

**DOI:** <https://doi.org/10.46296/ig.v9i17.0344>

## **APLICACIÓN DE LA RED GENERATIVA ADVERSARIAL CONDICIONAL PARA LA GENERACIÓN DE SOLUCIONES ARQUITECTÓNICAS**

### **APPLICATION OF CONDITIONAL GAN FOR THE GENERATION OF ARCHITECTURAL PLANNING SOLUTIONS**

Baque-De Los Santos Patricio Ivan <sup>1</sup>

<sup>1</sup> Universidad Politécnica de San Petersburgo "Pedro el Grande", Instituto de Ingeniería Civil.  
San Petersburgo, Rusia. Correo: bakedelossantos.p@edu.spbstu.ru.  
ORCID ID: <https://orcid.org/0009-0000-3465-6380>.

#### **Resumen**

El uso de modelos generativo con inteligencia artificial dentro de la arquitectura está en un constante auge con un principal problema: generar variantes o diseños no siempre y normativamente construible. El trabajo de investigación implementa y evalúa el uso de la red generativa adversarial condicional como alternativa al uso de modelos generativo dentro de la arquitectura, con el objetivo de generar modelos a partir de una base de datos de 400 pares de planos arquitectónicos, los cuales contienen criterios de diseño en zonificación y distribución espacial de edificios residenciales. La implementación del modelo cGan reveló que las imágenes obtenidas poseen fidelidad con los planos de referencia, además de un aprendizaje progresivo y secuencial en su etapa de entrenamiento y generación de imágenes. El estudio también revela que el uso de estos modelos es ajustable a objetivos dependientes de cada proyecto, posicionándose como una herramienta que es capaz de generar diseños coherentes y optimizar el tiempo de modelación en tareas repetitivas y mecánicas en el desarrollo de planos de planta.

**Palabras clave:** modelos generativos; soluciones de diseño; zonificación espacial.

#### **Abstract**

The use of generative models with artificial intelligence in architecture is experiencing constant growth, with one main problem: generating design variants that are not always functionally or normatively feasible for construction. This research evaluates the use of the Conditional Generative Adversarial Network as an alternative to generative models in architecture, with the objective of generating models from a database of 400 pairs of architectural floor plans, which contain design criteria related to zoning and spatial distribution of residential buildings. The implementation of the cGAN model revealed that the generated images maintain fidelity to the reference floor plans, as well as demonstrating progressive and sequential learning during the training and image generation stages. The study also reveals that the use of these models can be adjusted to project-specific objectives, positioning itself as a tool capable of generating coherent designs and optimizing modeling time for repetitive and mechanical tasks in the development of floor plans.

**Keywords:** generative models; architectural layout generation; spatial planning.

#### **Información del manuscrito:**

**Fecha de recepción:** 09 de marzo de 2026.

**Fecha de aceptación:** 11 de mayo de 2026.

**Fecha de publicación:** 15 de junio de 2026.



## 1. Introducción

El uso de modelos generativos dentro de la arquitectura, han demostrado una optimización en procesos como el diseño conceptual [1], diseño de formas en 3 dimensiones [2], [3], [4] modelación de planos de planta residenciales y de vivienda [5], [6], planos de planta escolares [7], fachadas y secciones [8], [9] e inclusive procesos de modelación estructural [10], [11], [12]. Sin embargo, dichos resultados de optimización no son obtenidos a partir de una sola herramienta generativa tal como lo revela el artículo [13], en el que los autores analizan varias investigaciones que implementan modelos generativos y de los cuales sus resultados surgen a partir de una combinación con otros modelos de programación y modelación paramétrica, tales como Dynamo, Grasshopper, Rhino y plugin de Revit.

Bajo ese contexto, la red generativa adversarial (Gan), aunque aparece por primera vez en 2014, con el objetivo de desarrollar muestras “samples” usando Data sets de dígitos, rostros e imágenes de objetos, mediante un entrenamiento

entre dos agentes: un generador que se encarga de crear imágenes a partir de la base de datos y un discriminador que estima la congruencia y originalidad de las imágenes creadas por el generador clasificándolas como falsas o reales, los autores revelaron que el modelo Gan es capaz de desarrollar muestras coherentes y una viabilidad al uso de la red generativa adversarial [14].

Hoy en día la red generativa adversarial ha desarrollado modificaciones y variaciones en su arquitectura de entrenamiento [15], uno de ellos es el modelo condicional Gan “cGan”, que surge del modelo original, pero con la variante “condición” (diagramas o etiquetas con niveles de detalle gráfico y técnico), el cual se integra a la base de datos de entrenamiento y tanto el generador como el discriminador están limitados a generar y evaluar datos sujetos al objetivo o dato referencial [16]. Entre los principales resultados e investigaciones alineados a nuestro objetivo están: [17] en donde el autor logra generar variantes de diseños (planos de vivienda, con mobiliario) bajo el criterio de “segmentación por

colores" en un proceso de tres etapas con un data set diferente y acorde a cada progreso, a diferencia de [18] que solo aplica un entrenamiento de dos etapas, partiendo de un data set con muros sin mobiliario, para luego generar el plano amueblado final. Otro trabajo que sigue la misma línea de investigación es [19], con el objetivo de generar planos de planta con una lógica espacial de distribución, a partir de un dataset el cual denominaron ALPLAN (Architectural-like Plan Dataset), donde la condición está definida por un plano de contorno (perímetro de cada sección de una planta) para que el generador interprete cada información específica como las coordenadas del centro de los espacios y su área, obteniendo muestras de la imagen objetivo.

En ese sentido, la actual investigación implementa el modelo generativo cGan con el objetivo de evaluarlo para el desarrollo de muestras híbridas referentes a planos de planta residenciales diseñados bajo normativa rusa siguiendo una lógica de distribución entre zonas comunes, zonas privadas y zonas de acceso y

evacuación. Además de fomentar el uso de modelos generativos dentro del área de modelación y conceptualización de espacios a profesionales ajenos al campo de la programación.

## 2. Metodología

El modelo cGan se divide básicamente de 4 partes:

- Componentes de entrenamiento: Generador (G) y discriminador (D);
- Adquisición de datos pareados: Input (x) "imagen de entrada" y Output (y) "imagen objetivo";
- Arquitectura de regulación y estabilización adversarial;
- Y entornos de programación

### Componentes de entrenamiento

Los componentes principales de entrenamiento de la red generativa cGan son los modelos: Generador (G) y Discriminador (D), en donde el generador crea muestras parecidas a la imagen objetivo y el discriminador evalúa las muestras del generador, creando un proceso de "competición" de ahí el termino adversarial, es decir entre mejor sea el resultado del generador, mayor

será la dificultad del discriminador juzgar las muestras y en ese sentido sea incapaz de diferenciar entre lo real y lo falso [16], [20]. Dicho proceso adversarial se define mediante la ecuación 1.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log (1 - D(x, G(x, z)))] \quad (1)$$

Donde:

$G(x, z)$ : es el generador que aprende un mapeo desde una imagen de entrada ( $x$ ) y un vector de ruido aleatorio ( $z$ ) para generar muestras con referencia a la imagen objetivo ( $y$ ).  $D(x, y)$ : es el discriminador que observa tanto la condición (imagen de entrada  $x$ ) como la imagen resultante (ya sea la real y o la generada) para decidir si el par es real.  $\mathbb{E}_{x,y}[\log D(x, y)]$ : es la probabilidad logarítmica esperada de que el discriminador identifique correctamente los pares de datos reales.  $\mathbb{E}_{x,y}[\log(1-D(x, G(x, z)))]$ : es la probabilidad logarítmica esperada de que el discriminador identifique correctamente que la imagen producida por el generador es falsa.

Dentro del proceso adversarial se debe añadir una función de pérdida  $L1$  junto con un coeficiente

( $\lambda$ ), esto ayuda a estabilizar la división de tareas entre ambos componentes, justificado por [20], obteniendo como resultado la ecuación 2.

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (2)$$

Donde:

$\min G$ : es la expresión del generador para minimizar datos de  $LcGan$ , y engañar al discriminador.

$\max D$ : es la expresión del discriminador que busca maximizar los datos de  $LcGan$  para clasificar correctamente las muestras reales y falsas.

$\lambda$ : factor que equilibra la importancia entre la calidad de la imagen y la fidelidad con las imágenes referenciales.

$L1(G)$ : función de pérdida que se calcula para estimar qué tan bien produce el generador imágenes similares a las reales.

**Adquisición de datos pareados:**  
**Input (x)** “imagen de entrada” y  
**Output (y)** “imagen objetivo”

El estudio parte de la necesidad de acelerar procesos mecánicos y

repetitivos dentro de la modelación arquitectónica, específicamente a la distribución espacial de planos de planta en edificios, bajo esa necesidad, el objeto de este estudio consiste en modelar planos de planta basados en las normativas [21], [22], [23], las cuales regulan estructuras y edificios públicos, espacios residenciales y presentan soluciones arquitectónicas y de planificación para edificios de varios apartamentos. Dentro de los criterios generales, se consideran los siguientes:

- Un edificio residencial se considera óptimo, cuando son de tipo seccional o seccional-corredor, es decir el edificio se divide por bloques donde cada uno actúa individualmente ante necesidades o situaciones de emergencia.
- Se considera solamente dos tipos de piso: estructura del primer piso y estructura de un piso típico (segundo piso y superiores)
- Las zonas comunes deben diseñarse como un solo bloque y poseer acceso independiente desde el exterior entre las escaleras o rutas de evacuación y entrada principal, con la

característica de ser un espacio amplio y además de diseñar cabinas en la entrada principal para reducir climas el acceso a temperaturas bajas.

- Un piso debe responder ante la densidad poblacional, lo que en la práctica se define a los espacios (apartamentos) como: estudio, apartamento de un dormitorio (1κ), apartamento de dos dormitorios (2κ), de 3 (3κ) y de 4 dormitorios (4κ).
- Cada espacio interno del apartamento debe poseer contacto con el exterior para satisfacer necesidades de ventilación e iluminación.

En la parte técnica, se consideran lo siguiente:

- Se debe proveer espacios exclusivos para sistemas de calefacción, conexiones eléctricas y bajadas sanitarias cercanos a baños y cocina.
- Para edificios entre 1 a 19 pisos se debe proporcionar dos elevadores y para edificios mayores a 19, entre 3 a 4 elevadores.
- Dimensiones mínimas (tabla 1)

Tabla 1. Dimensiones mínimas de elementos arquitectónicos

| Descripción                                          | Dimensión (m) |
|------------------------------------------------------|---------------|
| Tramo de escalera                                    | >1.2          |
| Elevadores                                           | 2.1 x 1.1     |
| Ancho de vestíbulo de elevadores                     | 1.5 - 2.5     |
| Ancho de pasillo (corredor) con longitud menor a 40m | >1.4          |
| Ancho de pasillo (corredor) con longitud mayor a 40m | >1.6          |
| Cabinas de acceso o entrada                          | 1.6x1.6       |

El trabajo de investigación no considera dimensiones específicas de cocina, habitaciones, balcón, baños y cantidades específicas de apartamentos por planta porque pertenecen a un diseño exhaustivo de planos de vivienda y residenciales por lo que se toman valores referenciales y generales, además de que son criterios ajenos al objeto de investigación.

Una vez definidos todas las especificaciones de conceptualización y distribución de espacios para edificios multifamiliares, se procede a la modelación de cada criterio, tal como se muestra en la Fig. 1, donde se le asigna un color específico a cada tipo de espacio

Figura 1. Elementos arquitectónicos de un plano de planta en edificios residenciales, Elaboración propia.



Por consiguiente, para el procesamiento de datos pareados se elabora como Input a una imagen que contiene el contorno de un plano, en el que la variante "condición" es la distribución de la zona común y la zona privada además de la etiqueta de cada

elemento espacial (Fig. 2.A). Y como Output se define a un plano de planta con los diferentes tipos de espacios definidos anteriormente (Fig. 2.B), permitiéndole al modelo comprender la lógica de ubicación y distribución: entrada-zona común central-zonas privadas.

**Figura 2.** Conjunto de Datos de entrenamiento: Imagen de entrada(2A-izquierda)-Imagen objetivo(2B-derecha). Elaboración propia



Se desarrollo un total de 400 imágenes pareadas en AutoCAD para la adquisición del Dataset, donde cada imagen pasa un proceso de redimensionamiento de 256 x 512 pixeles, considerado como buena

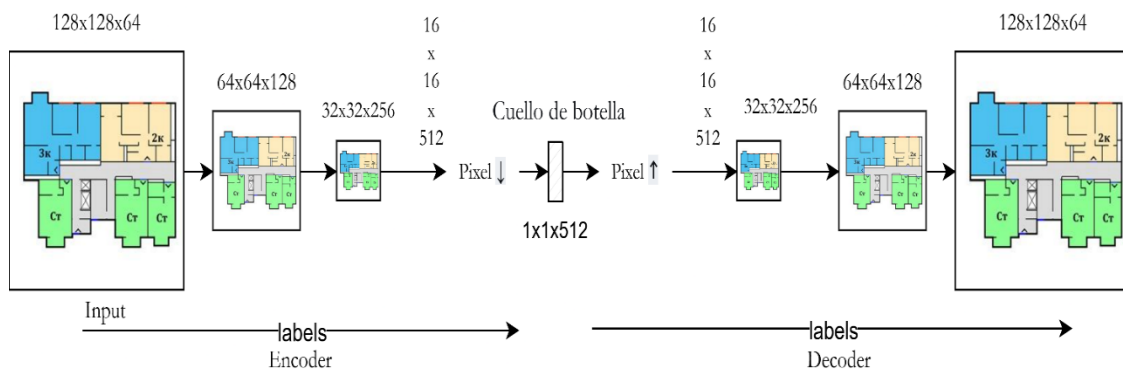
práctica para el procesamiento de datos dentro del modelo [7], [17], [20].

## Arquitectura de regulación y estabilización adversarial

El proceso de entrenamiento (arquitectura de entrenamiento) está conformado por funciones que regulan el procesamiento de datos, ejecución, generación y evaluación del modelo, entre ellas se consideran:

— Arquitectura U-Net con conexiones de salto (skip connections): proceso de redimensionamiento (encoder y decoder) a imágenes que el generador aplica por capas, donde se aplican conexiones de salto entre cada una de ellas para preservar detalles pequeños en cada redimensionamiento de la imagen (Fig. 3) [16], [20].

**Figura 3.** Arquitectura de redimensionamiento Downsampling-Upsampling (encoder-decoder). *Elaboración propia.*



- Discriminador PathGan: produce un conjunto de “parches” (división en cuadrícula de igual tamaño distribuidos en toda la imagen) que actúan como mini-discriminadores enfocados de manera local en cada uno de ellos, escalando mejores las resoluciones de una imagen [16].
- Optimizador Adam: mejora significativamente la velocidad de convergencia y la estabilidad del

entrenamiento, especialmente en problemas con dimensiones altas [15], [16].

- Normalización por lotes (Batch Normalization): ayuda a que los modelos más profundos funcionen reduciendo el cambio de covariable interna (internal covariate shift), lo que acelera el entrenamiento y es crucial para evitar el colapso [15], [16].

- Aumento de Datos (Data Jittering): permite realizar variantes y aumentar el tamaño de datos con redimensionamiento y giros a las imágenes. [20]
- LeakyReLU (Stack de bajada "downsampling" del Discriminador): permite un pequeño gradiente positivo para los valores negativos, evitando que "mueran" (se vuelvan cero) durante el entrenamiento, lo que produce mejores resultados en el stack de bajada [15], [20].
- ReLU (Stack de subida "Upsampling" del Generador): reconstruye y genera la imagen de salida, proporcionando un aprendizaje eficiente en las capas upsampling de la red [15], [20].

Finalmente, el proceso interno de la red termina con una retro-propagación de valores que recae en el discriminador, es decir el discriminador recibe un mini-lote compuesto por datos reales (y) y datos falsos ( $x^*$ ) producidos por el generador y mediante la función de pérdida determina qué tan lejos están sus predicciones de las reales (1 para real, 0 para falso). El

Generador por su parte aprende exclusivamente a través de la retroalimentación del discriminador, basándose en la incapacidad del discriminador para detectar la falsificación (Fig. 4) [16].

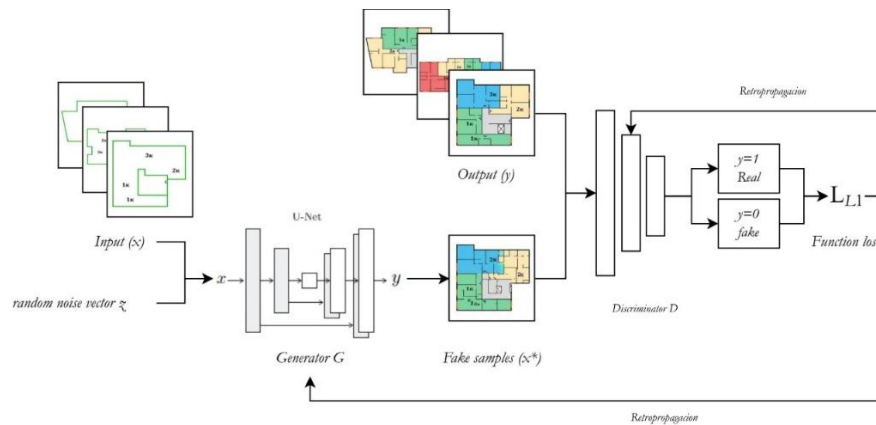
### Entornos de programación

Un modelo Gan, al ser una red neuronal adversarial que manipula grandes cantidades de datos necesita de lenguajes de programación y bibliotecas que le permitan empaquetar y abstraer funciones complejas detrás de las redes en líneas de código importables [16]. La forma correcta de ejecutar estos modelos responde a la necesidad de componentes de software y hardware, tales como el uso de librerías de TensorFlow, compatibles con sistemas operativos a Linux, Android e iOS [24]. En cuanto a hardware las características de procesamiento deben ser de al menos un procesador Intel Core i7 de 12 núcleos y una tarjeta gráfica Nvidia RTX 2060 con 6 GB de memoria [25]. Ante estas limitaciones el modelo cGan puede ejecutarse en entornos de nube gratuito como Google Colab, entorno que permite una interacción directa entre Google

drive para la integración de datos y almacenamiento, además de solucionar problemas locales de hardware y de cómputo. En él, el modelo genera muestras a partir de iteraciones, es decir el modelo toma una imagen y genera su

corresponsal, a esto se lo denomina una iteración, pero se considera una “buena” muestra cuando ha completado un sin número de iteraciones a la cual se le define como época [7].

**Figura 4.** Arquitectura de entrenamiento del modelo cGan. Elaboración propia.

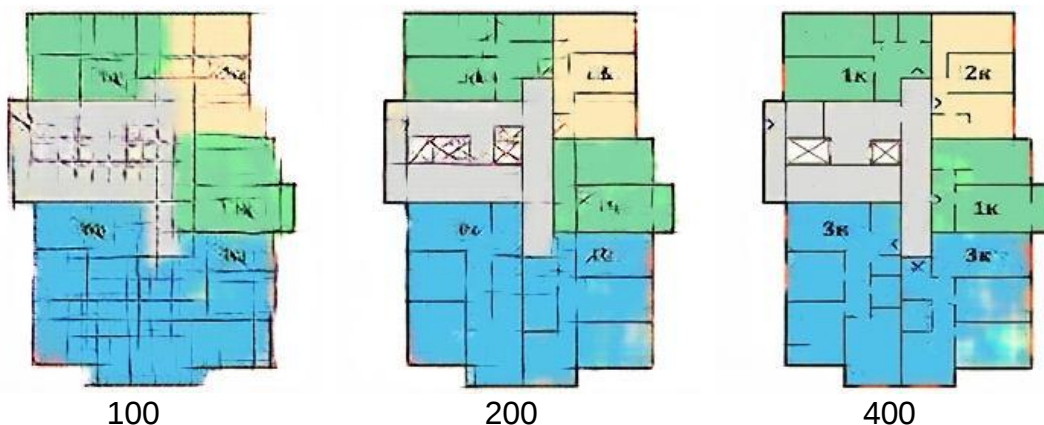


### 3. Resultados y discusión

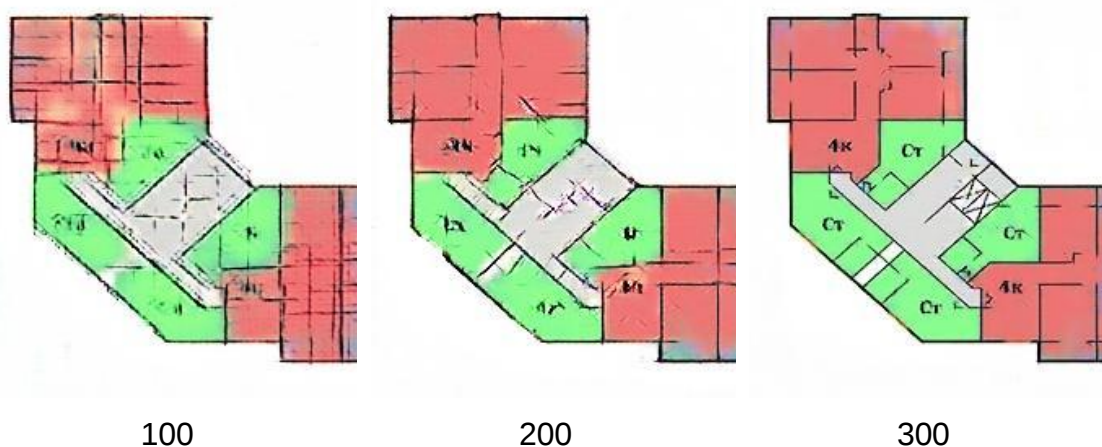
Se estableció al modelo presentar muestras (imágenes híbridas), en las épocas 100, 200 y 400 (Figs. 5-8). Se observa que en la época 400 el modelo es capaz de generar

muestras con una fidelidad coherente a la “condición” que se estableció en el Dataset, asignando colores correctamente a cada elemento y la lógica jerárquica de distribución: Entrada-zona común central-zonas privadas.

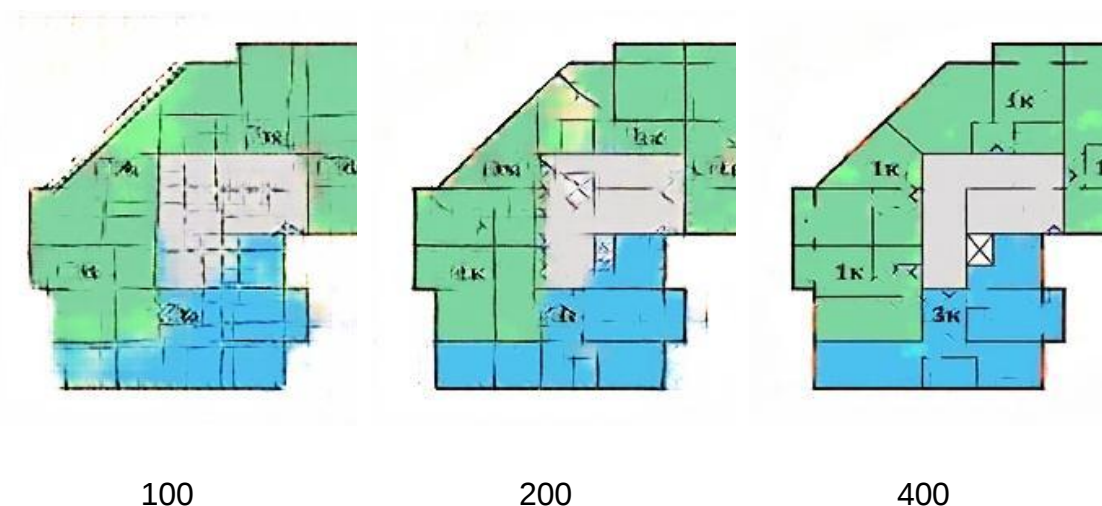
**Figura 1.** Plano de planta 1 con épocas 100, 200 y 400. Elaboración propia



**Figura 2.** Plano de planta 2 con épocas 100, 200 y 400. Elaboración propia



**Figura 3.** Plano de planta 3 con épocas 100, 200 y 400. Elaboración propia



**Figura 4.** Plano de planta 4 con épocas 100, 200 y 400. Elaboración propia



## Discusión

El desarrollo y uso de modelos generativos, juega un papel importante para modeladores, especialmente en arquitectura e ingeniería, clasificándose como: modelos para tareas específicas (task-specific models) y modelos base (foundation models) [13]. En la actualidad son conocidos los modelos Dall-e, Imagen, y Stable difusión dentro de la categoría de modelos base, los cuales son usados principalmente para la generación de imagen, sin conocer que estos modelos nacen a partir de una arquitectura de entrenamiento de modelos de difusión [26], [27].

El principal problema de estas herramientas para generar imágenes es que Dall-e e Imagen están diseñadas y evaluadas para maximizar la atención del usuario y el atractivo visual, más que la precisión técnica, funcional o normativa [26], [27]. Y aunque en arquitectura estos modelos se hayan integrado a otros, tales Chatgpt, Gemini, Midjourney y Nano banana (grandes modelos de lenguaje) para afinar resultados mediante técnicas de “text-to-image” a partir de instrucciones (prompts) dadas por un

usuario, los desafíos por controlar y obtener imágenes coherentes siguen siendo un obstáculo en la parte generativa de diseños que requieren características técnicas y normativas [28], [29].

La investigación destaca y corrobora que el desafío de generar imágenes congruentes y fieles al objetivo depende de la calidad del conjunto de imágenes [19], el cual se basa en definir anticipadamente criterios técnicos y acorde al objeto de investigación, permitiéndole al modelo no solamente a replicar formas, sino a integrar criterios dentro de sus muestras. Y no siempre depende del manejo grande de datos mayores a 1000 pares de imágenes [30].

El riesgo de generalización por parte del generador durante el redimensionamiento de imágenes para dataset pequeños también es una expectativa o problema que se puede esperar, es decir como la cantidad de imágenes es limitada para el entrenamiento, el modelo cGan tiende a enfocarse demasiado en la correlación entre las imágenes de entrada y la imagen objetivo, lo que tiende a fallar en aprender “la distribución completa”, cuando se

ejecuta el modelo o imágenes no vistas [31]. En ese sentido, para mitigar la escasez de datos sugieren técnicas como la de aumento de datos (Data Jittering) [20].

Otra observación durante el proceso de entrenamiento del modelo que se notó fue que los entornos de nube también son susceptibles a sufrir problemas de cómputo al ser entornos gratuitos, tales como: el tiempo de espera o tiempo de ejecución agotado. Bajo esa circunstancia el uso de otras plataformas como kaggle son una de las alternativas, pero con consideraciones de edición en los códigos de integración de datos, rutas de revisiones y almacenamiento para futuras implementaciones de la red generativa.

#### **4. Conclusiones**

El estudio concluye que la implementación de redes neuronales generativas, específicamente el modelo cGAN, representa una alternativa viable y eficiente para diseños de soluciones arquitectónica a pesar de tener partes extensas

como la adquisición y depuración de datos.

Aunque el desafío para profesionales ajenos a la programación es el uso y la implementación de modelos de lenguaje, se demuestra que estas herramientas pueden automatizar tareas mecánicas y repetitivas, permitiendo a los arquitectos concentrarse en la evaluación de variantes y en la toma de decisiones conceptuales.

El modelo cGAN probó ser capaz de generar planos que respetan la lógica de distribución interna y la organización funcional, manteniendo una alta concordancia con las imágenes objetivo y los requisitos normativos. Además, la investigación valida que las GANs son directamente aplicables y adaptables a los objetivos individuales de cada proyecto arquitectónico.

#### **Contribuciones de los autores**

Baque Patricio: Conceptualización (desarrollo del marco investigativo y referencias sobre la aplicación del modelo cGAN en arquitectura); Formalización de datos; Análisis

formal; Metodología (creación y depuración de una base de datos de 400 pares de imágenes para el entrenamiento del modelo generativo); Software (uso de AutoCAD para la modelación de planos, IrfanView para el redimensionamiento de pixel en lotes y el entorno de nube Google Colab para la implementación y ejecución del modelo cGAN); Análisis e interpretación de datos; Redacción, revisión y edición.

#### **Declaración de disponibilidad de datos**

- El modelo generativo usado en la investigación, además del referenciado en el cuerpo de la investigación está disponible en el repositorio Zenodo con acceso abierto: <https://doi.org/10.5281/zenodo.20085663>
- El dataset está disponible previa solicitud al autor (en casos excepcionales)

#### **Bibliografía**

- [1] Jiang, H., Xiao, Y., Hurley, R., Liu, S. (2023). The Impact of Generative AI on Architectural Conceptual Design: Performance, Creative Self-Efficacy and Cognitive Load. <https://doi.org/10.48550/arXiv.2601.10696>
- [2] De Miguel Rodríguez, J., Villafañe M. E., Piškorec, L, Sancho Caparrini, F. (2020). Generation of geometric interpolations of building types with deep variational autoencoders. Design Science. <https://doi.org/10.1017/dsj.2020.31>
- [3] Zhong, C., Shi, Y., Cheung, L. H., Wang, L. (2024). AI-Enhanced performative building design optimization and exploration. Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), 1, 59-68. <https://doi.org/10.52842/conf.caadria.2024.1.059>
- [4] Zhuang, J., Li, G., Xu, H., Xu, J., Tian, R. (2024). Text-to-City: Controllable 3D Urban Block Generation with Latent Diffusion Model. Proceedings of the 29th International Conference of the Association for Computer-Aided

- Architectural Design Research in Asia (CAADRIA), 2, 169-178. <https://doi.org/10.52842/conf.caadria.2024.2.169>
- [5] Zeng, P., Gao, W., Yin, J., Xu, P., Lu, S. (2024). Residential floor plans: Muti-conditional automatic generation using diffusion models. *Automation in Construction*, 162. <https://doi.org/10.1016/j.autcon.2024.105374>
- [6] Zhang, H., Zhang, R. (2025). Generating accesible multi-occupancy floor plans with fine-graind control using a diffusion model. *Automation in construccion*, 177. <https://doi.org/10.1016/j.autcon.2025.106332>
- [7] Karadag, I., Güzelci, O. Z., Alacam, S. (2023). EDU-AI: a twofold machine learning model to support classroom layout generation. *Construction Innovation: Information Process Management*, 23(4), 898-914. <https://doi.org/10.1108/CI-02-2022-0034>
- [8] Zhang, J., Fukuda, T., Yabuki, N., Li, Y. (2023). Synthesizing style-similar residential facade from semantic labeling according to the User-provided example. *HUMAN-CENTRIC*, Proceedings of the 28th International Conference of the Asociation for Computer-Aided Architectural Design Research in Asia (CAADRIA), 1, 139-148. <https://doi.org/10.52842/conf.caadria.2023.1.139>
- [9] Lin, H., Huang, L., Chen, Y., Zheng, L., Huang, M., Chen, Y. (2023). Research on the application of CGAN in the Design of historic Building Facades in Urban Renewal-Taking Fujian Putian Historic District as an example, China, *Buildings*, 13(6), 1478. <https://doi.org/10.3390/buildings13061478>
- [10] Gu, Y., Huang Y., Liao W., Lu, X. (2024). Intelligent design of shear wall layout based on diffusion models. *Computer-Aided Civil and Infrastructure Engineering*, 39, 3610-3625. <https://doi.org/10.1111/mice.13236>
- [11] Kösenciğ, K. Ö., Okuyucu, E. B., Balaban, Ö. (2024). Structural Plan Schema Generation Through Generative Adversarial Networks. *Nexus Network Journal*, 26, 409-427. [10.1007/s00004-024-00766-z](https://doi.org/10.1007/s00004-024-00766-z)
- [12] Leng, H., Gao, Y., Zhou, Y. (2024). ArchiDiffusion: A novel diffusion model connecting architectural layout generation from sketches to Wall Design. *Journal of Building Engineering*, 98. <https://doi.org/10.1016/j.jobbe.2024.111373>

- [13] Li, C., Zhang T., Du, X., Zhang Y., Xie, H. (2025). Generative AI models for different steps in architectural design: A literature review. *Frontiers of Architectural Research*, 14(3), 759-783.  
<https://doi.org/10.1016/j.foar.2024.10.001>
- [14] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative Adversarial Nets.  
<https://doi.org/10.48550/arXiv.1406.2661>
- [15] Jabbar, A., Li, X., Omar, B. A. (2020). A Survey on Generative Adversarial Networks: variants, application, and training. *ACM Computing Surveys (CSUR)*, 54(8), 1-49.  
<https://doi.org/10.1145/3463475>
- [16] Langr, J., Bok, V. (2019). GANs in Action: Deep learning with Generative Adversarial Networks. MANNING.  
<https://dokumen.pub/gans-in-action-deep-learning-with-generative-adversarial-networks>
- [17] Chaillou, S. (2019). ArchiGAN: Artificial Intelligence x Architecture. *Architectural Intelligence: Selected Papers from the 1st International Conference on Computational Design and Robotic Fabrication (CDRF)*, p. 117-127, 2019.  
[https://papers.cumincad.org/data/works/att/artificial\\_intelligence2019\\_117.pdf](https://papers.cumincad.org/data/works/att/artificial_intelligence2019_117.pdf)
- [18] Shi, Y.; Hong, T. C. (2026). Benchmarking pix2pix on floor Plans: Shape Grammar as a Deterministic Standard for Models Limits. In *Conference Paper: CAADRIA*.  
[https://www.researchgate.net/publication/403581693\\_Benchmarking\\_pix2pix\\_on\\_Floor\\_Plans\\_Shape\\_Grammar\\_as\\_a\\_Deterministic\\_Standard\\_for\\_Model\\_Limits](https://www.researchgate.net/publication/403581693_Benchmarking_pix2pix_on_Floor_Plans_Shape_Grammar_as_a_Deterministic_Standard_for_Model_Limits)
- [19] Guo, Y., Fang, T., Cui, Z., Stouffs, R. (2025). A dual-aspect evaluation framework for architectural-like plan generation via pix2pix series algorithms. *Frontiers of Architectural Research*, 14(6), 1516-1535.  
<https://doi.org/10.1016/j.foar.2025.04.006>
- [20] Isola, P., Yan-Zhu, J., Zhou, T., Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks.  
<https://doi.org/10.48550/arXiv.1611.07004>
- [21] Центр методологии нормирования и стандартизации в строительстве (ФГУП ЦНС), (2005). СП 31-107-2004: Архитектурно-Планировочные решения

- многоквартирных жилых  
зданий. Москва.  
<https://ohranatruda.ru/upload/iblock/9de/4294813059.pdf>
- [22] Центральный научно-исследовательский и проектно-экспериментальный институт промышленных зданий и сооружений. (2022). СП 118.13330.2022: Общественные здания и сооружения.  
[https://rkc56.ru/attach/orenburg/docs/Gosstandart\\_RF/sp\\_118\\_13330\\_2022\\_.pdf](https://rkc56.ru/attach/orenburg/docs/Gosstandart_RF/sp_118_13330_2022_.pdf)
- [23] Министерство строительства и жилищно-коммунального хозяйства российской федерации. (2022). СП 54-13330.2022: Здания жилые многоквартирные. Москва.  
[https://rkcsn-kbr.ru/images/docs/dostup\\_sreda/sp\\_54133302022.pdf](https://rkcsn-kbr.ru/images/docs/dostup_sreda/sp_54133302022.pdf)
- [24] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. (2016). TensorFlow: A system for large-scale machine learning.  
<https://doi.org/10.48550/arXiv.1605.0869>
- [25] Cira, C., Manso-Callejo, M., Alcarria, R., Pareja, T. F., Sanchez, B. B., Serradilla, F. (2021). Generative Learning for Postprocessing Semantic Segmentation Predictions: A Lightweight Conditional Generative Adversarial Network Based on Pix2Pix to Improve the Extraction of Road Surface Areas, buildings, 10(1).  
<https://doi.org/10.3390/land10010079>
- [26] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y. (2023). Improving Image Generation with Better Captions.  
<https://cdn.openai.com/papers/dall-e-3.pdf>
- [27] Imagen 3 Team Google (2024). Imagen 3.  
<https://doi.org/10.48550/arXiv.2408.07009>
- [28] Horvath, A.S., Pouliou, P. (2024). AI for conceptual architecture: Reflections on designing with text-to-text, text-to-imagen, and image-to-image generators. Frontiers of Architectural Research, 13(3), 593-612.  
<https://doi.org/10.1016/j.foar.2024.02.006>
- [29] Chen, J., Wang, D., Shao, Z., Zhang, X., Ruan, M., Li, H., Li, J. (2023). Usign Artificial intelligence to generate Master-Quality Architectural

Designs from text  
Descriptions. buildings, 13(9),  
2285.  
<https://doi.org/10.3390/buildings13092285>

- [30] Rodriguez, R. C., Duarte, R. B. (2022). Generating floor plans with deep learning: A cross-validation assessment over different dataset sizes. *International Journal of Architectural Computing*, 20(3), 630-644. <https://doi.org/10.1177/14780771221120842>
- [31] Nadari, M., Karami, N., Emami, A., Shirani, S., Samavi, S. (2022). Dynamic-Pix2Pix: Noise Injected cGan for Modeling Input and Target Domain Joint Distributions with Limited Training Data. <https://doi.org/10.48550/arXiv.2211.08570>