecuador, la convergencia de la Inteligencia Artificial (IA) y el framework Building Information Model (BIM) ha ofrecido perspectivas innovadoras para la automatización del diseño de edificaciones. Sin embargo, el impacto directo de las políticas locales en la configuración arquitectónica ha suscitado desafíos considerables. En específico, según la Dirección General de Planificación Territorial Cuenca (2022) se definen normativas de diseño como los retiros de la planta baja y del primer piso, el ancho mínimo de los pasillos y el número mínimo de parqueaderos con relación a la edificación; estas normas cambian según la ubicación de la edificación, ya sea en zonas rurales, urbanas o centro de la ciudad, estas normas hacen que las edificaciones en Cuenca tengan una volumetría muy específica para la ciudad.

En este contexto, se identifica la falta de herramientas específicas y adaptadas a las normativas locales de la ciudad de Cuenca en Ecuador, como un obstáculo clave. La desconexión

entre la innovación tecnológica y las restricciones regulatorias locales ha generado un vacío en la capacidad de los arquitectos para aprovechar plenamente las ventajas de la IA y el diseño generativo. La pregunta central que guía esta investigación es cómo superar esta brecha y facilitar la implementación efectiva de estas tecnologías en la configuración arquitectónica de Cuenca. La respuesta a este interrogante no solo implica el desarrollo de extensiones de código adaptadas, sino también la exploración de soluciones que reconcilien la creatividad arquitectónica con las regulaciones locales para lograr automatizar la creación de edificaciones contextualmente pertinentes.

1.3. Solución propuesta

La solución propuesta en este estudio consiste en el desarrollo de extensiones de código específicas, adaptadas a las normativas locales de Cuenca, Ecuador, para facilitar la aplicación efectiva de la Inteligencia Artificial (IA) mediante técnicas de diseño generativo en el proceso arquitectónico. Estas extensiones se integran con el software de diseño arquitectónico Autodesk Revit y su complemento Autodesk DynamoBIM utilizando piezas de código en el lenguaje de programación Python y clases en lenguaje C#, permitiendo a los arquitectos que colaboran con este trabajo de titulación aprovechar plenamente las capacidades de la IA a través del diseño generativo mientras se ajustan a las regulaciones locales.

En la fase inicial, se lleva a cabo un análisis de la información regulatoria de las construcciones de la ciudad de Cuenca con el fin de delimitar las restricciones especificadas para cada zona y espacio de la ciudad. La información fundamental se tabula de manera que se resumen las restricciones reglamentarias del diseño. Estas tablas de datos son las entradas para un flujo de trabajo encargado de la generación aleatoria de la estructura y volumetría de las edificaciones, teniendo en cuenta parámetros topográficos del terreno, número de pisos deseados y la ubicación específica del edificio en la ciudad de Cuenca. El generador de diseños se complementa con la aplicación de diseño generativo utilizando algoritmos genéticos para la búsqueda de las soluciones candidatas. Por ende, el problema recae en un caso de optimización multivariable, donde los objetivos corresponden a maximizar o minimizar las métricas de evaluación de los diseños generados según lo deseado. Estas métricas incluyen el Coeficiente de Ocupación del Suelo (COS), que indica el porcentaje de ocupación

de la primera planta respecto al terreno disponible, el Coeficiente de Utilización del Suelo (CUS), que representa el porcentaje total de ocupación de la edificación respecto al terreno potencial, y el área provisional para ascensor, la cual muestra la desventaja de emplear más pisos de los óptimos. Debido a los retiros establecidos, no es posible ocupar la totalidad del terreno en ninguno de los pisos, por lo que el objetivo es maximizar la ocupación de la edificación a través de estas métricas, con el fin de poder evaluar la rentabilidad de la solución.

Además, con base en la metodología propuesta por Das et al. (2016) se implementa la planificación de espacios bajo un documento de planificación donde constan los requerimientos de número de zonas y espacios en la planta, su área y en casos específicos sus dimensiones exactas. Esta información se organiza para formar una estructura de datos que permita la gestión de las geometrías generadas.

El generador de diseños se fundamenta en la metodología propuesta por Espín et al. (2020), la cual utiliza un algoritmo codicioso para resolver el problema de ubicación de polígonos irregulares dentro de un espacio contenedor. Un algoritmo codicioso toma decisiones en cada paso basándose en la mejor opción disponible en ese momento, buscando una solución óptima de manera eficiente. En este contexto, el algoritmo selecciona iterativamente la posición más adecuada para cada polígono, minimizando el espacio desperdiciado y maximizando el uso del área disponible. El autor combina los algoritmos con el problema de la mochila (Knapsack) con el objetivo de ubicar la totalidad de los polígonos sin exceder el área del espacio contenedor, y maximizando el peso total acumulado de dichos polígonos.

El generador posteriormente se emplea en el estudio de diseño generativo basado en algoritmos genéticos, este tiene como objetivo determinar la disposición óptima de los espacios dentro del sitio de planificación considerando los obstáculos establecidos. Para llevar a cabo dicha optimización, se utilizaron métricas específicas que incluyeron el porcentaje de ocupación de espacio del sitio, el número de espacios colocados, una métrica combinada integrando el valor de prioridad y la adyacencia de los espacios, así como una métrica de distancia promedio entre los espacios y los obstáculos predefinidos, tales como salidas, entradas, columnas y ascensores, para así reducir la distancia de circulación dentro de la

planta.

De esta manera, el objetivo principal de estas soluciones es conciliar la creatividad arquitectónica con las restricciones regulatorias, permitiendo a los profesionales encargados de la etapa de diseño generar propuestas estéticas, optimizadas, contextualmente pertinentes en el marco urbano de Cuenca y en un tiempo mucho más bajo de lo habitual. La efectividad de esta solución se aprobará mediante la comparación visual de los diseños y de las métricas de evaluación de los modelos generados con las extensiones de código y aquellos previamente diseñados por arquitectos.

Además, basándose en el trabajo de Bocco et al. para este caso de estudio se opta por un enfoque empírico de evaluación mediante encuestas dirigidas al equipo de arquitectos colaboradores del proyecto. Esto se debe a que, en la literatura no se cuentan hasta el momento con propuestas que evalúen la incidencia de las soluciones en un ambiente real de aplicación. La evaluación empírica, como indica Bocco, es un enfoque ideal para este tipo de herramientas novedosas. Vale la pena aclarar que solo el estudio “A y D Estudio” estuvo dispuesto no solo a probar la solución una vez lista, sino a realizar el proceso completo de evaluación propuesto en este trabajo, el cual tiene una alta demanda de tiempo y requiere una comunicación continua con el estudio. Y durante la etapa de desarrollo, estuvieron dispuestos a compartir conocimientos y trabajos previos de forma abierta para asegurar una correcta comprensión de la problemática y la selección de métricas para las extensiones de código propuestas. El objetivo es capturar métricas clave como la percepción de facilidad de uso, utilidad e intención de uso. Las preguntas de la encuesta están diseñadas para obtener información sobre la experiencia de los usuarios, la calidad de los resultados generados, el ahorro de tiempo en comparación con los métodos manuales y la adecuación de la herramienta a los requisitos normativos locales. La evaluación se lleva a cabo con los 16 profesionales del estudio de arquitectura, quienes tienen diversos perfiles y niveles de experiencia.

Con esta solución, se busca no solo superar los desafíos regulatorios, sino también promover la innovación y la automatización en el diseño arquitectónico local.

1.4. Objetivos

Aquí se detallan tanto el propósito general como las metas específicas de este proyecto de titulación.

1.4.1. Objetivo general

Mejorar la eficiencia en el proceso de creación y diseño de estructuras y volumetría de edificios y sus espacios internos de planta a través de extensiones de código bajo el software DynamoBIM, centradas en diseño generativo, adaptadas a la normativa y topografía de las gobernaciones de Ecuador, específicamente en la ciudad de Cuenca.

1.4.2. Objetivos específicos

1. Realizar un estudio del estado actual de la investigación respecto a metodologías que apliquen algoritmos de diseño generativo sobre datos de edificios bajo el framework BIM y el software de diseño DynamoBIM.
2. Definir las etapas de una metodología para crear flujos de trabajo de generadores de diseños de volumetría de edificios y planificación de espacios interiores en DynamoBIM, e implementar diseño generativo basado en algoritmos genéticos en dichos flujos de trabajo. Permitiendo así, la generación de múltiples soluciones potenciales con métricas optimizadas.
3. Implementar diseño generativo con base en algoritmos genéticos que tengan en cuenta las limitantes establecidas por las políticas de gobernaciones locales de Ecuador, específicamente en la ciudad de Cuenca.
4. Evaluar las extensiones de código desarrolladas para validar su funcionamiento y usabilidad dentro del ámbito profesional en conjunto con el estudio de arquitectos que apoyan este trabajo de titulación.

1.5. Estructura del trabajo

Este trabajo de titulación se estructura de la siguiente manera. En la sección 2 se presenta el marco teórico donde se detallan conceptos relacionados con el área de arquitectura, específicamente al uso de frameworks para la gestión de edificios, y el uso de su informa-

ción para aplicaciones de IA. En la sección 3 se indica el estado del arte de la aplicación de técnicas de IA para la planificación de espacios de planta y la generación de volumetrías de edificaciones. En la sección 4 se indica la metodología, en ella se especifican las representaciones de información utilizadas, los procesos seguidos y las técnicas implementadas para dar forma a las extensiones de código planteadas y como se evaluaron. Además, se especifican las variables, restricciones y métricas necesarias para la implementación de las extensiones de código en el programa de diseño arquitectónico Autodesk DynamoBIM y su uso con el programa de diseño generativo Project Refinery en Autodesk Revit. En la sección 5 se presentan los resultados, en la sección 6 se discuten los resultados y se da enfoque a los beneficios y dificultades observadas tras la evaluación empírica del funcionamiento de las extensiones de código y la calidad de sus soluciones. Finalmente, en la sección 7 se detallan las conclusiones.

2. Marco Teórico

A continuación se presentan los principales conceptos teóricos para entender los diferentes componentes que constituyen el trabajo de titulación.

2.1. Building Information Modeling (BIM)

Según el National BIM Standard. , BIM es un framework para el desarrollo de diseños y proyectos arquitectónicos que se basa en la representación digital de las características físicas y funcionales de una edificación. Es un recurso de conocimiento compartido entre profesionales para obtener información sobre una construcción que constituye una base de alta fiabilidad para el proceso de toma de decisiones a lo largo del ciclo de vida de la construcción. Una premisa básica del BIM es la colaboración entre diferentes partes interesadas en diferentes fases del ciclo de vida, permitiendo insertar, extraer, actualizar o modificar información del modelo para respaldar y reflejar los roles de cada parte involucrada en el proyecto.

Originalmente, el concepto de BIM nace en 1974, con el trabajo de Eastman et al. , en su trabajo titulado Sistema de Descripción de Edificios. En él destaca la dependencia de los dibujos en planos en el diseño, la construcción y la operación de edificios, con la intención de reemplazarlos con un sistema informático para el almacenamiento y manipulación detallados de la información de diseño de edificios.

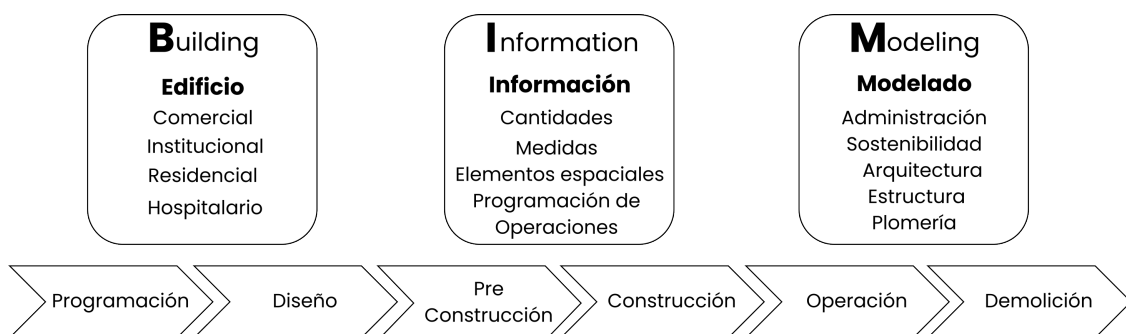


Figura 1: Representación del concepto BIM. Fuente: Adaptado de Salman et al. (2012).

BIM es un framework elemental para arquitectos y diseñadores de los proyectos, según Hardin. Hardin (2009), ellos pueden aprovechar el BIM para realizar estimaciones de cantidad y

costos, la identificación temprana de errores de diseño mediante la detección de conflictos, la planificación de la construcción y el análisis de constructibilidad. Además de la verificación en el lugar, la orientación y el seguimiento de las actividades de construcción. La prefabricación y modularización fuera del sitio, la planificación de la seguridad en el lugar, la ingeniería de valor y la implementación de conceptos de construcción eficientes, así como una comunicación mejorada con el propietario del proyecto, el diseñador, los subcontratistas y los trabajadores en el sitio. A través de estas aplicaciones, los constructores pueden obtener beneficios significativos, como una mayor rentabilidad, mejor servicio al cliente, compresión de costos y plazos, mayor calidad de producción, toma de decisiones más informada y una planificación y gestión de la seguridad mejoradas.

En la Figura 2 se aprecia un resumen del uso del BIM durante todo el ciclo de vida de una edificación, incluyendo etapas clave como la planificación, el diseño, el costo y el mantenimiento.

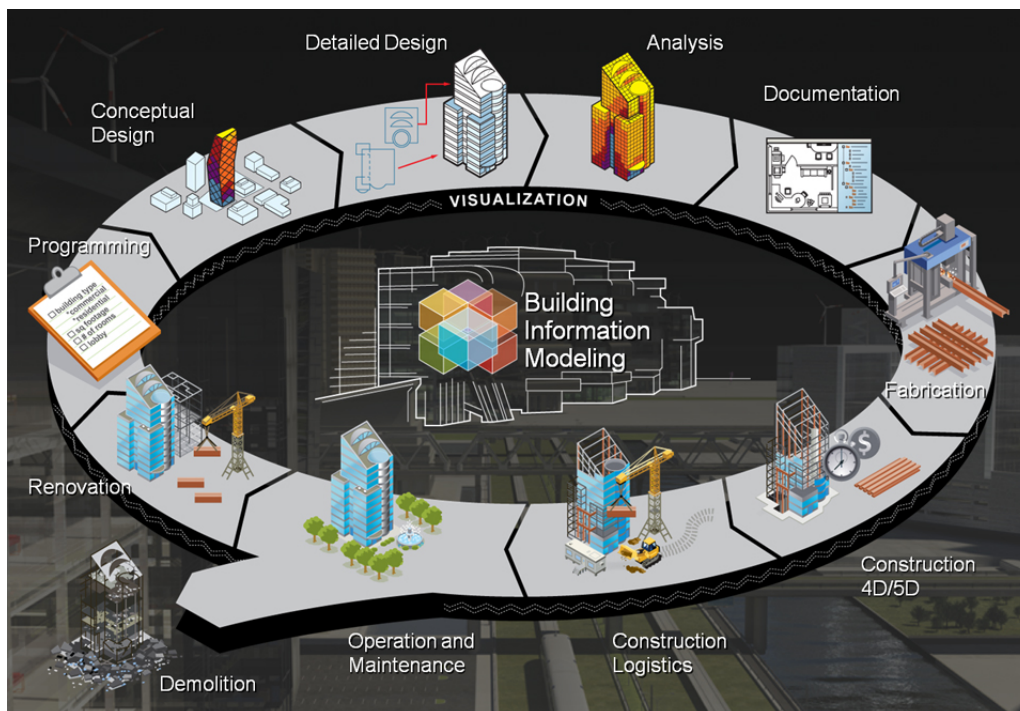


Figura 2: Ciclo de vida de una edificación con BIM. Fuente: Adaptado de BIM (2015).

Los diferentes software BIM que se ofrecen en el mercado (por ejemplo, Revit o ArchiCAD, entre otros) ofrecen adicionalmente una Interfaz de Programación de Aplicaciones (API) que permite acceder, extraer, seleccionar y modificar la información de las edificaciones

que van desde la estructura y materiales utilizados, hasta los complementos para el diseño de interiores (muebles, puertas, decoración, entre otros).

Además, la API permite la extensión de las funcionalidades del software a través del uso de interfaces de programación para la generación de flujos de trabajo propios.

2.1.1. Autodesk Revit

Autodesk Revit es una herramienta que permite a los arquitectos, ingenieros y diseñadores trabajar dentro de un entorno colaborativo durante la creación de modelos tridimensionales de edificaciones y sus interiores Autodesk (2024). Aunque, está disponible únicamente para sistemas operativos Windows de Microsoft.

Aunque es un software de paga, se disponen de licencias gratuitas para el uso de sus productos con fines de investigación.

Este software permite la creación de planos arquitectónicos detallados hasta la simulación de la construcción. Además, destaca por estar basado en el framework BIM, por lo que los elementos del proyecto se encuentran interconectados y se actualizan en tiempo real.

El funcionamiento de Revit se basa en la creación de elementos como muros, losas, ventanas, puertas, entre otros, los cuales se modelan y analizan tanto en el plano bidimensional como el tridimensional.

Además, Revit ofrece la expansión de su funcionamiento a través de extensiones de programa que permiten al usuario codificar diferentes procesos dentro de sus proyectos. Esto agiliza de manera significativa el flujo de trabajo.

Para ello, se ofrece el software DynamoBIM el cual cuenta con una amplia gama de nodos y bibliotecas de terceros que permiten tareas que van desde el análisis de datos hasta la generación de geometrías.

Además, brinda a los usuarios expertos la capacidad de personalizar y crear sus propias herramientas mediante la programación visual, o la programación de módulos en lenguaje Python o C++ para mejorar las capacidades de Revit y optimizar sus flujos de trabajo (AUGI, AUGI).

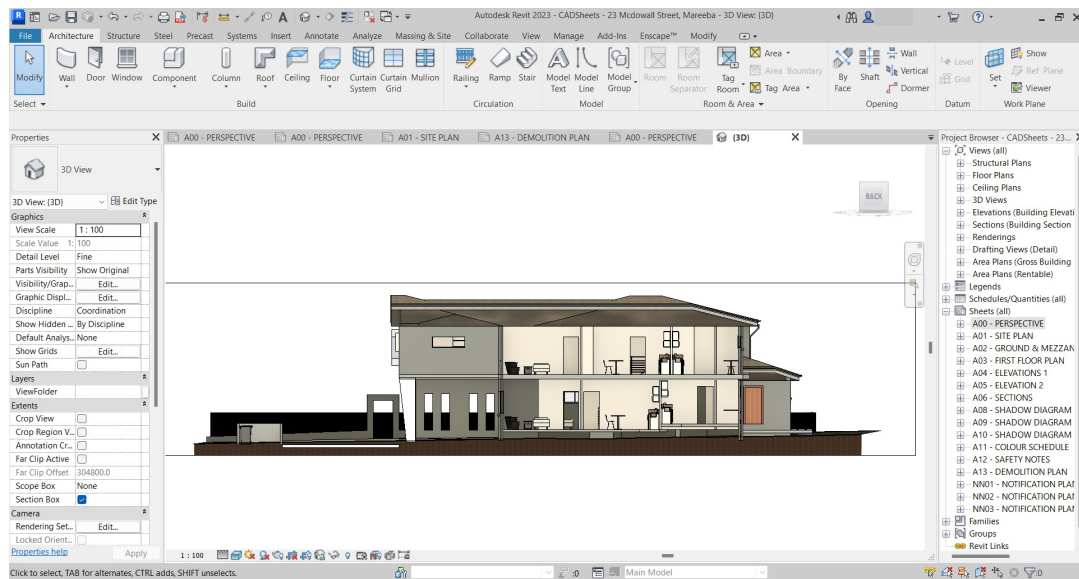


Figura 3: Interfaz del software Revit. Fuente: Adaptado de Sheets (2023).

2.1.2. Autodesk DynamoBIM

DynamoBIM es una plataforma de software de programación visual diseñada específicamente para la industria de la construcción. Es un proyecto de código abierto concebido para permitir a los usuarios programadores y no programadores crear flujos de trabajo complejos y automatizar tareas repetitivas a través de bloques de código acoplables. Gracias a su interfaz gráfica, se facilita la creación de algoritmos y lógica de programación.

La herramienta se integra perfectamente con otros software BIM y otros productos de Autodesk. Por lo que, permite a los profesionales de la construcción, como arquitectos, ingenieros y gestores de proyectos, optimizar sus procesos de diseño, análisis y documentación. Mediante la definición de flujos de trabajo personalizados, DynamoBIM posibilita la automatización de tareas.

Además, ofrece una biblioteca de nodos y paquetes que amplían sus capacidades a través de clases y métodos específicos para diseño o gestión de construcciones.

Por otro lado, ofrece la ampliación de las funcionalidades a través de código en lenguaje Python. Para ello, se utiliza el intérprete integrado CPython que utiliza la versión de Python 3, y permite mover datos entre objetos .NET y Python.

De igual forma, se pueden realizar paquetes y nodos completos utilizando el lenguaje de

programación C#. Este enfoque se denomina Zero-Touch en DynamoBIM y permite importar incluso bibliotecas que no necesariamente se desarrollaron para este software.

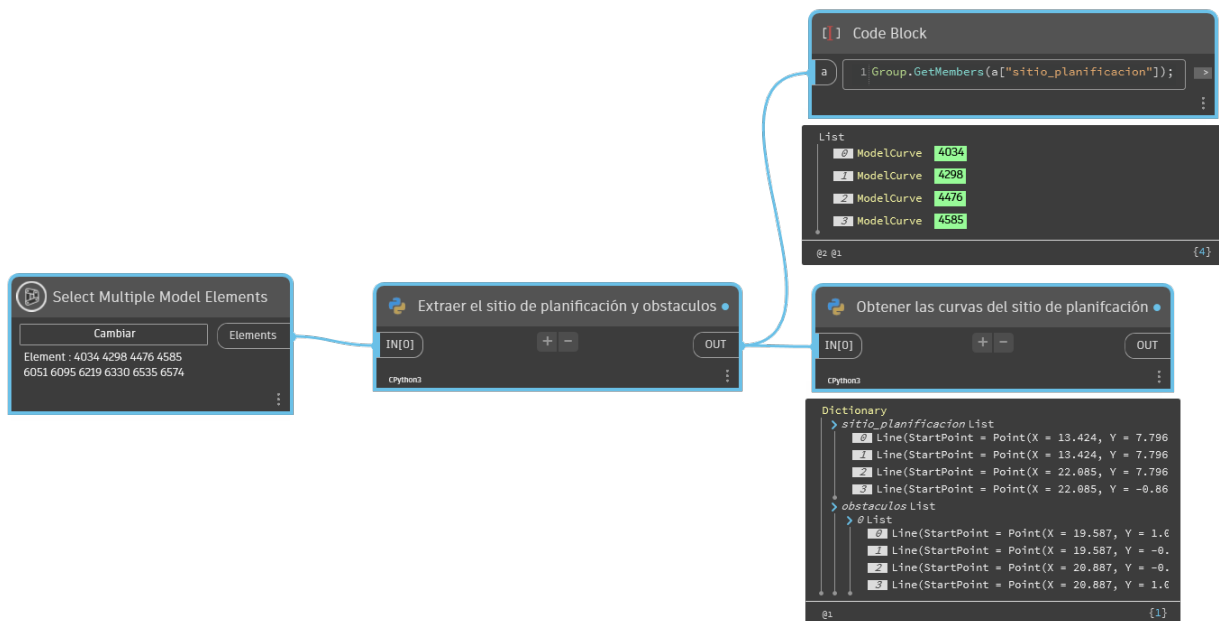


Figura 4: Ejemplo de flujo de trabajo en DynamoBIM. Fuente: Elaboración propia.

La programación visual en DynamoBIM se sustenta sobre una estructura modular interconectable. Los elementos básicos que forman esta arquitectura son los nodos, los cables y los puertos de entrada y salida.

Los nodos representan la unidad básica de funcionamiento. Cada nodo representa una tarea, función o acción específica que puede ser utilizada en el flujo de trabajo. A su vez, los nodos se configuran y parametrizan para lograr los resultados deseados por el usuario.

Los cables, por su parte, establecen la conexión entre los nodos. Así pues, representan el flujo de datos, permitiendo que la información se transfiera de un nodo a otro. De esta manera, se construye la lógica algorítmica que vertebra el proceso de automatización.

Los puertos de entrada y salida de los nodos juegan un papel fundamental en el intercambio de datos. Los puertos de entrada reciben los datos de entrada que alimentan al nodo, mientras que los puertos de salida emiten los resultados procesados. La correcta interconexión de estos puertos a través de los cables es crucial para asegurar la coherencia y el correcto funcionamiento del flujo de trabajo diseñado.

De manera complementaria, se dispone de una colección de nodos por defecto que ofrece una amplia gama de funcionalidades, desde operaciones matemáticas básicas hasta complejos algoritmos de análisis y generación de geometría.

2.2. Inteligencia Artificial (IA) en Arquitectura y Construcción.

La inteligencia artificial (IA) abarca un amplio espectro de técnicas que pueden ser aplicadas en múltiples campos de la ciencia y la industria. Por ende, el área de la construcción y la arquitectura no se queda atrás. Es así que, la IA se puede aplicar de diversas maneras para simplificar la gestión de proyectos y facilitar la generación de soluciones sostenibles y optimizadas (Petrova, 2019). Por ejemplo, se ha utilizado Deep Learning para la clasificación y optimización de cargas estructurales en geometrías complejas. De esta manera, se usan redes neuronales profundas para manejar una gran cantidad de parámetros de entrada, lo que permite la categorización precisa de la distribución de cargas y la reutilización de soluciones optimizadas en múltiples iteraciones de diseño (Tamke et al., 2018).

Así mismo, la IA puede automatizar tareas repetitivas, optimizar la asignación de recursos, ayudar en la estimación de costos, selección de materiales, optimización de energía, y proporcionar información precisa y oportuna para la planificación de proyectos (Stanojević et al., 2023).

Según Sacks et al. (2020), los modelos BIM de edificios e infraestructura son adecuados para la manipulación con herramientas de software inteligentes que incorporan visión por computadora, inferencia de reglas, aprendizaje automático, razonamiento basado en casos y otras estrategias de IA.

La integración de IA en entornos BIM da lugar al desarrollo de herramientas inteligentes para el diseño y gestión de la construcción, medición en tiempo real, evaluación e interpretación del estado del proyecto, aseguramiento y control de calidad, así como el control de producción. Además, la IA puede proporcionar análisis avanzados y capacidades predictivas, lo que a su vez permite una mejor toma de decisiones y optimización de los procesos de construcción (Pan, 2022).

Las aplicaciones potenciales para integrar BIM e IA son diversas e incluyen la planificación

y el control de seguridad, la sostenibilidad y las evaluaciones de costos del ciclo de vida, la adquisición de modelos BIM a partir de datos de nubes de puntos, así como las operaciones y administración de instalaciones utilizando gemelos digitales (Sacks et al., 2020).

No obstante, con base en las investigaciones de Harris (2021), la aplicación de la inteligencia artificial en el campo de la arquitectura, a través del diseño generativo, está experimentando un notable auge. Este enfoque se considera una técnica adaptable de optimización que está transformando la manera en que los diseñadores abordan sus proyectos. El diseño generativo implica que un diseñador establezca una serie de requisitos para un proyecto, a partir de los cuales el software de diseño genera múltiples soluciones potenciales. Esto brinda una mayor flexibilidad y adaptabilidad en el proceso de diseño, ya que no se limita a devolver una sola solución óptima, sino que se dispone de un conjunto de soluciones que buscan ser lo más próximas al frente de Pareto el cual hace referencia al conjunto de soluciones óptimas en el sentido de que se equilibran los diferentes objetivos que se están optimizando.

No obstante, es crucial resaltar que las personas siguen siendo responsables de la toma de decisiones y del trabajo en las demás fases del ciclo de vida de la edificación. El sistema de inteligencia artificial simplemente ayuda a comprender el impacto de dichas decisiones, lo que significa que el factor humano continúa siendo fundamental en el proceso de diseño y construcción.

2.3. Diseño Generativo en Arquitectura.

Bohnacker et al. (2009) en su libro “Diseño Generativo: Diseñar, Programar, Visualizar” define el diseño generativo como un proceso cíclico basado en una idea abstracta que se aplica a través de un algoritmo, el cual es traducido a código fuente y genera salidas en serie.

Los inicios de este enfoque se dieron a finales de la década de 1980 con el trabajo del arquitecto Peter Eisenman, quien aplicó técnicas influenciadas por la teoría de la deconstrucción. A medida que las herramientas computacionales se desarrollaron, Eisenman incorporó nuevas técnicas como la morfología de imágenes. Posteriormente, otros arquitectos como Greg Lynn e instituciones como UNstudio también contribuyeron al campo del diseño generativo,

utilizando avances en animación y superficies polisómicas isomórficas, que son superficies con propiedades topológicas similares a pesar de tener diferentes formas geométricas (Agkathidis, 2015).

La figura 5 describe el diseño generativo como un proceso cíclico que depende de la retroalimentación entre el diseñador y el sistema de diseño.

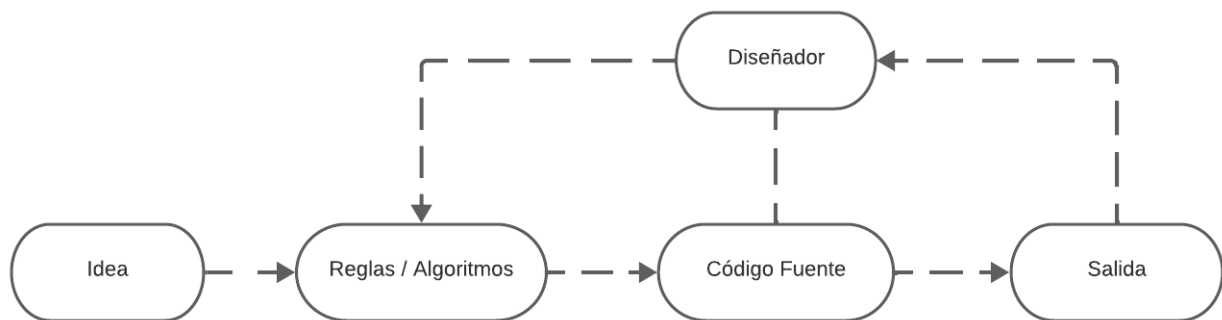


Figura 5: Proceso de diseño generativo. Inicia con la idea del diseño a optimizar, el diseñador escribe el código fuente para el generador de soluciones, y estas son evaluadas por el mismo diseñador. Fuente: Adaptado de Bohnacker et al. (2009).

Actualmente, el diseño generativo en arquitectura implica aprovechar la potencia computacional, especialmente a través del uso de algoritmos evolutivos como los algoritmos genéticos, para buscar soluciones potenciales de diseño, teniendo en cuenta las restricciones del proyecto y los objetivos del diseñador. Este enfoque permite explorar un rango más amplio de posibilidades de diseño y generar soluciones innovadoras que podrían no ser factibles mediante métodos tradicionales (Smorzhenkov, Nikita and Ignatova, Elena, 2021).

Aunque, también se han experimentado con redes neuronales generativas adversarias para la generación de conjuntos de soluciones para proyectos debido a su capacidad para capturar las características y estilos implícitos de los conjuntos de datos de entrada y generar soluciones coherentes. Sin embargo, como menciona Zhuang et al. (2023) este enfoque tiene mucho potencial, pero las aplicaciones actuales se ven limitadas por la falta de datos de entrenamiento de alta calidad, la falta de representaciones precisas de la información de los proyectos y la dificultad para generar soluciones de diseño viables.

2.3.1. Algoritmos Genéticos

Los algoritmos genéticos son una técnica de optimización inspirada en el proceso de la evolución natural. Estos algoritmos buscan encontrar soluciones óptimas a problemas complejos mediante la simulación de la selección natural y la genética. El principio fundamental es que, a partir de una población inicial de posibles soluciones, se aplican operadores de selección, cruce y mutación, de manera iterativa, para generar nuevas generaciones que se aproximen cada vez más a la solución óptima (Katoch et al., 2021).

En cada iteración, se evalúa la “aptitud” de cada individuo de la población, es decir, qué tan bien resuelve el problema. Los individuos más aptos tienen mayor probabilidad de ser seleccionados para reproducirse y pasar sus características a la siguiente generación. Mediante el cruce de los individuos seleccionados (combinación de características o variables de los individuos) y la introducción aleatoria de mutaciones (alteraciones aleatorias de las características o variables de los individuos), se genera una nueva población que hereda tanto las características exitosas como la diversidad necesaria para explorar el espacio de soluciones.

Este proceso iterativo permite que la población evolucione gradualmente hacia soluciones cada vez mejores. A medida que avanzan las generaciones, los individuos de la población se van adaptando cada vez más al problema, convergiendo hacia la solución óptima o un conjunto de soluciones cercanas a la óptima. Para ello, se recurre a la utilización de ciertos criterios de parada para poner fin al proceso de optimización.

Los principales criterios de parada comúnmente utilizados incluyen alcanzar un número máximo de generaciones, lograr la convergencia de la población hacia individuos similares, evidenciar un estancamiento en la mejora de la aptitud de los individuos, agotar un tiempo máximo de ejecución predefinido, o alcanzar un umbral de aptitud objetivo (Katoch et al., 2021).

Una vez que se cumple cualquiera de estos criterios, el algoritmo genético finaliza y se procede a utilizar la mejor solución encontrada, la cual generalmente corresponde al individuo con la mayor aptitud dentro de la población final. Este resultado representa la mejor aproxi-

mación obtenida al problema de optimización planteado, de acuerdo con los parámetros y el diseño del algoritmo aplicado.

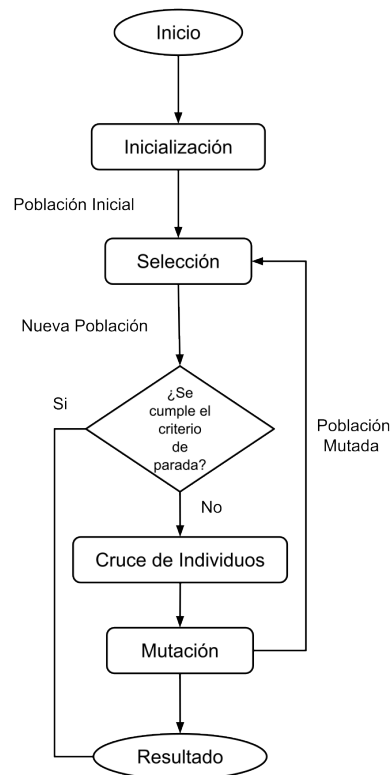


Figura 6: Esquema general de un algoritmo genético. Inicia con una población inicial de sujetos. Itera seleccionando los individuos más aptos, aplicando cruce y mutación para generar la siguiente generación, hasta que se cumplen los criterios de parada, obteniendo una solución mejorada. Fuente: Adaptado de Dastanpour et al. (2014).

La primera fase del proceso corresponde a la Inicialización (Initialization). En esta etapa fundamental, se define cuidadosamente el conjunto de parámetros y restricciones que delimitarán el espacio de diseño a explorar. Esto incluye la especificación precisa de los objetivos de rendimiento deseados, las limitaciones físicas y constructivas aplicables, así como cualquier otra variable relevante que guiará el posterior proceso generativo. Adicionalmente, se establecen los valores iniciales para estos parámetros, dando lugar a una población inicial de soluciones potenciales, las cuales servirán como punto de partida para el algoritmo (Dastanpour et al., 2014).

Posteriormente, en la fase de Evaluación (Evaluation), las alternativas de diseño generadas son sometidas a un proceso de valoración automatizada. Mediante el uso de modelos

computacionales o simulaciones avanzadas, se cuantifica el desempeño de cada solución en función de los objetivos y restricciones definidos en la fase de inicialización. Este procedimiento de evaluación genera una puntuación o aptitud que refleja el grado de cumplimiento o aptitud de los requisitos establecidos para el problema de diseño (Dastanpour et al., 2014).

Una vez evaluadas las soluciones, la siguiente fase corresponde a la Selección (Selection). Haciendo uso de mecanismos inspirados en los procesos de selección natural, los algoritmos generativos priorizan y filtran las alternativas más prometedoras, es decir, aquellas que mejor se adaptan a los objetivos y limitaciones definidos previamente. Las soluciones seleccionadas servirán como base para generar nuevas variaciones en la siguiente iteración del proceso (Dastanpour et al., 2014).

Seguidamente, en la fase de Cruzamiento / Reproducción (Crossover / Reproduction), los algoritmos aplican operadores de recombinación que permiten combinar características de diferentes soluciones seleccionadas. Este proceso de cruzamiento da lugar a nuevas alternativas potencialmente mejoradas, explorando de manera sistemática el espacio de posibilidades de diseño (Dastanpour et al., 2014).

Finalmente, la fase de Mutación (Mutation) introduce cambios aleatorios en las características de las soluciones, lo cual permite explorar nuevas regiones del espacio de diseño y evitar que el proceso quede atrapado en óptimos locales. De este modo, se fomenta la diversidad de las alternativas generadas, enriqueciendo el conjunto de soluciones disponibles (Dastanpour et al., 2014).

Por otro lado, existen desafíos fundamentales en este proceso. Uno de ellos radica en la selección de la población inicial, pues esta constituye el punto de partida del proceso evolutivo. La elección de una población inicial adecuada puede tener un impacto significativo en la convergencia y la calidad de las soluciones finales obtenidas (Reeves, 2010).

Otro aspecto crucial es la determinación del número de generaciones a simular. Este parámetro debe equilibrar la necesidad de explorar suficientemente el espacio de soluciones, sin incurrir en un costo computacional excesivo. Asimismo, la selección de la semilla inicial para la generación de números aleatorios puede afectar el comportamiento del algoritmo y

la diversidad de las soluciones exploradas.

Adicionalmente, los algoritmos genéticos pueden presentar dificultades en la optimización de problemas con restricciones complejas o con múltiples objetivos en conflicto. En tales casos, puede ser necesario el uso de técnicas de manejo de restricciones o de algoritmos multiobjetivo para garantizar soluciones factibles y equilibradas.

Por último, la definición adecuada de los operadores de selección, cruce y mutación también constituye un desafío importante. Estos parámetros deben ajustarse cuidadosamente para lograr un balance entre la explotación de soluciones prometedoras y la exploración de nuevas regiones del espacio de búsqueda, evitando así la convergencia prematura a óptimos locales.

2.3.2. Autodesk y el Diseño Generativo

Autodesk trabaja para generar y fomentar tecnologías que ayuden a los diseñadores del área de la construcción en la automatización de sus actividades. Específicamente, en el área de la construcción y diseño se han destinado esfuerzos para integrar soluciones de diseño generativo en sus productos. Como resultado de esto se ha desarrollado Project Refinery, una extensión de software incluida en el paquete AEC (Architecture, Engineering, and Construction) de Autodesk. Esta extensión adapta el algoritmo genético NSGA II para la optimización de diseños y automatización de tareas a usuarios no expertos.

El algoritmo genético NSGA II se definió en el trabajo de Deb et al. (2002), este algoritmo emplea un enfoque de ordenamiento no dominado para clasificar las soluciones en diferentes niveles de no dominación, en busca de soluciones cercanas a la óptima de Pareto (no existe una solución que mejore un objetivo sin empeorar al menos otro). Además, NSGA-II utiliza una estrategia de selección elitista, manteniendo una población de élite que se preserva a lo largo de las generaciones, garantizando así la calidad de las soluciones. Los operadores genéticos, como el cruce uniforme y la mutación de punto, junto con la medida de distancia de hacinamiento para mantener la diversidad, son esenciales para explorar el espacio de soluciones y evitar la convergencia prematura a soluciones no óptimas. Asimismo, el algoritmo maneja problemas con restricciones mediante un enfoque de dominación restringida,

lo cual amplía su aplicabilidad.

Entre las ventajas del algoritmo NSGA-II destacan su capacidad para lograr convergencia y diversidad en las soluciones, intentando obtener soluciones cercanas al óptimo de Pareto. Aunque teóricamente es imposible garantizar que se encontrará este óptimo, el algoritmo mantiene una buena distribución de soluciones. No requiere parámetros de penalización para manejar restricciones, simplificando su implementación, y se ha aplicado con éxito en diversos problemas de optimización multiobjetivo. Sin embargo, presenta desventajas como la complejidad computacional, especialmente en grandes poblaciones.

Por su parte, con Project Refinery se facilita la configuración y el uso de este algoritmo a través de una interfaz intuitiva. Esta interfaz permite configurar tanto las entradas del algoritmo (variables y constantes) como definir los objetivos a minimizar o maximizar y sus restricciones. Además, ofrece la flexibilidad de ajustar el tamaño de la población inicial del algoritmo, y el número de generaciones y semilla para las etapas de cruce y mutación.

En la Figura 7 se muestra un ejemplo del menú de configuración de Project Refinery.

The image shows two side-by-side panels from the Project Refinery software interface.

Left Panel: Definir estudio

- Colocación de ejemplares de familia aleatorios**
 - Descripción:* Coloca un conjunto aleatorio de ejemplares de familia en una cara seleccionada con un tipo de familia base, una densidad y una semilla de aleatoriedad especificados.
 - Autor:* Autodesk
 - URL:* https://help.autodesk.com/view/RVT/2025/ESP?contextId=GRAPH_GD_RANDOM_FAMILY
 - Nom. est.: Colocación de ejemplares de familia aleatorios 0
 - Método: Optimizar
- 03 | Densidad**
 - ☒ Variable: 10 a 200
 - Introduzca un número superior para un diseño de ejemplar de familia más denso
- 04 | Semilla de aleatorización**
 - ☒ Variable: 0 a 100
 - Un número que inicia el proceso de aleatorización (puede ser cualquier número)

Right Panel: Definir objetivos

- Definir objetivos**
 - ☒ Resultados ☒ Minimizar ☐ Maximizar
- Definir restricciones**
 - ☐ Resultados Min. [] Máx. []
- Configuración de generación (Avanzado)**
 - Tamaño de la población: 20
 - Generaciones: 10
 - Semilla: 1
- Problemas**

Buttons: Atrás, Generar

Figura 7: Menú de configuración de un estudio de diseño generativo con Autodesk Project Refinery. Se necesita definir las entradas, objetivos de diseño, y configurar tamaño de población, generaciones y semilla. Fuente: Elaboración propia.

Por otro lado, los estudios de diseño generativo se desarrollan utilizando flujos de trabajo en el software DynamoBIM. Dentro del proceso de diseño generativo en Autodesk, estos flujos de trabajo se denominan Generadores, ya que su función es definir y modificar los diseños que se buscan optimizar. En esta etapa, es crucial definir correctamente las entradas y salidas del generador. Las entradas pueden ser constantes o variables, siendo estas últimas los genes propios de cada individuo dentro del algoritmo genético. A su vez, los cromosomas del algoritmo corresponden a la unidad de diseño generada por el flujo de trabajo bajo los genes ingresados. Dentro del proceso generativo, en la primera iteración los genes toman valores aleatorios para formar una población inicial de n individuos. Esta población atraviesa por la etapa de selección usando la función de aptitud simulada por las métricas del Evaluador.

Los estudios de diseño generativo se desarrollan utilizando flujos de trabajo en el software DynamoBIM. En el proceso de diseño generativo de Autodesk, estos flujos de trabajo se denominan Generadores, ya que su función es definir y modificar los diseños que se buscan optimizar. En esta etapa, es crucial definir correctamente las entradas y salidas del generador. Las entradas pueden ser constantes o variables, siendo estas últimas los genes propios de cada individuo dentro del algoritmo genético. Los cromosomas del algoritmo corresponden a la unidad de diseño generada por el flujo de trabajo bajo los genes ingresados. Dentro del proceso generativo, en la primera iteración los genes toman valores aleatorios para formar una población inicial de n individuos. Esta población atraviesa la etapa de selección utilizando la función de aptitud simulada por las métricas del Evaluador.

En el proceso de diseño generativo de Autodesk, el Evaluador corresponde a un flujo de trabajo que devuelve los valores numéricos de las métricas evaluadas para los diseños generados. Estas métricas se calculan según los criterios y definiciones establecidos por el arquitecto que creó el flujo de trabajo. Finalmente, estas métricas se comparan entre cada diseño (cromosoma) para determinar los mejores diseños, de acuerdo al criterio de minimización o maximización definido.

Además, con los resultados del Evaluador, se seleccionan los mejores diseños de la población. Luego, se realiza una etapa de cruce entre los genes de estos diseños seleccionados.

Además, se agrega una mutación o cambio aleatorio en los genes, con el objetivo de explorar más el espacio de soluciones posibles y así intentar conseguir una mejor solución, teniendo como resultado una nueva generación de individuos.

Este proceso se repite iterativamente según la configuración de número de generaciones y el tamaño de la población inicial escogido.

Finalmente, los flujos de trabajo se importan como un modelo de diseño generativo para poder ejecutarse en el Solucionador, que corresponde al algoritmo NSGA-II implementado en Project Refinery.

En la Figura 8 se resume el proceso de ejecución de un estudio de diseño generativo en Project Refinery.

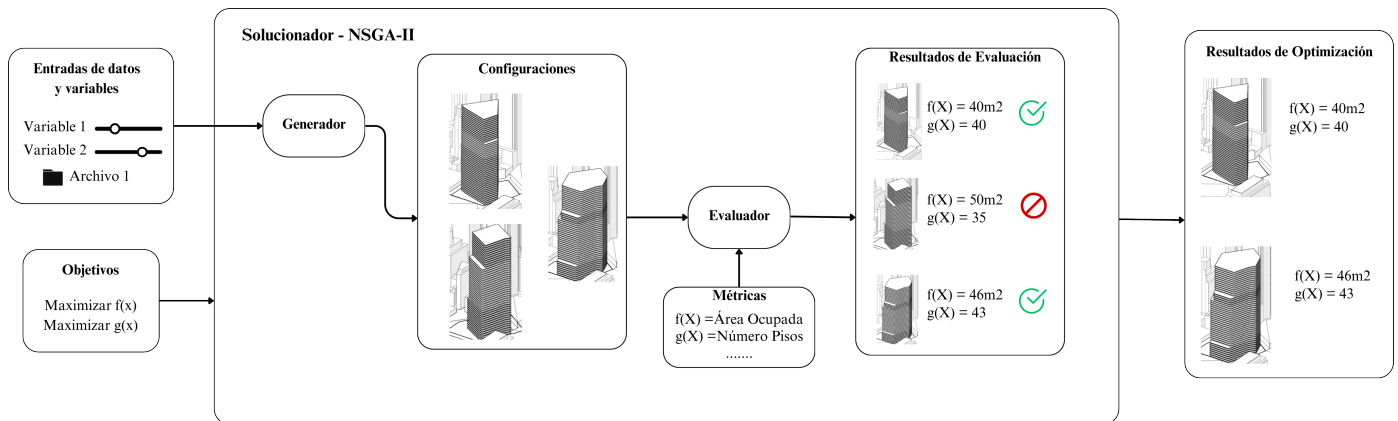


Figura 8: Estudio de diseño generativo que inicia con un generador que recibe los datos de entrada y variables. Este genera diferentes configuraciones, que son evaluadas por métricas y enviadas a un solucionador para obtener resultados optimizados de manera iterativa. Fuente: Elaboración propia.

2.4. Planificación de espacios

De acuerdo con Dutta et al “la planificación de espacios en arquitectura se define como la búsqueda de ubicaciones viables para un conjunto de objetos interrelacionados como (habitaciones, paredes, etc. sin necesariamente tener pre dimensiones); que no solo cumplan con los requisitos de diseño y maximicen la calidad del diseño de acuerdo con las preferencias de diseño, sino que también satisfagan los requisitos estéticos y de usabilidad” (Dutta and Sarthak, 2011).

Además, menciona que el diseño de la distribución de espacios es uno de los problemas más difíciles en la arquitectura debido a las múltiples restricciones que se tienen para su desarrollo. A su vez, a los arquitectos se les confía la responsabilidad de planificar el espacio que va desde un solo piso hasta un edificio en su totalidad.

Por ende, esta tarea toma gran parte del tiempo de diseño en el ciclo de vida del proyecto arquitectónico, y necesita de la colaboración activa del inversor y los diseñadores para que estos diseños logren aprobarse y seguir con las siguientes etapas.

De esta manera, este problema muestra un potencial adecuado para generar herramientas de automatización y optimización de los diseños, para así buscar reducir el tiempo de desarrollo y facilitar este proceso dentro del flujo de trabajo de los arquitectos.

2.5. Normativas de construcción locales

El desarrollo urbano de las ciudades está sujeto a una serie de regulaciones gubernamentales que buscan garantizar la seguridad, la calidad de vida y el bienestar de los ciudadanos. Estas regulaciones incluyen normas sobre la construcción de edificios, que tienen como objetivo asegurar que las edificaciones sean seguras, eficientes y compatibles con el entorno urbano.

En Cuenca, Ecuador, la normativa sobre restricciones gubernamentales para la construcción de edificios está establecida en el Reglamento de Zonificación y Usos de Suelo del Cantón Cuenca. Este reglamento establece una serie de normas que deben cumplir las edificaciones según su ubicación en la ciudad. Toda la información se encuentra en el Anexo 6.3 Fichas de polígonos de intervención territorial del suelo urbano, el cual se incluye en los anexos. También es importante aclarar que varias zonas de Cuenca, Ecuador, son consideradas Patrimonio Cultural de la Humanidad, y que el turismo es una gran fuente de ingresos para la ciudad, gracias a su belleza y armonía tanto natural como arquitectónica. Esta situación explica por qué la normativa de construcción en Cuenca es una de las más complejas, completas, exigentes y poco flexibles. Los arquitectos enfrentan grandes desafíos al desarrollar planos que optimicen el área construida del terreno debido a todas las limitaciones y complicaciones que existen.

El reglamento de zonificación de Cuenca divide la ciudad en siete zonas, cada una con un uso de suelo específico. Las zonas son:

- Zona central: Uso residencial, comercial y de servicios.
- Zona periférica: Uso residencial, comercial y de servicios.
- Zona industrial: Uso industrial.
- Zona rural: Uso agrícola y ganadero.
- Zona de conservación: Usos recreativos y turísticos.
- Zona de protección: Usos de protección ambiental.

Las zonas se diferencian en función de varios criterios, entre ellos:

- Densidad poblacional: La densidad poblacional es el número de personas por unidad de área. Las zonas centrales suelen tener una mayor densidad poblacional que las zonas periféricas.
- Actividades económicas: Las zonas centrales suelen albergar actividades económicas más intensivas, como el comercio y los servicios.
- Impacto ambiental: Las zonas de protección tienen un alto valor ambiental y, por lo tanto, se establecen restricciones para proteger el entorno.

Las edificaciones en Cuenca deben cumplir una serie de restricciones en función de su ubicación en la ciudad. Estas restricciones incluyen retiros y normas por piso.

Los retiros, establecidos en la normativas de construcción de Cuenca, Ecuador, se dividen en tres tipos: retiro frontal, retiros laterales y retiro posterior.

Los retiros frontales son distancias mínimas que deben existir entre la edificación y la línea de edificación. La línea de edificación marca el límite entre la propiedad privada y el espacio público. Los retiros frontales se establecen en función de la altura de la edificación y de la zona en la que se encuentra ubicada. Generalmente, los retiros frontales son mayores para las edificaciones más altas y para las zonas más centrales de la ciudad.

Los retiros laterales son distancias mínimas que deben existir entre la edificación y los límites

laterales de la propiedad. Al igual que los retiros frontales, se establecen en función de la altura de la edificación y de la zona en la que se encuentra ubicada. En general, los retiros laterales son mayores para las edificaciones más altas y para las zonas más centrales de la ciudad. Es importante agregar que la ciudad de Cuenca, Ecuador, permite lo que la normativa denomina “Adosamiento”. Esto indica que, si una persona lo desea, en la planta baja del edificio puede no aplicar los retiros laterales. Sin embargo, a partir del segundo piso de construcción, los retiros laterales son obligatorios.

Los retiros posteriores son distancias mínimas que deben existir entre la edificación y el límite posterior de la propiedad. Al igual que los retiros frontales, se establecen en función de la altura de la edificación y de la zona en la que se encuentra ubicada. En general, los retiros posteriores son mayores para las edificaciones más altas y para las zonas más centrales de la ciudad.

Las normas por piso establecen los máximos permitidos para la altura o el número de pisos de las edificaciones, pero nunca limitan ambas simultáneamente. La normativa determina la altura máxima o el número máximo de pisos que una edificación puede tener, dependiendo de la zona en la que se encuentra ubicada y el área del terreno donde se desea construir. Generalmente, en las zonas más periféricas de la ciudad, se permiten edificaciones más altas o con más pisos. La superficie máxima por piso también se determina en función de la zona, con superficies mayores permitidas en las áreas periféricas.

Las normativas de construcción en Cuenca están diseñadas para asegurar que las edificaciones se integren de manera adecuada en su entorno, garantizando la seguridad, la eficiencia y la compatibilidad con el uso del suelo definido para cada zona. Estas regulaciones son fundamentales para el desarrollo urbano ordenado y sostenible de la ciudad.

3. Estado del Arte

El avance en el diseño automatizado de edificaciones ha sido marcado por el framework Building Information Model (BIM), el cual proporciona una representación digital de las características físicas y funcionales de un edificio. Se han utilizado diferentes enfoques para realizar este proceso, en esta sección se exploran las técnicas actualmente aplicadas y analizadas en la literatura.

3.1. Trabajos Relacionados

Actualmente, se exploran técnicas de diseño generativo en el marco del BIM (Building Information Model), donde el diseñador establece parámetros y restricciones para obtener soluciones de diseño óptimas o cercanas a las óptimas. Estas técnicas se aplican en software de diseño arquitectónico como Revit, permitiendo la creación y simulación de modelos digitales de edificios y sus componentes, lo que reduce el trabajo manual y ahorra tiempo y dinero.

3.1.1. Planificación de espacios usando diseño generativo

La planificación de la disposición de espacios en arquitectura implica una comprensión profunda de las relaciones espaciales topológicas. A pesar de su naturaleza repetitiva y laboriosa, los avances en inteligencia artificial (IA) ofrecen nuevas perspectivas a través de la planificación automatizada de la disposición espacial (ASLP), alejándose de los enfoques tradicionales centrados en humanos. En un estudio reciente, Ko et al. (2023) tras una revisión de más de 589 artículos acerca de las tendencias en la planificación automatizada de espacios de planta empleando inteligencia artificial, proponen que se debe dar un mayor enfoque a técnicas basadas en imágenes, redes de grafos, rendimiento y agentes inteligentes. Además, instan a implementar la ASLP mediante metodologías de IA, pero preservando el papel de los arquitectos como refinadores de los resultados devueltos por las aplicaciones. Se destaca el uso de grafos para resumir la información de las superficies y utilizarla con algoritmos de diseño generativo.

Por su parte, Gan (2022) presenta un modelo de datos basado en grafos para la represen-

tación teórica de características esenciales en edificios modulares, con el objetivo de automatizar y optimizar el diseño generativo para la construcción modular. El método propuesto implica el establecimiento de la definición de modelo de vista (MVD) de Industry Foundation Classes (IFC) para describir de manera exhaustiva la información requerida para el diseño de edificios modulares. A continuación, el MVD de IFC se mapea a un modelo de datos de grafos, que incluye la formulación explícita de atributos espaciales ausentes, relaciones topológicas, geometrías y semántica para módulos volumétricos. Se desarrollan algoritmos de transformación de datos para convertir la información requerida en BIM al modelo de grafos para consultas de información variada. Además, desarrolló un sistema prototipo para el diseño generativo de edificios modulares.

A su vez, Zhang et al. (2021) proponen un algoritmo generativo para la creación automática de esquemas de diseño de espacios de planta en edificios residenciales, basándose en la optimización del rendimiento espacial.

En el estudio de Wang et al. (2023) se describe un enfoque para generar diseños automatizados de edificios utilizando aprendizaje profundo y algoritmos de grafos. Se crea un conjunto de datos único de diseños de edificios para respaldar el enfoque propuesto. Se utilizan algoritmos de grafos y aprendizaje profundo para seleccionar y ajustar los diseños de manera precisa. El modelo de Red Neuronal U-Net con atención espacial controlada por entrada segmenta con precisión la región del edificio y el diseño resultante se perfecciona a través de operaciones de imagen, lo que permite generar modelos BIM completos para los diseñadores. Adicionalmente, a través de dos estudios de casos generativos y un experimento comparativo con redes neuronales, este artículo demuestra la eficacia del enfoque que puede ayudar a los diseñadores durante las etapas iniciales del diseño y permitir una generación rápida de diseños completos para edificios individuales.

Por otro lado, Aupy. presenta un caso de estudio en el que se utilizó el diseño generativo y la automatización para optimizar el diseño de una farmacia hospitalaria. Se utilizó Dynamo, Project Refinery y programación para llevar a cabo estudios paramétricos y generar un modelo BIM en 3D. El objetivo fue mejorar la configuración arquitectónica de la farmacia y su entorno de trabajo. Se describen los desafíos logísticos de una farmacia y las reglas

arquitectónicas involucradas, así como los algoritmos utilizados para resolver problemas logísticos complejos. También se explora la automatización arquitectónica a través de la programación y se muestra cómo se generaron las soluciones de diseño.

Así mismo, en el trabajo de Aalaei et al. (2023) se desarrolló un modelo de Redes Adversarias Generativas Condicionales (cGAN) basado en grafos. Este modelo toma como entrada los límites del edificio, la ubicación de los espacios y sus relaciones de adyacencia, y genera diseños de distribución de espacios en formato vectorial. Para lograr esto, se implementa un enfoque iterativo de tres etapas, donde cada generador del modelo refina progresivamente los diseños generados. Este modelo puede generar diseños competitivos considerando restricciones topológicas y geométricas, en comparación con los diseños reales del conjunto de datos RPLAN utilizado. Además, el artículo introduce una técnica de postprocesamiento que convierte los diseños generados en formato pixelado a formato vectorial, facilitando así su integración en herramientas de diseño arquitectónico. Aunque, con este enfoque todavía existe la limitación de tener una distribución inicial de diseño para generar el plan de espacios.

La literatura presentada hasta el momento aporta con diferentes enfoques para atacar el problema de la automatización de la planificación de espacios. Aunque, se cuenta con la limitación de la adaptabilidad al software Autodesk DynamoBIM el cual es indispensable para la conexión con el software de diseño arquitectónico Revit. Esto se debe principalmente a la representación de la información utilizada, la cual resume numéricamente las relaciones topográficas y geométricas, mientras que en Revit se ocupa una manipulación directa de las formas, estructuras y geometrías necesarias.

Frente a esta problemática, Das et al. (2016) presentan el Sistema Generador de Planificación de Espacios (SPG), una metodología integral para la elaboración de planes de espacios mediante el uso del software DynamoBIM. El SPG integra estrategias de diseño generativo, estructuras de datos eficientes y algoritmos de geometría computacional con el objetivo de asistir a los equipos de diseño en la toma de decisiones informadas. El enfoque clave del SPG es su estrategia jerárquica de asignación de espacios. Primero se colocan los departamentos en el sitio, luego se asignan los programas dentro de ellos, y finalmente se calcula

la circulación. Para implementar esta jerarquía, el sistema utiliza una versión modificada de la estructura de datos K-d Tree, la cual permite un almacenamiento y recuperación eficiente de los datos espaciales. Así mismo, el SPG emplea diversas estrategias personalizadas de división de policurvas para distribuir los elementos del programa en el sitio, considerando criterios como distancia, proporción y área. Adicionalmente, el sistema construye una grilla de celdas que facilita el análisis de la circulación y la evaluación de los planes de espacio generados. Aunque, cabe mencionar que esta metodología no incluye la validación empírica de los resultados, ni la implementación de algoritmos de diseño generativo.

Finalmente, como parte del problema de asignación de espacios, se han encontrado metodologías que abordan problemas similares en otras áreas. Es así que, Espín et al. (2020) presenta un enfoque evolutivo para resolver el problema de asignación de polígonos irregulares dentro de un área específica, combinado con el Problema de la Mochila. Se destaca el uso de mutaciones como agregar, eliminar y modificar la posición y rotación de los objetos, así como un operador de cruce que divide el contenedor en dos regiones e intercambia los objetos entre los progenitores.

Además del algoritmo evolutivo, el documento presenta dos algoritmos complementarios. El primero es un algoritmo codicioso (greedy), este algoritmo se enfoca en tomar decisiones con base en la mejor opción de diseño disponible en cada iteración, con el objetivo de construir una solución de manera progresiva y que la misma sea cercana a la óptima para el problema. Específicamente, el algoritmo codicioso selecciona y agrega los polígonos en un área delimitada, basándose en el área del objeto, su orientación y en una puntuación que combina el peso dado al objeto y su área;

El segundo algoritmo que propone el autor se denomina “reversible”. Esto se debe a que el algoritmo tiene la misma base que el algoritmo codicioso, pero adicionalmente realiza de manera aleatoria adiciones, eliminaciones y modificaciones en la posición de los polígonos ya establecidos. Aunque, se pueden revertir dichos cambios en caso de que no se logre obtener una mejor solución que la ya disponible.

Finalmente, el autor evaluó estos tres algoritmos y tras un análisis comparativo demostró que el algoritmo evolutivo obtiene los mejores resultados en términos de número de iteracio-

nes para converger, tiempo de ejecución y por presentar una solución de ubicación de las geometrías muy cercana a la solución real planteada. Aunque, se tiene un mayor consumo de recursos computacionales.

3.1.2. Diseño de volumetrías usando diseño generativo

Dentro del campo del diseño automático de estructuras de edificaciones, Zheng et al. (2021) sugiere una metodología de diseño arquitectónico y urbano basado en redes neuronales artificiales. Se utilizan redes neuronales profundas (Deep Learning) para generar geometrías tridimensionales con mayor precisión y velocidad de cálculo. Se desarrolla una red neuronal artificial específica que aprende y genera características de diseño para formas de edificios.

Se utiliza una estructura de datos personalizada con parámetros de características para describir las propiedades del diseño. La red neuronal se entrena con datos de diseño generados y luego se prueba ajustando los parámetros de características. Los resultados muestran la capacidad generativa básica de la red neuronal.

Adicionalmente, al ser entrenada con datos de diseño recopilados de edificios existentes, la red neuronal aprende e infiere las características geométricas del diseño arquitectónico con diferentes parámetros de características, lo que proporciona un método basado en datos para que los diseñadores generen y analicen formas arquitectónicas.

Por otro lado, Zhuang et al. (2023) presenta varias contribuciones clave para la generación automática de edificios. En primer lugar, explora dos formatos de representación de datos tridimensionales, vóxeles y función de distancia con orientación (Signed Distance Function SDF), para entrenar los modelos de aprendizaje de máquina. Los autores encuentran que la representación SDF tiene mejor resolución y eficiencia que los vóxeles.

Además, utilizan dos tipos de modelos de redes neuronales generativas, el decodificador automático y la red adversaria generativa (GAN), para generar nuevas formas de edificios a partir de los datos de entrada. Demuestran que ambos modelos pueden capturar las características implícitas de los edificios y generar nuevas soluciones de diseño coherentes con el estilo de los datos de entrada.

Adicionalmente, los autores proponen utilizar los modelos generativos para construir un es-

pacio rico de exploración de soluciones de diseño. Pueden interpolar y extrapolar en el espacio latente para generar múltiples candidatos de diseño que no se han visto en la realidad, pero son razonables.

Por último, el artículo reconoce el problema del sesgo de datos en el aprendizaje automático y discute estrategias para aumentar la diversidad y representatividad de los conjuntos de datos, como reconstrucción masiva en 3D a partir de imágenes satelitales, transferencia de conocimiento y generación de datos sintéticos.

De acuerdo con el National BIM Report NBSEnterprises (2020), el software más utilizado a nivel mundial es Revit, debido principalmente a su compatibilidad con otros sistemas de diseño de construcción de la familia Autodesk. Mediante su API, Revit permite crear flujos de trabajo para automatizar procesos y aplicar algoritmos generativos, esto lo vuelve el software ideal para la generación de las extensiones de código, ya que realizar trabajos similares en otros software involucran la creación de las bases de funcionamiento las cuales corresponden a un proceso mucho más amplio de desarrollo que se escapa de los objetivos de este trabajo de titulación (NBSEnterprises, 2020).

Por este motivo, se ha escogido Revit como la base para la realización de este trabajo de titulación, además de que es la herramienta principal dentro del estudio de arquitectura colaborador y se conoce que está presente en múltiples estudios dentro de la ciudad de Cuenca.

3.2. Limitaciones dentro de la Literatura Actual

Si bien se posee una amplia aplicación de las técnicas de diseño generativo en la literatura, hasta el momento no se han encontrado aplicaciones que evalúen la factibilidad de la construcción de las edificaciones y las divisiones de planta de manera empírica en conjunto con miembros del área de la arquitectura que permitan identificar las falencias o fortalezas de los métodos y los resultados que arrojan. Además, las aplicaciones existentes se evalúan en sus propios contextos locales, lo que implica que las normativas de construcción varían con base en el país donde se desarrolló la investigación. Es por esto que se buscan adaptar las soluciones a contextos locales en Ecuador, específicamente en la ciudad de Cuenca. A

continuación, se detallan estas problemáticas.

3.2.1. Adaptación de soluciones a normativas específicas

En el ámbito del diseño generativo aplicado a la arquitectura, se evidencia una laguna sustancial de conocimiento en cuanto a la integración efectiva de soluciones generativas con normativas específicas. Este vacío de investigación se manifiesta especialmente en la escasez de enfoques detallados que aborden de manera precisa y contextualizada la adaptación de algoritmos de diseño generativo a políticas urbanísticas particulares. Este problema adquiere mayor relevancia en contextos regionales como Ecuador, donde las normativas locales imponen restricciones específicas en términos de retiros, densidad y otros requisitos urbanos. La falta de literatura que aborde de manera específica este desafío puede limitar el desarrollo y la implementación efectiva de soluciones generativas que respeten y se ajusten a los marcos regulatorios locales, destacando así la necesidad de investigaciones que aborden este vacío y proporcionen orientación detallada para diseñadores y profesionales del urbanismo.

3.2.2. Evaluación Empírica de los Resultados Generados por las Extensiones de Código

En la perspectiva de llevar a cabo una evaluación empírica de los resultados derivados de las extensiones de código propuestas, se vislumbra un vacío de conocimiento relativo a la precisión y validez de las métricas empleadas para valorar la calidad estética y estructural de las soluciones arquitectónicas generadas. Este hueco se presenta como una limitación significativa en el corpus de literatura existente, ya que no se ha abordado de manera exhaustiva la eficacia y fiabilidad de dichas métricas en el contexto específico de la evaluación de propuestas arquitectónicas generadas mediante algoritmos de diseño generativo. La literatura actual en este ámbito podría no ofrecer un análisis detenido de la idoneidad y aplicabilidad de las métricas utilizadas, lo que plantea interrogantes sobre la confiabilidad de los resultados derivados de dicha evaluación.

4. Metodología

En esta sección se especifican los procesos realizados para la generación y evaluación pertinente de las extensiones de código para la generación automática de volumetría y la generación automática de diseños de distribución de espacios de planta.

4.1. Selección del enfoque para la solución del problema

En primer lugar, hasta el momento en la literatura revisada no se han encontrado soluciones factibles y compatibles con los software de diseño arquitectónico de libre distribución o disponible en el mercado.

En el contexto del desarrollo de planos mediante la metodología BIM, existe una amplia gama de software disponible, como Revit, ArchiCAD, FreeCAD, entre otros. Todas estas herramientas poseen características y limitaciones propias, las cuales fueron evaluadas para conocer la mejor alternativa para construir la solución propuesta.

De acuerdo con el National BIM Report NBSEnterprises (2020), el software más utilizado a nivel mundial es Revit el cual permite extender su funcionalidad a través de flujos de trabajo para automatizar procesos y aplicar algoritmos generativos, esto lo vuelve el software ideal para la generación de las extensiones de código, ya que realizar trabajos similares en otros software involucran la creación de las bases de funcionamiento las cuales corresponden a un proceso mucho más amplio de desarrollo que se escapa de los objetivos de este trabajo de titulación.

Por este motivo, se ha escogido Revit como la base para la realización de este trabajo de titulación, además de que es la herramienta principal dentro del estudio de arquitectura colaborador y se conoce que está presente en múltiples estudios dentro de la ciudad de Cuenca.

Por otro lado, los enfoques empleados en la literatura abarcan diversos contextos y tipos de información. Las representaciones de información sobre diseños arquitectónicos incluyeron grafos, conjuntos de puntos tridimensionales, imágenes y geometrías. Si bien esta variedad

de representaciones resulta útil, también supone una limitación importante en la selección de un método de generación de diseños, ya que no se adecuan al software Revit.

Esto se debe a que se cuenta con un número limitado de 40 diseños de edificios y sus distribuciones de espacio, desarrollados únicamente con dicho software.

Esta información presenta dificultades para ser exportada a otras representaciones fuera del software, ya que no se dispone de una API (Interfaz de Programación de Aplicaciones) que permita la manipulación externa de los objetos y clases del programa. DynamoBIM, definido por Autodesk, es la solución para la manipulación de información y la integración de código fuente a los procesos de Revit.

Cabe recalcar que existió un proceso de selección previo del enfoque de desarrollo, y la cantidad de información junto con su representación y simplificación mediante el uso de polígonos resultó en un pobre rendimiento bajo redes neuronales de grafos, redes neuronales profundas y redes neuronales generativas. Además, como se mencionó anteriormente, la exportación de los resultados hacia el software arquitectónico resultó en un proceso mucho más laborioso. Por este motivo, se decidió abordar el problema de diseño como un problema de optimización, con sus respectivos objetivos de diseño medibles bajo métricas establecidas para cada extensión de código. Este enfoque permitió generar un conjunto de diseños cercanos al óptimo, parametrizando el proceso y sin la necesidad de contar con una muestra de edificios grande, ya que únicamente se recurre a una evaluación visual de las soluciones y una comparativa de métricas.

4.2. Desarrollo de la extensión de código para la generación automática de volúmenes de edificios

En la Figura 9 se proporciona un esquema que resume la metodología diseñada para la elaboración de la herramienta y su respectiva evaluación. En ella se describe

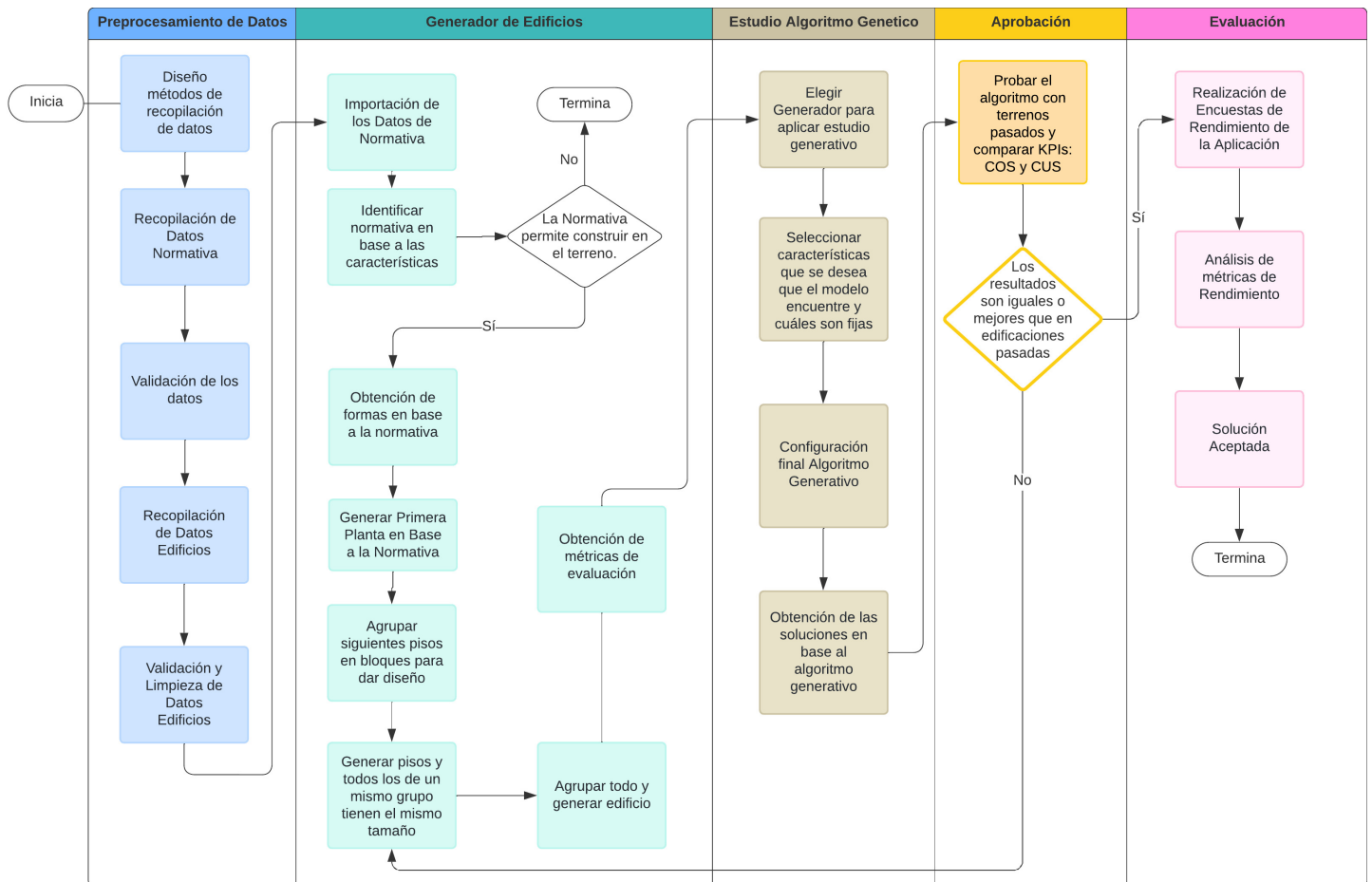


Figura 9: Metodología para generar y evaluar una extensión de código automatizada para volumetrías de edificios. Se procesa información y normativas, se diseñan plantas y bloques, se usan algoritmos genéticos para mejorar resultados previos, culminando con evaluación empírica.

4.2.1. Recopilación de datos de diseños de edificios y normativa

En el apartado de recolección de datos, se explicarán los datos disponibles para este trabajo de titulación. Para empezar, se habla de los edificios con los que se dispone, todos ellos considerados optimizados, ya que cumplen con la normativa y fueron creados buscando maximizar el área total construida. De estos edificios, se tomarán las características necesarias para la evaluación inicial.

Como se había indicado en la selección del enfoque, la cantidad de edificios con los que se cuenta son 40, ubicados en diversas zonas de la ciudad de Cuenca. Estos edificios servirán para una evaluación inicial y preliminar que permitirá determinar si la solución es óptima, en otras palabras, aprobarla, para pasar a la evaluación empírica. En esta etapa, se tomarán

los edificios y se extraerán las siguientes características: altura en pisos del edificio, retiros correspondientes, área total del terreno, área total del primer piso, área total construida y si el edificio está adosado. Estas características se utilizarán en las próximas etapas para determinar las métricas necesarias.

El siguiente paso a tomar es la recopilación de las normativas de la ciudad de Cuenca. Estas normativas serán recopiladas y almacenadas en un archivo de Excel que contiene el formato necesario para su uso futuro en el script. No se almacenarán todos los datos del documento, ya que no son necesarios. En las filas, se presentará la normativa por sector y pisos. Dado que la normativa varía según la cantidad de pisos, es posible que se tenga más de una fila con un mismo código de sector, pero la cantidad de pisos siempre será variable, así como las normativas asociadas.

Las columnas seleccionadas fueron las siguientes:

- **Código:** Esta columna contiene el código que la ciudad emplea para los sectores existentes. Por ejemplo, “E-1” se refiere a todos los terrenos ubicados en lo que la ciudad denomina como Este 1.
- **Altura en pisos:** Indica la altura en pisos de la edificación de la cual obtendremos la normativa. Es importante mencionar que, en el caso de las edificaciones industriales, la altura se expresa en metros. Por lo tanto, almacenamos la cantidad seguida de la letra “m” para poder distinguirla y seleccionarla correctamente en el script.
- **Lote Mínimo:** Esta columna indica el tamaño mínimo que debe tener el lote en metros cuadrados para calificar a la altura en pisos seleccionada.
- **Frente Mínimo:** Indica el tamaño mínimo que debe tener el frente del lote en metros cuadrados para calificar a la altura en pisos seleccionada.
- **Retiro Frontal:** Contiene el retiro frontal mínimo obligado por la ley, expresado en metros cuadrados.
- **Retiro Lateral:** Contiene el retiro lateral mínimo obligado por la ley, expresado en metros cuadrados.

- Retiro Trasero: Indica el retiro trasero mínimo obligado por la ley, expresado en metros cuadrados.

Vale la pena recalcar que toda la normativa se obtuvo del documento oficial publicado por la alcaldía de Cuenca, el cual se incluye en los anexos.

4.2.2. Selección de las métricas relevantes para la evaluación de los diseños generados.

Las métricas seleccionadas para evaluar los diseños de volumetría generados son el **COS**, que indica el coeficiente o porcentaje de ocupación de la primera planta respecto al terreno disponible, y el **CUS**, que representa el coeficiente o porcentaje total de ocupación de la edificación respecto al terreno potencial. Además, se incluye el área provisional para ascensor. Esta métrica indica la desventaja de emplear más pisos de los óptimos, debido a que seleccionar más pisos de los necesarios para la zona disminuye el CUS, o incluso lo mantiene, pero el precio aumenta por el costo de un piso extra que no era necesario para maximizar el retorno de inversión. Debido a los retiros establecidos, no es posible ocupar la totalidad del terreno en ninguno de los pisos. Por lo tanto, se utilizan estas métricas con el objetivo de maximizar la ocupación de la edificación y se pueda emplear la solución para evaluar su rentabilidad. Bustillos (Bustillos)

4.2.3. Desarrollo del generador de diseños de volumetrías con DynamoBIM

El generador se construye empleando DynamoBIM, el cual funciona sobre el software Autodesk Revit, como se explicó anteriormente. El código comienza con la recepción de las entradas para el generador, las cuales pueden ser fijas o variables, dependiendo de los requisitos. Todas estas entradas deben ser proporcionadas por el usuario para garantizar la ejecución adecuada del generador y la optimización. La primera entrada es el archivo que contiene las normas en el formato deseado, el cual tiene un formato CSV (Comma Separated Value). Luego, se ingresan los límites de la construcción en Revit, ya que DynamoBIM permite seleccionar las líneas correspondientes al terreno sobre el cual se desea construir directamente desde el plano que se está construyendo. Posteriormente, se especifica la zona de construcción, que corresponde al código del lugar donde se desea la construcción.

Las siguientes entradas son: un indicador para determinar si el edificio se desea adosado, la cantidad de pisos y la altura entre los pisos de la edificación. Estos valores pueden variar durante la ejecución del algoritmo, por lo que en un punto de la optimización pueden ser 3 pisos y 2 metros de altura, mientras que en otro momento pueden ser 20 pisos y 2.31 metros de altura. La validación de estas combinaciones con respecto a la normativa se explicará a continuación.

Después de recibir las entradas, se necesita filtrar el archivo con la normativa y conservar solo las filas para las que aplica la construcción actual. El primer paso es cargar el documento y convertirlo a un DataFrame de pandas (librería de python que permite un fácil manejo de datos) para facilitar el filtrado. Luego, filtramos para obtener todas las normativas para el lugar donde se está construyendo. Una vez hecho esto, filtramos el rango de normativa para las que aplica el terreno según su frente y ancho. Con esto, verificamos el DataFrame. Si está vacío, se devuelve 0, lo que indica que no se puede construir en dicho terreno porque la normativa no lo permite y los resultados estarán vacíos. Luego, en caso de no ser cero, verificamos si la altura en pisos no contiene una "m". En caso afirmativo, buscamos la fila que corresponde a las especificaciones actuales y se devuelven las restricciones de retiro correspondientes. En caso de no ser válida, se entrega la cantidad máxima de pisos posible y la normativa para dicha cantidad. Por otro lado, si la altura se limita por metros, tomamos la altura del piso y calculamos la cantidad de pisos que se puede tener con dicha altura, devolviendo las restricciones de la normativa y la cantidad de pisos que se usarán.

El siguiente paso es crear en DynamoBIM los offsets personalizados en los límites del terreno. Aunque esto podría parecer algo muy sencillo, DynamoBIM no permite aplicar offset diferencial, por lo que se debe hacer manualmente. Primero se clonan las líneas del límite del terreno y luego se trasladan las líneas de retiro frontal y trasero en el eje de las X, y las líneas de retiro lateral en el eje de las Y. Luego, se extienden las líneas según el lado más largo del terreno, para asegurarse de que las líneas se superpongan en algún punto. Una vez que las líneas están trasladadas y extendidas, se buscan los puntos de intersección y se unen, generando así la superficie del edificio con los retiros variables, que cumple con los límites de la normativa. En la ciudad de Cuenca, como se explicó anteriormente, se permite

el adosamiento; en este punto, si se selecciona adosamiento, se crea otra superficie que muestra el terreno solo con el retiro frontal y trasero.

En caso de los edificios adosados, se generará el primer piso con la superficie adosada; caso contrario, con la superficie con todos los retiros.

A continuación, se debe crear un array de longitud aleatoria, cuya longitud será un número entre 1 y la cantidad de pisos seleccionada menos uno. Este array se llena de una forma peculiar, se llena con valores aleatorios que pueden ir desde 0 hasta la cantidad de pisos seleccionada menos uno, pero siempre se valida que la suma de todos los valores del array de la altura del edificio en pisos menos uno. En este proceso se busca generar grupos de pisos para poder dar diseño al edificio. El siguiente paso es ir tomando estos valores del array, seleccionar si se le dará o no un balcón; en caso de dárselo, de qué tamaño será, que puede ser desde 0.5 metros hasta 3 metros. Después se generará la cantidad de pisos según el valor tomado del array, luego se tomará el siguiente valor y se realizará el mismo proceso hasta tomar todos los valores del array. Esto genera todos los pisos desde el segundo piso hasta el último piso.

En el punto en el que se tienen ya todos los pisos listos, se los une en el orden correspondiente y se los agrupa en una sola forma. Esta forma se convierte en nuestro edificio generado. De este edificio se extraen las métricas de COS, CUS, área total construida y el espacio provisional para ascensor. Estos valores, junto con el edificio terminado, son las salidas del generador que nos permiten aplicar el algoritmo generativo.

A continuación, se presenta un diagrama de flujo que facilita la comprensión del proceso que sigue el generador 10.

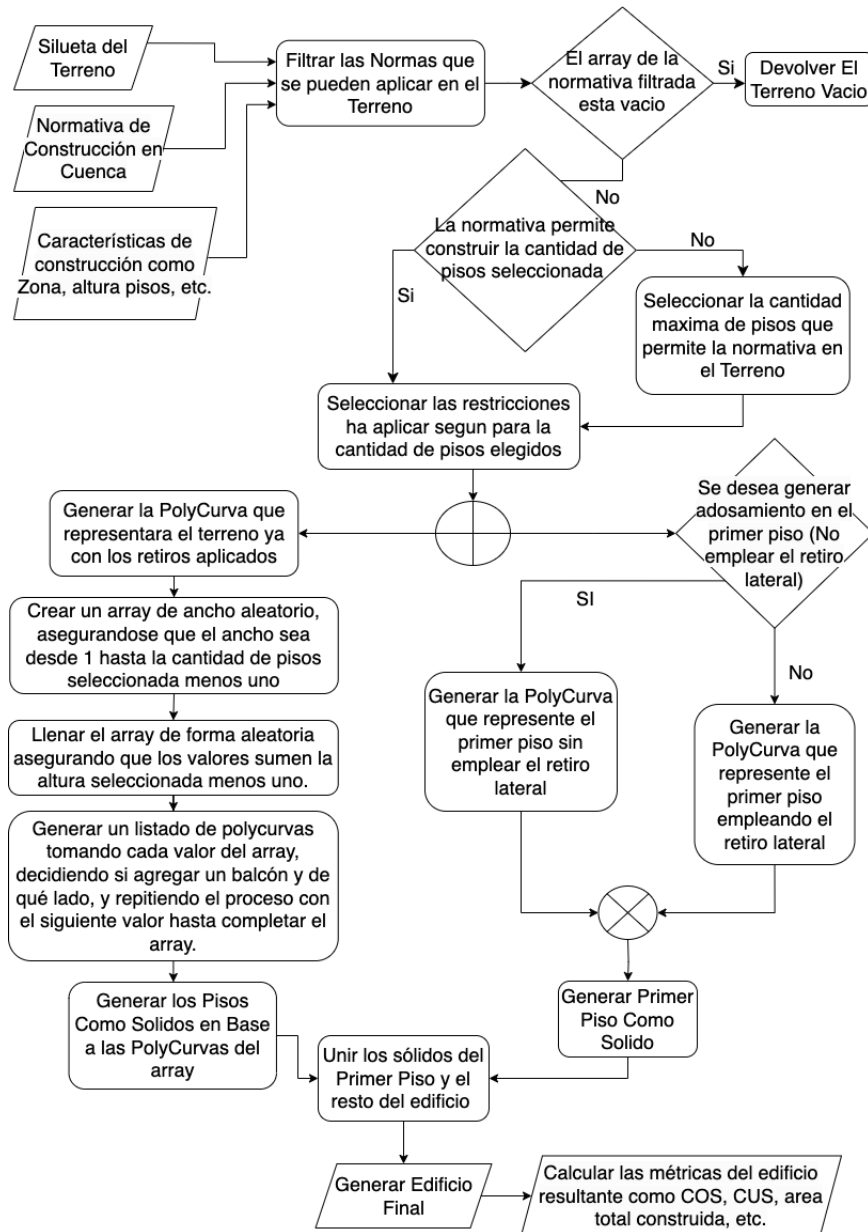


Figura 10: Flujo del generador de volumetrías: Se seleccionan y filtran normas de construcción, se genera una PolyCurva del terreno con retiros, se crea un array de dimensiones aleatorias, se diseñan pisos sólidos y se unen al edificio, calculando métricas como COS, CUS y área construida, garantizando normativas y diseño funcional.

4.2.4. Aplicación del Algoritmo Genético con Project Refinery

El primer paso implica abrir el código de Revit generado previamente y guardar el archivo indicando que será utilizado como generador para el algoritmo genético. A continuación, se abre Revit y dentro de este entorno se accede al estudio generativo con Project Refinery. Posteriormente, se busca el generador creado y se procede a abrirlo. Se selecciona el archi-

vo de normativas en la ubicación correspondiente, luego se especifica la zona de la ciudad donde se realizará el edificio, y en Revit se seleccionan los límites del terreno donde se aplicará el algoritmo. Finalmente, se indica la maximización del COS y el CUS, así como la minimización del espacio provisional para el elevador.

Una vez que estas configuraciones están listas, se establece una población y generaciones. En la sección de resultados, se mostrarán las combinaciones que han dado buenos resultados y luego se inicia el estudio. Tras ejecutar el algoritmo, se obtiene como resultado un número “n” de edificios que cumplen con la normativa. Este valor de “n” dependerá de la combinación de población y generaciones seleccionadas. Posteriormente, estos edificios se ordenan de menor a mayor según la métrica deseada. Al hacer clic sobre cada uno de ellos, se pueden visualizar más detalles como los pisos seleccionados por el algoritmo, la altura, entre otros.

4.2.5. Aprobación de las soluciones de la extensión de código

Antes de proceder con la evaluación empírica de la extensión de código, fue necesario asegurarse de que sus resultados fueran adecuados. En este apartado, se realizó un procedimiento simple que se detalla a continuación.

En primer lugar, se tomaron los 40 edificios y de cada uno de ellos se extrajeron: el terreno, la ubicación del edificio, los pisos seleccionados, si está adosado o no, el área total construida, el espacio provisional para ascensor, el COS del edificio y el CUS del mismo; todo esto se almacenó en un archivo Excel. Posteriormente, se pasó a la solución el terreno y la ubicación del edificio, se aplicó el algoritmo generativo y se extrajeron los pisos seleccionados, si está adosado o no, el área total construida, el espacio provisional para ascensor, el COS del edificio y el CUS del mismo; todo esto también se pasó a un archivo Excel.

El primer paso fue revisar el levantamiento inicial que tienen los arquitectos del terreno. En la figura 11, tenemos una referencia de cómo se ve este levantamiento. Vale la pena recalcar que no siempre se usa el mismo formato, ya que puede variar según el arquitecto, pero los datos que contiene son los mismos. Posteriormente, con base en este levantamiento, se tomó el sector y las medidas del terreno y se transfieren a un archivo de Excel. Este proceso

se repitió en todos los 40 planos que se tienen para este estudio.

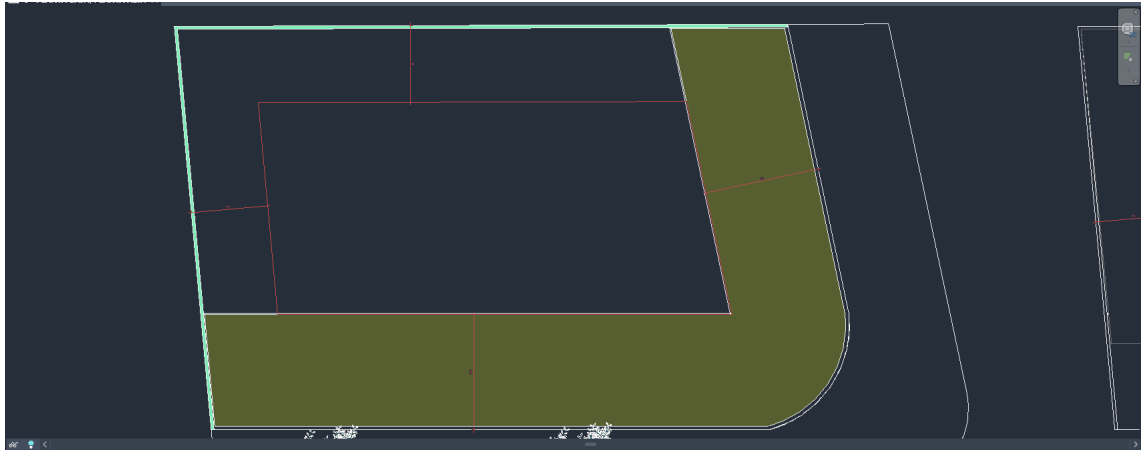


Figura 11: Levantamiento inicial del terreno con retiros inicialmente propuestos.

Para la carga de información se creó una representación de la silueta del terreno levantado previamente con líneas de modelo (Model Lines) las cuales son la representación de las rectas dentro del espacio tridimensional de Revit. Y se creó un estudio de diseño generativo. En la figura 12, se observa el proceso.

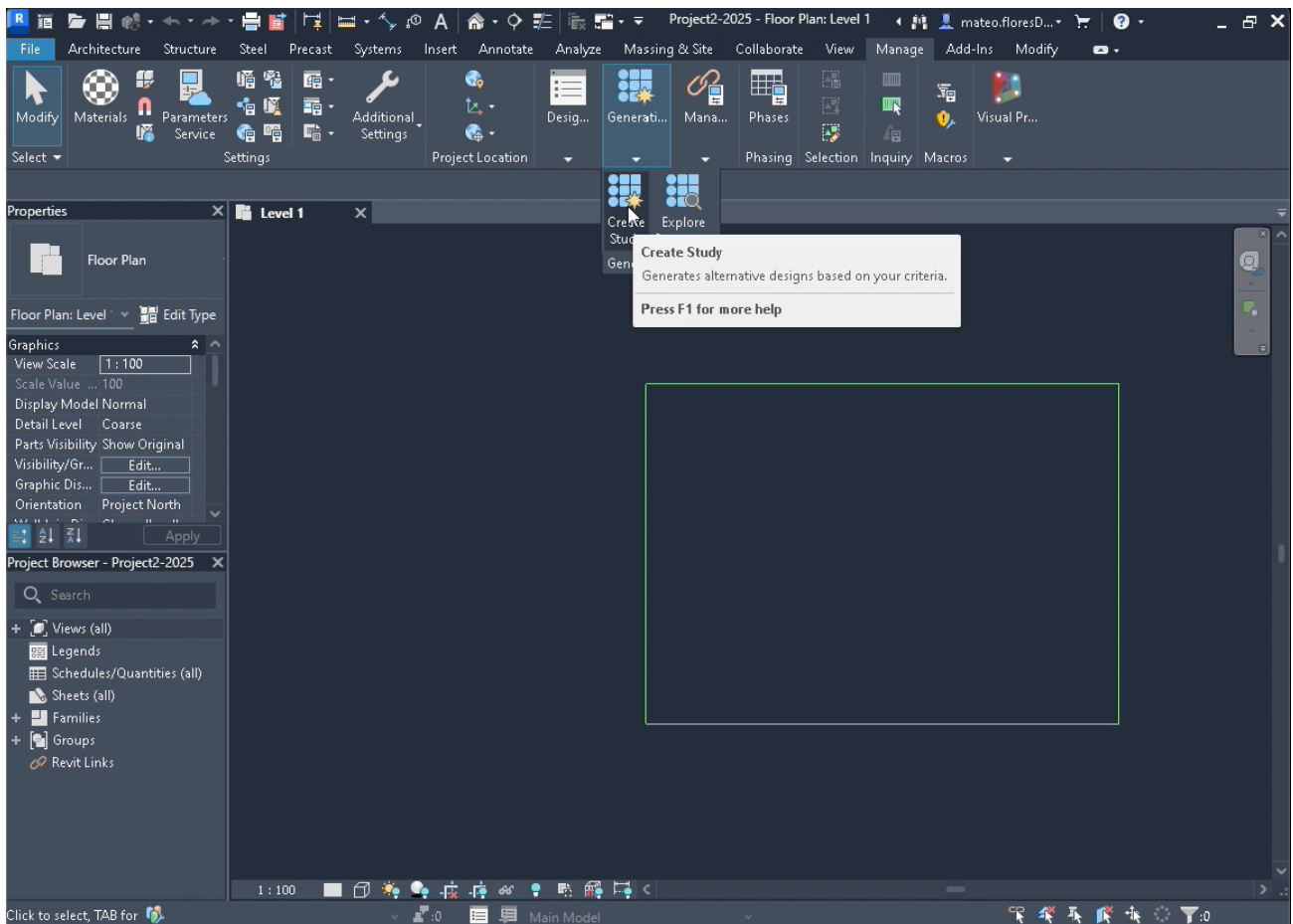


Figura 12: Ubicación en Revit para ingresar al estudio de diseño generativo.

Una vez dentro del menú de estudios de diseño generativo, se cargó la extensión de código previamente desarrollada como un nuevo estudio generativo. En la figura 13, se indica cómo seleccionar la extensión de código para crear un nuevo estudio. Cabe recalcar que la extensión de código corresponde al generador aleatorio de los diseños.

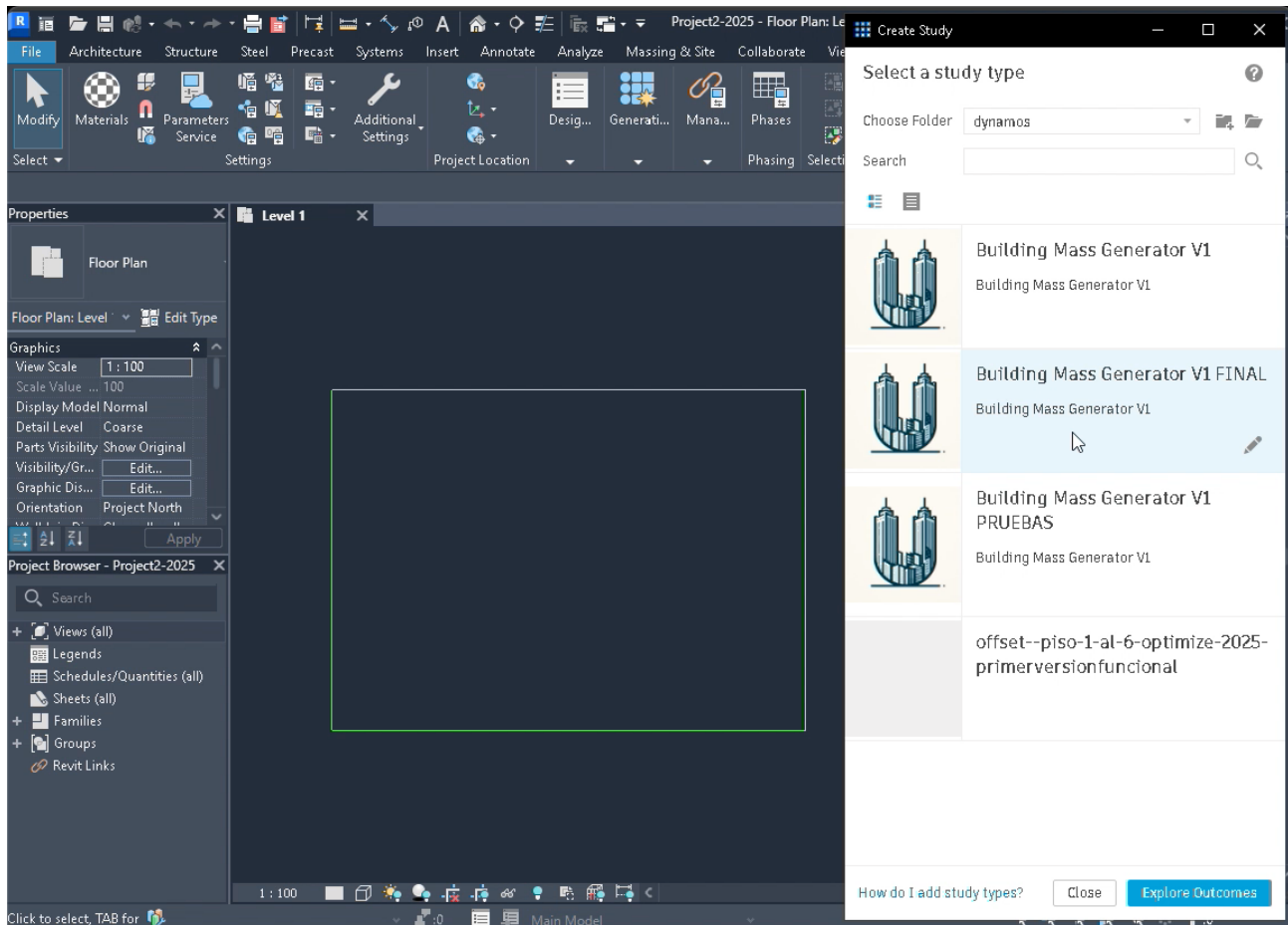


Figura 13: Selección del generador de volumetrías en Revit para el estudio de diseño generativo, indicando la versión final utilizada en esta tesis.

Seguidamente, se presenta el menú de configuración del algoritmo generativo la cual se detalla a continuación.

En primer lugar, se configuran las variables y constantes del algoritmo. Tal como se expuso previamente, el algoritmo generativo tiene como objetivo optimizar determinadas variables de salida, entre ellas el coeficiente de ocupación del suelo (**COS**), el coeficiente de uso del suelo (**CUS**) y el área total construida. Asimismo, el algoritmo recibe un conjunto de variables de entrada que establecen los parámetros iniciales del sistema.

En la Figura 14 se presenta el menú de configuración correspondiente, con el objetivo de facilitar la comprensión de esta sección. La primera entrada del menú corresponde a la altura de cada piso, la cual puede estar seleccionada o no. En el primer caso, el algoritmo considera dicha entrada como una variable y la ajusta automáticamente en cada iteración.

Por el contrario, en el segundo caso, se puede establecer un valor fijo para la altura de los pisos, lo cual generará diseños con una altura constante.

La siguiente variable es la cantidad de pisos, que debe estar siempre seleccionada.

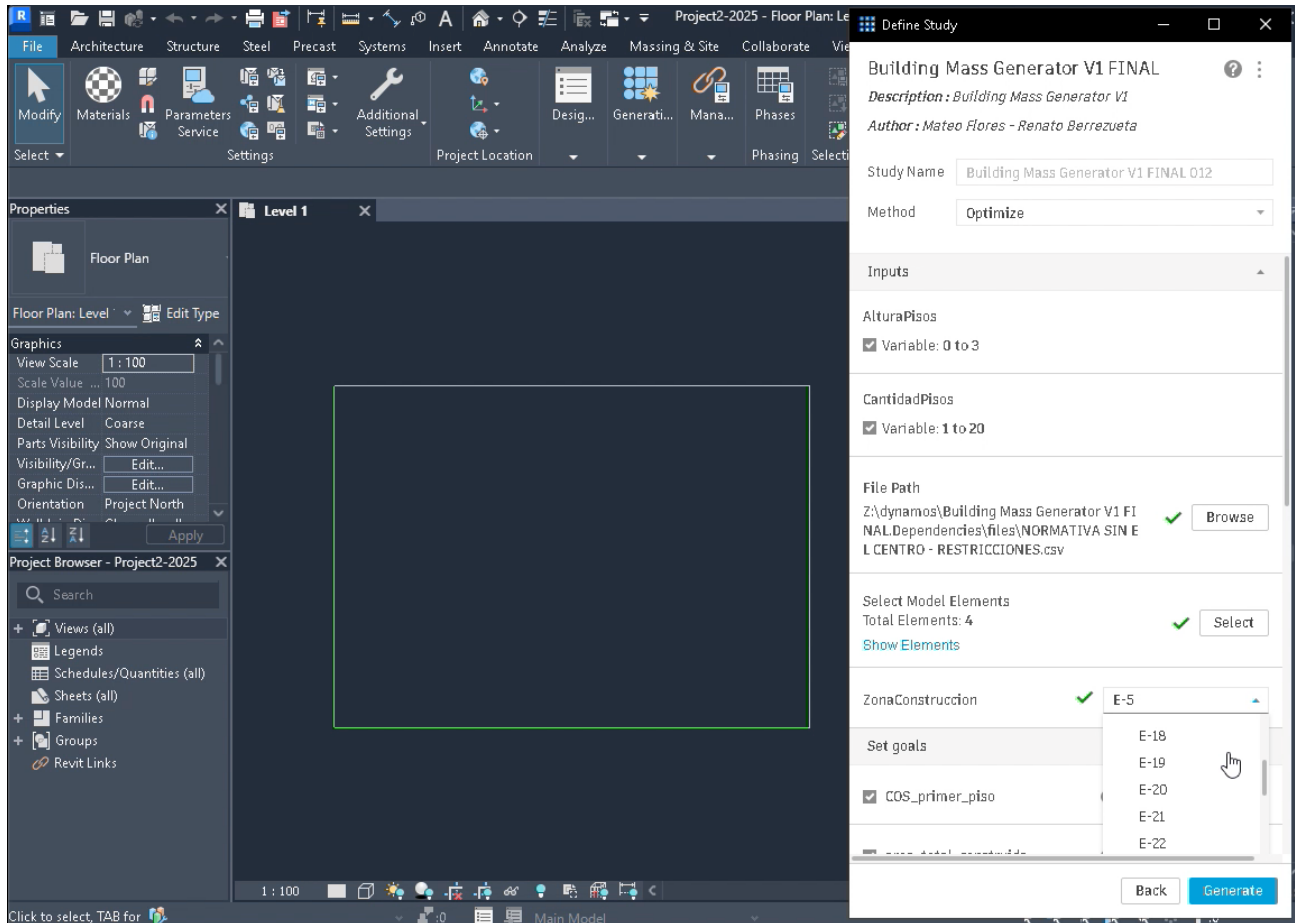


Figura 14: Menú de configuración del algoritmo generativo en Revit, mostrando las variables de entrada y opciones de configuración necesarias para optimizar la generación de volumetrías.

Luego, se seleccionó el archivo generado con las normas, el cual venía precargado por defecto y se eligió el terreno. En la figura 15, se presenta dicho proceso. Por último, se seleccionó la zona dentro de la ciudad de Cuenca donde se deseó generar la construcción.

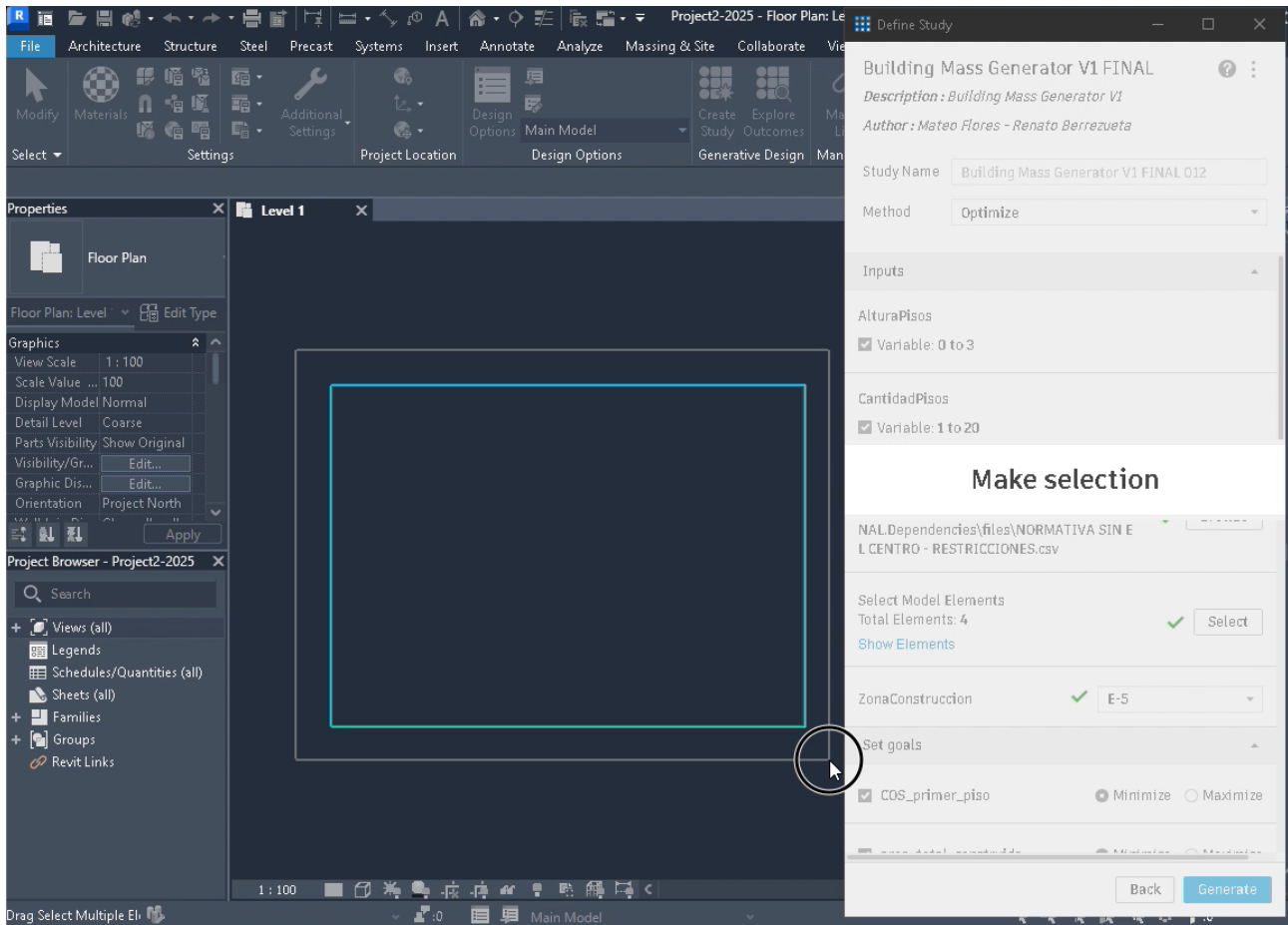


Figura 15: Selección del terreno en el menú de configuración del algoritmo generativo en Revit, mostrando la simplicidad del proceso.

Posteriormente, se configuraron los objetivos de diseño a ser optimizados por el algoritmo genético de Project Refinery. En la figura 16, se muestra dicha configuración. El generador se encarga de calcular estas métricas en cada diseño. Estas variables que vemos fueron calculadas por nuestro generador y marcadas como salidas del modelo.

La primera variable, COS, se deseaba maximizar, por lo que elegimos “Maximize”. Luego, teníamos el área total construida, que también se deseaba maximizar, así que la marcamos como “Maximize”. La siguiente variable, área provisional para ascensor, calcula un área provisional para el ascensor y su principal función es penalizar la creación de pisos, ya que se buscaba el menor número de pisos posibles por temas de costos, la marcamos como “Minimize”. La siguiente variable, área total terreno, no era útil para el algoritmo, pero sí como salida del modelo, por lo que la desmarcamos para que no la tomara en cuenta en la optimización. Luego teníamos el CUS, que buscábamos maximizar, y por ende, lo marcamos

como “Maximize”. Por último, pisos finales era una variable que mostraba la selección de pisos, útil como salida del modelo, pero no para los cálculos de optimización, ya que esta era la principal a modificar en busca de la optimización, por lo que la desmarcamos. Con esto, quedó lista la configuración del generador del modelo.

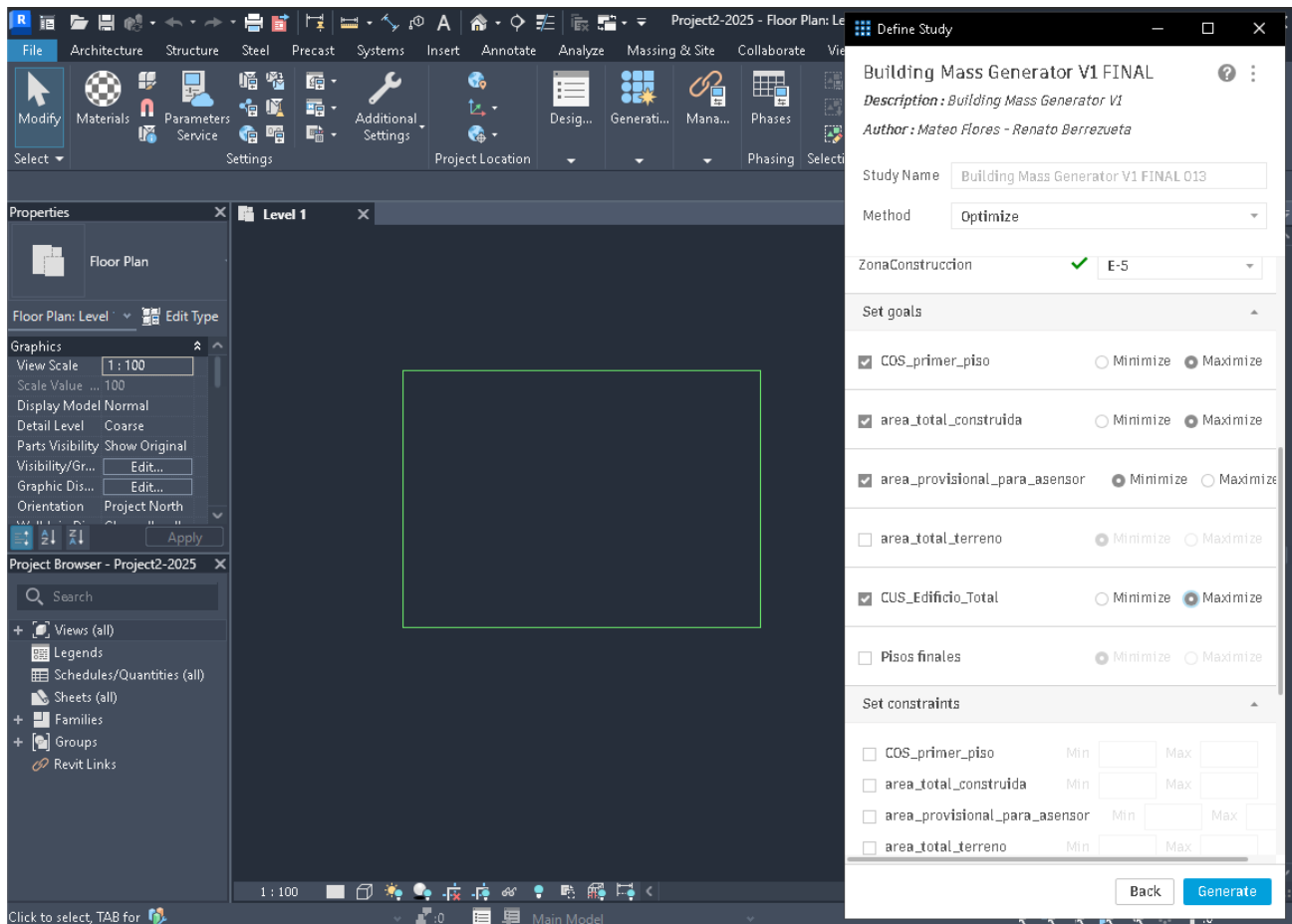


Figura 16: Configuración de los objetivos del algoritmo generativo para la generación de volumetrías en el estudio.

El siguiente paso fue el último de configuración y se dividía en dos partes. Primero, teníamos un sector llamado “Set Constraints”, el cual permitía establecer límites mínimos o máximos para las variables de salida del modelo. Por ejemplo, si el arquitecto solo deseaba soluciones con un mínimo de 50 de CUS y 3 pisos de altura, podía seleccionar CUS en las opciones y marcar el valor mínimo en 50, y lo mismo con los pisos.

Luego, configuramos el algoritmo generativo en términos de población y generaciones. En el apartado del marco teórico se explicó cómo estas variables influían en el algoritmo, pero de manera resumida, la población era la cantidad de individuos iniciales, por lo que a mayor

población, más soluciones cercanas a la óptima propuestas al final por el algoritmo. Por otro lado, las generaciones representaban cuántas veces deseábamos que estos individuos se cruzaran; a más generaciones, más ajustado sería el algoritmo y, por ende, las soluciones serían más cercanas a la óptima.

En la figura 17, se puede observar una configuración sin restricciones y con una población inicial de 20 individuos y 10 generaciones. También se puede observar que al final teníamos una semilla. Dado que este algoritmo tenía un componente aleatorio, el uso de esta semilla permitía generar repeticiones y controlar este suceso cuando deseábamos obtener la misma respuesta y cuándo queríamos obtener respuestas completamente nuevas para la misma configuración en varias ejecuciones. Por último, se debía hacer clic en "Generate".

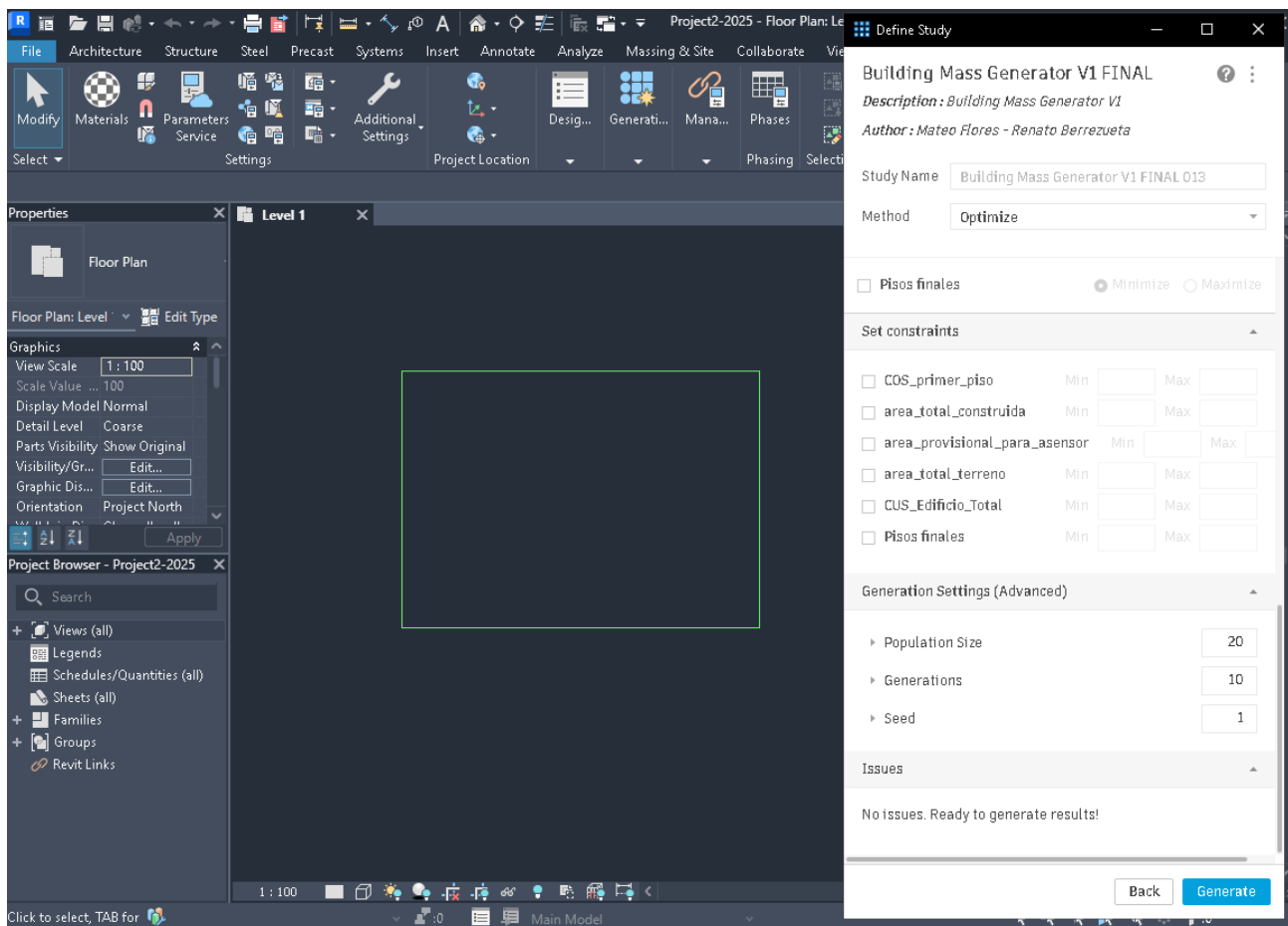


Figura 17: Configuración del algoritmo generativo y establecimiento de restricciones para el estudio de generación de volumetrías.

Después de realizar todo el proceso de configuración, se abrió automáticamente una pestaña donde pudimos observar generación tras generación las soluciones óptimas que el

algoritmo iba mostrando, como se ilustra en la figura 18. En esta misma figura, se puede observar que teníamos la opción de ordenar las soluciones propuestas según alguna métrica de salida del modelo. Para esta aprobación, necesitábamos seleccionar la solución con la mayor área total construida, por lo que elegimos esta métrica y la ordenamos de manera descendente.

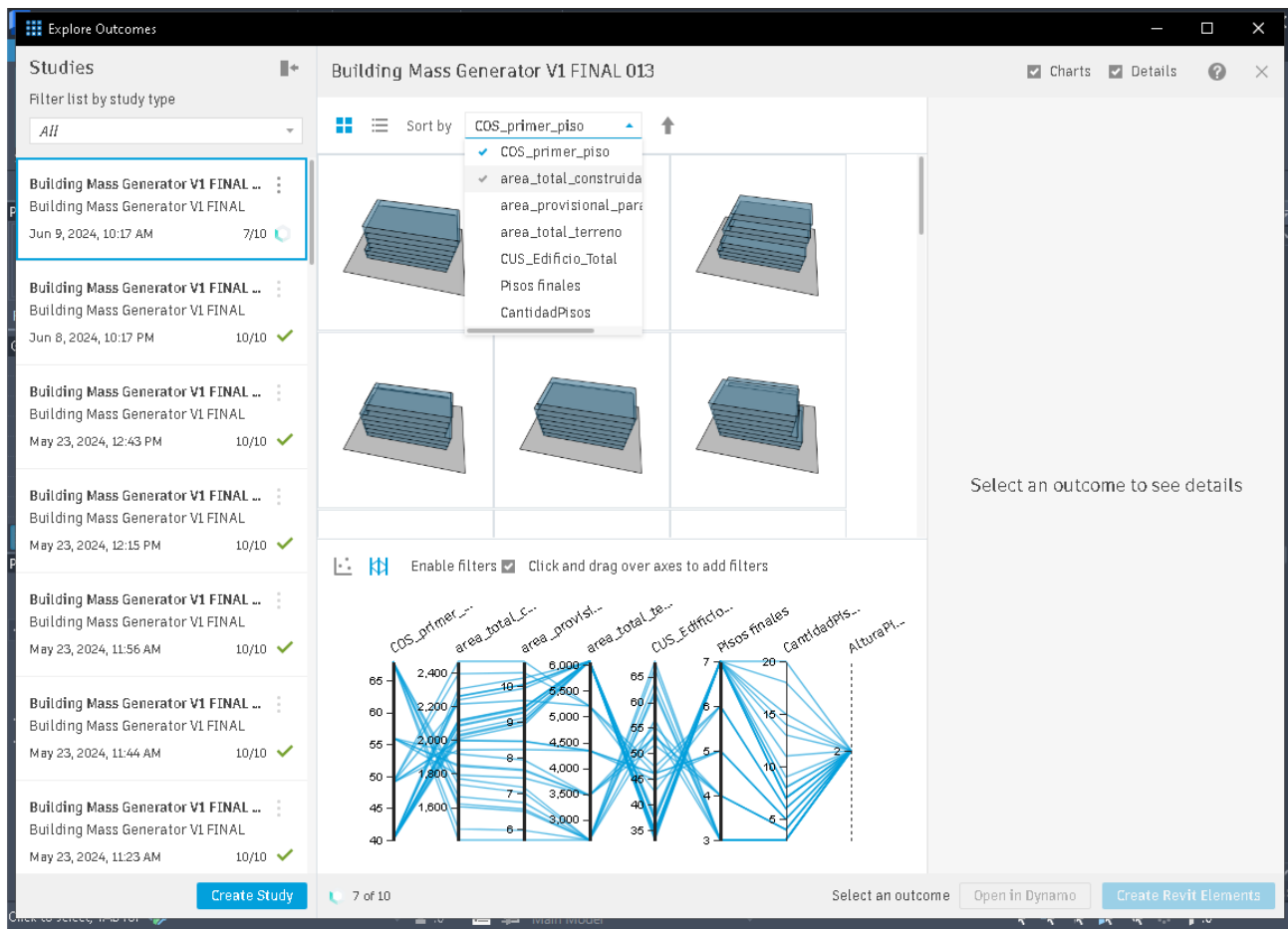


Figura 18: Interfaz de visualización de generaciones del algoritmo generativo, mostrando las soluciones óptimas para la aprobación, con opción de ordenar por métricas de salida del modelo.

El último paso en la extensión del código es la visualización de la solución más óptima para el terreno en cuestión. En esta ventana, la solución se muestra en la esquina superior izquierda. Al hacer clic en esta solución, en la parte derecha, podremos visualizar de manera más detallada la volumetría y las métricas finales que serán útiles para la aprobación del modelo. En la figura 19 se muestra esta ventana con las métricas y salidas del modelo. En esta sección derecha, con las métricas y salidas de la solución seleccionada, observamos

que los pisos finales son 6, maximizando el área construida. Sin embargo, esta zona permite hasta 7 pisos, y al considerar el cambio en los retiros, resulta más óptimo y proporciona una mayor área total construida realizar la edificación con solo 6 pisos, en lugar de los 7 permitidos como máximo. A partir de estas métricas obtenidas, extraemos el COS, CUS y el área total construida, y los agregamos al archivo Excel donde ya teníamos las dimensiones del terreno como nuevas columnas, marcando que provienen del resultado de la solución.

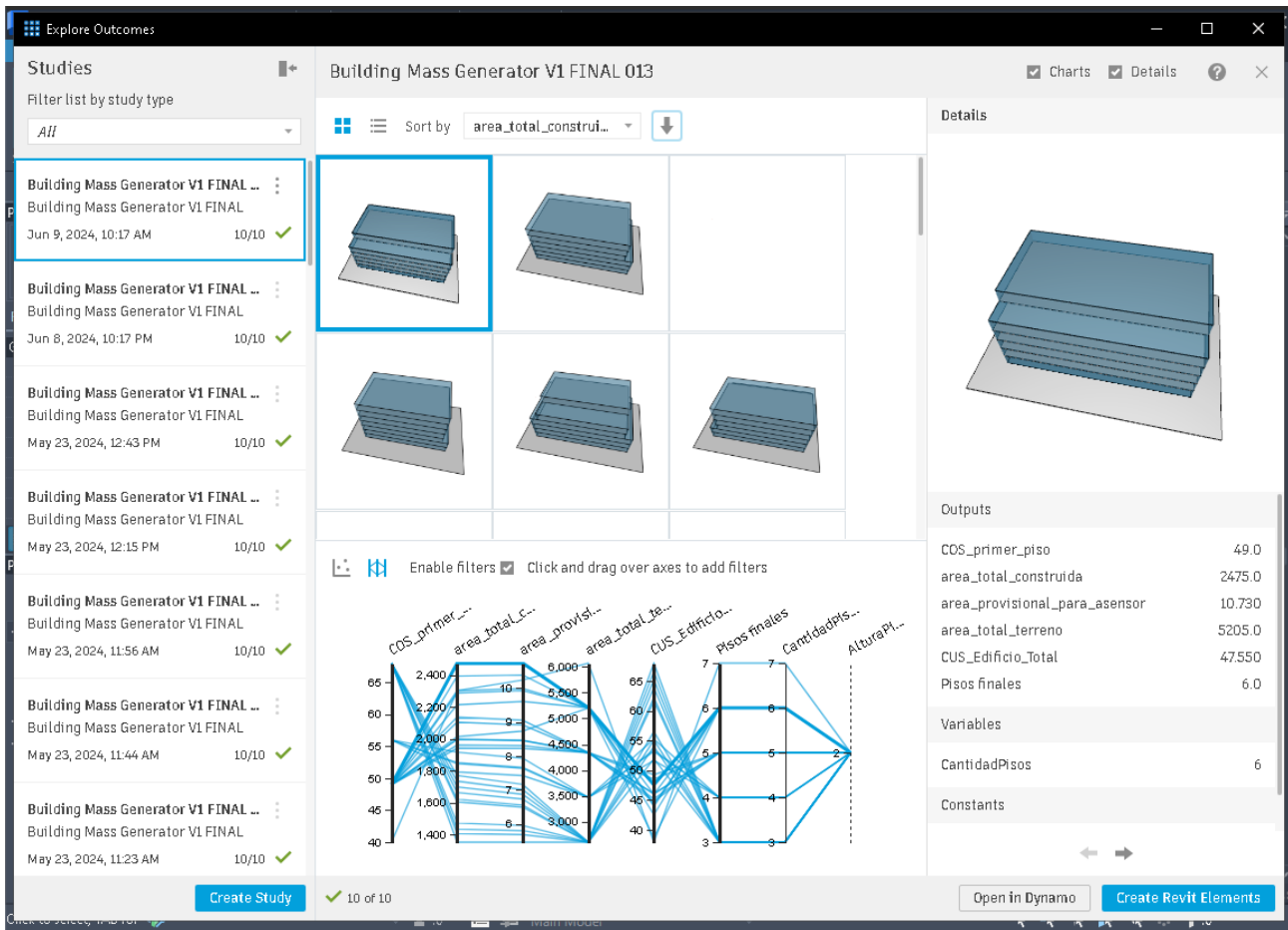


Figura 19: Interfaz de visualización de la solución óptima para el terreno seleccionado, con métricas y salidas del modelo detalladas, incluyendo número de pisos, COS, CUS y área total construida.

Dentro del levantamiento propuesto por los arquitectos, podíamos encontrar la volumetría final generada por ellos. En la figura 20, se mostraba esta volumetría final. En ella, encontramos las métricas de COS, CUS y el área total construida. Luego, transferimos estas métricas al Excel que estábamos generando, indicando que pertenecían a la volumetría final de los arquitectos. Finalmente, en el Excel, realizamos la resta entre el COS y CUS de la

extensión de código y los valores reales de la volumetría de los arquitectos.



Figura 20: Volumetría final generada por los arquitectos, con métricas de COS, CUS y área total construida.

Este proceso se repitió para las 40 volumetrías disponibles. El punto clave residía en la resta entre el COS y CUS de la extensión de código y los valores reales. Se buscaba que esta diferencia fuera siempre positiva, ya que indicaba que la herramienta obtenía mejores resultados que los entregados por el estudio, o al menos que tendiera a cero.

4.2.6. Evaluación empírica de la extensión de código

Con base en el trabajo de Bocco et al. se planteó el caso de estudio para evaluar la facilidad de uso y funcionalidad de la extensión de código.

Se definió como objetivo del estudio evaluar el rendimiento y la satisfacción de los profesionales de la construcción del estudio de arquitectura “A y D Estudio” con respecto al uso de la extensión de código propuesta para generación automática de volumetrías para proyectos de construcción en la ciudad de Cuenca.

La pregunta principal que guió el estudio es: “¿Qué nivel de intención de uso tiene la solución propuesta para el diseño de volumetrías en la ciudad de Cuenca, qué tan eficiente es y cuánto tiempo ahorra en un uso diario y real? ”

Extendiendo un poco más la pregunta de investigación, se buscó comprobar si la solución propuesta es eficiente, refiriéndose a qué tan buenos resultados ofrece y cómo maneja la

normativa de construcción de la ciudad de Cuenca. Por otro lado, también se buscó verificar si la extensión de código tiene una intención de uso real por parte del estudio de arquitectura. Finalmente, se evaluó cuánto tiempo se podría ahorrar en el estudio de arquitectura al usar esta extensión de código en las tareas de generación automática de volumetría. Debido a la complejidad de la pregunta principal, se decidió dividir la pregunta principal en tres sub preguntas independientes:

- PI1: ¿Cómo afecta el uso de la extensión de código de generación automática de volumetrías al tiempo en comparación con el método manual?
- PI2: ¿La extensión de código cumple con las expectativas de los usuarios en términos de eficiencia y cumplimiento de la normativa de construcción de Cuenca?
- PI3: ¿Los profesionales de la construcción están satisfechos con los resultados y recomendarían el uso de la extensión de código a otros colegas?

Con el fin de alcanzar los objetivos planteados y ser capaces de responder a las preguntas de investigación de una manera más clara, se formuló un conjunto de preguntas, las cuales fueron fundamentales para alcanzar el objetivo inicial:

- ¿Con qué frecuencia necesita generar volumetrías para proyectos de construcción? (relacionado con la PI1)
- ¿La extensión de código es intuitiva y fácil de usar? (relacionado con la PI2)
- ¿Cuánto tiempo le tomó aprender a usar la extensión de código eficientemente? (relacionado con la PI1)
- ¿La extensión de código cumple con sus expectativas en cuanto a la generación automática de volumetrías? (relacionado con la PI3)
- ¿Qué tan bien maneja la extensión de código la normativa de construcción de Cuenca en sus cálculos? (relacionado con la PI2)
- En comparación con el método manual, ¿cuánto tiempo ahorra al usar la extensión de código? (relacionado con la PI1)
- ¿Considera que la extensión de código maximiza el área total construida, el COS y el

CUS de manera más eficiente que el método manual? (relacionado con la PI2)

- ¿Qué tan satisfecho está con la extensión de código en general?(relacionado con la PI3)
- ¿Recomendaría esta extensión de código a otros profesionales del sector? (relacionado con la PI3)

A continuación, en la etapa de recolección de datos se generó una encuesta con las preguntas planteadas para cubrir la pregunta de investigación. La encuesta fue dirigida a los arquitectos miembros del estudio de arquitectura y permitió obtener datos cualitativos y cuantitativos que resumieron la experiencia de cada uno utilizando la extensión de código.

En primer lugar, se convocó a todos los arquitectos y se les brindó una breve capacitación sobre el uso de la extensión de código dentro del software Revit, así como los principales detalles sobre su funcionamiento interno. Posteriormente, se llamó individualmente a cada arquitecto y se les solicitó que utilizaran la aplicación de manera independiente, aplicando una serie de ejemplos de sitios de construcción en diferentes zonas de la ciudad. Una vez finalizada la prueba de la extensión de código, se realizó una encuesta de forma anónima con el fin de evitar sesgos y garantizar la sinceridad de las respuestas. Es importante destacar que la encuesta consistió únicamente en preguntas cerradas; no obstante, se recopiló retroalimentación adicional para futuros trabajos.

La encuesta completa realizada se resume en la Tabla 1.

No.	Pregunta
Preguntas genéricas	
1.	¿Cuál es su nivel de experiencia en el diseño y construcción de edificios? Alto – Medio – Bajo
Preguntas sobre Tiempo ahorrado	
2.	¿Con qué frecuencia necesita generar volumetrías para proyectos de construcción? 1 vez al mes – 2 veces al mes – 3 veces al mes – 4 veces al mes o más
3.	En comparación con el método manual, ¿cuánto tiempo ahorra al usar la extensión de código? (En minutos, ingresar los minutos en números sin nada extra) Cantidad de minutos ahorrados
4.	¿Cuánto tiempo le tomó aprender a usar la extensión de código eficientemente? 5-10min – 10-15 min – 15-20min – +20min
Preguntas sobre eficiencia y Normativa	
5.	¿La extensión de código es intuitiva y fácil de usar? Sí – No
6.	¿La extensión de código cumple con sus expectativas en cuanto a la generación automática de volumetrías? Sí – No
7.	¿Qué tan bien maneja la extensión de código la normativa de construcción de Cuenca en sus cálculos? Alto – Medio – Bajo
8.	¿Considera que la extensión de código maximiza el área total construida, el COS y el CUS de manera más eficiente que el método manual? Totalmente en desacuerdo – En desacuerdo – Neutral – De acuerdo – Totalmente de acuerdo
Preguntas sobre satisfacción e intención de uso	
9.	¿Qué tan satisfecho está con la extensión de código en general? Nada satisfecho – No satisfecho – Neutral – Satisfecho – Totalmente satisfecho
10.	¿Recomendaría esta extensión de código a otros profesionales del sector? Sí – No

Tabla 1: Preguntas formuladas para la encuesta de evaluación empírica empleada a los miembros del estudio de arquitectura “A y D Estudio”.

4.3. Desarrollo de la extensión de código para la planificación automática de espacios

En esta sección se resaltan todos los procesos seguidos para la construcción, validación y evaluación de la extensión de código para la planificación automática de espacios de planta.

En la Figura 21 se adjunta un diagrama que proporciona una vista general de los procesos involucrados antes de sumergirse en la explicación detallada de cada uno de ellos.

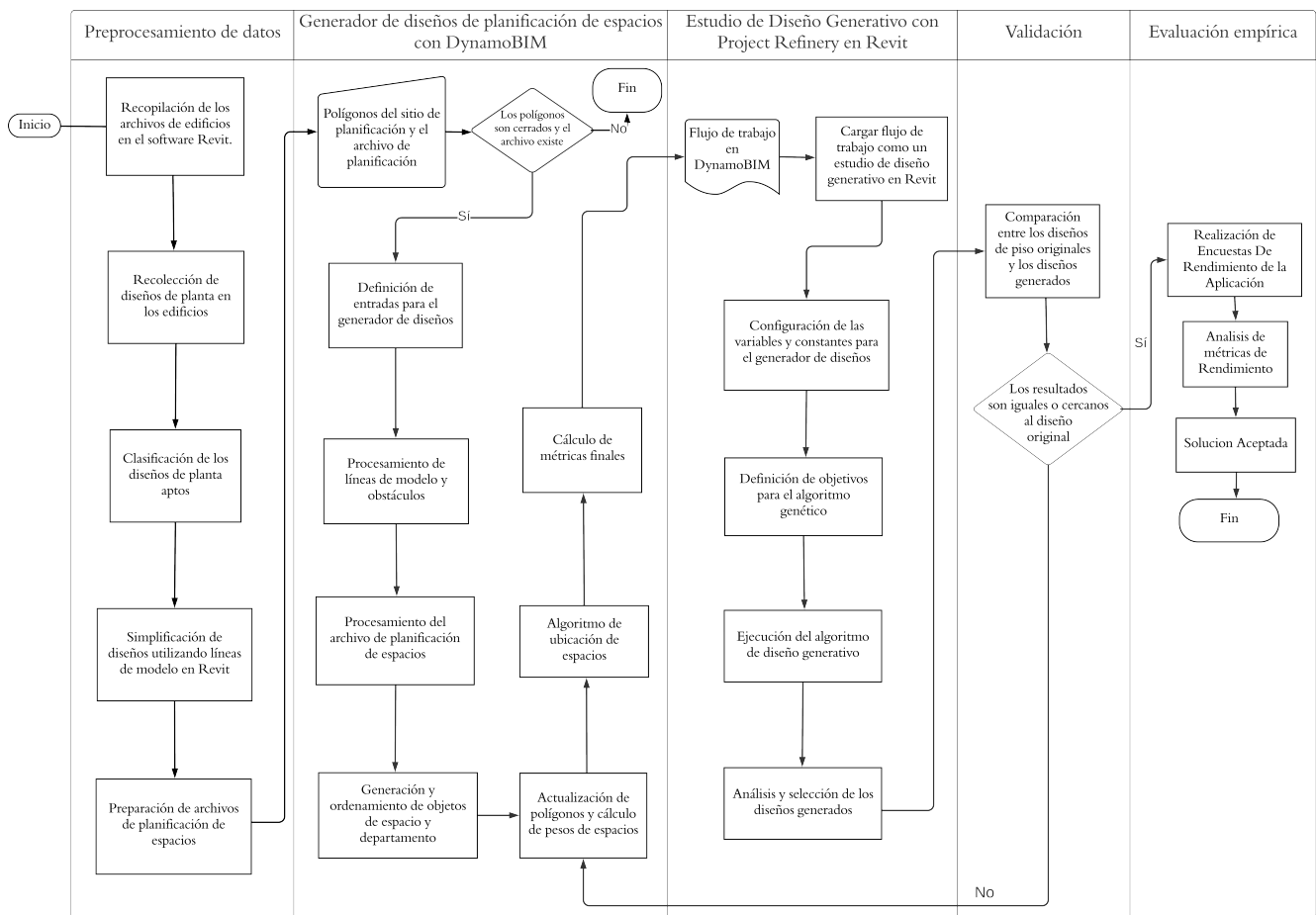


Figura 21: El diagrama de flujo describe el proceso de generar y validar diseños de planificación de espacios usando DynamoBIM y Project Refinery en Revit. Inicia con la recopilación de datos, sigue con la generación de diseños por algoritmos de ubicación de espacios, los compara con los originales, y si son satisfactorios, realiza una evaluación empírica final para validar la solución.

4.3.1. Recopilación de datos de planes de espacio

En esta etapa, se recolectaron los diseños de planta disponibles dentro de los proyectos arquitectónicos ya ejecutados por el estudio de arquitectura. En total, se contaba con 40 edificios que contenían la distribución parcial de los espacios planificados, ya que en su mayoría se tenían maquetados de la volumetría de los edificios. Estos diseños se habían realizado con el software de diseño arquitectónico Autodesk Revit, y de estos edificios se seleccionaron las plantas y distribuciones de departamentos que se encontraban finalizadas y no eran repetidas, ya que el conjunto de diseños no contemplaba todo el ciclo de vida

del proyecto y el estudio tenía su propia plantilla de diseño de distribuciones para departamentos. Al final, se contó con 10 diseños de distribución de espacios en pisos de plantas y departamentos en edificios aptos.

A continuación, utilizando el software Revit, se simplificaron los diseños utilizando líneas de modelo agrupadas y etiquetadas para generar los polígonos. De esta manera, se representó tanto la silueta del sitio de planificación como los espacios ubicados y los obstáculos correspondientes a pasillos, escaleras, ascensores, columnas, entradas, salidas y el resto de espacios no ocupables.

Cabe recalcar que, para los espacios, se restringió el uso de formas de 4 lados. Esta decisión se debió a que el generador implementado se limitaba a generar únicamente este tipo de formas poligonales. La complejidad de generar formas con más de 4 lados que cubran un área específica representaba una limitación en el alcance del algoritmo utilizado. En la Figura 22 se muestra la representación simplificada de un plan de espacio disponible del conjunto de edificios.

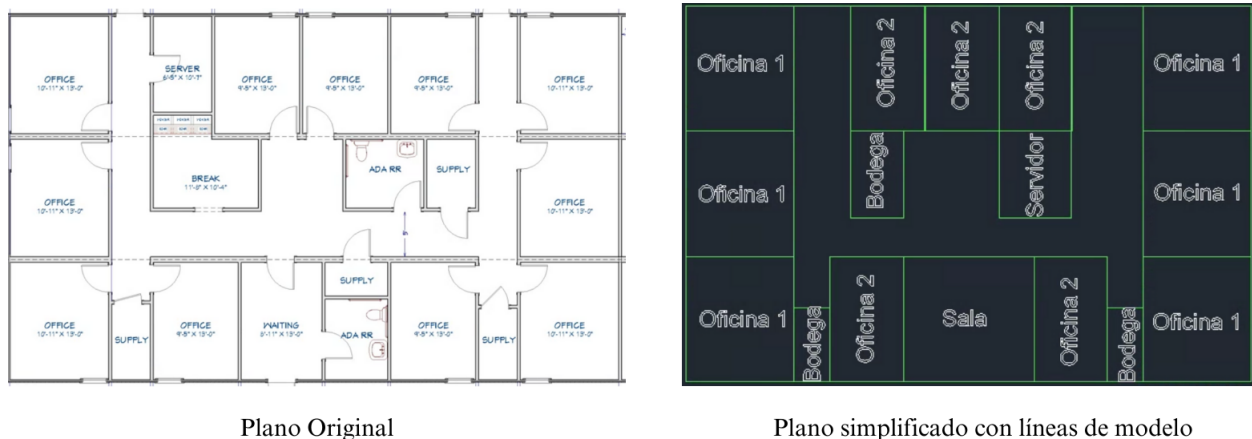


Figura 22: Representación simplificada de un plano de espacio. Cada espacio corresponde a un polígono independiente con sus etiquetas y sus relaciones.

Estas representaciones son la base para iniciar el proceso de creación del generador aleatorio de diseños de planificación y la comparación con los diseños generados por el proceso de diseño generativo.

Adicionalmente, se prepararon los archivos de planificación de espacios correspondientes a cada diseño disponible. Cada archivo corresponde a un archivo de valores separado por

coma (CSV) el cual detalla los espacios que se buscan ubicar dentro del sitio de planificación y constó de las siguientes columnas:

- ID: corresponde al identificador del espacio, este es un valor numérico que se debe asignar en orden según como se especifiquen los espacios, esto es importante, ya que al momento de generar las estructuras de datos los identificadores se utilizan como índices de posición dentro de listas ligadas.
- NOMBRE ESPACIO: corresponde al nombre que se le asigne a cada espacio, este nombre posteriormente se muestra el espacio al momento de generar cada diseño.
- NOMBRE DEPARTAMENTO: corresponde al nombre del departamento al que pertenece el espacio, un departamento puede estar compuesto por más de un espacio. Con esto se logran organizar los espacios en bloques diferentes.
- CANTIDAD: indica la cantidad de espacios con las mismas características especificadas que se buscan ubicar en el sitio de planificación, es un valor entero que parte de 1 hasta el valor deseado.
- AREA: indica el área en metros cuadrados que ocupa cada espacio del documento de planificación.
- VALOR PREFERENCIA: es un valor en la escala de 1 a 10 que indica como tal lo preferible que es agregar un espacio, un valor más alto le da al espacio mayor prioridad para ser ubicado sobre los espacios con valores de prioridad más bajos.
- TIPO ESPACIO: se especifican dos tipos de espacios, el espacio Principal identificado con la letra mayúscula P, y el espacio Regular identificado con la letra mayúscula R. La diferencia principal entre estos tipos de espacios radica en que al espacio principal se le da mayor prioridad de asignación sobre los espacios regulares, además de que los espacios principales deben tener predefinidas las dimensiones para cubrir el área especificada del espacio. Por otro lado, los espacios regulares no tienen ninguna prioridad adicional y las dimensiones se las asigna de manera aleatoria dentro del generador de diseños.

- ADYACENCIA ID: corresponde al identificador de cada espacio que se busca que esté cerca del espacio indicado. Cada identificador se debe separar por un punto y se puede indicar la adyacencia con los espacios que se deseen.
- ANCHO: este campo está reservado para los espacios principales y define el ancho en metros para este espacio.
- ALTO: este campo está reservado para los espacios principales y define el largo en metros para este espacio.

Es importante definir correctamente todos los campos en el documento de planificación, ya que cualquier error en el mismo involucra un mal funcionamiento del generador.

4.3.2. Selección de las métricas relevantes para la evaluación de los diseños generados.

En la planificación de espacios, la consigna principal fue tener la mayor ocupación y aprovechamiento del espacio posible. Debido a esto, se plantearon como objetivos a maximizar el porcentaje de espacio del sitio de planificación ocupado y el porcentaje de espacios que se especificaron en el documento de planificación y que fueron colocados dentro del sitio de planificación.

Además, se buscó que los diseños generados respetaran la adyacencia y prioridad que se les había asignado. Por este motivo, se planteó una métrica que ponderaba la preferencia, adyacencia y el área de cada espacio, la cual debía maximizarse para así garantizar la asignación de espacios prioritarios antes que espacios de menor jerarquía.

Finalmente, la circulación fue un factor clave en el proceso de diseño de distribución de espacios. Se buscó una distancia de circulación mínima entre los espacios definidos y los respectivos obstáculos del sitio de planificación, ya que estos obstáculos delimitaban la circulación en el sitio y funcionaban como puntos estratégicos.

4.3.3. Desarrollo del generador de diseños de planificación de espacios con Dynamo-BIM

En primer lugar, se definieron las entradas para el generador de diseños en el software DynamoBIM. Se estableció la ruta del archivo de planificación de espacios. Luego, utilizando las librerías de conexión con Revit y las funciones de selección de elementos, se escogieron todos los grupos de líneas de modelo que representaban el sitio de planificación y sus obstáculos. Previamente, en Revit, se había asignado el nombre de “sitio de planificación” al grupo de líneas de modelo que formaban la silueta del sitio de planificación, para poder diferenciarlo de los grupos de obstáculos.

Posteriormente, se utilizó una variable entera en un rango de 1 a 1000000, esta definiría la semilla de generación del diseño. Además, se utilizaron dos variables reales en un rango de 0 a 10 y un paso de 0.01, para definir tanto el ancho mínimo de cada espacio a generar como la distancia mínima entre espacios colindantes, ambos en metros. Finalmente, se definió una variable entera en un rango de 1 a 100 para especificar el número máximo de iteraciones que se ejecutarían para la ubicación de los espacios en el sitio de planificación.

Por otro lado, se procesaron las líneas de modelo empleadas para el sitio de planificación, así como los obstáculos identificados. En primer lugar, se separaron los grupos de obstáculos de la silueta del sitio de planificación. Para esta última, se extrajeron los miembros que la conformaban, es decir, las líneas de modelo. Mediante los métodos del objeto de línea de modelo de DynamoBIM, se obtuvieron los puntos que componen dichas líneas, los cuales se enlistaron utilizando el lenguaje de programación Python en forma de una lista de tuplas, donde cada tupla almacena las coordenadas X, Y de cada punto.

A su vez, para cada uno de los obstáculos, se llevó a cabo un proceso similar. Al final, se agrupó en una sola lista a cada lista de tuplas que contiene las coordenadas de cada obstáculo. Este procesamiento se realizó para posteriormente ser utilizado en el algoritmo de ubicación de espacios.

En cambio, el archivo de planificación de espacios se procesó fila por fila para generar la estructura de datos que organizaba los espacios y departamentos. Para ello, se desarrollaron

clases específicas para Departamentos y Espacios, así como las funciones necesarias para la organización de dicha información en el lenguaje C#. Posteriormente, estos elementos fueron importados como módulos dentro de DynamoBIM. De cada fila, se extrajo la información de cada espacio y departamento para ser agregada a sus clases correspondientes.

El módulo de departamento resume la información del mismo. En él, se almacenan los valores de nombre, área total que ocupaba (suma de áreas de todos los espacios que contiene), tipo de departamento (correspondiente al tipo de espacios que guardaba), y los espacios que pertenecían a dicho departamento.

Por su parte, el módulo de espacio resume la información del mismo. En él, se almacenan los valores de identificador, nombre, área ocupada, tipo de espacio, el ancho y alto si aplicaba, el valor de preferencia, la lista de adyacencia con otros espacios, el departamento al que pertenecía y la cantidad de espacios de ese tipo.

Al momento de crear el objeto de espacio, se definió el ancho y largo que ocupa, y se le asignó el polígono inicial de su forma con base en las coordenadas de las cuatro esquinas, partiendo del origen y utilizando la clase PolyCurve de DynamoBIM. Para ello, se comprobó el tipo de espacio, y en caso de ser un espacio principal, se registró el valor de ancho y largo cargados. Para el caso de los espacios regulares, se realizó la selección aleatoria de estos valores utilizando la semilla de generación de diseños. Como base, se tomó el valor del ancho del espacio de planificación. Este ancho se podría encontrar en un rango entre el valor de ancho mínimo establecido en las entradas del generador y el valor de la raíz cuadrada del área del espacio. Así, la altura se calculó basándose en el ancho seleccionado en el rango mencionado y el área ya definida.

A continuación, se generó y almacenó el valor de peso de adyacencia de cada espacio tomando la lista de identificadores de espacios adyacentes definida para cada espacio del documento de planificación, este peso se definió realizando el conteo de veces que cada espacio constó en la lista de adyacencia de otro espacio. Al final, el conteo para cada espacio se normalizó en una escala de 0 a 10.

Enseguida, se obtuvieron los departamentos más frecuentes a partir de la lista de espacios

y departamentos. Para ello, primero se creó una lista de adyacencia de los nombres de departamentos asociados a cada espacio. Luego, se contó la frecuencia de aparición de cada departamento adyacente, eliminando la autoadyacencia y la adyacencia a los departamentos principales. Finalmente, se ordenó la lista de nombres departamentos más frecuentes de forma descendente.

A su vez, esta lista permitió ordenar los objetos de departamento en base a la frecuencia encontrada, además de calcular el valor de peso de adyacencia para cada departamento. Para ello, se recorrió la lista de departamentos más frecuentes y se asignó un peso elevado (10,000,000) a los departamentos que se encontraban en esta lista, con un valor decreciente (múltiplo de 1,000) según su posición en la lista.

Posteriormente, se recorrió la lista de datos de departamentos y se asignó el peso correspondiente a cada departamento. Si no se encontraba el departamento en la lista de preferidos, se le asignó un peso de 0. Para calcular el peso combinado de cada departamento, se consideraron dos factores con base en la metodología de Das et al. (2016):

Surplus: Si el departamento es de tipo Principal, se le asigna un valor muy alto (1,000,000,000 + un valor aleatorio entre 10 y 50). Si es de tipo Regular, se le asigna el peso de adyacencia calculado anteriormente.

Peso final del departamento: Se calculó un valor que consideraba el 25 % del área necesitada por el departamento más el surplus calculado anteriormente.

Este peso combinado final se actualizó para cada departamento y se ordenó la lista de instancias de departamento en orden descendente, para mantener por encima a los departamentos con mayor peso.

Con la lista de departamentos ordenada se procedió a ordenar el conjunto de espacios contenidos en cada departamento. Para ello, de cada departamento, se obtuvo la lista de espacios almacenados y se creó una lista de estructuras que contenían el objeto de espacio y un valor de peso combinado de adyacencia calculado de la siguiente manera:

Peso Combinado de adyacencia = Valor de preferencia del espacio + α + Peso de adyacencia del espacio.

El valor de alpha se inicializa en 0 se incrementaba en 0.01 para cada espacio.

Finalmente, la lista de estructuras se ordenó de manera descendente basándose en el valor del peso combinado de adyacencia y se extrajo cada espacio de esa estructura para almacenarlo de manera ordenada en el objeto de departamento específico. Se recorrió la lista de espacios y se les asignó un identificador propio a cada uno correspondiente a su posición en la lista.

La lista de espacios ordenada es fundamental para dar prioridad a los espacios con base en su adyacencia con otros espacios, área, frecuencia de aparición y tipo.

A continuación, se actualizaron los polígonos generados para cada espacio y se les aplicó un desplazamiento (offset) en todos sus lados, con un valor correspondiente a la distancia mínima entre espacios establecida en los parámetros de entrada del generador. Después, se generó una lista que resumía, para cada espacio, la lista de coordenadas de las esquinas de su polígono de forma, el valor de peso del espacio, el valor de área del polígono y el identificador propio para cada espacio.

Esta lista, en conjunto con la lista de puntos correspondientes al sitio de planificación y a los obstáculos, el valor del deslizador de número de iteraciones y la semilla de diseño, fueron entradas para el algoritmo de ubicación de espacios realizado en el lenguaje de programación Python.

El algoritmo de ubicación de espacios fue una adaptación del algoritmo codicioso propuesto por Espín et al. (2020), el cual se caracterizó por seguir una estrategia voraz a la hora de colocar los espacios en el sitio de planificación. En primer lugar, se establecieron las representaciones geométricas del sitio de planificación y sus obstáculos usando multipolígonos con la librería Shapely de Python y las coordenadas enviadas en los parámetros. Con esto, se delimitó un polígono irregular y sus agujeros correspondientes a cada obstáculo definido. De igual forma, se utilizaron polígonos de Shapely para representar a cada espacio a ubicar en el sitio de planificación.

En cuanto al funcionamiento, en cada iteración, el algoritmo seleccionó de manera aleatoria un espacio que aún no se había colocado, dando mayor prioridad a aquellos espacios con

un mayor “puntaje codicioso”. Este puntaje se calculó en función del valor, el peso y el área de cada espacio, de modo que se favorecía la selección de espacios valiosos, pero se penalizaba su peso y área, ya que estos dos últimos eran recursos limitados en el sitio de planificación.

Una vez seleccionado el espacio, el algoritmo intentaba colocarlo en una posición y orientación aleatorias dentro del sitio de planificación, siempre y cuando dicha colocación no generara superposiciones con otros espacios o excediera el área disponible en el sitio de planificación. Si la colocación era válida, el espacio se agregaba al sitio de planificación; en caso contrario, se descartaba. Este proceso se repetía hasta agotar los intentos máximos de colocación (el autor marcaba que la convergencia se aseguraba con un máximo de 100,000 iteraciones con una parada temprana sin cambios de máximo 30,000 iteraciones), se haya alcanzado la capacidad del sitio de planificación o no queden más espacios por intentar colocar.

Al finalizar la ejecución del algoritmo, se obtuvieron los polígonos que representan los espacios colocados en su posición definitiva, junto con el identificador correspondiente a cada espacio. Posteriormente, se procedió a reasignar estos identificadores al objeto que representa cada espacio. Luego, se eliminó el desplazamiento establecido en los parámetros de entrada del generador, con el objetivo de separar adecuadamente los espacios colindantes.

Finalmente, se asignó un color a cada superficie del espacio, siguiendo un criterio booleano: con un valor “falso” cada espacio tendrá un color, con un valor “verdadero” cada espacio tendrá el color del departamento al que pertenece. Además, de agregar el nombre de cada espacio sobre su superficie.

Una vez definido el diseño aleatorio, se procedió a extraer las métricas previamente establecidas. La primera de ellas fue el porcentaje de área utilizada, que consistió en la proporción entre la suma del área de cada espacio colocado y el área total disponible en el sitio de planificación.

La segunda métrica fue el porcentaje de espacios colocados, calculado como la proporción entre la cantidad de espacios colocados y la cantidad de espacios definidos en el archivo

de planificación.

La tercera métrica fue el peso combinado, obtenido mediante la suma del peso combinado de cada espacio colocado.

Finalmente, se calculó la métrica de circulación entre espacios y obstáculos utilizando el siguiente proceso. Primero, se utilizó la librería Space Analysis disponible para DynamoBIM la cual fue desarrollada por Goldstein et al. De esta librería se utilizó lo que el autor denomina Space Lattice. Este es una estructura de datos de cuadrícula que permite generar trayectorias y calcular distancias entre puntos de un espacio delimitado con obstáculos. Aplicando el algoritmo de Dijkstra sobre esta estructura de datos, se generaron las trayectorias más cortas entre los puntos medios de los lados de cada espacio y los puntos medios de los lados de cada obstáculo. Las trayectorias calculadas no atravesaban los límites del resto de espacios y obstáculos del sitio de planificación. Posteriormente, se calculó el promedio de distancia de todas las trayectorias encontradas entre todos los espacios y obstáculos. Este valor fue asignado como la métrica de circulación entre espacios y obstáculos.

A continuación, en la Figura 23 se presenta un diagrama de flujo que facilita la comprensión del proceso que sigue el generador.

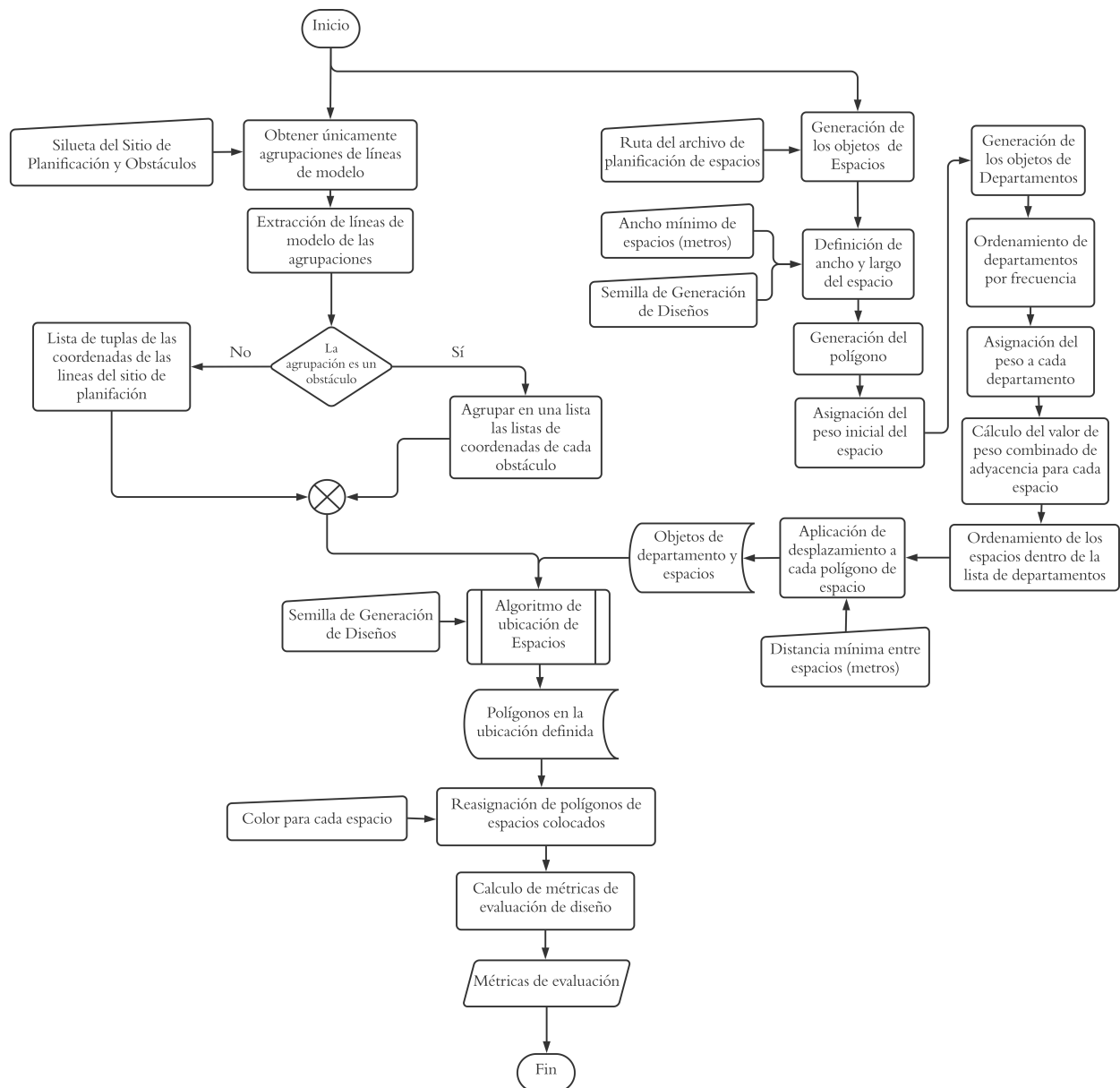


Figura 23: El diagrama de flujo resume el proceso de generación de diseños de distribución de espacios. Comienza con la extracción de datos del sitio y la generación de objetos de espacios y departamentos. Estos se procesan y organizan para alimentar el algoritmo de ubicación de espacios, que calcula nuevas posiciones. Por último, se evalúan las métricas definidas para el diseño resultante.

4.3.4. Aplicación del Algoritmo Genético con Project Refinery

El flujo de trabajo en DynamoBIM que resultó de la aplicación de la metodología se guarda configurándolo como un grafo de diseño generativo, las dependencias utilizadas en el grafo se almacenaron en una carpeta independiente, y en este caso específico se publicó el paquete generado con C# como un archivo de extensión .dll (Dynamic Link Library), ya que

corresponde a bibliotecas de código usadas en software para el sistema operativo Windows.

De esta manera, se definió la extensión de código para usar en Project Refinery dentro de Revit.

Dentro del software Revit, se definieron los grupos de líneas de modelo del sitio de planificación a utilizar, así como los obstáculos presentes. Para aplicar el diseño generativo, se utilizó la herramienta “Crear un estudio generativo” dentro de Revit. En esta herramienta, se seleccionó la carpeta donde se encontraba el archivo de flujo de trabajo de DynamoBIM y la carpeta de dependencias.

Con el generador cargado, se escogieron las entradas para el diseño generativo previamente especificadas. Es importante mencionar que, para este generador, el único parámetro variable fue la semilla de generación de diseños, ya que el ancho mínimo de espacios, la distancia mínima entre espacios, el número de iteraciones del algoritmo y la decisión de asignar o no un color a cada espacio se mantuvieron constantes para garantizar el adecuado funcionamiento del generador.

Finalmente, se definieron los objetivos a cumplir. Por un lado, se buscó maximizar el puntaje de peso combinado total, el porcentaje de área del sitio de planificación utilizado y el porcentaje de espacios colocados. Por otro lado, se buscó minimizar el puntaje de circulación entre espacios y obstáculos.

Una vez establecidas las configuraciones iniciales, se definió una población inicial para el estudio, así como el número de generaciones de cruce y mutación, y la semilla correspondiente. Tras ejecutar el algoritmo, se obtiene como resultado un conjunto de n diseños de planes de espacio que buscan satisfacer los requerimientos establecidos. El valor de n dependerá de la combinación de población y generaciones seleccionadas, mientras mayor sea la población inicial mayor exploración del universo de soluciones habrá pero con un mayor gasto de recursos. Posteriormente, estos diseños de espacios se ordenan de menor a mayor según la métrica deseada. Al seleccionar cada uno de ellos, se pueden visualizar más detalles referentes a las métricas evaluadas y los valores de parámetros utilizados para obtener ese diseño en particular.

4.3.5. Aprobación de las soluciones de la extensión de código

Antes de proceder con la evaluación empírica de la extensión de código, fue necesario asegurarse de que sus resultados fueran adecuados. En este apartado, se llevó a cabo un procedimiento sencillo que se detalla a continuación.

En primer lugar, en base a lo especificado en la etapa de recolección de datos se refinaron y procesaron los 10 diseños de distribución de espacios en pisos de plantas y departamentos de edificios que contenían la información completa, junto con sus representaciones simplificadas en forma de polígonos del sitio de planificación y los obstáculos correspondientes, así como los archivos de planificación asociados a cada diseño.

Posteriormente, utilizando la funcionalidad Project Refinery dentro de Revit 2025, se seleccionó la opción “Crear estudio” en la sección “Gestionar”. Luego, en la ventana de estudios generativos disponibles, se eligió el estudio de la extensión de código encargada de la generación de diseños de planificación de espacios, denominada “PlanificadorEspacios2025”. La Figura 24 ilustra los pasos descritos anteriormente.

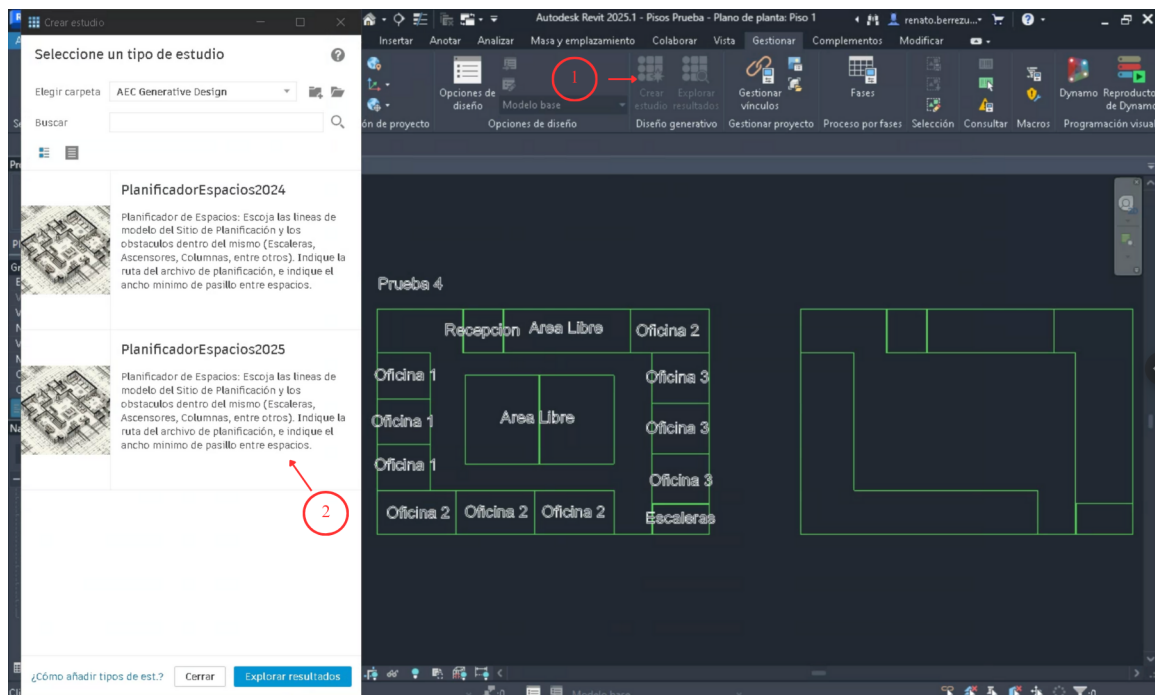


Figura 24: Selección del estudio de diseño generativo realizado con la extensión de código utilizando Project Refinery. Se siguieron los pasos numerados.

Tras esto se realizó la selección del tipo de estudio, todas las pruebas fueron realizadas

con el método de optimización (Optimizar) el cual ejecuta el algoritmo genético sobre el flujo de trabajo de la extensión de código. Aunque, en primer lugar se definió cada constante y variable del problema, y los objetivos de optimización. Las Figuras 2 y 3 resumen las configuraciones generales para todas las pruebas de aprobación realizadas.

Entradas			
Nombre	Tipo	Valor	Descripción
Sitio de planificación (Elementos)	Constante	Elementos de Revit	Elementos de Revit correspondientes al sitio de planificación y obstáculos
Archivo de Planificación de Espacios	Constante	Ruta de archivo de planificación de espacios	Archivo CSV con la configuración de espacios
Ancho mínimo de espacios (Metros)	Constante	1.9	Ancho mínimo para los espacios regulares a generar
Separación mínima entre espacios (Metros)	Constante	0	Separación aplicada a espacios colindantes
Mostrar cada Departamento	Constante	False	Valor booleano para definir si colocar un color a cada espacio (False) o departamento (True)
Número iteraciones algoritmo ubicación de espacios	Constante	10	Número de iteraciones máximas realizadas por el algoritmo de ubicación de espacios, se definieron 10 iteraciones ya que los resultados a partir de ese número no varían lo suficiente
Semilla de generación de diseños	Variable		Valor de semilla que cambiará aleatoriamente internamente en el programa

Tabla 2: Configuración de las entradas del programa para el estudio generativo.

Definir Objetivos		
Nombre	Objetivo	Descripción
Valor de Peso Combinado de Espacios Asignados (Maximizar)	Maximizar	Peso combinado de preferencia de espacio, adyacencia con otros espacios y área
Porcentaje de Área Utilizada (Maximizar)	Maximizar	Porcentaje de área ocupada del sitio de planificación
Porcentaje de Espacios Colocados (Maximizar)	Maximizar	Porcentaje de espacios colocados con base en el documento de planificación.
Métrica Distancia Rutas (Minimizar)	Minimizar	Distancia promedio entre espacios y obstáculos, se busca minimizar la distancia de circulación.
Área del sitio de planificación	No marcado	Salida informativa que muestra el área del sitio de planificación
Generación del Algoritmo Genético		
Nombre	Valor	Descripción
Tamaño de población	100	Se genera el estudio con 100 aleatorios iniciales
Generaciones	20	Se llevan a cabo 20 etapas de cruce y mutación
Semilla	2024	Semilla de generación de números aleatorios

Tabla 3: Configuración de los objetivos, los cuales se optimizan simultáneamente en cada iteración, y la generación del programa para el estudio generativo.

Los resultados de diseño se presentaron en la ventana de Project Refinery, en la Figura 25

se resumen todas las características del estudio.

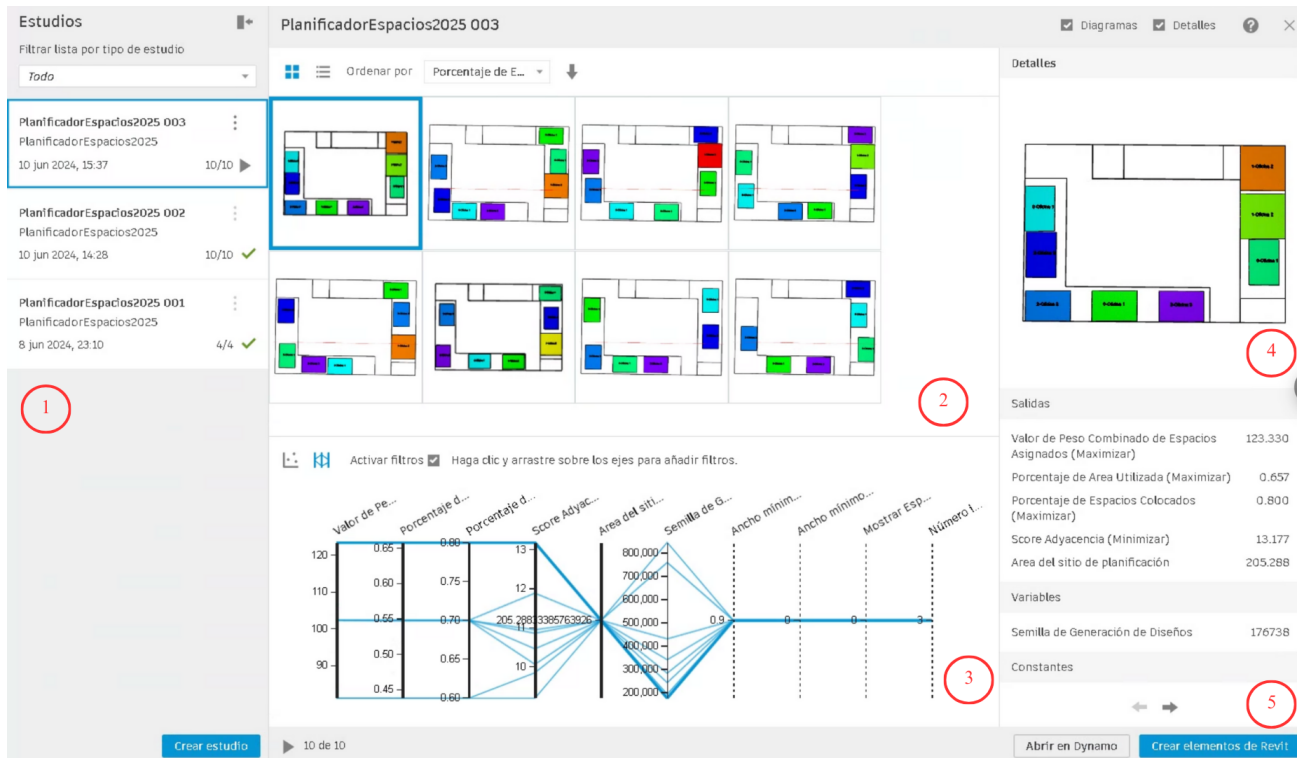


Figura 25: Interfaz del estudio. Se incluyen los estudios generados, los diseños del generador, los valores de variables, un diseño seleccionado y las métricas de cada diseño.

Según se observa en la Figura 25, el módulo Project Refinery presenta la siguiente estructura y funcionalidades: la Sección 1 enumera todos los estudios de diseño generativo realizados; la Sección 2 muestra los diseños devueltos por el generador, indicando la cantidad de propuestas en función de la configuración de la población inicial y el número de generaciones. Por su parte, la Sección 3 detalla las combinaciones de variables empleadas para cada diseño. Asimismo, la Sección 4 profundiza en el diseño seleccionado, mientras que la Sección 5 despliega las métricas correspondientes a dicha propuesta.

Finalmente, tras generar las propuestas de diseño de planificación de espacios, se realizó un análisis visual y de las métricas obtenidas para seleccionar la mejor solución disponible. Con esta solución se realizó la comparación con el diseño original en términos de la cantidad de espacios colocados, así como de la posición y orientación de dichos espacios.

El objetivo de este procedimiento fue determinar qué tan cerca se encontraron las soluciones propuestas de la solución original considerada como óptima. Esto permitió definir una exten-

sión de código con capacidades de diseño deseables antes de proceder con la evaluación empírica por parte de los profesionales.

4.3.6. Evaluación empírica de la extensión de código

Con base en el trabajo de Bocco et al. se planteó el caso de estudio para evaluar la facilidad de uso y funcionalidad de la extensión de código. El objetivo de este caso de estudio fue evaluar el rendimiento y la satisfacción de los profesionales de la construcción del estudio de arquitectura “A y D Estudio” con respecto al uso de la extensión de código para la planificación automática de espacios de planta dentro de proyectos de construcción.

La pregunta principal que guió el estudio fue: “¿Qué nivel de intención de uso tiene la solución propuesta para la distribución automática de espacios, qué tan eficiente es y cuánto tiempo ahorra con el uso diario y en un contexto real? “

Es decir, se buscó comprobar si la extensión de código propuesta resultaba eficiente, considerando la calidad de los resultados ofrecidos y la manera en que manejaba las restricciones del usuario referentes a los espacios a ser ubicados y las relaciones entre los mismos. Asimismo, se procuró verificar que la extensión de código tuviese una intención de uso real por parte del estudio de arquitectura. Finalmente, se evaluó y estimó la cantidad de tiempo que los miembros del estudio de arquitectura pudieron ahorrar al utilizar la extensión de código en tareas de planificación de espacios.

Debido a la complejidad de la pregunta de investigación, se decidió dividirla en tres subpreguntas independientes:

- PI1: ¿Cómo afecta el uso de la extensión de código de planificación automática de espacios al tiempo para desarrollar los diseños de distribución de espacios de planta en comparación con el método manual?
- PI2: ¿La extensión de código cumple con las expectativas de los arquitectos en términos de eficiencia y cumplimiento de las restricciones establecidas referentes a los espacios a ubicar, y sus relaciones de adyacencia?
- PI3: ¿Los arquitectos están satisfechos con los resultados y recomendarían el uso de

la extensión de código a otros colegas del área?

Para alcanzar los objetivos planteados, y dar una respuesta a la pregunta de investigación, se formuló una serie de preguntas que cubrieron todas las sub preguntas de investigación, a continuación se detallan esas preguntas:

- ¿Con qué frecuencia necesita generar planificaciones de espacios para proyectos de construcción? (relacionado con la PI1)
- ¿La extensión de código es intuitiva y fácil de usar? (relacionado con la PI2)
- ¿Cuánto tiempo le tomó aprender a usar la extensión de código eficientemente? (relacionado con la PI1)
- ¿La extensión de código cumple con sus expectativas en cuanto a la planificación de espacios generada? (relacionado con la PI3)
- ¿Qué tan bien maneja la extensión de código los requerimientos y restricciones impuestas? (relacionado con la PI2)
- En comparación con el método manual, ¿cuánto tiempo usted ahorra al usar la extensión de código? (relacionado con la PI1)
- ¿Considera que la extensión de código entrega soluciones con valores de área ocupada, porcentaje de espacios colocados y adyacencia mejores o iguales que el método manual? (relacionado con la PI2)
- ¿Qué tan satisfecho está con la extensión de código en general?(relacionado con la PI3)
- ¿Recomendaría esta extensión de código a otros profesionales del sector? (relacionado con la PI3)

A continuación, en la etapa de recolección de datos se generó una encuesta con las preguntas planteadas para cubrir la pregunta de investigación. La encuesta fue dirigida a los arquitectos miembros del estudio de arquitectura y permitió obtener datos cualitativos y cuantitativos que resumieron la experiencia de cada uno utilizando la extensión de código.

En primer lugar, se convocó a todos los arquitectos y se les brindó una breve capacitación sobre el uso de la extensión de código dentro del software Revit, así como los principales detalles sobre su funcionamiento interno. Posteriormente, se llamó individualmente a cada arquitecto y se les solicitó que utilizaran la aplicación de manera independiente, aplicando una serie de ejemplos de sitios y documentos de planificación. Una vez finalizada la utilización de la extensión de código, se realizó una encuesta de forma anónima con el fin de evitar sesgos y garantizar la sinceridad de las respuestas. Es importante destacar que la encuesta consistió únicamente en preguntas cerradas; no obstante, se recopiló retroalimentación adicional para futuros trabajos.

La Tabla 4 resume las preguntas de la encuesta que se formuló.

No.	Pregunta
Preguntas Genéricas	
1.	¿Cuál es su nivel de experiencia en el diseño y construcción de edificios? Alto – Medio – Bajo
Preguntas sobre Tiempo Ahorrado	
2.	¿Con qué frecuencia necesita generar planificaciones de espacios para proyectos de construcción? 1 vez al mes – 2 veces al mes – 3 veces al mes – 4 veces al mes o más
3.	En comparación con el método manual, ¿cuánto tiempo usted ahorra al usar la extensión de código? (En minutos, ingresar los minutos en números sin nada extra) Cantidad de minutos ahorrados
4.	¿Cuánto tiempo le tomó aprender a usar la extensión de código eficientemente? 5-10min – 10-15 min – 15-20min – +20min
Preguntas sobre Eficiencia y Requerimientos	
5.	¿La extensión de código es intuitiva y fácil de usar? Sí – No
6.	¿La extensión de código cumple con sus expectativas en cuanto a la planificación de espacios generada? Sí – No
7.	¿Qué tan bien maneja la extensión de código los requerimientos y restricciones impuestas? Alto – Medio – Bajo
8.	¿Considera que la extensión de código entrega soluciones con valores de área ocupada, porcentaje de espacios colocados y adyacencia mejores o iguales que el método manual? Totalmente en desacuerdo – En desacuerdo – Neutral – De acuerdo – Totalmente de acuerdo
Preguntas sobre Satisfacción e Intención de Uso	
9.	¿Qué tan satisfecho está con la extensión de código en general? Nada satisfecho – No satisfecho – Neutral – Satisfecho – Totalmente satisfecho
10.	¿Recomendaría esta extensión de código a otros profesionales del sector? Sí – No

Tabla 4: Preguntas formuladas para la encuesta de evaluación empírica empleada a los miembros del estudio de arquitectura “A y D Estudio”.

5. Resultados

Este capítulo se centra en los resultados de la investigación, dividiéndose en dos partes principales: la evaluación de la extensión de código para la generación de volumetrías y la evaluación de la extensión de código para la planificación de espacios. Cada una de estas subsecciones aborda tanto los resultados de la aprobación de las soluciones propuestas como los hallazgos de la evaluación empírica realizada.

5.1. Extensión de código para la generación de volumetrías

En esta sección, se presentan los resultados obtenidos en las etapas de aprobación de los diseños devueltos por la extensión de código para la generación automática de volumetrías de edificios. Además, se muestran los resultados obtenidos tras realizar la evaluación empírica con el estudio de arquitectos. A continuación, se detallan estos apartados.

5.1.1. Aprobación de los resultados de la extensión de código

En el inicio de los resultados, se presenta la aprobación de la solución propuesta. Inicialmente, en la sección de metodología, se describió el proceso y los pasos para obtener esta aprobación para una única volumetría. Por cuestiones de espacio, no se detalló el proceso paso a paso de los 40 estudios de volumetría disponibles. El análisis del resto de las edificaciones se centrará exclusivamente en los resultados obtenidos.

En este punto, todos los datos necesarios están disponibles en un archivo Excel. La tabla final, referenciada como Tabla 5, es la empleada para generar la aprobación.

Al momento que tenemos esta tabla, es fácil calcular el promedio de la diferencia entre el COS y CUS generado por la extensión de código y el real. Usando una función simple de Excel, calculamos el promedio de ambas diferencias. En el COS, todas las respuestas tienen una diferencia positiva, lo que indica que la extensión de código siempre obtiene mejores resultados que los obtenidos por los arquitectos. El promedio de la diferencia a favor en COS es del 9.57 %. En el CUS sucede lo mismo, y el promedio es del 15.37 % de diferencia a favor. Podemos ver que la extensión de código siempre otorga resultados mejores a las

Campo	Descripción	Ejemplo
ID	Identificador único de la edificación.	ED1
FRENTE	Ancho frontal del terreno (en metros).	36
FONDO	Profundidad del terreno (en metros).	30.5
SECTOR	Sector donde se ubica la edificación.	E5
PISOS	Número de pisos de la edificación.	6
COSREAL (%)	Porcentaje real del Coeficiente de Ocupación del Suelo (COS).	58
CUSREAL (%)	Porcentaje real del Coeficiente de Uso del Suelo (CUS).	38.71
COSSOLUCIÓN	Coeficiente de Ocupación del Suelo (COS) propuesto por la solución.	62
CUSSOLUCIÓN	Coeficiente de Uso del Suelo (CUS) propuesto por la solución.	48.9
PISOS SOLUCIÓN	Número de pisos propuesto por la solución.	6
CUS-CUSR	Diferencia entre CUS de la solución y CUS real (%).	10.19
COS-COSR	Diferencia entre COS de la solución y COS real (%).	4

Tabla 5: Tabla final para realizar la comparación entre los resultados reales y los generados por la extensión de código.

volumetrías propuestas por los arquitectos, pero este resultado es un poco engañoso, ya que en algunas de las construcciones el cliente no quería más de 5 pisos o daba un área total construida máxima por cuestiones de presupuesto. Todo esto se puede manejar en la extensión de código, pero en comparación directa con edificaciones, la solución presentada es muy superior a lo entregado por los arquitectos. Lo mencionado anteriormente nos indica por qué la necesidad de una evaluación empírica, porque necesita probarse en un entorno real para encontrar qué tan útil realmente es la extensión de código. Además, se necesita evaluar otros factores como tiempos ahorrados, tiempos de aprendizaje y eficacia. En este punto, vemos que el COS y CUS son mejores, pero no vemos si las volumetrías generadas son realmente útiles y se pueden construir en un entorno real.

5.1.2. Evaluación empírica de la extensión de código

En esta sección, se presentan los resultados obtenidos tras aplicar la evaluación empírica planteada en la metodología de la extensión de código para la planificación automática de espacios. Estos hallazgos proporcionan información clave para responder a la pregunta de investigación planteada y alcanzar los objetivos establecidos para este trabajo de titulación.

A continuación, se detallan los principales resultados organizados de forma estructurada, con el apoyo de tablas, gráficos y análisis pertinentes que permiten una mejor comprensión de los datos recopilados.

Durante la evaluación empírica de la extensión de código para la generación de volumetrías, se contó con el valioso apoyo y conocimiento del estudio de arquitectura “A y D estudio arquitectura + diseño”, cuya colaboración fue fundamental para validar y enriquecer los resultados obtenidos. Agradecemos sinceramente su participación y aportes, los cuales contribuyeron significativamente al éxito de este proyecto.

Partimos conociendo el nivel de experiencia de los arquitectos participantes de la encuesta, en la Figura 26 se resume la cantidad de arquitectos por cada nivel de experiencia que poseen en tareas de planificación de espacios dentro de sus proyectos arquitectónicos.

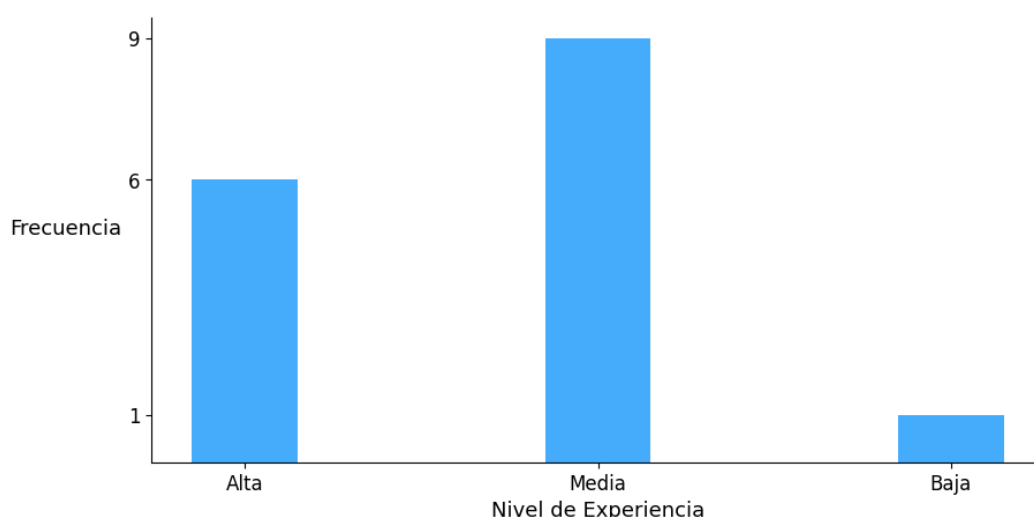


Figura 26: Distribución de la experiencia en tareas de generación de volumetría entre los arquitectos evaluados. Predomina la experiencia media, aunque también hay arquitectos con experiencia alta y baja.

De acuerdo a la Figura 26, el nivel de experiencia autopercebido por los participantes se distribuyó de la siguiente manera: en general, se tienen más arquitectos con un nivel de experiencia medio, aunque también hubo una presencia considerable de arquitectos que indicaron tener un nivel de experiencia alto o bajo en la planificación de espacios de planta. Esta diversidad en los perfiles de los participantes resulta fundamental, ya que permite contar con diferentes perspectivas a nivel profesional en cuanto a la utilidad y eficiencia de

la herramienta. Cabe resaltar que los arquitectos más experimentados aportaron un criterio más realista, dada la complejidad inherente a sus proyectos.

Por otro lado, al enfocarse en la primera pregunta de investigación, que abordaba cómo el uso de la extensión de código para la generación automática de volumetría afecta el tiempo en comparación con el método manual, se analizaron primero los resultados de la pregunta sobre la frecuencia con la que los arquitectos necesitan generar volumetrías en sus proyectos. Dichos resultados, presentados en la Figura 27, se indica que las planificaciones se realizan más frecuentemente una, dos o hasta cuatro veces al mes, mientras que una frecuencia de tres veces al mes es menos común. Por lo tanto, cualquier reducción en el tiempo de creación de volumetrías tendría un impacto mensual significativo.

En otras palabras, cada mejora en el tiempo, por ejemplo, si reducimos 120 minutos por generación de volumetrías, multiplicado por 2, nos dará un ahorro de 240 minutos al mes con la extensión de código. Esto sin tener en cuenta las mejoras que la solución propuesta pueda generar en otras áreas.

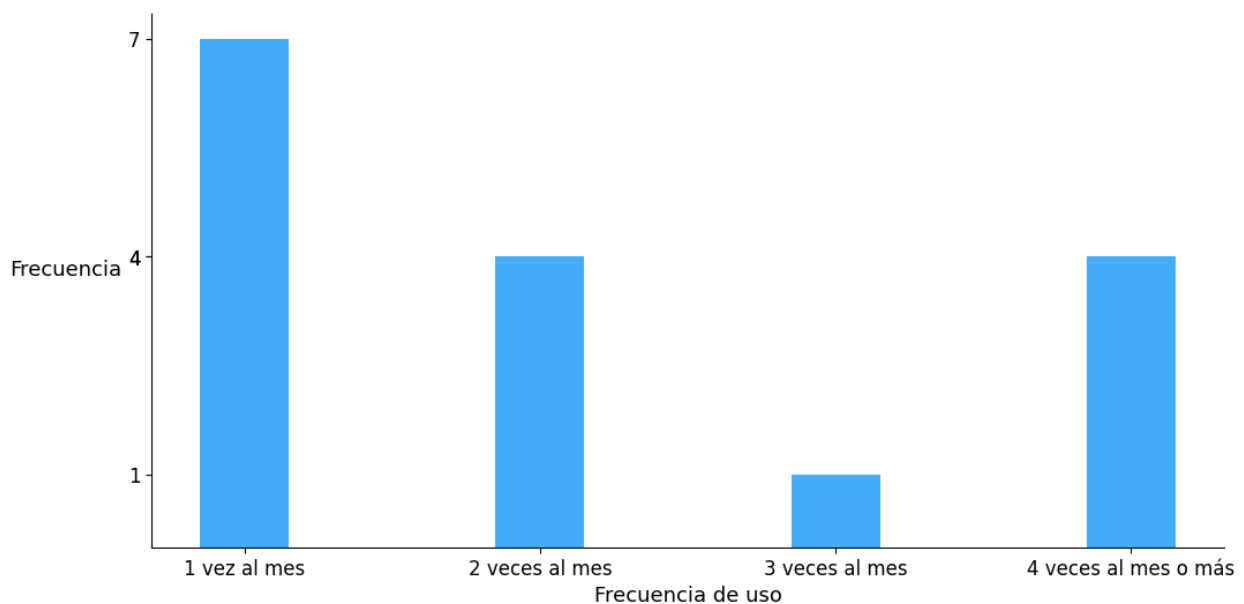


Figura 27: Frecuencia con la que los arquitectos generan volumetrías de edificios según la encuesta. La planificación se realiza más frecuentemente una, dos o cuatro veces al mes, siendo menos común tres veces al mes.

En cuanto a la pregunta “En comparación con el método manual, ¿Cuánto tiempo ahorra al usar la extensión de código?”, los resultados se presentan en la Figura 28. En general,

gran parte de los arquitectos ahorran entre unos 30 y 178 minutos al utilizar la extensión de código. No obstante, se observan dos casos extremos donde se llegan a ahorrar entre 425 y 475 minutos para la creación de volumetrías.

Para explicar este último punto se toma en cuenta que los arquitectos más experimentados dentro del estudio reciben proyectos más complejos que abarcan terrenos considerados difíciles. Esto se debe a que deben realizar pruebas con varios pisos para generar y determinar la volumetría final del edificio, de modo que el cliente pueda decidir si está dispuesto a sacrificar cierta cantidad de metros cuadrados a cambio de la forma del edificio.

En contraste, los arquitectos menos experimentados suelen recibir proyectos cuya ejecución y elaboración no involucran estudios ni restricciones tan complejas, por lo que el tiempo empleado en estos casos suele ser mucho menor, según lo indicado por los propios miembros del estudio.

Por ejemplo, considerando un tiempo promedio ahorrado de 162 minutos, multiplicado por 2.5, que es el promedio de volumetrías generadas al mes, obtenemos que, en promedio, los arquitectos ahorran 405 minutos o 6.75 horas al mes con el uso de la extensión de código. Para los arquitectos a cargo de volumetrías más complejas, que ven un ahorro de 475 minutos por las 2.5 volumetrías mensuales promedio (que en este caso podrían ser más), el ahorro mensual asciende a 1187.5 minutos o 19.79 horas.

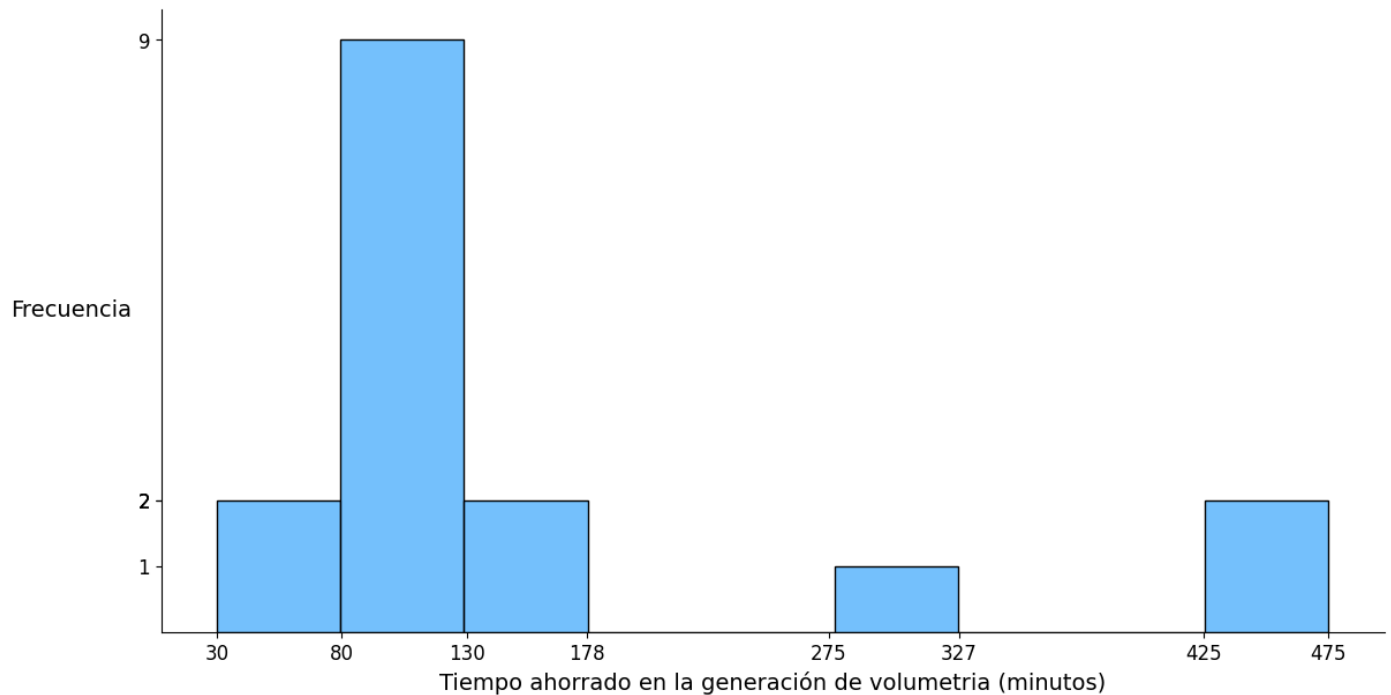


Figura 28: Tiempo ahorrado empleando la solución de planificación de espacios según la encuesta de evaluación empírica. El ahorro de tiempo más común al usar la extensión de código de generación de volumetría fue de 30 a 178 minutos (13 participantes). Además, se reportaron ahorros de 275 a 327 minutos (1 participante) e incluso de 425 a 475 minutos (2 participantes).

En relación con la última pregunta de investigación enfocada en el ahorro de tiempo, ¿Cuánto tiempo le tomó aprender a usar la extensión de código eficientemente?, los resultados se presentan en la Figura 29. En general, los tiempos empleados para aprender a utilizar la extensión de código se concentran entre los 5 a 15 minutos. En los apartados anteriores se observó que la cantidad de tiempo ahorrado por mes es considerablemente grande. Dado que el tiempo requerido para aprender a usar la extensión de código es tan reducido y es un proceso que solo se realiza una vez, el tiempo de aprendizaje resulta insignificante en comparación con los beneficios generados. Por lo tanto, no afecta los beneficios de la implementación de la solución.

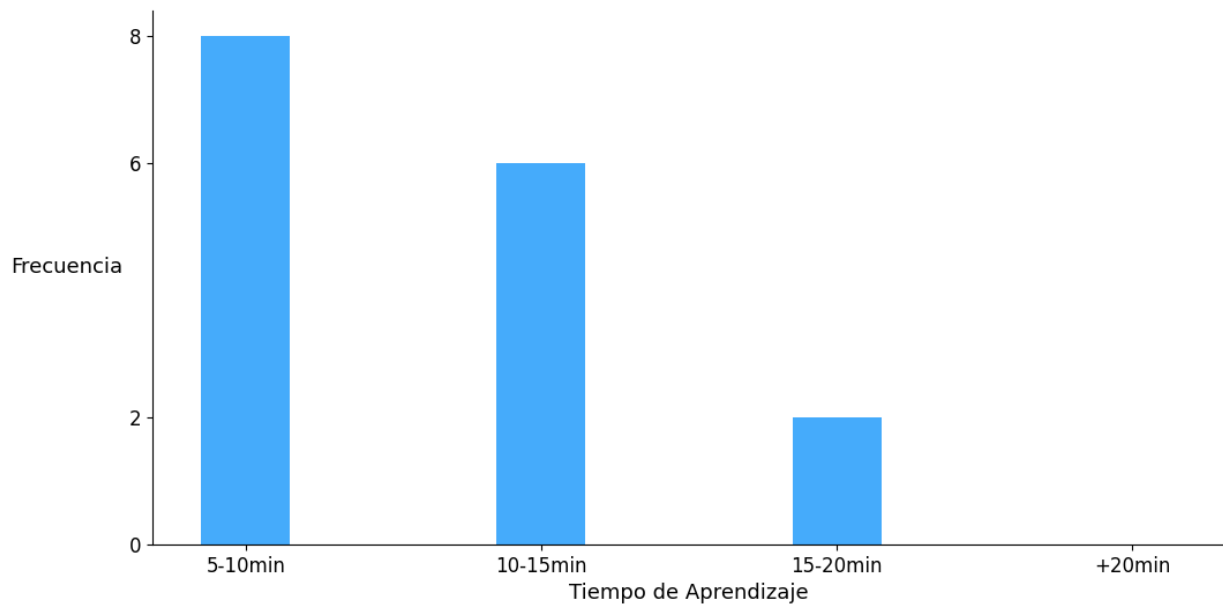


Figura 29: Tiempo empleado para aprender a usar la extensión de código de generación de volumetría según la encuesta de evaluación empírica. Se observa que el tiempo de aprendizaje se concentra entre los 5 a 15 minutos.

La primera pregunta en torno a la investigación sobre eficiencia y normativa fue la siguiente: ¿La solución es intuitiva y fácil de usar? Los resultados de dicha investigación se observan en la Figura 30. En ella, se aprecia que 14 arquitectos indicaron que la solución es de fácil utilización e intuitiva, mientras que únicamente 2 participantes señalaron lo contrario. Los arquitectos que consideraron que la solución no era intuitiva proporcionaron retroalimentación, manifestando que se sentían ligeramente abrumados por la gran cantidad de variables que presenta el software en la pantalla de configuración, así como por la explicación del algoritmo genético, que les resultó igualmente compleja. No obstante, estos profesionales señalaron que, a pesar de estos aspectos, consideraban la extensión de código muy útil.

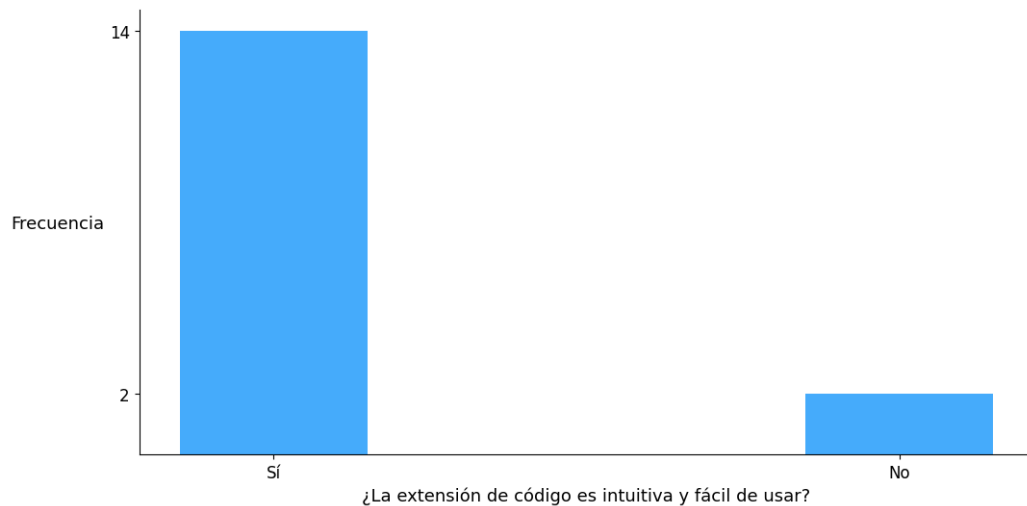


Figura 30: Percepción de la facilidad de uso de la extensión de código para la generación automática de volumetrías. En ella 14 participantes concuerdan que la extensión es intuitiva y fácil de utilizar, mientras que 2 de ellos no.

La siguiente pregunta abordada en el estudio sobre eficiencia y normativa fue la siguiente: ¿La extensión de código cumple con las expectativas de los usuarios en cuanto a la generación automática de volumetrías? Los resultados de dicha investigación se pueden observar en la Figura 31. Al igual que en la pregunta anterior, 14 participantes expresaron su satisfacción con las volumetrías obtenidas, mientras que 2 personas no lo hicieron. En este caso, la retroalimentación proporcionada por estos últimos indicó que las volumetrías no cumplían con sus expectativas, dado que deseaban formas más complejas. Se les explicó que el enfoque de la herramienta se centra en optimizar la construcción, lo que puede implicar la simplificación de las formas. Es importante recalcar que las personas que encontraron dificultoso el uso de la extensión de código, sí se mostraron satisfechas con los resultados obtenidos, por lo que no se evidencia una relación directa entre ambas variables.

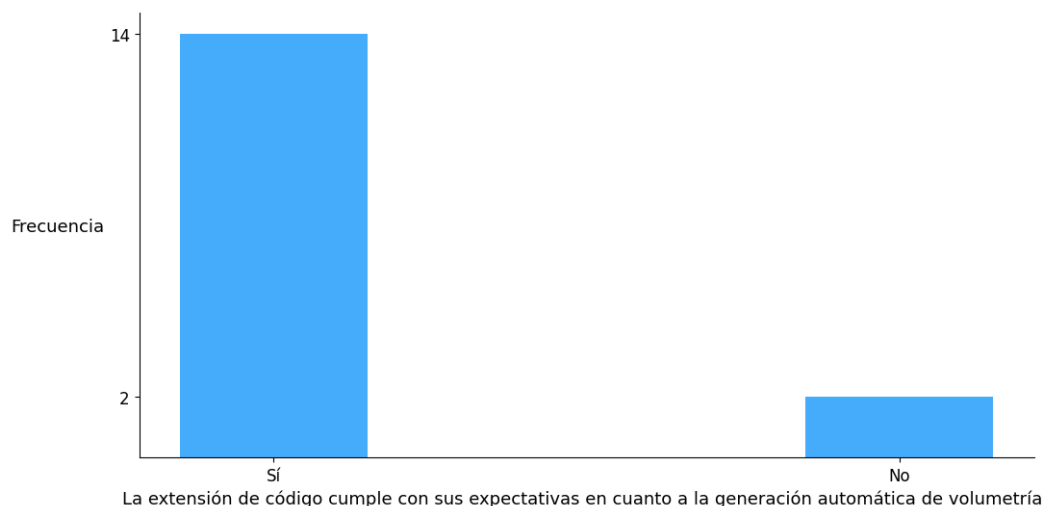


Figura 31: Cumplimiento de las expectativas de la extensión de código para la generación automática de volumetrías según la encuesta de evaluación empírica. En su mayoría las expectativas fueron cumplidas aunque 2 arquitectos experimentados esperaron resultados más complejos que no pudieron ser cubiertos por la extensión de este trabajo.

La tercera pregunta se refiere a la eficiencia y el cumplimiento de la normativa. Específicamente, se cuestiona qué tan bien maneja la solución la normativa de construcción de Cuenca en sus cálculos. Los resultados se pueden observar en la Figura 32.

Según los datos, todos los arquitectos encuestados coinciden en que el manejo de la normativa por parte de la extensión de código es muy satisfactorio. Esto se debe a que las volumetrías generadas siempre cumplen con la normativa vigente. De hecho, si la extensión detecta que la normativa no permite la construcción en un terreno determinado, o que las limitaciones impuestas por el usuario no lo permiten, devolverá el terreno vacío, sin generar una propuesta que no se ajuste a las regulaciones.

La retroalimentación de los arquitectos indicó que este aspecto es crucial, ya que en un peritaje futuro, obviar o incumplir la normativa de manera no intencional puede resultar en sanciones económicas significativas o en la imposibilidad de desarrollar un proyecto. Por lo tanto, el hecho de tener la seguridad de que la solución siempre maneja correctamente la normativa disminuye la presión en las propuestas entregadas y brinda mayor tranquilidad de que la normativa está siendo cumplida.

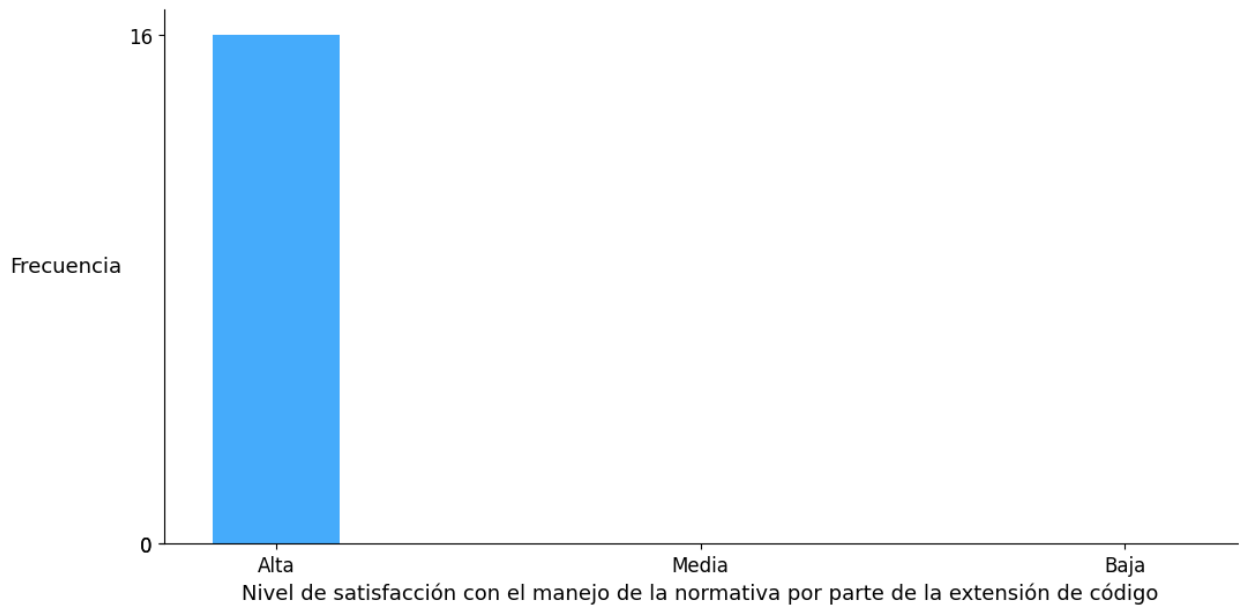


Figura 32: Cumplimiento de la normativa de construcción de Cuenca de la solución de generación de volumetrías según la encuesta de evaluación empírica. Los arquitectos encuestados consideran que dichas restricciones se cumplen en una alta medida.

Finalmente, en relación con la pregunta sobre si la solución maximiza de manera más eficiente el área total construida, el COS y el CUS en comparación con el método manual, los resultados se pueden observar en la Figura 33. En esta, se aprecia que la totalidad de los arquitectos encuestados están de acuerdo en que el uso de la extensión de código devuelve diseños de manera más eficiente que el método manual. No obstante, también se evidencia que algunos arquitectos se muestran más convencidos que otros.

La retroalimentación proporcionada por los arquitectos menos satisfechos sugiere que la extensión de código los limita en ciertas técnicas o leyes aplicables a sectores o situaciones muy específicas, como la compra de metros al gobierno. Este es un caso poco común que no aplica a todos los contextos, pero en esos escenarios excepcionales, la extensión de código no resultaría más eficiente que el método manual. Sin embargo, en la gran mayoría de las situaciones, la solución digital es más eficiente.

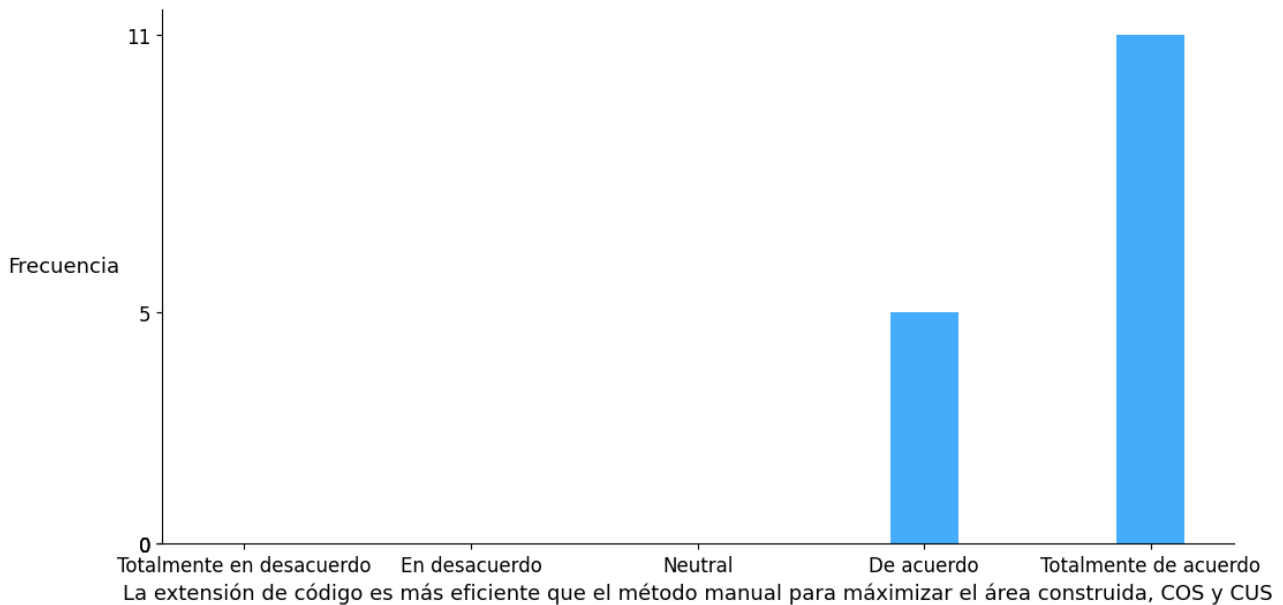


Figura 33: Satisfacción con la maximización de COS y CUS en la generación de volumetrías según la encuesta. En general, los arquitectos están totalmente de acuerdo con los resultados obtenidos.

Por último, se analizan las preguntas relacionadas con la satisfacción y la intención de uso de la extensión de código. Respecto a la pregunta “¿Qué tan satisfecho está con la extensión de código en general?”, los resultados se pueden observar en la Figura 34. En ella se aprecia que, en general, los arquitectos se encuentran satisfechos con el funcionamiento de la extensión de código, sus aplicaciones y los resultados obtenidos.

La retroalimentación proporcionada por los arquitectos en este apartado indica que desearían que la extensión tuviera más funcionalidades, más allá de la generación de volumetrías, incluyendo la creación de otros elementos como ventanas, puertas, entradas, entre otros. No obstante, debido a las limitaciones de tiempo del trabajo de titulación, no fue posible implementar estas funciones adicionales.

En general, esta pregunta revela una alta satisfacción de los usuarios con la extensión de código, ya que la falta de satisfacción se debe únicamente al deseo de agregar más funcionalidades. Esto sugiere una alta intención de uso, que deberá ser confirmada por la siguiente pregunta.

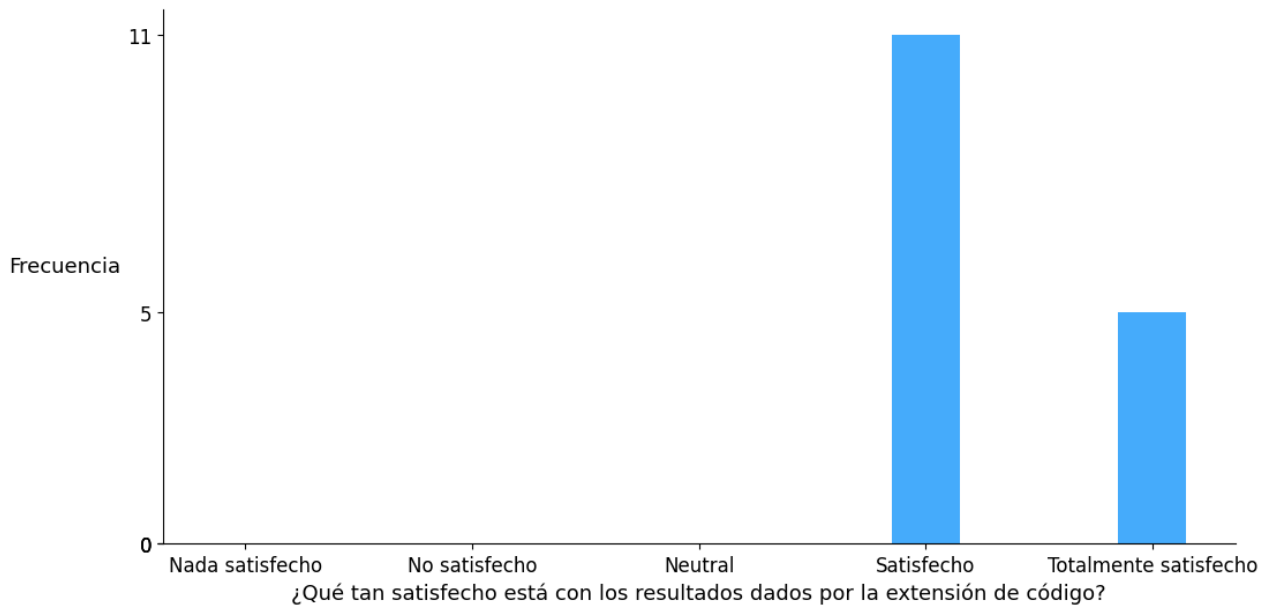


Figura 34: Satisfacción con la extensión de código en general según la encuesta de evaluación empírica. En general, gran parte de los arquitectos se encuentran satisfechos con la extensión de código.

Finalmente, con relación a la pregunta sobre si ¿Recomendaría esta solución a otros profesionales del sector? Los resultados se observan en la Figura 35, y se observa que todos los arquitectos concuerdan en recomendar el uso de la extensión de código a otros profesionales del sector. Esto muestra una intensidad de uso clara de la extensión de código para la generación automática de volumetrías, ya que los arquitectos encontraron potencial en la misma y sus capacidades de expansión con futuras funcionalidades.

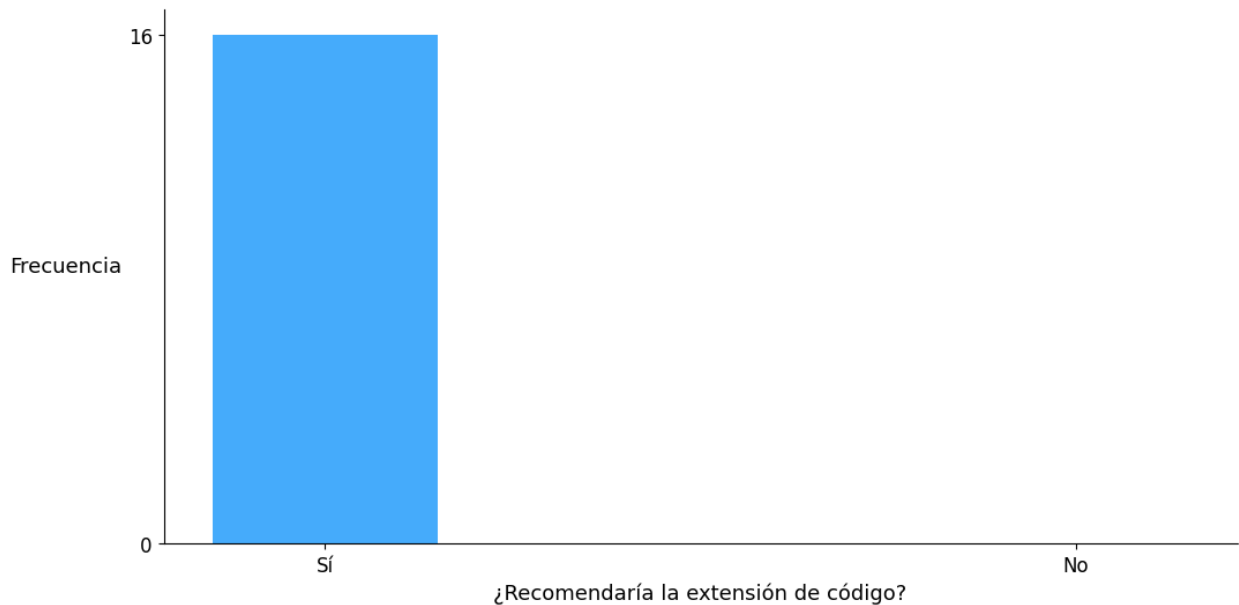


Figura 35: Cantidad de personas que recomendarían la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Todos los arquitectos concuerdan en recomendar la extensión de código a otros profesionales.

Por último, se desea analizar la relación entre el tiempo ahorrado por la extensión de código y el nivel de experiencia del arquitecto. La Figura 36 nos permite ver esta relación de mejor manera. En la misma, se pueden observar ciertas particularidades, como que algunos usuarios con experiencia media o baja tienen altos tiempos ahorrados. Según la retroalimentación proporcionada, esto se debe a la falta de experiencia al realizar el proceso manual, lo cual lógicamente indica que estos arquitectos verán una gran cantidad de tiempo ahorrado con la extensión de código.

Otra particularidad aún más interesante encontrada es que gran parte de los arquitectos con un nivel medio de experiencia indican que la extensión de código ahorra menos tiempo en comparación con algunos arquitectos con alta experiencia. Según la retroalimentación dada, esto se debe a dos factores.

Primero que los arquitectos con menos experiencia tienden a obviar ciertas normativas por falta de conocimiento o no generan la propuesta completa que se entrega al cliente, sino solo un borrador. Además, estos arquitectos generalmente reciben las volumetrías más sencillas de realizar o aquellas donde ya se tienen pocas opciones debido a la normativa, lo que facilita su generación.

Por otro lado, los arquitectos más expertos reciben volumetrías más complejas que requieren más tiempo para ser calculadas. Sin embargo, debido a su experiencia, algunos con más de 20 años, los tiempos ahorrados son menores que en niveles de experiencia más bajos, a pesar de la mayor complejidad de sus volumetrías.

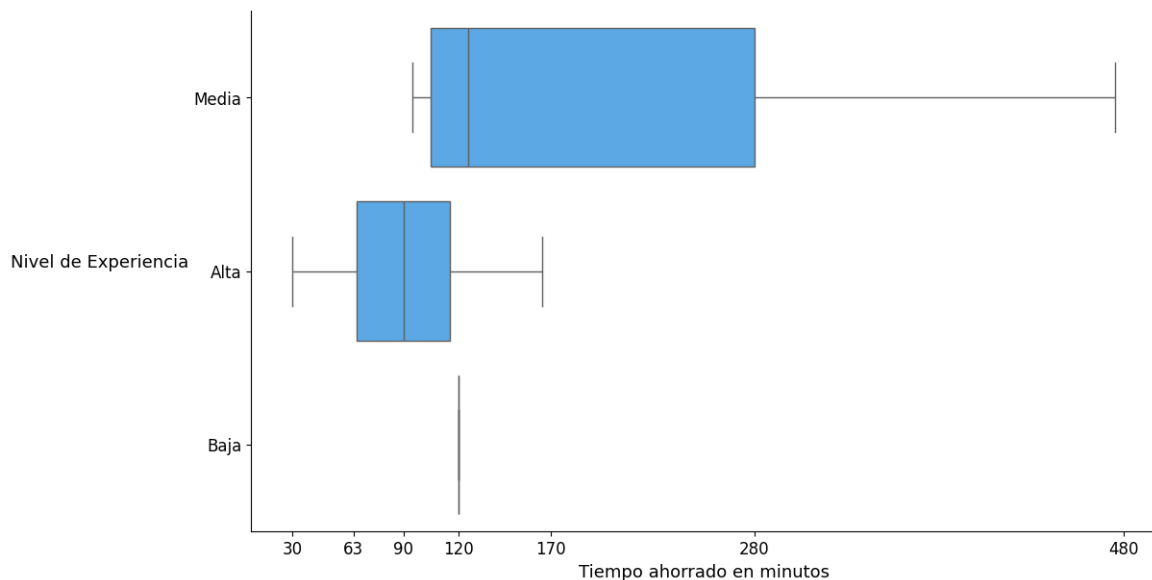


Figura 36: Relación entre la experiencia de los arquitectos y el tiempo que ahorran empleando la extensión de código. Se destaca que los arquitectos con un nivel medio de experiencia ahorran un mayor tiempo pese a que sus problemas de diseño tienen menos restricciones que la de los arquitectos con mayor experiencia.

Finalmente, la extensión de código resulta en un mayor ahorro de tiempo para arquitectos con menos experiencia debido a su falta de familiaridad con el proceso, mientras que los arquitectos más experimentados aprecian el ahorro de tiempo en tareas más complejas y detalladas.

5.2. Extensión de código para la planificación automática de espacios

En esta sección, se presentan los resultados obtenidos en las etapas de aprobación de los diseños devueltos por la extensión de código para la planificación automática de espacios. Además, se muestran los resultados obtenidos tras realizar la evaluación empírica con el estudio de arquitectos. A continuación, se detallan estos apartados.

5.2.1. Aprobación de los resultados de la extensión de código

Se aprobó la implementación de la extensión de código para la planificación de espacios, utilizando el conjunto de diseños de distribución de espacios proporcionados por el estudio de arquitectura. Dichos diseños fueron simplificados mediante el empleo de policurvas y líneas de modelo.

A continuación, el comportamiento de la extensión con los diez planes de distribución de espacios disponibles se pueden resumir a través de tres planes generados. En las Figuras 37, 38 y 39 se resumen los resultados de la generación de estos planes de espacio.

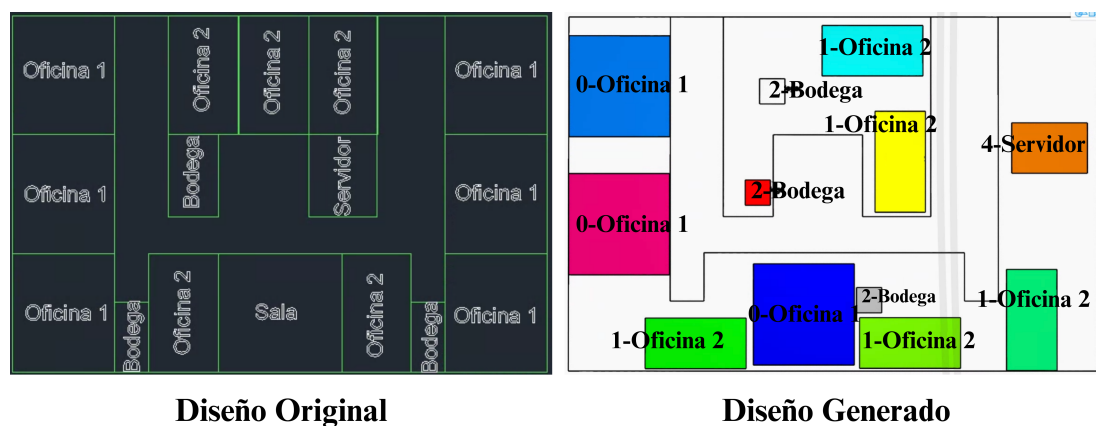


Figura 37: Prueba realizada de la extensión de código. El diseño generado obtuvo un 44 % de porcentaje de área utilizada, un 75 % de porcentaje de espacios colocados y un tiempo de generación y selección de 54 minutos.

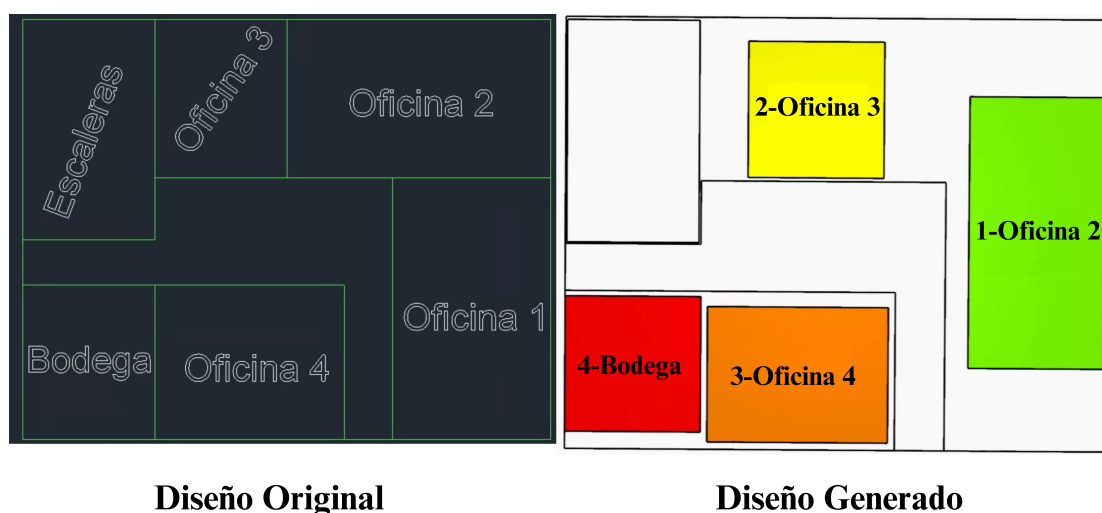


Figura 38: Prueba realizada de la extensión de código. El diseño generado obtuvo un 60 % de porcentaje de área utilizada, un 80 % de porcentaje de espacios colocados y un tiempo de generación y selección de 1 hora y 12 minutos.

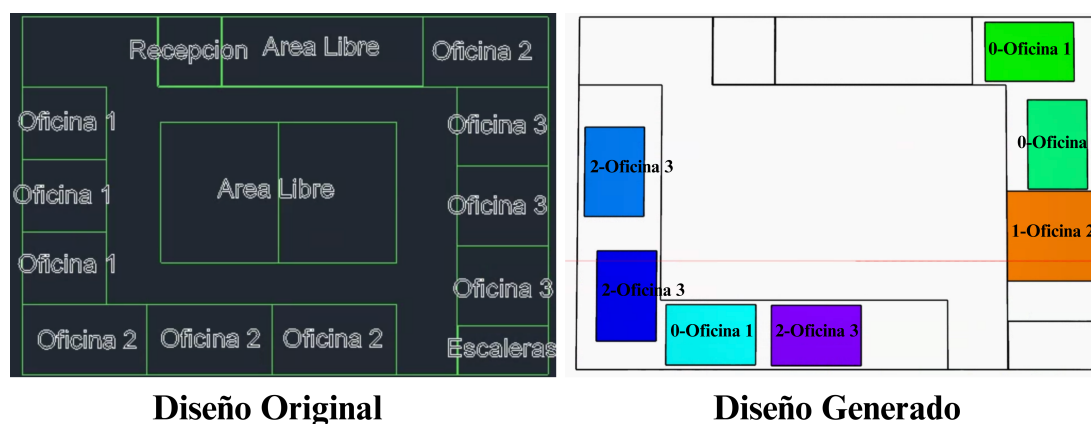


Figura 39: Prueba realizada de la extensión de código. El diseño generado obtuvo un 54 % de porcentaje de área utilizada, un 70 % de porcentaje de espacios colocados y un tiempo de generación y selección de 45 minutos.

En general, la configuración del proceso de generación de población y las etapas de selección y mutación resultaron ser exigentes. Las soluciones, en promedio, tardaron aproximadamente una hora en generarse y seleccionarse manualmente. El criterio de selección, como se explicó en la metodología, fue que las soluciones se asemejaran o superaran el diseño original. No obstante, en la mayoría de los casos se logró colocar entre un 70 y 80 % de los espacios especificados en los documentos de planificación. Esto evidenció una dificultad para la selección de diseños, a pesar de que se lograron diseños con distribuciones de espacios cercanas a la original. La solución se aprobó en base a estos resultados, y se prosiguió con la evaluación empírica utilizando configuraciones más exigentes, sobre todo en el número de iteraciones del algoritmo de colocación de espacios.

5.2.2. Evaluación empírica de la extensión de código

En esta sección, se presentan los resultados obtenidos tras aplicar la evaluación empírica planteada en la metodología de la extensión de código para la planificación automática de espacios. Estos hallazgos proporcionan información clave para responder a la pregunta de investigación planteada y alcanzar los objetivos establecidos para este trabajo de titulación. A continuación, se detallan los principales resultados organizados de forma estructurada, con el apoyo de tablas, gráficos y análisis pertinentes que permiten una mejor comprensión de los datos recopilados.

Durante la evaluación empírica de la extensión de código para la planificación de espacios, se contó con el valioso apoyo y conocimiento del estudio de arquitectura “A y D estudio arquitectura + diseño”, cuya colaboración fue fundamental para validar y enriquecer los resultados obtenidos. Agradecemos sinceramente su participación y aportes, los cuales contribuyeron significativamente al éxito de este proyecto.

Partimos conociendo el nivel de experiencia de los arquitectos participantes de la encuesta, en la Figura 40 se resume la cantidad de arquitectos por cada nivel de experiencia que poseen en tareas de planificación de espacios dentro de sus proyectos arquitectónicos.

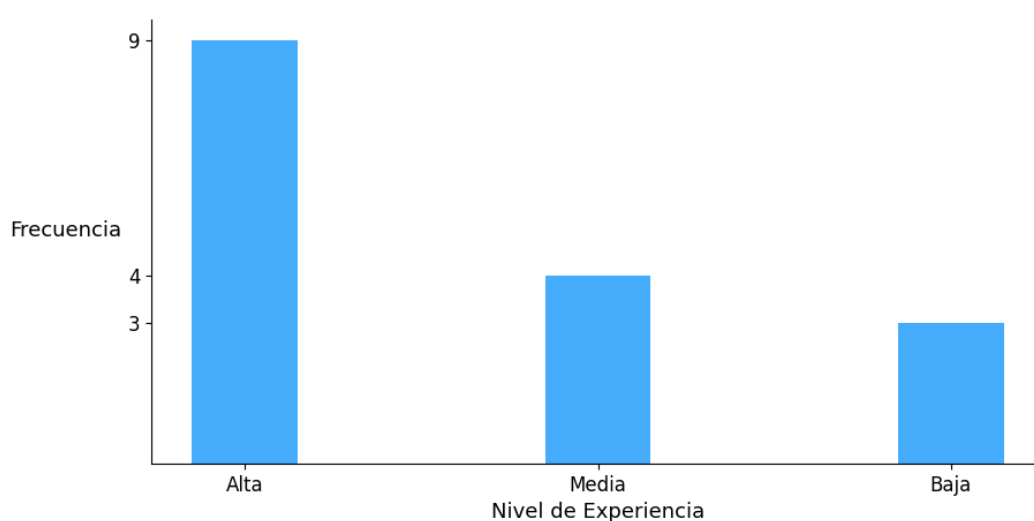


Figura 40: Distribución de los niveles de experiencia en planificación de espacios de los arquitectos que participaron en la evaluación empírica. La mayoría tenía un nivel de experiencia alto, pero también hubo una presencia considerable de arquitectos con experiencia media y baja.

De acuerdo a la figura, el nivel de experiencia autopercebido por los participantes se distribuyó de la siguiente manera: en general, se tienen más arquitectos con un nivel de experiencia alto, aunque también hubo una presencia considerable de arquitectos que indicaron tener un nivel de experiencia medio o bajo en la planificación de espacios de planta. Esta diversidad en los perfiles de los participantes resulta fundamental, ya que permite contar con diferentes perspectivas a nivel profesional en cuanto a la utilidad y eficiencia de la herramienta. Cabe resaltar que los arquitectos más experimentados probablemente aportaron un criterio más realista, dada la complejidad inherente a sus proyectos.

Por otro lado, al enfocarse en la primera pregunta de investigación, que abordaba cómo el

uso de la extensión de código para la planificación automática de espacios afecta el tiempo en comparación con el método manual, se analizaron primero los resultados de la pregunta sobre la frecuencia con la que los arquitectos necesitan generar planificaciones de espacios para sus proyectos. Dichos resultados, presentados en la Figura 41, indican que las planificaciones se realizan más frecuentemente una o dos veces al mes, mientras que una frecuencia mayor es menos común. Por lo tanto, cualquier reducción en el tiempo de planificación de espacios tendría un impacto mensual significativo.

A partir de los resultados de la extensión de código para la generación automática de volúmetrías, se observa que, si se reduce en 120 minutos el tiempo empleado en el diseño de la distribución de espacios en una planta, el cual se realiza entre 2 y 3 veces al mes, esto conllevaría un ahorro promedio de 240 a 360 minutos mensuales por cada planta diseñada. Cabe mencionar que, en proyectos de edificios completos, el número de pisos y las condiciones de planificación pueden variar, lo que implicaría ajustar y escalar estos ahorros de tiempo en consecuencia.

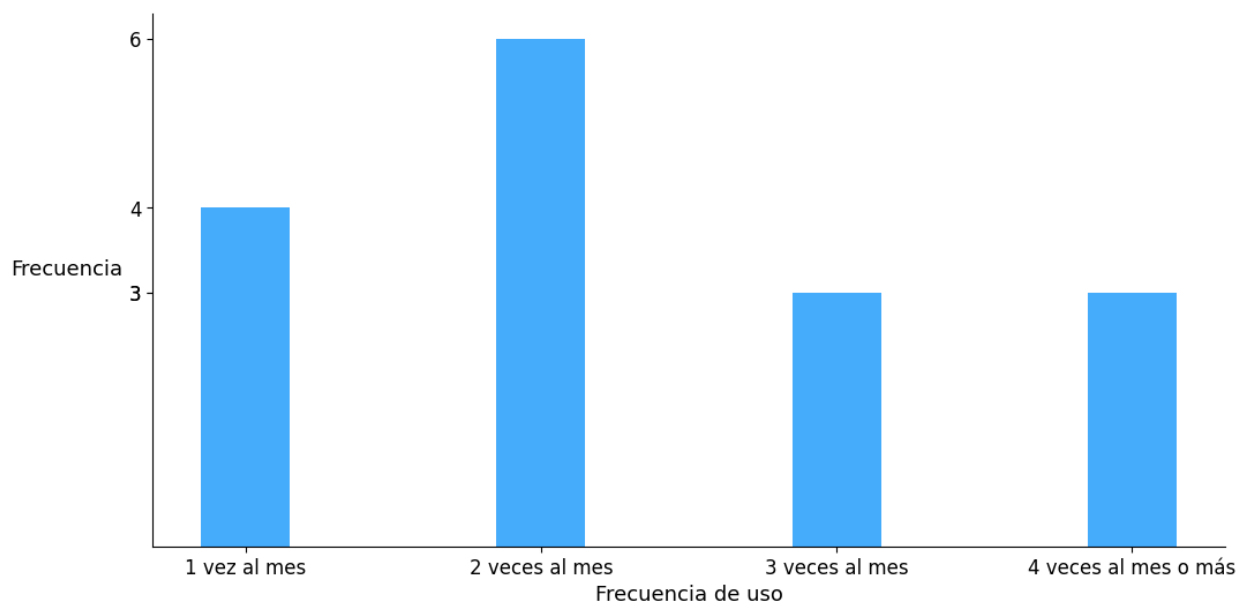


Figura 41: Frecuencia con la que los arquitectos generan planificaciones de espacios en sus proyectos según la encuesta de evaluación empírica. Se indica que las planificaciones se realizan más frecuentemente una o dos veces al mes, mientras que una frecuencia mayor es menos común

En cuanto a la pregunta “En comparación con el método manual, ¿cuánto tiempo se ahorra

al usar la extensión de código? (Ingresar los minutos en números)”, los resultados se presentan en la Figura 42. En general, gran parte de los arquitectos ahorran entre unos 30 y 170 minutos al utilizar la extensión de código. No obstante, se observan dos casos extremos donde se llegan a ahorrar entre 445 y 515 minutos para cada planificación de espacio realizada.

Para explicar este último punto se toma en cuenta que los arquitectos más experimentados dentro del estudio reciben proyectos de planificación más largos y que abarcan más detalles referentes a posicionamiento y estudios de iluminación, mientras que los arquitectos menos experimentados, reciben proyectos cuyo ejecución y elaboración no involucran estudios complejos ni restricciones elaboradas por lo que el tiempo empleado suele ser mucho menor según indicaron los propios miembros del estudio.

Por ejemplo, considerando un tiempo promedio ahorrado de 143 minutos, multiplicado por 2.5, que es el promedio de planificaciones de espacio generadas al mes, obtenemos que, en promedio, los arquitectos ahorran 357 minutos o 5.95 horas al mes con el uso de la extensión de código por cada planta planificada. Para los arquitectos a cargo de planificaciones de espacios más complejas, que ven un ahorro de 515 minutos por las 2.5 planificaciones mensuales promedio (que en este caso podrían ser más), el ahorro mensual asciende a 1287.5 minutos o 21.46 horas.

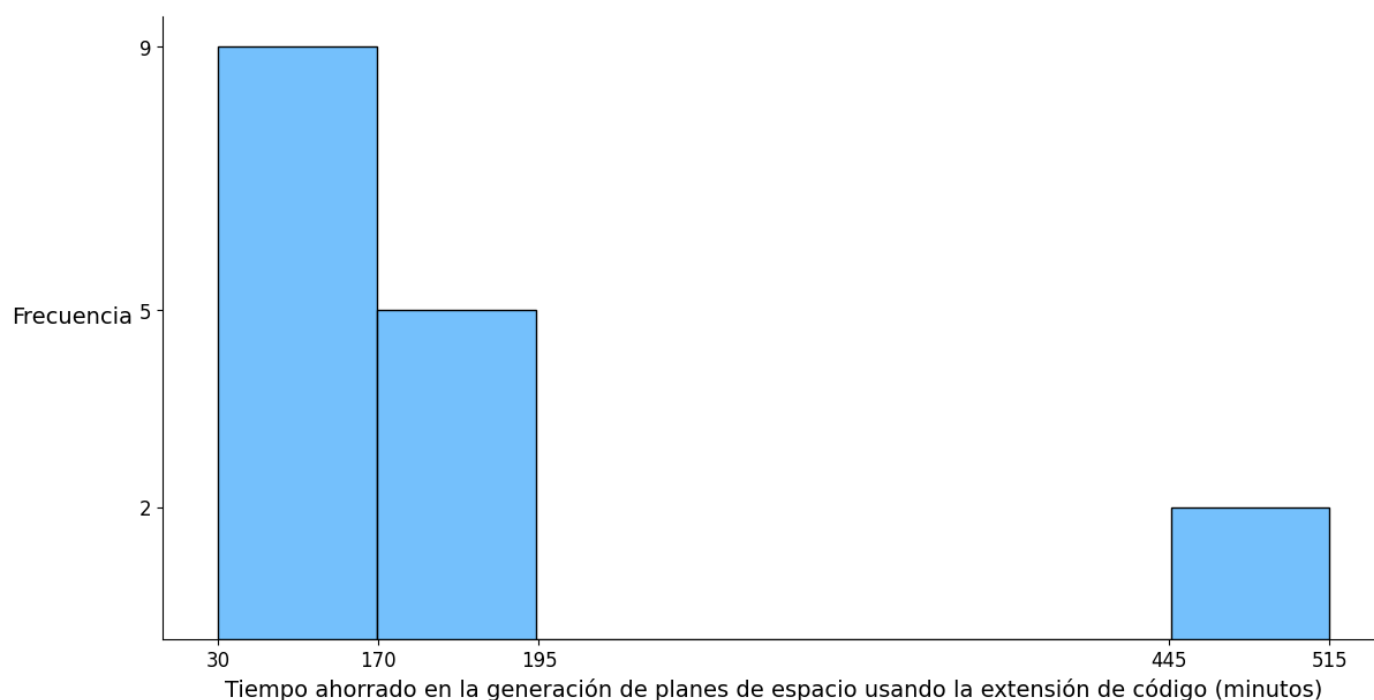


Figura 42: Tiempo ahorrado empleando la solución de planificación de espacios según la encuesta de evaluación empírica. El ahorro de tiempo más común al usar la extensión de código de planificación de espacios fue de 30 a 170 minutos (9 participantes). Además, se reportaron ahorros de 170 a 195 minutos (5 participantes) e incluso de 445 a 515 minutos (2 participantes).

En relación con la última pregunta de investigación enfocada en el ahorro de tiempo, ¿Cuánto tiempo les tomó a los usuarios aprender a usar la solución eficientemente?, los resultados se presentan en la Figura 43. En general, los tiempos empleados para aprender a utilizar la extensión de código se concentran entre los 5 a 20 minutos. Dado que este proceso de aprendizaje solo se realiza una vez, el tiempo invertido resulta insignificante en comparación con los considerables beneficios generados por el uso de la solución. Por lo tanto, el tiempo de aprendizaje no afecta de manera significativa los beneficios de la implementación de la extensión de código.

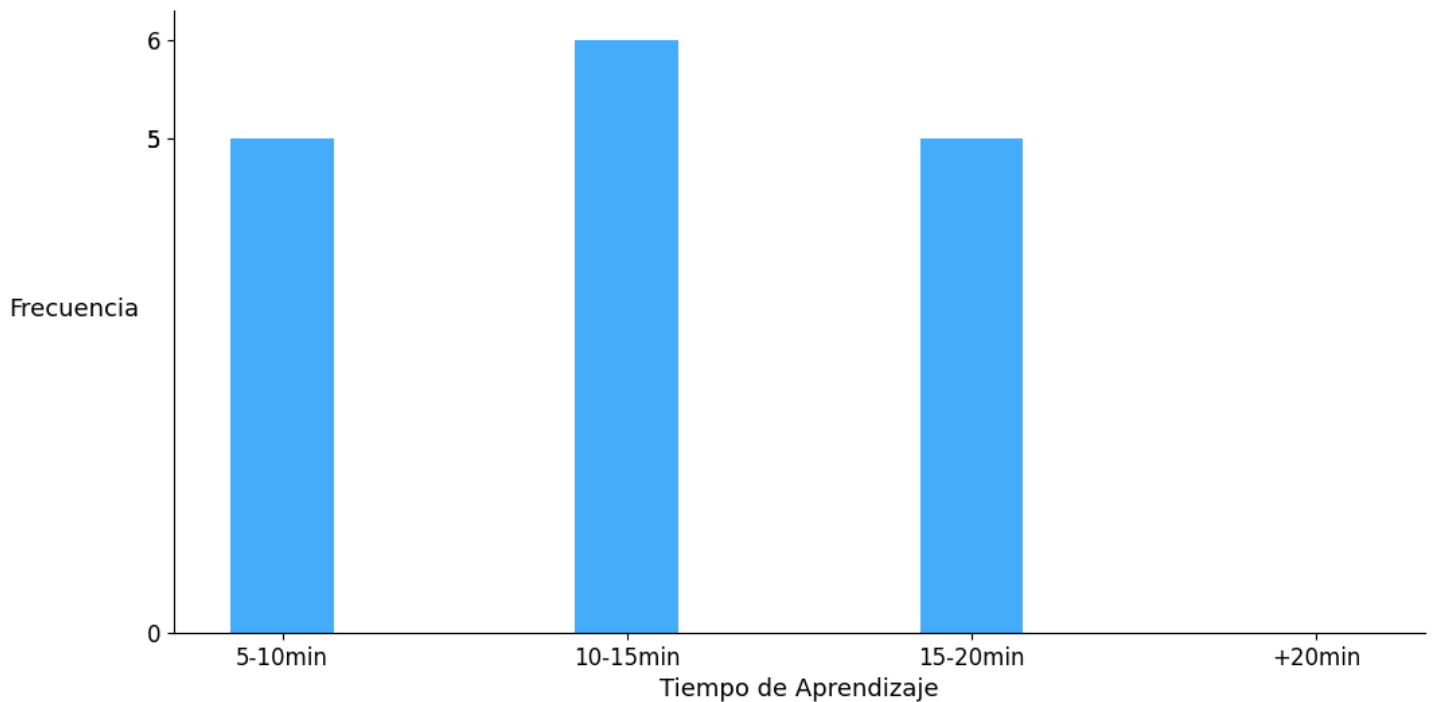


Figura 43: Tiempo empleado para aprender a usar la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Se observa que el tiempo de aprendizaje se concentra entre los 5 a 20 minutos.

Por otra parte, en relación con las preguntas sobre eficiencia y requerimientos, se planteó la interrogante: ¿La solución es intuitiva y fácil de usar? Los resultados se presentan en la Figura 44. Se observa que la totalidad de los participantes concuerdan en que la extensión de código es intuitiva y fácil de utilizar. Los arquitectos proporcionaron retroalimentación indicando que el documento de requerimientos asociado a la extensión de código es útil, ya que les permite visualizar los espacios que desean o necesitan ubicar juntos, así como los tamaños y otros detalles relevantes. Además, señalaron que la utilidad de la extensión de código es elevada, dado que facilita la organización y visualización de las áreas, así como la planificación previa a su incorporación en el plano.

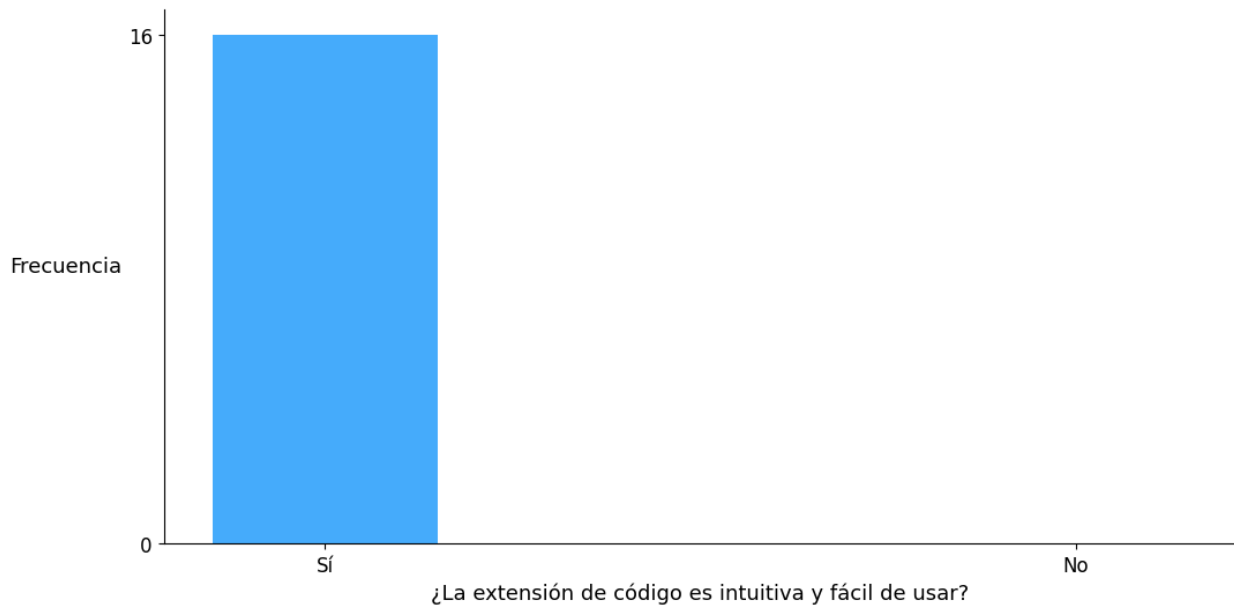


Figura 44: Percepción de la facilidad de uso de la extensión de código para la planificación de espacios. Todos los participantes concuerdan en que la extensión es intuitiva y fácil de utilizar.

A continuación, con referencia a la pregunta ¿La solución cumple con las expectativas en cuanto a la planificación de espacios generada?, los resultados se presentan en la Figura 45. En esta pregunta, 11 arquitectos se mostraron satisfechos con los resultados, mientras que 5 no lo estuvieron.

Las diferencias entre las expectativas de esta extensión de código radicarón en el nivel de detalle que buscaron obtener en las soluciones, sobre todo los arquitectos con mayor experiencia en la planificación de espacios. Estos participantes proporcionaron retroalimentación relacionada con la futura implementación de análisis de luminosidad y posicionamiento de espacios utilizando atractores dentro del sitio de planificación. Originalmente, estas funcionalidades fueron consideradas, pero debido a la extensión del trabajo, se decidió recortarlas y tomarlas en cuenta para desarrollos futuros.

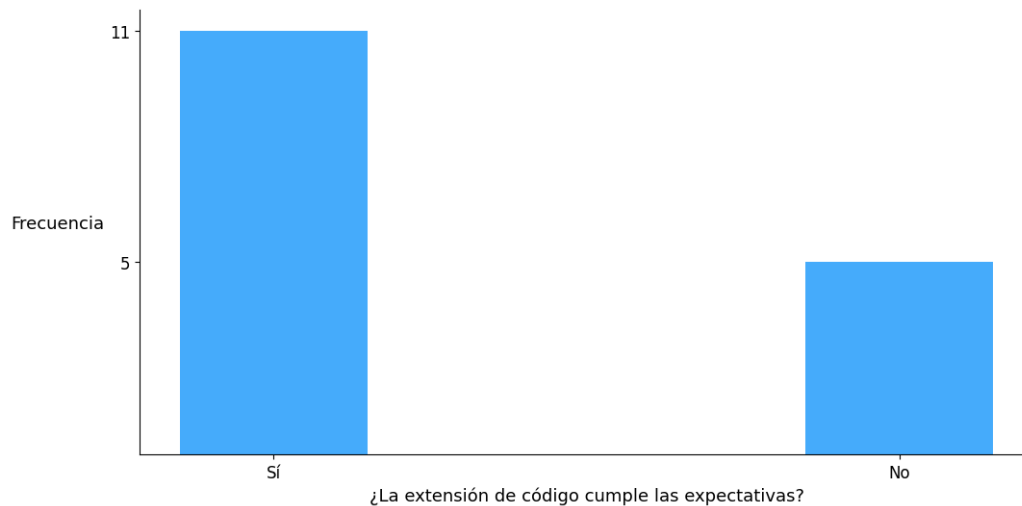


Figura 45: Cumplimiento de las expectativas de la extensión de código para la automatización de la planificación de espacios según la encuesta de evaluación empírica. En su mayoría las expectativas fueron cumplidas aunque 3 arquitectos experimentados esperaron resultados y métricas avanzadas que fueron recortadas de los alcances de la extensión debido a su complejidad.

Asimismo, en cuanto a la pregunta ¿Qué tan bien maneja la extensión de código los requerimientos y restricciones impuestas?, los resultados se pueden observar en la Figura 46. Se aprecia que las respuestas se encuentran distribuidas de manera equitativa entre percepciones medias y altas, con 8 arquitectos que seleccionaron dichas categorías. Esto indica que la extensión de código maneja de manera adecuada las restricciones impuestas dentro del archivo de planificación de espacios.

Aunque, ciertamente, los arquitectos indicaron limitaciones relacionadas con las formas dadas a los espacios y obstáculos dentro del sitio de planificación. Pero comprendieron que dichos objetos fueron simplificados a polígonos complejos, pero con una tendencia a mantenerse como rectángulos o cuadrados, debido a la simplicidad de su generación automática.

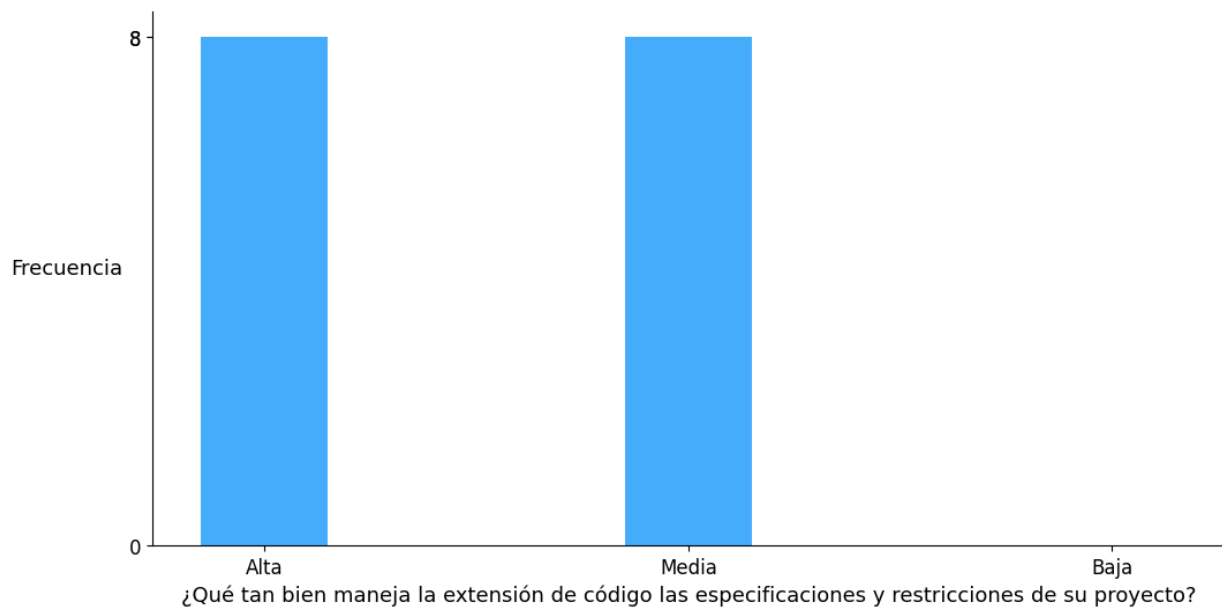


Figura 46: Cumplimiento de las restricciones y requerimientos establecidos en el documento de planificación por parte de las soluciones devueltas por la extensión de código para la planificación automática de espacios según la encuesta de evaluación empírica. Los arquitectos encuestados consideran que dichas restricciones se cumplen en una medida entre alta y media.

Finalmente, en relación con la pregunta sobre si ¿Considera que la extensión de código entrega soluciones con valores de área ocupada, porcentaje de espacios colocados y adyacencia mejores o iguales que el método manual?, los resultados se pueden observar en la Figura 47. Se aprecia que nueve arquitectos adoptaron una postura neutral, es decir, los diseños entregados no les convencieron lo suficiente como para tomar una decisión clara. Por otro lado, siete arquitectos estuvieron de acuerdo en que la extensión de código entrega soluciones mejores o iguales al método manual.

En relación con estas decisiones, se recopilieron comentarios y observaciones importantes. Por ejemplo, se identificó que los diseños al ser guiados por un algoritmo de posicionamiento codicioso a menudo no aprovecha de manera eficiente los espacios sobrantes en el sitio de planificación. No obstante, las soluciones generadas representan, en general, una fuente de inspiración para los trabajos de los arquitectos, ya que se observa un conjunto de posibles soluciones que exploran el universo de opciones disponibles.

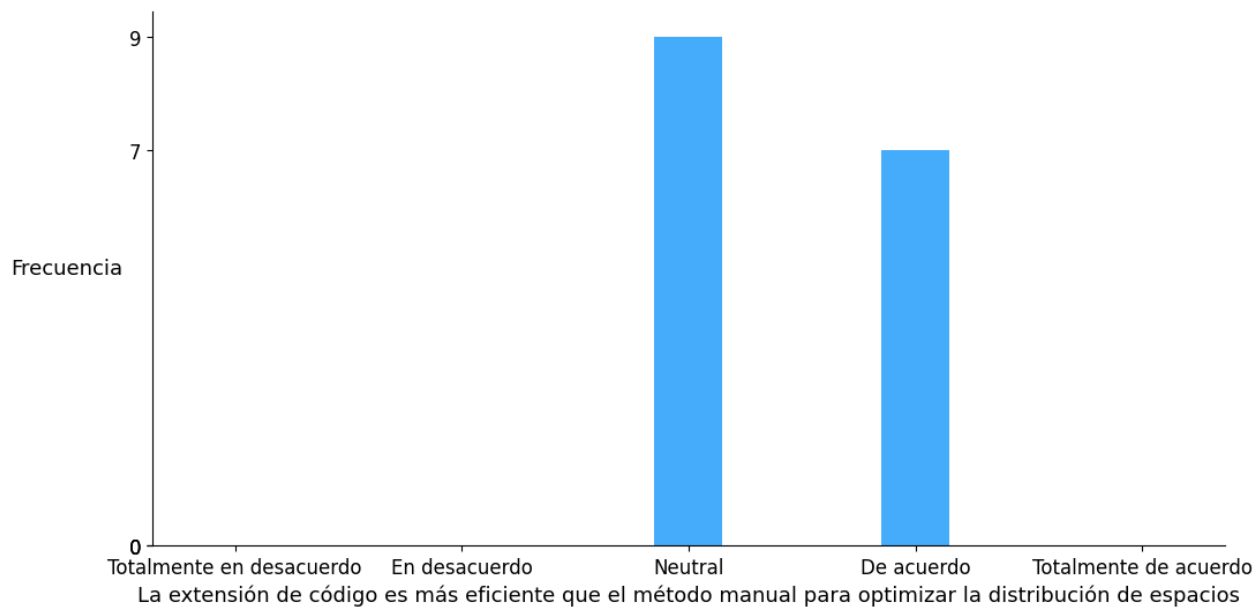


Figura 47: Satisfacción con el área ocupada, porcentaje de espacios colocados y adyacencia de los diseños entregados por la extensión de código para la planificación de espacios según la encuesta de evaluación empírica. En general, los arquitectos mantienen una posición neutral sobre la satisfacción de los resultados obtenidos.

Por último, se analizan las preguntas relacionadas con la satisfacción y la intención de uso de la extensión de código. Respecto a la pregunta “¿Qué tan satisfecho está con la solución en general?”, los resultados se pueden observar en la Figura 48. En ella se aprecia que, en general, los arquitectos se encuentran satisfechos con el funcionamiento de la extensión de código, sus aplicaciones y los resultados obtenidos.

Por otro lado, cuatro de los arquitectos encuestados se mostraron neutrales en cuanto a esta pregunta. Esto se debe a que varios de estos usuarios se mantuvieron a la expectativa de una aplicación más rigurosa de las reglas de diseño que ellos siguen habitualmente. No obstante, la solución resultó cumplir con las expectativas de la mayor parte de los arquitectos participantes en el estudio, quienes la consideran una propuesta inicial que resuelve una gran parte de los desafíos específicos de cada proyecto de diseño.

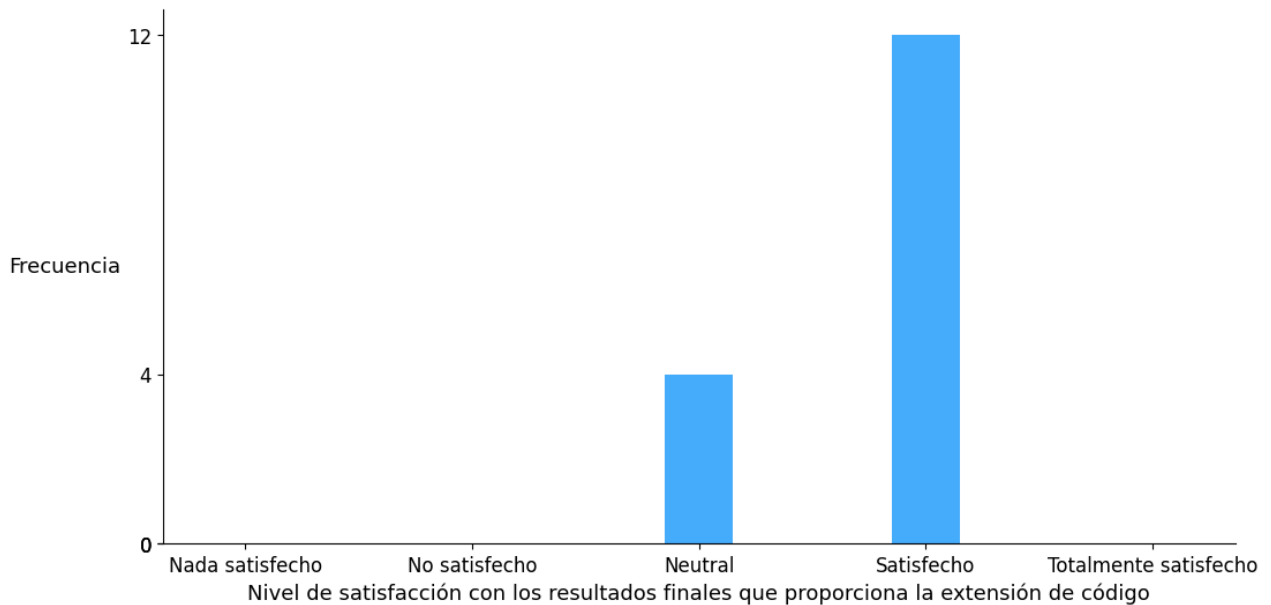


Figura 48: Satisfacción con la extensión de código de planificación de pisos según la encuesta de evaluación empírica. Gran parte de los arquitectos se encuentran satisfechos con la extensión de código, aunque se tienen posiciones neutrales por parte de una parte del estudio.

Finalmente, con relación a la pregunta sobre si ¿Recomendaría esta solución a otros profesionales del sector? Los resultados se observan en la Figura 49, y se observa que todos los arquitectos concuerdan en recomendar el uso de la extensión de código a otros profesionales del sector. Esto muestra una intensidad de uso clara de la extensión de código para la planificación automática de espacios, ya que los arquitectos encontraron potencial en la misma y sus capacidades de expansión con futuras funcionalidades.

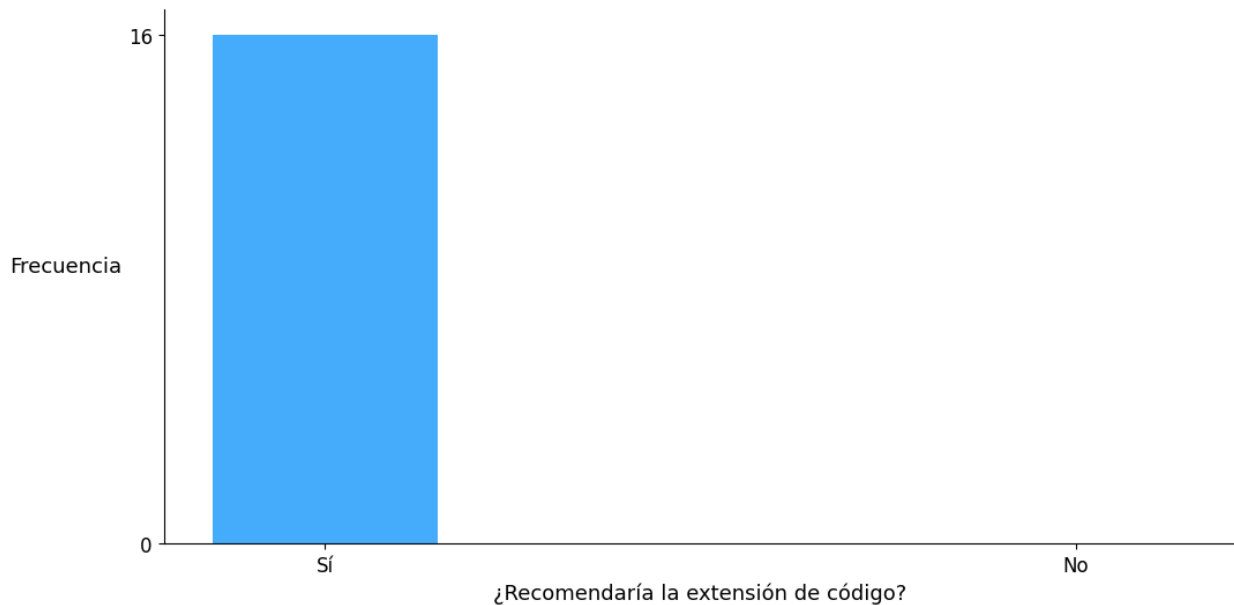


Figura 49: Cantidad de personas que recomendarían la extensión de código de planificación de espacios según la encuesta de evaluación empírica. Todos los arquitectos concuerdan en recomendar la extensión de código a otros profesionales.

En última instancia, la extensión de código resulta en un mayor ahorro de tiempo para los arquitectos con menor experiencia, debido a su falta de familiaridad con el proceso de planificación de espacios. Por el contrario, los profesionales más experimentados valoran el ahorro de tiempo que les proporciona la herramienta para abordar tareas más complejas y minuciosas.

Además, en este ámbito, los arquitectos tienden a guiarse de manera subjetiva. Por consiguiente, la extensión de código ofrece también una forma cuantificable y parametrizada de comparar las distintas propuestas de planificación, lo cual resulta beneficioso para el desarrollo de sus proyectos.

6. Discusión

Este estudio demostró que las extensiones de código propuestas, desarrolladas conjuntamente en Autodesk Revit y DynamoBIM, son fáciles de usar, intuitivas y eficientes para los profesionales del área de la arquitectura. La mayoría de los usuarios las encontraron sencillas de utilizar y recomendarían su empleo a otros profesionales del sector en la ciudad de Cuenca, Ecuador; indicando una clara aceptación de estas herramientas. Esta información es fundamental, ya que la facilidad de uso es un factor clave al implementar/iniciar el uso de nuevas tecnologías. Además, la mayoría de los usuarios indicó que la solución cumplió sus expectativas, destacando la utilidad práctica de estas herramientas para resolver tareas específicas de diseño y construcción. Sin embargo, se abstuvieron de recomendar nuevas funcionalidades que potencien aún más estas extensiones.

Es particularmente significativo no solo el ahorro de tiempo promedio de 6.75 horas al mes en la extensión de código de generación de volumetrías y de 5.95 horas al mes en la extensión de código de planificación de pisos, sino también el ahorro que se obtiene en volumetrías difíciles y con arquitectos de poca experiencia, que llega a ser de 19.79 horas en la solución de volumetrías y 21.46 horas en la solución de planificación de espacios. Esto se debe a dos factores:

Primero, un arquitecto experto puede ahorrar mucho tiempo porque algunos trabajos solicitados pueden tener un mayor grado de dificultad, pero para estos arquitectos con gran experiencia, esos trabajos son parte de su día a día. Por lo tanto, el ahorro de tiempo que otorga la herramienta es muy significativo en estos casos. Además, en la solución de planificación de espacios, la herramienta entrega métricas y valores que facilitan aún más elegir la mejor opción. Cabe destacar que la herramienta no distingue si se trata de un terreno o una zona de planificación muy compleja o muy simple, y las soluciona de la misma manera y con el mismo grado de dificultad, apoyando a los arquitectos que, cuando los trabajos son más complejos, sienten un aumento significativo en el tiempo necesario para desarrollarlos.

El segundo factor es cómo ayuda a los arquitectos menos expertos a competir con archi-

tectos con experiencia en la entrega de respuestas a clientes y en el desarrollo más rápido de sus planos. Se sabe que a un arquitecto le tomará tiempo ganar experiencia, aprender a buscar en la normativa y encontrar técnicas de búsqueda de resultados optimizados, pero al usar las extensiones de código presentadas, verá esta brecha de experiencia, traducida en dinero y confianza de los clientes, reducirse significativamente, permitiéndole salir al mundo laboral de forma más competitiva.

En cuanto al cumplimiento de expectativas en las extensiones de código, se observó que estas cumplen con las expectativas de la mayoría de los usuarios. Sin embargo, algunos no vieron satisfechas sus expectativas porque esperaban formas más complejas en el caso de la volumetría y planificaciones más llamativas en el caso de la otra solución. No obstante, la finalidad de las herramientas no es proporcionar resultados complejos o llamativos, sino resultados optimizados y eficientes.

Por otro lado, se observó que la solución de generación de volumetrías cumplió más con las expectativas que la solución de planificación de espacios. Esto se debe a que la generación de volumetrías está muy sujeta a la normativa, mientras que la planificación de espacios no. Al tener una normativa tan flexible y poco rigurosa, la planificación de espacios tiende a ser más subjetiva. Incluso los arquitectos, en la retroalimentación / evaluación, indicaron que las métricas proporcionadas ayudan a establecer una mejor base de comparación. Esto resalta aún más la utilidad de las extensiones de código que permiten la generación de estructuras en base a ciertas características, ya que permiten una evaluación más objetiva y precisa de las diferentes opciones disponibles, lo que facilita la toma de decisiones en el proceso de diseño y planificación arquitectónica. Vale la pena destacar, que al momento de desarrollar estas extensiones de código, se encontró mayor dificultad en la planificación de espacios debido precisamente a la libertad y subjetividad intrínseca en la generación de respuestas.

En este estudio se observó que la normativa en la extensión de código de generación automática de volumetrías y las restricciones establecidas en los archivos de planificación de espacios utilizados en la extensión de planificación automática de espacios se cumplen de manera adecuada. Esto es crucial, ya que los errores pueden resultar en altos costos por multas para los arquitectos. Al destacar esta característica se resalta otra de las grandes

ventajas y apoyos que ofrece la solución. La fiabilidad y confianza que entregan los resultados de estas soluciones son fundamentales para los profesionales, ya que les brindan la seguridad de cumplir con la normativa y restricciones evitando así posibles penalizaciones.

Finalmente, la satisfacción general con la solución fue alta, con los arquitectos indicando un deseo de más funciones adicionales como la creación automática de otros elementos arquitectónicos. La intención de uso es alta, ya que todos recomendarían la herramienta a otros profesionales, indicando una fuerte aceptación y reconocimiento de su valor en el sector.

Los resultados de este estudio están en línea con investigaciones previas que han demostrado los beneficios de la automatización en la arquitectura. Los estudios expuestos en el apartado de estado del arte muestran que la implementación de tecnologías similares ha resultado en mejoras significativas en la eficiencia y la precisión de los proyectos. Estos hallazgos refuerzan la idea de que herramientas como Revit Dynamo pueden transformar la práctica arquitectónica.

Sin embargo, este estudio presenta algunas limitaciones, una de ellas es el tamaño de la muestra y la posible influencia de sesgos en las respuestas de los usuarios. Futuros estudios podrían ampliar el alcance de esta investigación a diferentes contextos geográficos y tipos de proyectos, así como explorar mejoras adicionales en la interfaz y funcionalidad de la herramienta. Además, sería recomendable investigar el desarrollo de nuevas extensiones de código basadas en la retroalimentación entregada por los arquitectos. Esto implicaría generar funcionalidades propuestas directamente a partir de las necesidades expresadas, lo que permitiría avanzar más de forma mas eficiente en la mejora de las soluciones existentes, evitando así repetir el proceso de investigación ya realizado en este trabajo de titulación.

7. Conclusiones

La investigación ha confirmado que las extensiones de código desarrolladas en Revit Dynamo son herramientas útiles y fáciles de usar, capaces de mejorar significativamente los procesos de diseño y construcción. Los resultados demuestran que estas soluciones no solo cumplen con los objetivos de proporcionar una interfaz intuitiva y funcional, sino que también ofrecen un ahorro de tiempo considerable y aseguran el cumplimiento de normativas y especificaciones.

Este estudio contribuye al campo de la construcción y el diseño al evidenciar que la automatización a través de generación de estructuras por medio de extensiones de código en Revit Dynamo puede transformar la manera en que se gestionan las especificaciones y restricciones en proyectos de construcción. La alta percepción de utilidad y la intención de uso futuro de la solución reflejan su potencial para ser adoptada ampliamente en la industria.

En términos prácticos, se recomienda la adopción de esta herramienta en prácticas de diseño y construcción para aumentar la eficiencia en el uso de suelo y garantizar el cumplimiento normativo, especialmente en la ciudad de Cuenca, Ecuador. Además, se sugiere una formación continua para los profesionales en el uso de esta tecnología, con el fin de maximizar los beneficios observados en términos de ahorro de tiempo y precisión en el manejo de especificaciones.

En resumen, las extensiones de código desarrolladas en este trabajo de titulación e incluidas en Revit Dynamo no solo facilitan el trabajo de arquitectos, sino que también pueden ser un factor decisivo en la evolución hacia métodos de construcción más eficientes y precisos. Este estudio establece una base sólida para futuras investigaciones y mejoras, promoviendo un enfoque más innovador y tecnológico en la industria de la construcción.

Referencias

- Aalaei, M., M. Saadi, M. Rahbar, and A. Ekhlassi (2023, 11). Architectural layout generation using a graph-constrained conditional generative adversarial network (gan). *Automation in Construction* 155.
- Agkathidis, A. (2015). *Generative design methods implementing computational techniques in undergraduate architectural education*. MaProceedings of the 33rd eCAADe Conference 2, 47–55.
- AUGI. Extending BIM design value using the revit api. <https://www.augi.com/articles/detail/extending-bim-design-value-using-the-revit-api>. Accessed: 2024-1-4.
- Aupy, D. Optimizing spaces flows dynamo refinery.
- Autodesk (2024). About revit.
- Azadi, S. and P. Nourian (2021). A modular generative design framework for mass-customization and optimization in architectural design. Volume 1, pp. 285–294. Education and research in Computer Aided Architectural Design in Europe.
- BIM, K. (2015, Feb). ¿que es el bim? building information modeling.
- Bocco, G., C. Lemus, and P. Velthuis. Métodos de investigación en ingeniería del software.
- Bohnacker, H., B. Groß, J. Laub, and C. Lazzeroni (2009). *Generative Gestaltung: entwerfen, programmieren, visualisieren*. Schmidt.
- Bustillos, F. R. A. La importancia del cos, cus y número de cajones de estacionamiento en el desarrollo urbano. *En Concreto* 81, 028.
- Castillo, A. and J. Martínez (2018). *Razones por las cuales los proyectos de construcción no se ejecutan como se diseñan originalmente*.
- Chaillou, S. (2020). *ArchiGAN: Artificial intelligence x architecture*. In Architectural Intelligence, pp. 117–127. Singapore: Springer Nature Singapore.

Cudzik, J. and K. Radziszewski (2018). *Artificial intelligence aided architectural design*. In eCAADe proceedings. eCAADe.

Cuenca, P. T. (2022). *Plan de uso y gestión de suelo del cantón cuenca anexo 8: Normas de arquitectura y urbanismo*. <https://n9.cl/dpb60>. Accessed: 2024-1-7.

Das, S., C. Day, A. Hauck, J. Haymaker, and D. Davis (2016). *Space plan generator. rapid generation evaluation of floor plan design options to inform decision making*.

Dastanpour, A., S. Ibrahim, and M. Mashinchi (2014, 03). *Used genetic algorithm for support artificial neural network in intrusion detection system*.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). *A fast and elitist multiobjective genetic algorithm: Nsga-ii*.

Dunaevskiy, E. Y., Odessa State Academy of Civil Engineering and Architecture, A. P. Lisova, A. R. Besedina, Odessa State Academy of Civil Engineering and Architecture, and Odessa State Academy of Civil Engineering and Architecture (2022, December). *Prospects for the development of the use of artificial intelligence in design and construction*. Regional problems of architecture and urban planning (16), 186–193.

Dutta, K. and S. Sarthak (2011, 12). *Architectural space planning using evolutionary computing approaches: A review*. Artificial Intelligence Review 36, 311–321.

Eastman, C., D. Fisher, G. Lafue, J. Lividini, D. Stoker, and C. Yessios. *An outline of the building description system*.

Espín, A., R. Advisor, . René, and A. Mancho (2020). *An evolutionary algorithm for solving the two-dimensional irregular shape packing problem combined with the knapsack problem*.

Gan, V. J. (2022, 2). *Bim-based graph data model for automatic generative design of modular buildings*. Automation in Construction 134.

Goldstein, R., S. Breslav, K. Walmsley, and A. Khan. *Spaceanalysis: A tool for pathfinding, visibility, and acoustics analyses in generative design workflowsaided design*; g.2.2 discrete mathematics: Graph theory.

- Hardin, B. (2009). BIM and Construction Management: Proven Tools, Methods, and Workflows. Hoboken: John Wiley & Sons.
- Harris, A. (2021, Jun). *la en la fabricación: su empleo e importancia en las fábricas del futuro*.
- Ismail Agha, D. (2019, 11). *The impact of artificial intelligence on the future of architecture architects (the revolution of artificial intelligence)*.
- Katoch, S., S. S. Chauhan, and V. Kumar (2021). *A review on genetic algorithm: past, present, and future*. Multimedia Tools and Applications 80, 8091–8126.
- Ko, J., B. Ennemoser, W. Yoo, W. Yan, and M. J. Clayton (2023, 10). Architectural spatial layout planning using artificial intelligence.
- [NBIMS], N. (2022). Building information modeling (bim) | wbdg - whole building design guide.
- NBSEnterprises (2020). National bim report 2020.
- Onur, Z. and D. F. Nouban (2019, September). BIM software in architectural modelling. *International Journal of Innovative Technology and Exploring Engineering* (11), 2089–2093.
- Pan, Y. (2022, 11). Integrating bim and ai for smart construction management: Current status and future directions. *Archives of Computational Methods in Engineering* 30, 1–30.
- Petrova, E. A. (2019). *AI for bim-based sustainable building design integrating knowledge discovery and semantic data modelling for evidence-based design decision support*.
- Reeves, C. R. (2010). Genetic Algorithms, pp. 109–139.
- Sacks, R., M. Girolami, and I. Brilakis (2020). *Building information modelling, artificial intelligence and construction tech*. Developments in the Built Environment 4, 100011.
- Salman, A., M. Khalfan, and T. Maqsood (2012, 12). Building information modeling (bim): Now and beyond. *Australasian Journal of Construction Economics and Building* 12, 15.
- Sheets, C. (2023, Nov). *Designs*.

Smorzhenkov, Nikita and Ignatova, Elena (2021). *The use of generative design for the architectural solutions synthesis in the typical construction of residential buildings*. E3S Web Conf. 281, 04008.

Stanojević, D., I. Peško, and V. Mučenski (2023). Improving construction projects and reducing risk by using artificial intelligence.

Tamke, M., P. Nicholas, and M. Zwierzycki (2018, 6). Machine learning for architectural design: Practices and infrastructure. *International Journal of Architectural Computing* 16, 123–143.

Wang, L., J. Liu, Y. Zeng, G. Cheng, H. Hu, J. Hu, and X. Huang (2023, 10). *Automated building layout generation using deep learning and graph algorithms*. Automation in Construction 154.

Zhang, J., N. Liu, and S. Wang (2021, August). Generative design and performance optimization of residential buildings based on parametric algorithm. *Energy Build.*@(111033), 111033.

Zheng, H. and P. F. Yuan (2021, 11). A generative architectural and urban design method through artificial neural networks. *Building and Environment* 205.

Zhuang, X., Y. Ju, A. Yang, and L. Caldas (2023). *Synthesis and generation for 3d architecture volume with generative modeling*. International Journal of Architectural Computing@ (2), 297–314.