

TRABAJO FINAL DE MÁSTER

"AUTOMATIZACIÓN DE PROPUESTAS DE REFORMA ARQUITECTÓNICA MEDIANTE DEEP LEARNING"

Máster en Inteligencia Artificial Avanzada y Aplicada

Autor: Víctor Manuel Sánchez Valero

Tutor: Valero Laparra Pérez-Muelas

Valencia, septiembre 2025

Índice

1. Introducción	3
1.1 Contexto y motivación	3
1.2 Problema detectado	4
1.3 Objetivos del trabajo	5
1.4 Resultados esperados y contribuciones	6
2. Estado del arte	8
2.1 Digitalización en la arquitectura y necesidad de automatización	8
2.2 Segmentación semántica en visión por computador	8
2.3 Modelos multimodales visión–texto	10
2.4 Datasets de planos arquitectónicos	11
2.5 Métodos existentes de generación automática de planos arquitectónicos	13
2.6 Integración de la normativa en la IA arquitectónica	14
2.7 Aplicaciones comerciales y posicionamiento del trabajo	15
2.8 Síntesis y diferenciación del proyecto	16
3. Metodología	17
3.1 Fase 1 – Segmentación de planos arquitectónicos	17
3.2 Fase 2 – Imagen a texto (Mask2Text)	22
3.3 Fase 3 – Motor de decisión	25
3.4 Fase 4 – Visualización final de la propuesta	27
4. Resultados	28
4.1 Resultados cuantitativos	28
4.2 Resultados cualitativos	49
4.3 Explicabilidad del modelo	54
5. Discusión	59
5.1 Fortalezas del sistema propuesto	59
5.2 Limitaciones del trabajo	61
5.3 Comparación con procesos manuales	62
5.4 Líneas de investigación futura	62
5.5 Contraste con Modelos de Lenguaje Generalistas (LLMs)	63
6. Conclusiones	68
7. Líneas futuras	71
8. Bibliografía	73

1. Introducción

1.1 Contexto y motivación

La disciplina arquitectónica, un campo que tradicionalmente ha entrelazado la creatividad artística con la precisión técnica, se encuentra en plena era de **transformación digital**. En las últimas dos décadas, hemos sido testigos de la adopción masiva de herramientas avanzadas que han optimizado los procesos de diseño y construcción, desde el **modelado de información de construcción (BIM)** y el diseño paramétrico, hasta la simulación de estructuras y el renderizado fotorrealista. Sin embargo, hay un sector específico de la práctica profesional donde gran parte del trabajo todavía sigue dependiendo de tareas manuales e intensivas en tiempo y que, a menudo no se realiza por técnicos: **los proyectos de reforma y rehabilitación**. La interpretación de planos existentes, la elaboración de múltiples propuestas iniciales y la validación de la viabilidad estructural o funcional son procesos que consumen una cantidad desproporcionada de recursos y dependen, casi exclusivamente, de la experiencia individual del arquitecto. Esta dependencia limita la capacidad de los estudios de arquitectura para explorar un número amplio de alternativas de diseño de manera ágil, lo cual afecta directamente a la satisfacción del cliente y a la eficiencia del negocio.

En este contexto, la **inteligencia artificial (IA)** emerge como una tecnología con un potencial disruptivo significativo. Modelos de **visión por computador** permiten a los sistemas informáticos "ver" e "interpretar" planos arquitectónicos, extrayendo información estructural como la ubicación de paredes, puertas, ventanas y estancias. Paralelamente, los avances en el **procesamiento del lenguaje natural (PLN)** y los **modelos generativos** posibilitan la transformación de dicha información visual en descripciones textuales coherentes o incluso en propuestas gráficas automatizadas. La convergencia de estas tecnologías abre la puerta a la creación de sistemas de asistencia que pueden procesar grandes volúmenes de datos, acelerar las fases preliminares de una reforma y, en última instancia, empoderar al arquitecto para que se enfoque en las tareas de mayor valor creativo.

El porqué del tema elegido para este Trabajo de Fin de Máster se entiende desde mi perfil profesional: soy arquitecto y ejerzo desde hace 15 años. Durante este tiempo he podido constatar que una parte significativa de la práctica arquitectónica, concretamente en el ámbito de **la reforma, permanece anclada en métodos tradicionales, con una escasa integración tecnológica**. En numerosos casos, son las empresas de reforma, constructoras o incluso los propios promotores quienes asumen este proceso de manera manual, sin apoyo en

herramientas digitales avanzadas y con una calidad técnica desigual. Esta observación, unida a la motivación de aplicar la inteligencia artificial en un campo donde existe un gran margen de mejora, ha sido uno de los principales motores para iniciar este proyecto en el marco del Máster en Inteligencia Artificial Avanzada.

La reforma de una vivienda constituye una experiencia común para gran parte de la población, y sin embargo pocas veces es un proceso dirigido íntegramente por técnicos cualificados. Incluso en aquellos casos donde se elaboran planos arquitectónicos, estos suelen carecer de un análisis automatizado que permita explorar múltiples alternativas de manera ágil y reproducible. Esta realidad evidencia la necesidad de soluciones que integren visión por computador, PLN y modelos generativos en un flujo de trabajo coherente, capaz de asistir en las fases preliminares de una reforma y de aportar un valor añadido tanto al profesional como al cliente final.

El presente Trabajo de Fin de Máster se sitúa en esta intersección fundamental entre arquitectura e inteligencia artificial. Su objetivo es explorar la posibilidad de construir un flujo de trabajo **multimodal** y automatizado que, partiendo de un plano de vivienda en formato digital (PDF), sea capaz de producir de forma autónoma una propuesta de reforma coherente y viable. Este proyecto busca ser un catalizador que acelere la transición de la práctica arquitectónica hacia una nueva era de diseño asistido por sistemas inteligentes.

1.2 Problema detectado

A pesar de la omnipresencia de herramientas digitales, la tarea de generar una propuesta de reforma sigue siendo un proceso manual y desarticulado. Un arquitecto debe interpretar la geometría de un plano (que para un ordenador es solo una colección de líneas), identificar las estancias y elementos constructivos (paredes, pilares, puertas, etc.), y luego, a partir de esta interpretación, elaborar mentalmente una propuesta de modificación que responda a las necesidades del cliente. Por ejemplo, una solicitud tan simple como “abrir la cocina al salón” requiere un proceso manual y detallado de análisis para identificar las paredes que deben ser demolidas, considerar la viabilidad estructural y redibujar el nuevo espacio. Este proceso no solo es intensivo en tiempo, sino que también depende de la experiencia individual del profesional, lo que dificulta la estandarización y la escalabilidad.

La dificultad de simular reformas de manera automatizada limita de forma crítica la exploración de alternativas de diseño. A día de hoy, la mayor parte de las herramientas de software arquitectónico (CAD, BIM) se centran en el **modelado preciso** y no en la **generación o interpretación automática** de propuestas. Para estas herramientas, un plano es un conjunto de

datos geométricos, no una representación semántica de una vivienda. Esta es la principal brecha que busca abordar el presente trabajo: la **ausencia de un puente semántico** entre los planos digitales y la posibilidad de procesarlos, analizarlos y transformarlos rápidamente en propuestas de reforma asistidas por IA.

1.3 Objetivos del trabajo

El objetivo general de este Trabajo de Fin de Máster es el desarrollo de un sistema modular basado en inteligencia artificial capaz de generar propuestas de reforma arquitectónica a partir de planos en formato PDF.

Para lograr este objetivo, se ha desglosado el trabajo en los siguientes objetivos específicos, cada uno representando una fase crítica del flujo de trabajo propuesto:

- **Segmentación automática de planos arquitectónicos:** Este es el paso fundacional y el punto de partida del *pipeline*. El objetivo es entrenar modelos de visión profunda (Deep Learning) para que, a partir de una imagen del plano, sean capaces de obtener **máscaras semánticas** de sus elementos principales, como estancias (cocina, baño, salón), estructura (paredes de carga, pilares), puertas y ventanas. Sin una segmentación precisa, el sistema no puede "entender" el plano, lo que haría inviable cualquier paso posterior.
- **Generación de descripciones arquitectónicas:** El segundo objetivo es transformar la información visual de las máscaras segmentadas en una representación simbólica que pueda ser procesada por un sistema lógico. Se busca entrenar un modelo que genere descripciones textuales estructuradas, reflejando de manera precisa las estancias presentes y sus características principales. Este texto actúa como un puente vital entre el mundo visual del plano y el mundo lógico del motor de decisión.
- **Motor de decisión de reformas:** Este módulo constituye el "cerebro" del sistema. Su función es interpretar las intenciones de reforma detectadas en fases previas y transformarlas en acciones estructuradas en formato JSON. Para ello aplica un conjunto de **reglas deterministas** que verifican la contigüidad de las estancias, la factibilidad geométrica de la intervención y, de manera crucial, el **cumplimiento de la normativa vigente**. En este punto se distinguen dos niveles de comprobación: reglas duras, derivadas de normativas como la *DC-09 de Valencia*, que imponen requisitos mínimos obligatorios (por ejemplo, superficies mínimas por estancia o espacio integrado), y reglas blandas, basadas en recomendaciones del *CTE*, que se registran como avisos o

sugerencias (como ventilación, accesibilidad o condiciones de uso). De esta forma, el motor de decisión no solo identifica posibles reformas, sino que también asegura que se planteen dentro de un marco reglamentario sólido antes de continuar a fases posteriores.

- **Visualización final de la propuesta:** El objetivo final es comunicar el resultado de una forma clara y visualmente intuitiva. Se busca superponer gráficamente la propuesta de reforma sobre el plano original, utilizando técnicas de renderizado simples para visualizar los cambios. Asimismo, se generará un texto narrativo explicativo de la reforma, que, además, incluye recomendaciones basadas en la normativa del CTE (Código Técnico de la Edificación). De esta manera, la propuesta técnica se convierte en una herramienta comunicativa útil para el usuario.

En la siguiente *Figura 1.1* se muestra un esquema gráfico del flujo de trabajo, desde el plano de entrada hasta el plano de salida, a través de las 4 fases del sistema:

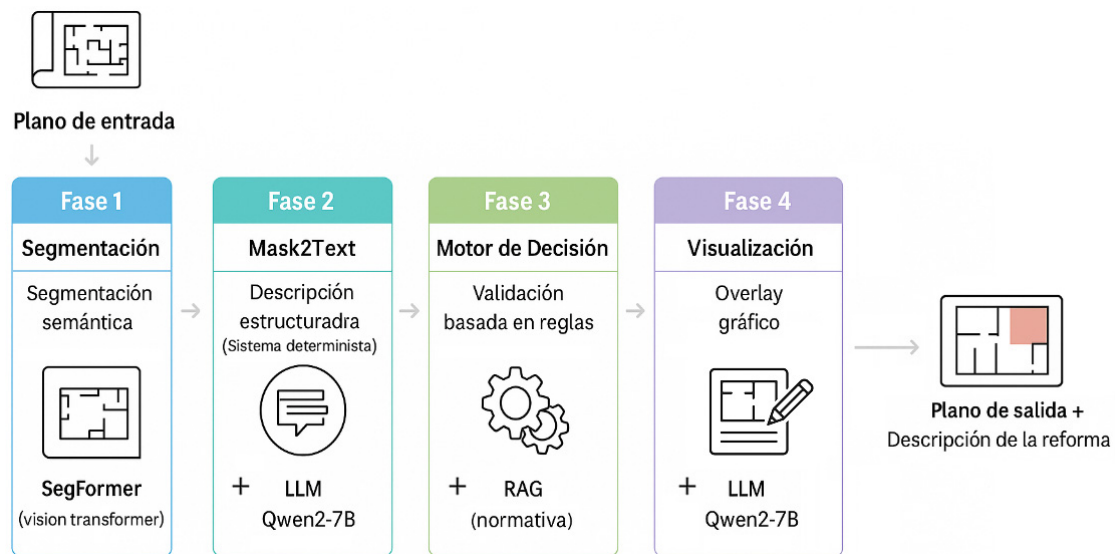


Figura 1.1 Esquema de flujo de trabajo

1.4 Resultados esperados y contribuciones

Se espera que este trabajo demuestre la viabilidad de un flujo de trabajo multimodal que combina de manera sinérgica visión por computador, PLN y generación gráfica para automatizar una parte significativa del proceso de reforma arquitectónica.

Las principales contribuciones previstas son:

- **Un *pipeline* completo e integrado:** Este trabajo no se centra en una única tecnología, sino en la integración de varias para crear un sistema funcional y escalable, desde la entrada en formato PDF hasta la salida de una propuesta reformada en imagen y texto.
- **Una metodología modular por fases:** La división del sistema en etapas discretas (segmentación, descripción, decisión y visualización) es una contribución metodológica clave. Este enfoque permite que cada fase pueda ser mejorada, escalada o incluso sustituida por modelos más avanzados en el futuro sin comprometer la integridad de todo el sistema.
- **Validación experimental en *datasets* de planos:** Se utilizarán conjuntos de datos públicos como MSD, así como ejemplos propios. La validación incluirá métricas cuantitativas (como la **IoU** para la segmentación) y un análisis cualitativo (evaluando la claridad de las descripciones generadas y la utilidad de las propuestas).
- **Aplicabilidad práctica:** Se explorará el potencial de esta tecnología en la práctica real. Por ejemplo, en **estudios de arquitectura**, podría ser una herramienta de pre-análisis rápido que reduzca el tiempo de las primeras reuniones con clientes. En **plataformas inmobiliarias**, podría ofrecer a los usuarios la posibilidad de visualizar rápidamente cómo una propiedad podría transformarse.

En conjunto, este trabajo busca sentar las bases de un sistema capaz de asistir al arquitecto en la fase inicial de diseño de reformas, integrando tecnologías de IA en un flujo que combina la precisión técnica con una notable capacidad generativa.

2. Estado del arte

2.1 Digitalización en la arquitectura y necesidad de automatización

La digitalización del sector de la construcción y la arquitectura no es un fenómeno reciente, sino un proceso evolutivo que ha transformado los flujos de trabajo en las últimas décadas. Desde la adopción inicial del **diseño asistido por ordenador (CAD)** en los años 80, que permitió la transición del tablero de dibujo a la pantalla, hasta la integración del **Building Information Modeling (BIM)** a principios del siglo XXI. El BIM representó un salto cualitativo al centralizar la información de un proyecto en un modelo 3D inteligente, facilitando la colaboración entre arquitectos, ingenieros y constructores. No obstante, a pesar de estos avances, existe una brecha persistente: la interpretación y la manipulación de planos en formatos no estructurados, como PDFs escaneados o dibujos CAD antiguos, sigue siendo una tarea eminentemente manual.

Estudios recientes indican que más del **70% de las reformas en vivienda en España** se tramitan todavía a partir de documentos no estandarizados. Este problema se agrava en el ámbito de la rehabilitación y la reforma, donde los planos existentes suelen ser una mezcla de archivos digitales antiguos, copias físicas escaneadas y notas manuales. La falta de estandarización exige una inversión de tiempo significativa por parte del arquitecto o del técnico. Se ven obligados a reconocer manualmente cada estancia, identificar los elementos constructivos (muros de carga, tabiques), medir superficies y, a partir de este análisis, elaborar una o varias propuestas de redistribución sin apenas apoyo automático. Este proceso no solo consume recursos y aumenta el coste del servicio, sino que también limita la exploración de alternativas de diseño, ya que cada propuesta adicional representa un esfuerzo considerable.

La introducción de técnicas avanzadas de **visión por computador y procesamiento del lenguaje natural (PLN)** abre una nueva posibilidad: la de crear herramientas inteligentes que no solo asistan, sino que automatizan las fases iniciales de análisis y diseño. La combinación de estas tecnologías permite el desarrollo de sistemas capaces de interpretar la información gráfica y transformarla en conocimiento estructurado, lo que podría reducir drásticamente los tiempos de análisis y agilizar la generación de propuestas preliminares, permitiendo al arquitecto enfocarse en tareas de mayor valor creativo.

2.2 Segmentación semántica en visión por computador

La **segmentación semántica**, que consiste en asignar una etiqueta de clase a cada píxel de una imagen, es el pilar fundamental del presente trabajo. Su aplicación en el ámbito arquitectónico

permite que un sistema automatizado “entienda” un plano, identificando y separando las estancias (salón, cocina, baño) y los elementos constructivos (paredes, puertas, ventanas) con precisión. La evolución de las arquitecturas de redes neuronales profundas ha sido clave en este campo.

- **Modelos clásicos basados en convoluciones:**

- **U-Net (Ronneberger et al., 2015):** Originalmente diseñada para segmentación de imágenes biomédicas, su arquitectura de tipo **encoder-decoder** con **skip connections** la hizo muy popular. El *encoder* comprime la información espacial, mientras que el *decoder* la reconstruye. Las *skip connections* transfieren información de alta resolución desde las capas del *encoder* a las del *decoder*, lo que es crucial para mantener los detalles finos de los planos (líneas, esquinas) que de otro modo se perderían.
- **DeepLabV3+ (Chen et al., 2018):** Este modelo mejoró las técnicas de segmentación al introducir las **convoluciones dilatadas** y el **Atrous Spatial Pyramid Pooling (ASPP)**. Las convoluciones dilatadas permiten expandir el campo de visión del modelo sin aumentar los parámetros, mientras que el módulo ASPP captura información a múltiples escalas. Esta arquitectura es especialmente robusta para reconocer objetos de diferentes tamaños, lo cual es útil para identificar tanto estancias grandes como elementos pequeños como puertas.

- **Modelos recientes basados en Transformers:**

- **SegFormer (Xie et al., 2021):** Representa un cambio de paradigma al reemplazar las convoluciones tradicionales por la arquitectura **Vision Transformer (ViT)**. Elimina la necesidad de *positional embeddings* complejos y utiliza un *Transformer* para procesar la imagen de manera eficiente. Su enfoque en la **atención** permite capturar dependencias a largo plazo entre píxeles, mejorando la coherencia de la segmentación en planos complejos con distribuciones variadas.
- **Mask2Former (Cheng et al., 2022):** Este modelo unifica las tres tareas de segmentación (semántica, por instancia y panóptica) bajo un único *framework* con una arquitectura de tipo *query-based*. En lugar de predecir la clase de cada píxel, utiliza **queries** que "buscan" las regiones de interés en la imagen. Esta versatilidad lo hace ideal para dominios heterogéneos como los planos, donde se requiere identificar tanto categorías generales (estancias) como instancias específicas (puertas individuales).

En el ámbito de la segmentación de planos arquitectónicos, la investigación más reciente ha demostrado que los modelos basados en Transformers, como **SegFormer**, ofrecen un equilibrio óptimo entre **precisión y eficiencia**. Su capacidad para capturar relaciones globales en el plano, combinada con su velocidad de inferencia, los convierte en candidatos ideales para *pipelines* que requieren resultados rápidos y de alta calidad.

2.3 Modelos multimodales visión–texto

Más allá de la segmentación, una línea de investigación crucial es la que aborda la **vinculación entre representaciones visuales y textuales**. El objetivo es crear un "puente" que permita a un sistema comprender un plano a nivel gráfico y, al mismo tiempo, generar descripciones textuales o responder preguntas sobre él.

- **Modelos de encodificación conjunta:**
 - **CLIP (Radford et al., 2021):** Este modelo de OpenAI fue pionero en el aprendizaje **contrastivo** a gran escala, entrenando un *encoder* de texto y uno de imagen para que los *embeddings* de imágenes y sus descripciones correspondientes se ubiquen en una posición cercana en un espacio latente común. Esto permite que el modelo asocie conceptos visuales con etiquetas de texto sin necesidad de un entrenamiento específico para cada tarea.
- **Modelos de generación de texto a partir de imagen:**
 - **BLIP (Li et al., 2022) y BLIP-2 (2023):** Estos modelos avanzaron la tarea de *image-to-text* integrando *encoders* visuales con grandes modelos de lenguaje (LLMs). BLIP-2, en particular, introdujo el **Querying Transformer (Q-Former)**, un módulo que extrae las características más relevantes de la imagen para ser procesadas por el LLM, mejorando la calidad de las descripciones generadas.
 - **Flamingo (DeepMind, 2022):** Este modelo se especializa en el aprendizaje multimodal con pocos ejemplos (*few-shot learning*). Su arquitectura, que entrelaza capas de atención visual y textual, le permite generar descripciones coherentes y dialogar sobre imágenes, adaptándose rápidamente a nuevos dominios con un mínimo de datos.
 - **Kosmos-1 (Microsoft, 2023):** Este modelo multimodal generalista procesa texto e imagen de manera conjunta, con un enfoque en la comprensión de documentos y la resolución de problemas lógicos.

Si bien estos modelos han demostrado capacidades impresionantes, presentan limitaciones cuando se aplican directamente a planos arquitectónicos. La mayoría han sido entrenados con datos de internet, por lo que carecen de un **vocabulario arquitectónico específico** (por ejemplo, podrían no distinguir un muro de carga de un tabique). Además, sus descripciones tienden a ser de **texto libre**, lo que dificulta su integración en un flujo de trabajo estructurado que requiere datos procesables como dimensiones o ubicaciones.

2.4 Datasets de planos arquitectónicos

El éxito de los modelos de *deep learning* depende de la disponibilidad de grandes y diversos *datasets* anotados. En el campo de los planos arquitectónicos, la creación de estos conjuntos de datos es un desafío debido a la complejidad de las anotaciones y a los derechos de autor. En la siguiente *Figura 2.1* se describen los más relevantes en la investigación académica.

Dataset	Nº planos	Tipo	Clases (ej.)	Ventajas	Limitaciones
RPLAN (Wu et al., 2019)	>80.000	Sintético	Estancias, puertas, ventanas	Gran escala, variedad de <i>layouts</i>	No realistas, carecen de elementos estructurales.
CubiCasa5K (Kalervo et al., 2019)	~5.000	Reales	Estancias, mobiliario	Fidelidad a la realidad, anotaciones vectoriales.	Tamaño limitado, no incluye elementos estructurales.
Modified Swiss Dwellings (MSD)	~19.000	Reales	Estancias, paredes, puertas, ventanas	Anotaciones detalladas y precisas.	Tamaño limitado, distribución geográfica restringida.
Others (DeepFloorPlan, HouseGAN++, etc.)	Variable	Sintético/Real	Variable	Complementan a los grandes datasets.	A menudo son pequeños y específicos.

Figura 2.1 Tabla comparativa datasets arquitectónicos

La elección del dataset constituye un aspecto fundamental en el desarrollo de este TFM, ya que determina tanto la calidad de los resultados obtenidos como la viabilidad del flujo de trabajo planteado. Inicialmente, se consideró el uso del dataset **RPLAN**, ampliamente citado en la literatura y con un volumen superior a 80.000 planos residenciales. Sin embargo, este recurso presenta limitaciones significativas para su aplicación práctica: su acceso es restringido, sujeto a autorización por parte de los autores, y los planos que lo conforman corresponden mayoritariamente a viviendas unifamiliares de estructuras ortogonales y relativamente simples. Además, se trata de un dataset **sintético**, lo que reduce su representatividad frente a situaciones arquitectónicas reales.

En contraste, el dataset **MSD (Modified Swiss Dwellings)** ofrece una serie de características que lo convierten en una alternativa más adecuada para el objetivo de este trabajo:

- **Accesibilidad:** se trata de un dataset abierto y de libre acceso, lo que permite su uso inmediato en contextos académicos sin depender de autorizaciones externas.
- **Escala y riqueza estructural:** incluye 5.372 planos de edificios residenciales, que suman 18.914 apartamentos individuales y 246.496 habitaciones segmentadas. Esta organización permite trabajar a diferentes niveles (edificio completo, apartamento o estancia), ofreciendo flexibilidad metodológica.
- **Diversidad y realismo arquitectónico:** a diferencia de RPLAN, los planos de MSD no se limitan a configuraciones ortogonales. Incorporan distribuciones irregulares, múltiples viviendas por planta, muros de carga, escaleras y elementos comunes propios de tipologías reales. Esto lo convierte en un banco de pruebas más exigente y representativo de la práctica profesional.
- **Datos procedentes de viviendas reales:** los planos de MSD provienen de **edificios residenciales existentes**, lo que aporta un valor añadido de **realismo** y permite que los experimentos del TFM estén más alineados con situaciones que un arquitecto afronta en proyectos de reforma. Esta diferencia es clave frente a RPLAN, que se basa en datos sintéticos generados artificialmente.
- **Multimodalidad de formatos:** además de imágenes rasterizadas, el dataset incluye representaciones vectoriales y grafos de relaciones espaciales, lo que facilita la integración de técnicas híbridas de visión por computador y procesamiento de grafos.
- **Información enriquecida por estancia:** cada habitación no solo está clasificada por tipología (cocina, dormitorio, baño, salón, etc.), sino que también incluye su **superficie**

en metros cuadrados. Este detalle permite generar descripciones arquitectónicas más completas y precisas, vinculando la segmentación espacial con datos cuantitativos que resultan esenciales en el análisis y en la Fase 2 del TFM (imagen a texto).

- **Orientación al objetivo del TFM:** el carácter complejo, detallado y realista de MSD aporta un mayor valor académico al proyecto, al aproximarse mejor a las condiciones que un arquitecto encuentra en proyectos de reforma reales.

En consecuencia, se ha decidido emplear **MSD como dataset principal** para este Trabajo de Fin de Máster en la fase de segmentación de planos, asegurando un punto de partida sólido y realista sobre el que desarrollar las fases posteriores de generación de propuestas de reforma y representación visual de las mismas.

2.5 Métodos existentes de generación automática de planos arquitectónicos

Una línea de investigación relacionada es la generación automática de *layouts* de vivienda, que busca producir distribuciones espaciales coherentes a partir de cero.

- **Merrell et al. (2011):** Este trabajo pionero propuso un algoritmo que, a partir de un conjunto de restricciones lógicas y funcionales (número de habitaciones, relaciones espaciales), generaba una distribución de vivienda viable. Su enfoque era simbólico y no visual, centrado en la coherencia de la planta.
- **HouseGAN (Nauata et al., 2020):** Utiliza redes generativas antagónicas (GANs) para generar planos completos a partir de un conjunto de especificaciones. Las GANs permiten que el modelo “aprenda” la distribución de elementos de un plano a partir de un gran dataset y genere nuevas distribuciones visualmente realistas.
- **ArchGAN (Chaillou, 2020):** Modelo generativo capaz de producir distribuciones residenciales adaptadas a parámetros específicos, demostrando la capacidad de la IA para explorar el espacio del diseño de una manera creativa.
- **Wu et al. (2019):** Presenta un método *data-driven* para la generación automática de planos residenciales a partir del contorno del edificio. Su flujo consiste en situar primero las estancias principales (como el salón), añadir el resto de habitaciones y posteriormente definir las paredes mediante un modelo encoder-decoder, con refinamiento vectorial final.

- **Graph2Plan (Hu et al., 2020):** Propone un enfoque híbrido que combina aprendizaje profundo e interacción con el usuario. A partir de un contorno y un *layout graph* que define las relaciones entre estancias, el sistema genera un plano rasterizado, lo refina mediante cajas delimitadoras y produce finalmente una versión vectorial. Destaca por permitir la edición del grafo, lo que ofrece al usuario control sobre la propuesta generada.
- **DStruct2Design (2024):** Introduce un marco de trabajo basado en **estructuras de datos** para guiar la generación de planos. En lugar de trabajar solo con imágenes, emplea representaciones intermedias (JSON con dimensiones y polígonos) que permiten a modelos de lenguaje como Llama3 producir distribuciones ajustadas a restricciones numéricas o espaciales.

Estas aproximaciones son muy relevantes en el campo del diseño generativo, ya que demuestran la capacidad de los modelos para crear nuevos diseños residenciales desde cero. Sin embargo, su principal limitación es que se centran en la **generación de layouts originales** y no en la **modificación de planos existentes**. Además, algunos de ellos priorizan la creatividad (ArchGAN, HouseGAN), otros el control estructural (Graph2Plan, DStruct2Design) o la coherencia funcional (Wu et al., Merrell), pero ninguno aborda el problema de la reforma arquitectónica como tal.

El presente trabajo, en cambio, se orienta a la **interpretación y reforma de un plano dado**, lo que requiere un enfoque distinto que combine:

- la **segmentación semántica** de planos existentes,
- la **traducción multimodal a texto** para describirlos,
- la **aplicación de reglas y normativa** para validar propuestas,
- y la **generación de visualizaciones de reforma**.

De este modo, se diferencia de los trabajos mencionados al situarse no en el diseño generativo puro, sino en la automatización de procesos prácticos de la reforma arquitectónica.

2.6 Integración de la normativa en la IA arquitectónica

La viabilidad de una propuesta de reforma no solo depende de su coherencia estética o funcional, sino también de su cumplimiento con la **normativa arquitectónica** vigente (como el Código Técnico de la Edificación - CTE - en España, o normativas más autonómicas como la DC-

09 de Valencia). La integración de este conocimiento en los sistemas de IA es un desafío crítico, ya que las normativas son documentos extensos, complejos y a menudo con excepciones.

Algunos proyectos incipientes han explorado la codificación de estas reglas en sistemas expertos o mediante lenguajes de programación basados en restricciones. No obstante, la conexión entre la normativa escrita en lenguaje natural y los modelos de IA sigue siendo un área en desarrollo. Una de las líneas más prometedoras es la aplicación de la **Retrieval-Augmented Generation (RAG)**, una técnica que permite a los LLMs consultar una base de datos externa de documentos (como la normativa) y generar respuestas verificadas. Aunque no existen aún implementaciones sólidas en el dominio de las reformas residenciales, este enfoque sienta las bases para futuras investigaciones que podrían validar automáticamente las propuestas del sistema.

2.7 Aplicaciones comerciales y posicionamiento del trabajo

Actualmente, el mercado de software para arquitectos ofrece herramientas con funcionalidades abordan aspectos relacionados con este proyecto, aunque con objetivos distintos:

- **MagicPlan:** Permite crear planos digitalizados a partir de fotografías tomadas con un dispositivo móvil y generar estimaciones de superficie o presupuestos. Su aportación es agilizar la digitalización, pero no incluye lógica de propuesta ni análisis de reforma.
- **Planner 5D:** Plataforma orientada al diseño de interiores y a la visualización 3D. Facilita la distribución de mobiliario y la producción de renders de alta calidad. Sin embargo, su foco es la personalización estética por parte del usuario, careciendo de un motor de decisión autónomo que plantee reformas arquitectónicas.
- **CubiCasa:** Centrada en la conversión de planos a modelos BIM de forma semiautomática, optimiza la integración de planos escaneados en flujos de trabajo profesionales. Aunque destaca por la fidelidad geométrica, no incorpora generación de propuestas creativas ni evaluaciones funcionales.

En todos los casos, se trata de herramientas **de código cerrado**, con algoritmos propietarios y limitaciones en cuanto a transparencia y capacidad de personalización. Además, su orientación se centra en la representación gráfica o en la digitalización geométrica, sin abordar la **interpretación semántica de un plano existente ni la generación automática de propuestas de reforma**.

En este sentido, el presente TFM se posiciona como una propuesta diferenciadora: un sistema de **IA multimodal** que combina segmentación semántica, generación textual y validación

normativa para automatizar la elaboración de reformas, aportando un nivel de inteligencia y aplicabilidad práctica que actualmente no está cubierto por las soluciones comerciales disponibles.

2.8 Síntesis y diferenciación del proyecto

La revisión del estado del arte pone de manifiesto la existencia de dos grandes corrientes de investigación: por un lado, la **interpretación visual de planos** mediante técnicas de visión por computador, y por otro, la **generación automática de diseños** a partir de modelos generativos. El presente trabajo se sitúa en un punto intermedio, aprovechando los avances de ambas líneas y articulándolos en un **pipeline modular y especializado**.

La principal diferenciación frente a las herramientas comerciales y frente a los modelos de lenguaje de propósito general (LLMs), que han emergido como tecnologías prometedoras, pero aún insuficientes para este campo, se concreta en tres aspectos:

- **Interpretación visual limitada:** Los LLMs multimodales, a pesar de aceptar imágenes, muestran dificultades para interpretar con precisión la geometría y simbología de un plano arquitectónico. No son capaces de segmentar a nivel de píxel ni de distinguir elementos críticos en la práctica profesional, como un muro de carga frente a un tabique ligero.
- **Falta de integración normativa y lógica:** Los modelos generalistas carecen de conocimiento arquitectónico formal y, al generar texto libre, pueden proponer reformas inviables o contrarias a la normativa.
- **Ausencia de salida estructurada:** El pipeline propuesto genera un resultado en formato **JSON**, estructurado y verificable, lo que habilita fases posteriores de validación y visualización. En contraste, la salida de los LLMs es texto libre, poco reproducible y difícil de integrar de forma automática en procesos posteriores.

En conclusión, mientras que los modelos generativos como HouseGAN se centran en la creación de planos nuevos y los LLMs generalistas se limitan a la redacción creativa de propuestas, el presente trabajo desarrolla un **sistema híbrido especializado** que integra la precisión técnica de la visión por computador con la capacidad generativa de los modelos de lenguaje. Este enfoque no solo demuestra la **viabilidad de la automatización del proceso de reforma arquitectónica**, sino que también establece un **marco de referencia para futuros sistemas de asistencia inteligente en arquitectura**, con un claro impacto tanto académico como profesional.

3. Metodología

El desarrollo del sistema para la automatización de propuestas de reforma arquitectónica se ha estructurado en un **pipeline secuencial y modular de cuatro fases**. Este enfoque metodológico permite no solo una clara división del trabajo, sino también la posibilidad de escalar o sustituir los modelos de cada fase de manera independiente, facilitando la investigación futura.

El entrenamiento de los modelos se llevó a cabo en la plataforma **Kaggle**, que permitió disponer de GPU durante sesiones prolongadas y con la estabilidad necesaria. Esta elección se justificó por la elevada demanda computacional de los modelos empleados y por el volumen de datos del dataset MSD, que hacía inviable el entrenamiento en CPU o en entornos con limitaciones de uso de GPU como **Google Colab**.

3.1 Fase 1 – Segmentación de planos arquitectónicos

El primer paso y la base de todo el sistema es la **segmentación semántica**. El objetivo principal de esta fase es la interpretación visual del plano de entrada. A partir de una imagen rasterizada (por ejemplo, un PDF escaneado o una imagen PNG), el modelo debe asignar una etiqueta de clase a cada píxel, identificando de manera automática las estancias y elementos constructivos principales, como baños, cocinas, salones, dormitorios, pasillos, puertas, ventanas y la estructura de la vivienda.

La precisión de esta segmentación es crítica, ya que cualquier error en la detección de un elemento o en la delimitación de una estancia condicionará directamente la calidad de las fases posteriores, desde la descripción textual hasta la viabilidad de la propuesta de reforma. Por ejemplo, una incorrecta segmentación de un muro como un tabique en lugar de una pared de carga podría invalidar toda la propuesta.

Dataset y preprocesado

El **dataset Modified Swiss Dwellings (MSD)** se proporciona en formato tabular mediante un archivo CSV, en el cual cada fila describe una estancia de un apartamento, incluyendo:

- El identificador del apartamento.
- El tipo de estancia (por ejemplo, dormitorio, cocina, salón, baño, etc.).
- La geometría de la estancia, codificada en formato WKT (Well-Known Text).
- La superficie de cada estancia en metros cuadrados.

Para poder entrenar modelos de segmentación, fue necesario **transformar esta información en imágenes rasterizadas y máscaras de segmentación**. El proceso implementado en el notebook consistió en los siguientes pasos:

1. Lectura del CSV

Se cargó el archivo `mds_V2_5.372k.csv` con la librería **pandas** para inspeccionar sus columnas, unificar etiquetas de tipos de estancia y preparar los datos para su procesado.

2. Normalización de etiquetas

Algunas clases de estancias se unificaron (por ejemplo, `dining` → `livingroom`, `entrance` → `corridor`) para simplificar la taxonomía final de clases y adaptarla a los objetivos del TFM.

3. Construcción geométrica de estancias

A partir de la columna geométrica en formato WKT, se utilizó la librería **Shapely** para reconstruir cada estancia como un polígono.

- Se aplicaron transformaciones afines y rotaciones necesarias para situar las estancias en el sistema de coordenadas correcto.
- Se agruparon las estancias pertenecientes a un mismo apartamento usando el identificador común.

4. Rasterización en imágenes y máscaras

Para cada apartamento:

- Se creó una **imagen en blanco y negro** donde se dibujaron las geometrías de todas las estancias, representando el plano de entrada.
- Se generó una **máscara de segmentación en color indexado**, donde cada estancia se pintó con el identificador de su clase (cocina, baño, dormitorio, etc.).
- La librería **PIL (Python Imaging Library)** y su módulo `ImageDraw` se utilizaron para rasterizar los polígonos en formato PNG.

5. Exportación a carpetas organizadas

Los resultados se guardaron en dos directorios principales:

- `images/`: planos rasterizados de cada apartamento.

- **masks/:** máscaras segmentadas en formato PNG, con cada píxel asociado a una clase de estancia.

6. Uso de metadatos

La información de superficie en m² y tipología, extraída del CSV, quedó asociada a cada apartamento para ser utilizada posteriormente en la **Fase 2 (imagen a texto)**, enriqueciendo las descripciones generadas automáticamente.

Con este proceso, el dataset quedó transformado en un formato estándar de entrenamiento para segmentación semántica (pares de imagen + máscara), a la vez que se conservaron los metadatos arquitectónicos para fases posteriores del TFM.

El dataset estaba organizado en bloques de viviendas, pero cada apartamento se trató como una unidad de análisis independiente con sus propias anotaciones vectoriales (líneas y polígonos) y raster (imágenes y máscaras).

Las transformaciones realizadas para preparar el dataset fueron meticulosas y cruciales para el rendimiento final:

- **Redefinición de clases:** Se realizó una unificación y limpieza para adaptar el conjunto de clases a las necesidades del proyecto. Se unificaron living room y dining room en una sola clase (living_dining). Se fusionaron entrance door y door en una única clase (door) y se eliminó la clase storeroom por su baja frecuencia. Se mantuvo la clase structure como fundamental para delimitar estancias y validar las reformas. Se descartaron clases minoritarias del dataset no relevantes para el propósito de este trabajo, como balcón o vestidor. Estas decisiones se fueron tomando a medida que se obtenían resultados, con el objetivo de simplificar los datos para que el modelo pudiera generalizar mejor.
- **Conversión de datos vectoriales a máscaras:** Las anotaciones vectoriales originales se convirtieron a imágenes y máscaras en formato PNG. Cada píxel de la máscara se etiquetó con un valor entero que identifica la clase correspondiente, un formato estándar para el entrenamiento de modelos de segmentación.
- **Normalización geométrica y escalado:** Para garantizar la uniformidad en los datos de entrada, se aplicó un *padding* cuadrado alrededor de cada plano para evitar distorsiones en la escala. Posteriormente, todas las imágenes y máscaras se redimensionaron a **512x512 píxeles**. Esta resolución representó un equilibrio óptimo entre la retención de

detalle (necesario para puertas y ventanas) y un coste computacional asumible para el entrenamiento.

- **División del dataset:** Se dividió el dataset en tres conjuntos: un **60 % para entrenamiento**, un **20 % para validación** y un **20 % para test**. Esta división aseguró que el modelo se evaluara sobre datos no vistos, garantizando la variabilidad y la reproducibilidad de los resultados.

El preprocesado aplicado (resize a 512×512 px) fue un compromiso entre detalle y coste computacional. Esta decisión redujo el tiempo de entrenamiento, pero también supuso una pérdida de resolución que posiblemente afectó a elementos finos como muros o carpinterías.

El dataset MSD, pese a su volumen y diversidad, presenta un reto particular en la segmentación de clases de pequeño tamaño, como puertas y ventanas. Estas categorías están ampliamente representadas en número de ejemplos, pero ocupan muy pocos píxeles en relación con el resto del plano, lo que dificulta el aprendizaje de los modelos y repercute en métricas más bajas. Se trata, por tanto, de una limitación ligada a la proporción espacial de estas clases en la imagen más que a su frecuencia en el dataset, y constituye un desafío intrínseco en la segmentación de planos arquitectónicos.

Modelos evaluados

Para esta fase se entrenaron y evaluaron tres arquitecturas de segmentación semántica ampliamente utilizadas en la literatura, cada una con un enfoque distinto para abordar el problema:

- **U-Net (ResNet34 como encoder):**

Se seleccionó como punto de partida debido a su simplicidad y eficacia en conjuntos de datos reducidos.

En esta primera fase, se consideraron 10 clases:

0 - Fondo | 1 - Baño | 2 - Salón | 3 - Dormitorio | 4 - Cocina | 5 - Pasillo | 6 - Trastero |
7 - Puerta interior | 8 - Ventana | 9 - Puerta entrada

- **DeepLabV3+ (ResNet50 como encoder):**

Se entrenó posteriormente con ResNet50 como backbone, aprovechando las convoluciones dilatadas y el contexto multiescala.

- **SegFormer-B1 (Vision Transformer):**

Finalmente, se adoptó SegFormer como modelo de referencia. Este modelo elimina las convoluciones y emplea un **Vision Transformer eficiente** capaz de procesar la imagen completa y capturar dependencias espaciales a largo plazo.

Ajustes en las clases y el dataset

Durante este proceso comparativo también se tomaron **decisiones críticas de modelado de las clases**:

- Analizando los resultados previos obtenidos en el modelo U-Net, se tomó la decisión de **ignorar la clase 6 (trastero) en el modelo DeepLabV3**, debido a su escasa relevancia arquitectónica y el fuerte desbalanceo de muestras que afectaba negativamente a la estabilidad del modelo. Cabe señalar que para esta prueba **no se regeneró el dataset** eliminando la clase 6, de manera que el modelo seguía “viendo” esta categoría, aunque se ignoraba al mostrar los resultados.
- Para SegFormer se **redefinió el dataset desde cero**, aplicando cambios importantes:
 - Unificación de *puerta interior* (7) y *puerta de entrada* (9) en una sola clase (7 Puerta).
 - Reemplazo de la clase *Trastero* (6) por la clase **Estructura**, que incluye fachadas y tabiquerías. Esta decisión fue especialmente relevante al constatar que la segmentación de tabiques divisorios era clave para poder detectar estancias de manera coherente en fases posteriores del pipeline.
- Las clases finales quedaron de la siguiente manera:
 - 0 Fondo
 - 1 Baño
 - 2 Salón
 - 3 Dormitorio
 - 4 Cocina
 - 5 Pasillo
 - 6 Estructura
 - 7 Puerta
 - 8 Ventana

Justificación de la elección de SegFormer

La decisión de adoptar **SegFormer** como modelo base del pipeline responde a una combinación de razones técnicas y empíricas. Desde el punto de vista arquitectónico, SegFormer constituye una propuesta más reciente basada en *Vision Transformers*, eliminando convoluciones y capturando dependencias globales de manera más eficiente. Esto se traduce en un mejor equilibrio entre precisión y coste computacional, con inferencias rápidas y escalables, así como una mayor flexibilidad para futuras ampliaciones (nuevas clases o intenciones de reforma).

En cuanto al rendimiento, los experimentos realizados (véase sección 4) muestran que SegFormer ofrece una mejora clara en la delimitación de estancias principales, alcanzando valores de segmentación notablemente superiores a los de U-Net y DeepLabV3+ en clases de gran superficie. Aunque las clases de menor tamaño, como puertas y ventanas, siguen presentando cierta dificultad, la estabilidad de entrenamiento y la calidad visual de las segmentaciones confirman la robustez del modelo.

Cabe señalar que la comparación con los modelos iniciales no es estrictamente directa, ya que en U-Net y DeepLabV3+ no se incluyó la clase **estructura**, indispensable para representar correctamente divisiones espaciales. Esta limitación fue identificada en los resultados preliminares y motivó un rediseño del dataset, lo que coincidió con la implementación y evaluación de SegFormer.

De este modo, los modelos previos cumplieron una función esencial como línea de base y de diagnóstico de limitaciones, mientras que SegFormer marca un **punto de inflexión** al proporcionar no solo métricas más sólidas en las clases principales, sino también la escalabilidad y consistencia necesarias para consolidar el pipeline completo.

3.2 Fase 2 – Imagen a texto (Mask2Text)

La segunda fase tiene como objetivo transformar las máscaras semánticas generadas en la fase anterior en una **descripción arquitectónica textual**. Esta conversión resulta clave para disponer de un inventario estructurado de estancias y superficies, utilizable en fases posteriores como la propuesta de reformas.

Generación de dataset sintético en JSONL

Dada la ausencia de un corpus real de descripciones arquitectónicas asociadas a planos, se generó automáticamente un **dataset sintético en formato JSONL**. Este archivo funcionó como la pieza central de la fase, ya que cada línea contenía la información estructurada de un plano:

- id: identificador único del plano.
- mask_path: ruta a la máscara segmentada.
- total_m2: superficie total de la vivienda.
- areas: diccionario con cada estancia y su superficie.
- text: descripción normalizada construida automáticamente a partir de los datos.

Para generar estas entradas se aplicaron varias reglas de consistencia:

- Conteo de estancias a partir de regiones cerradas por estructura, tabiques y puertas.
- Cálculo de áreas exactas de cada estancia en m² a partir de las máscaras.
- Frases normalizadas con un vocabulario controlado, evitando variaciones lingüísticas que dificultaran el aprendizaje.

Enfoque inicial: arquitectura encoder–decoder

En un primer momento se desarrolló un modelo Mask2Text basado en un esquema encoder–decoder:

- **Encoder:** SegFormer B0, encargado de extraer embeddings visuales de las máscaras.
- **Decoder:** mT5-small, modelo multilingüe de generación de texto.
- **Conexión:** proyección intermedia para alinear los embeddings del encoder con la entrada del decoder.

El objetivo era entrenar esta arquitectura de extremo a extremo para que generara descripciones en lenguaje natural directamente desde las máscaras.

Problemas encontrados

Durante los entrenamientos iniciales se observaron problemas de colapso en la generación: el modelo tendía a devolver siempre la misma descripción independientemente de la entrada. A pesar de introducir mejoras (dataset más estructurado, enriquecimiento con medidas, entrenamiento de 10 épocas y guardado del mejor modelo en validación), el rendimiento seguía siendo insuficiente y no superaba la calidad del sistema determinista.

Un factor decisivo en este resultado fue la **naturaleza sintética del dataset en JSONL**. No existían descripciones redactadas manualmente por arquitectos para cada plano, sino frases generadas

automáticamente a partir de los datos extraídos de las máscaras (áreas y conteos). Esto provocaba que:

- El estilo lingüístico fuera demasiado rígido y repetitivo, limitando la capacidad del modelo para aprender variaciones naturales.
- La cobertura semántica fuera incompleta, al centrarse solo en número de estancias y superficies, sin referencias contextuales propias de un lenguaje humano.
- El modelo no pudiera generalizar a descripciones más realistas, ya que nunca vio ejemplos auténticos.

Decisión final: sistema determinista + LLM baseline

Ante estos problemas, se optó por no incluir el modelo encoder–decoder en la versión final del sistema. En su lugar, se consolidó un pipeline basado en dos componentes:

1. **Sistema determinista**, que a partir de la máscara segmentada calcula la superficie total y las áreas por estancia, generando una primera descripción exacta.
2. **LLM baseline (Qwen2-7B-Instruct)**, que toma como entrada la información del JSONL (m^2 totales y estancias) y produce una descripción fluida en español, manteniendo fielmente las cifras y sin inventar datos.

Inferencia y exportación

El mismo esquema se emplea en inferencia sobre planos propios. El sistema solicita al usuario la superficie total construida de la vivienda y, a partir de la máscara segmentada, reparte las áreas de cada estancia según la proporción de píxeles cerrados por estructura, puertas y ventanas. Esto permite adaptar automáticamente la escala de cada plano sin necesidad de anotaciones adicionales.

Los resultados se exportan en formato **JSONL**, con un registro por plano que incluye:

- id: identificador del plano.
- mask_path: ruta de la máscara segmentada.
- total_m2: superficie total de la vivienda (proporcionada por el usuario en inferencia).
- areas: diccionario con el nombre de cada estancia y su superficie en m^2 .
- description: texto fluido generado por el modelo LLM a partir de los datos deterministas.

De este modo, tanto el dataset sintético de entrenamiento como los resultados de inferencia siguen una estructura uniforme, lo que facilita su reutilización directa en la **Fase 3 (Texto a texto: propuesta de reformas)**.

3.3 Fase 3 – Motor de decisión

La tercera fase representa el "cerebro" del sistema, un módulo de reglas diseñado para transformar la descripción textual de un plano en un archivo JSON que representa una acción de reforma. Este módulo dota al sistema de inteligencia lógica, ya que no se trata de un simple modelo de *text-to-text*, sino de un sistema experto que aplica **reglas deterministas y comprobaciones normativas**.

Funcionamiento

El funcionamiento del motor de decisión es el siguiente:

- **Input:** recibe la descripción textual generada en la fase 2 junto con las máscaras de segmentación originales y los metadatos del plano (áreas y superficies).
- **Procesamiento:** analiza la descripción textual en busca de intenciones de reforma (en este trabajo se ha implementado la intención *abrir cocina al salón*). Al detectarla, el módulo aplica un conjunto de reglas de validación sobre las máscaras segmentadas. Estas reglas incluyen:
 - Comprobación de contigüidad entre ambas estancias (cocina y salón).
 - Detección del tabique común y validación de que no afecte a ventanas ni a muros estructurales.
 - Verificación de que el tramo de tabique tenga la longitud suficiente para abrir el hueco solicitado.
 - **Integración de normativa:**
 - **Reglas duras:** se comprueba el cumplimiento de la *DC-09 de Valencia*, que establece superficies mínimas para recintos. En concreto, se verifica que la estancia resultante de integrar cocina y salón alcance al menos la superficie mínima exigida para un espacio de “estar-comedor-cocina” (18 m², o el valor extraído automáticamente del PDF oficial de la norma). Si no se cumple, la propuesta se descarta y el JSON refleja la invalidez de la acción.

- **Reglas blandas:** se incorporan recomendaciones del *CTE* que no pueden validarse geométricamente a partir de la máscara, como la necesidad de garantizar ventilación mecánica en la cocina (DB-HS 3) o un paso útil mínimo de 0,80 m en el hueco abierto (DB-SUA). Estas se añaden como avisos en el JSON.
- **Output:** si la acción de reforma es viable, el módulo produce un archivo JSON estructurado que incluye:
 - Un identificador único del plano.
 - Un listado de estancias y sus áreas en m².
 - El área total del plano.
 - La acción de reforma seleccionada y su tipología.
 - Las coordenadas exactas del tabique a demoler.
 - Un campo de validación que indica si la propuesta cumple las reglas duras.
 - Avisos normativos y referencias a los documentos oficiales consultados.

Descripción del funcionamiento

Para la intención *abrir cocina al salón*, el motor de decisión localiza ambas estancias en el plano a partir de las máscaras segmentadas. Si se verifica la contigüidad y el tabique común no coincide con fachada ni con elementos estructurales, se selecciona el tramo más adecuado para la apertura. Posteriormente, se calcula la superficie resultante de la integración de cocina y salón, comparándola con los mínimos de la DC-09.

- Si la propuesta cumple las condiciones geométricas y normativas, se genera un JSON válido con la acción y las referencias normativas aplicadas.
- Si no se cumplen, se genera igualmente un JSON pero marcado como no viable, indicando explícitamente las causas (por ejemplo, “superficie insuficiente según DC-09” o “segmento propuesto toca ventana”).

De este modo, la Fase 3 no solo traduce intenciones en acciones, sino que también actúa como **filtro normativo determinista**, garantizando que las propuestas de reforma respeten unos mínimos objetivos antes de pasar a la representación gráfica y a la descripción generada por la LLM en la Fase 4.

3.4 Fase 4 – Visualización final de la propuesta

La última fase del pipeline se centra en la **presentación comprensible de la reforma validada en fases previas**. El objetivo es comunicar el resultado de manera clara, combinando recursos gráficos y textuales.

Técnicas empleadas

Para la visualización, se utilizó la librería *OpenCV*, aprovechando sus capacidades de procesamiento de imágenes. A partir del archivo JSON generado en la Fase 3, el sistema genera overlays gráficos que se aplican sobre el plano original. Las propuestas de reforma se representan mediante recursos visuales adaptados a cada tipo de intervención: líneas, rectángulos semitransparentes o anotaciones, que permiten localizar con precisión la modificación sin perder la legibilidad del plano base.

De forma complementaria, se integra un **modelo de lenguaje (LLM, Qwen2-7B-Instruct)** que construye una narrativa breve y profesional a partir de los datos del JSON. Este texto cumple una doble función:

- **Explicativa**, ya que describe de forma clara y concisa la intervención propuesta (continuidad espacial, mayor luminosidad, mejora de la circulación, etc.).
- **Normativa suave**, porque introduce recomendaciones relacionadas con el *Código Técnico de la Edificación* (CTE), como la ventilación y extracción en cocinas (DB-HS) o la accesibilidad y el paso útil en los huecos (DB-SUA). Estas sugerencias no son reglas vinculantes dentro del sistema, pero aportan un valor añadido como recordatorio de buenas prácticas y de aspectos a comprobar en la fase de proyecto o ejecución.

Este formato mixto de **imagen + texto + recomendaciones**, facilita la comunicación de la propuesta a clientes o usuarios no especializados, al mismo tiempo que ofrece a los técnicos una referencia normativa que complementa las validaciones duras ya aplicadas en la Fase 3.

4. Resultados

La evaluación del sistema desarrollado se planteó en dos niveles complementarios. En primer lugar, se analizaron las **métricas cuantitativas** para validar objetivamente el rendimiento de los modelos y, adicionalmente, se realizaron pruebas de inferencia sobre planos propios con el fin de valorar su capacidad de generalización en condiciones reales. En segundo lugar, se presentan los resultados cualitativos a través de un **caso completo end-to-end aplicado a un plano real**, que permite demostrar la integración efectiva de todas las fases del sistema y su aplicabilidad práctica en un contexto arquitectónico.

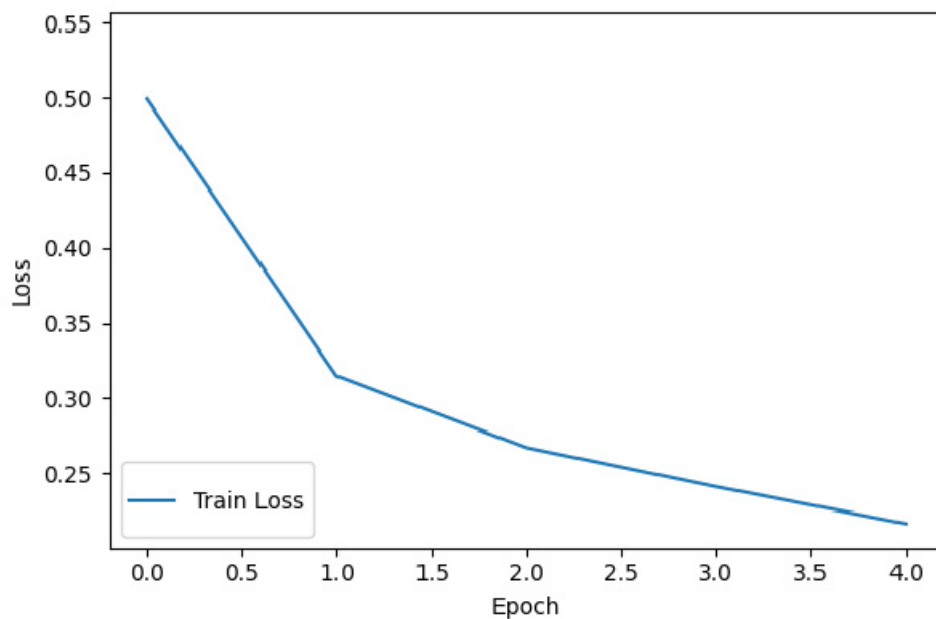
4.1 Resultados cuantitativos

Fase 1 – Segmentación de planos arquitectónicos

El rendimiento de los modelos de segmentación se evaluó utilizando la métrica **Intersection over Union (IoU)**, que mide la superposición entre la máscara predicha y la máscara real (*ground truth*), y la **función de pérdida (loss)**, que indica la precisión del modelo durante el entrenamiento y la validación.

Resultados parciales

U-NET



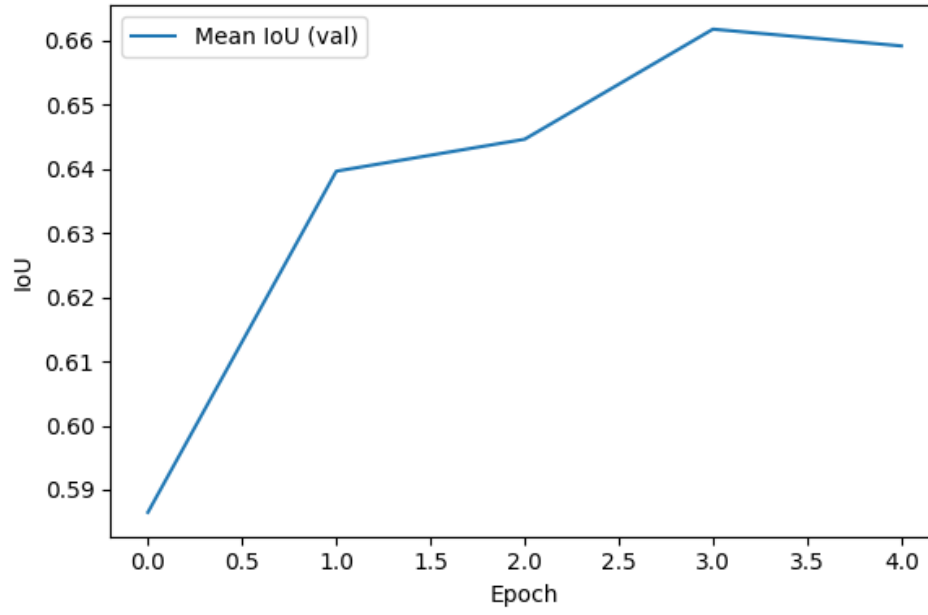


Figura 4.1 Curvas de entrenamiento (loss y mIoU) para U-Net

	Clase	IoU (val)		Clase	IoU (test)
0	background	0.973803	0	background	0.973119
1	bathroom	0.778141	1	bathroom	0.772810
2	livingroom	0.562418	2	livingroom	0.560315
3	bedroom	0.739742	3	bedroom	0.736305
4	kitchen	0.491314	4	kitchen	0.491693
5	corridor	0.863924	5	corridor	0.856946
6	storeroom	0.107880	6	storeroom	0.119364
7	door	0.566924	7	door	0.570431
8	window	0.758269	8	window	0.757100
9	entrance door	0.749443	9	entrance door	0.753662

Figura 4.2 Tablas IoU por clase en epoch 5

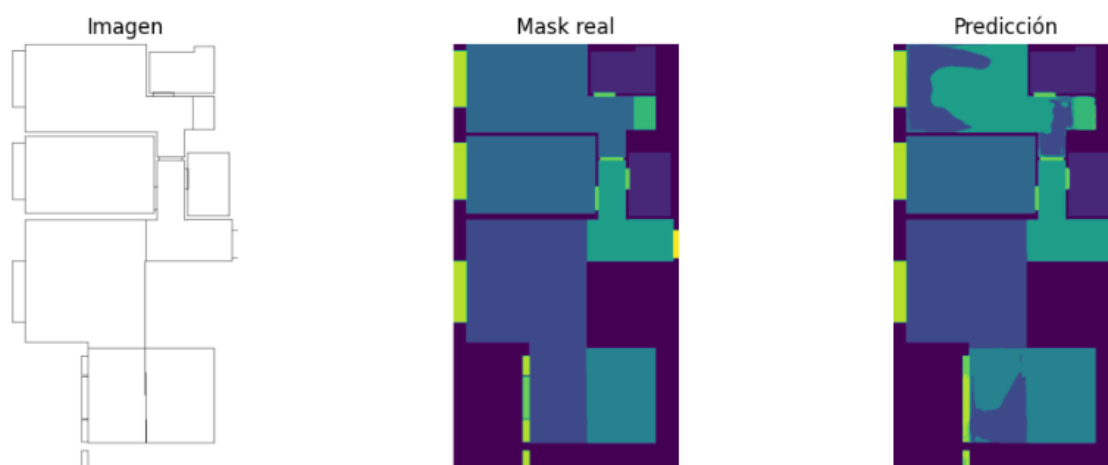


Figura 4.3 Visualización de muestras de validación

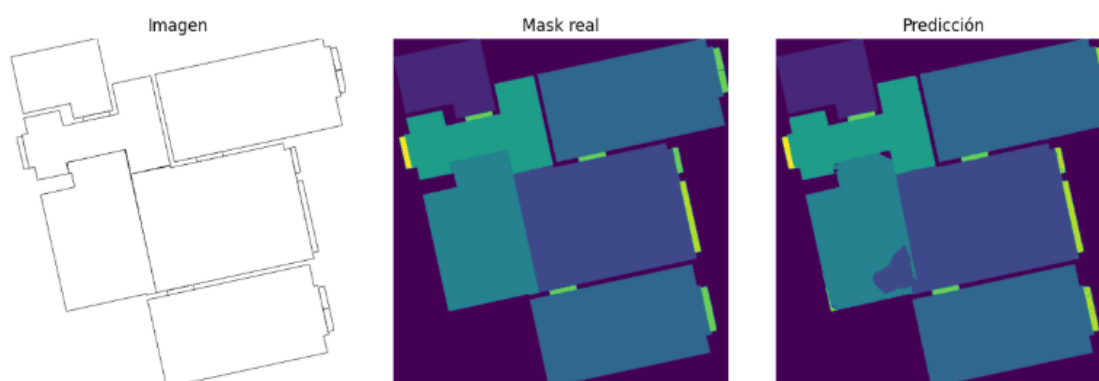


Figura 4.4 Visualización de muestras de test

Análisis de resultados y limitaciones de U-Net

El modelo **U-Net**, entrenado durante 5 épocas, mostró un comportamiento aceptable en clases de gran tamaño (salón, cocina), alcanzando valores de IoU razonables (véase Figura 4.2). Sin embargo, los resultados en elementos más pequeños como puertas y ventanas fueron irregulares, reflejando la dificultad de la arquitectura para capturar detalles finos en planos de alta complejidad.

En los resultados gráficos de entrenamiento se aprecia una disminución progresiva de la pérdida, lo que confirma que el modelo va aprendiendo las características esenciales para la segmentación (véase Figura 4.1). En las visualizaciones de muestras, se observa que la U-Net es capaz de reproducir de forma adecuada la geometría básica de las estancias principales, si bien se detectan errores en los límites entre clases y cierta confusión en estancias minoritarias (véanse Figuras 4.3–4.4).

Estos resultados ponen de manifiesto las limitaciones de la arquitectura, lo que motivó la **exploración de alternativas más modernas y complejas** con el objetivo de obtener mejores resultados.

DEEPLABV3

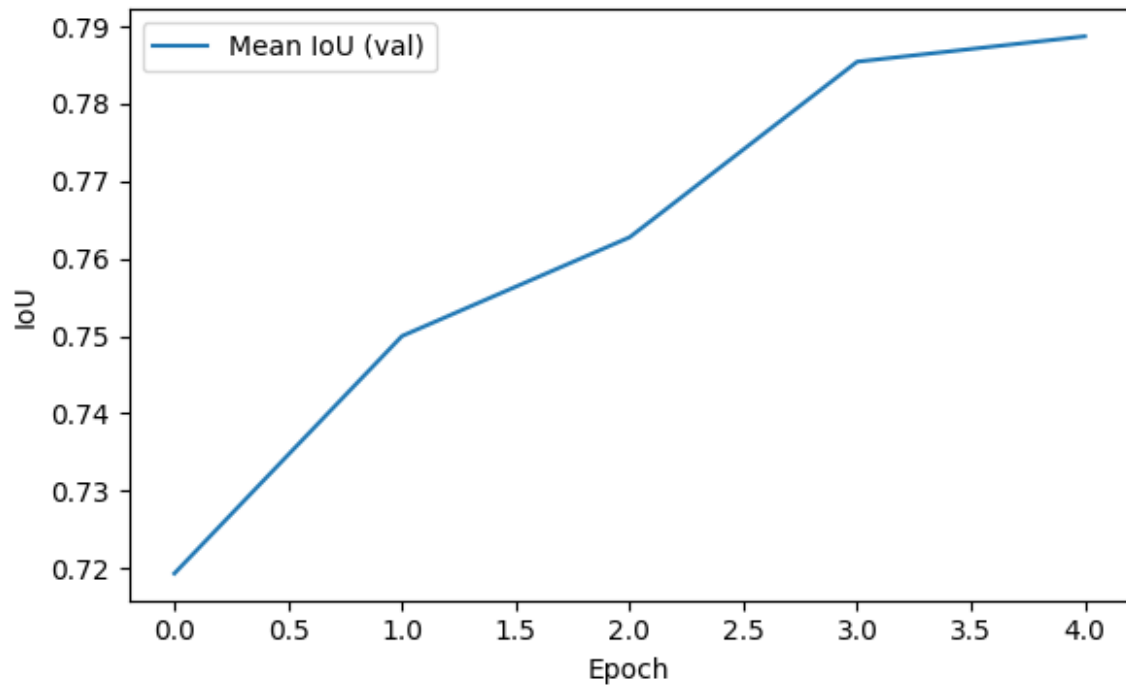
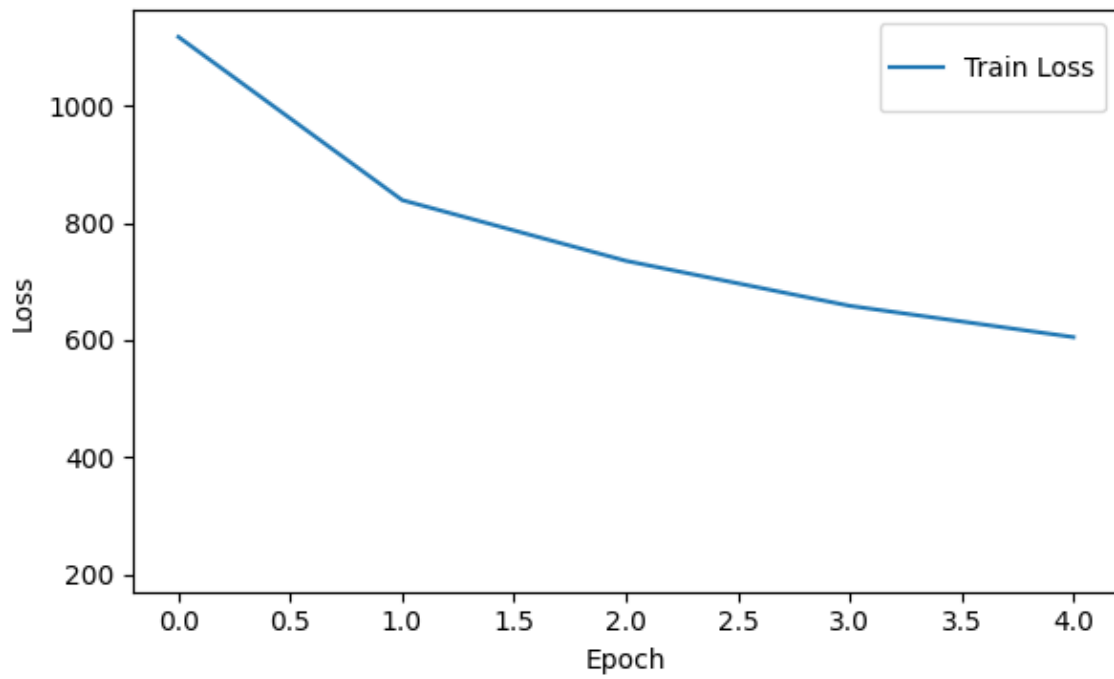


Figura 4.5 Curvas de entrenamiento (loss y mIoU) para DeeplabV3

	Epoch 6	Epoch 7	Epoch 8	Epoch 9
Fondo	0.9623	0.9671	0.9671	0.9702
Baño	0.8445	0.8462	0.8526	0.8548
Salón	0.7504	0.7618	0.7236	0.7682
Dormitorio	0.8221	0.8269	0.8140	0.8284
Cocina	0.8268	0.8090	0.8227	0.8269
Pasillo	0.8996	0.9050	0.9058	0.9053
Puerta interior	0.5554	0.5581	0.5711	0.5723
Ventana	0.7490	0.7625	0.7718	0.7646
Puerta entrada	0.7574	0.7609	0.7742	0.7716
mIoU	0.7964	0.7997	0.8003	0.8069

Figura 4.6 Tablas Mean IoU por clase hasta epoch 9

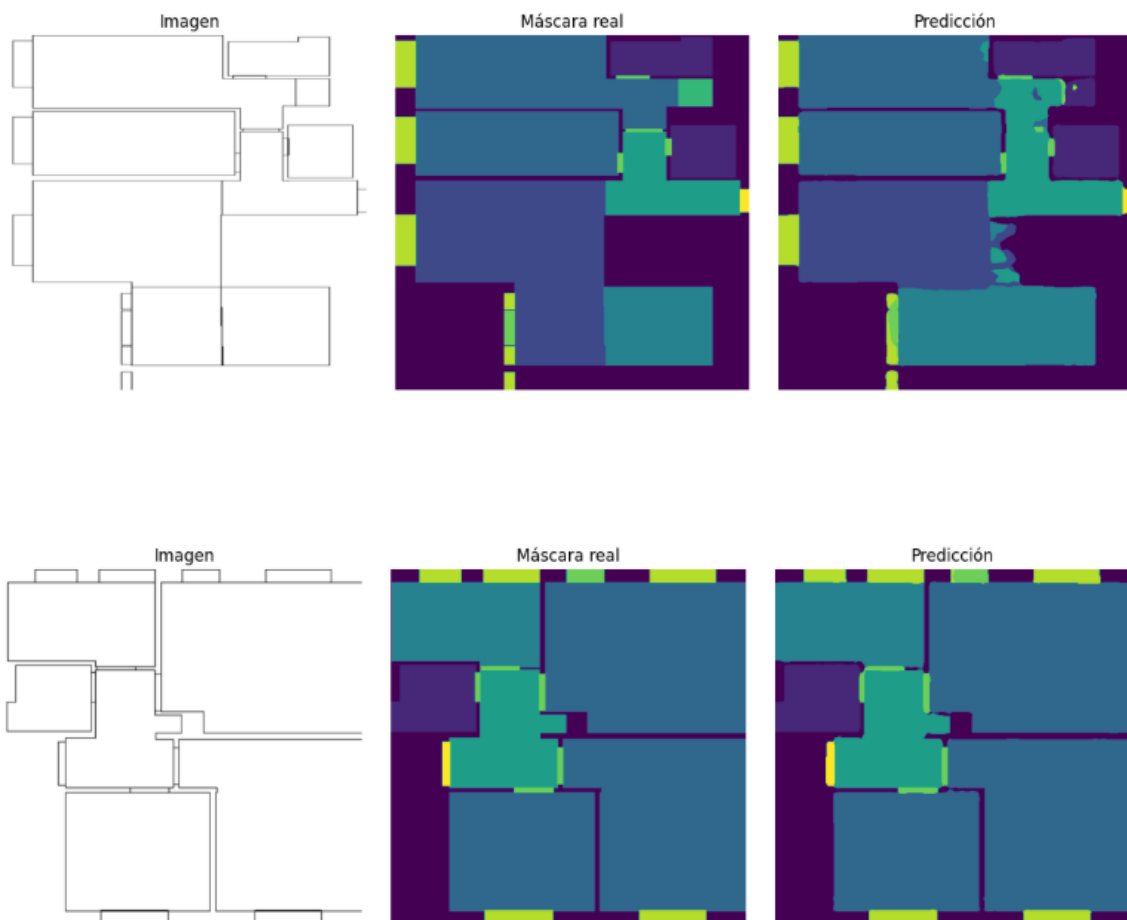
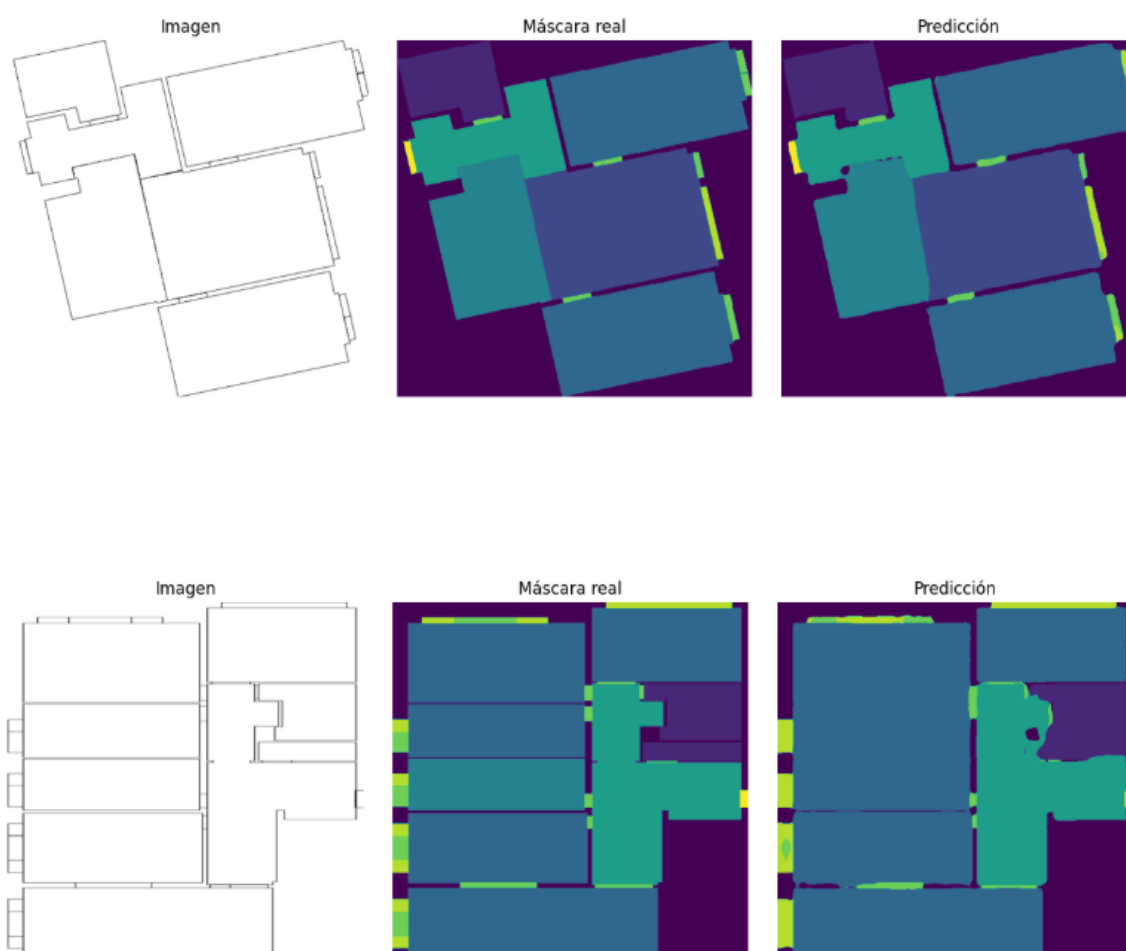




Figura 4.7 Visualización de muestras de validación (entrenado hasta epoch 5)



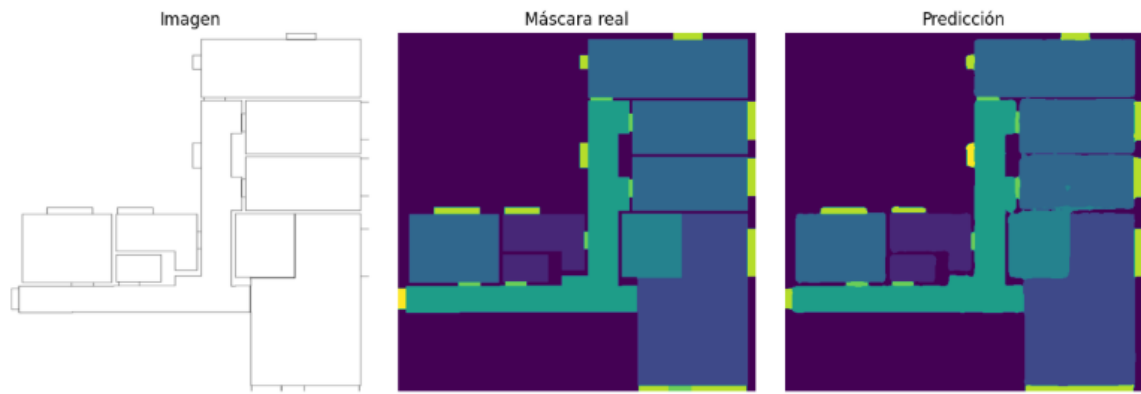


Figura 4.8 Visualización de muestras de validación (entrenado hasta epoch 5)

Inferencia plano propio real

Se preprocesa un plano de una vivienda real para que el grafismo sea similar a las imágenes del dataset MSD preparadas para el entrenamiento del modelo DeepLabV3.

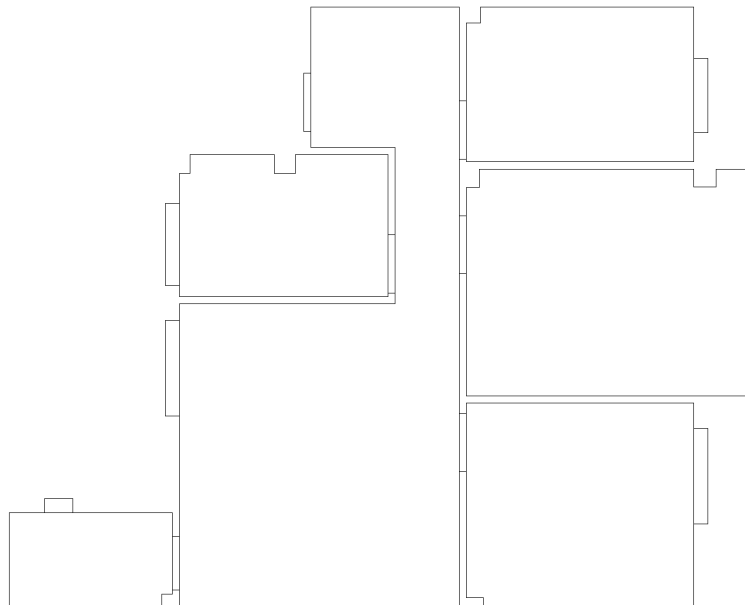


Figura 4.9 Plano propio de entrada al modelo

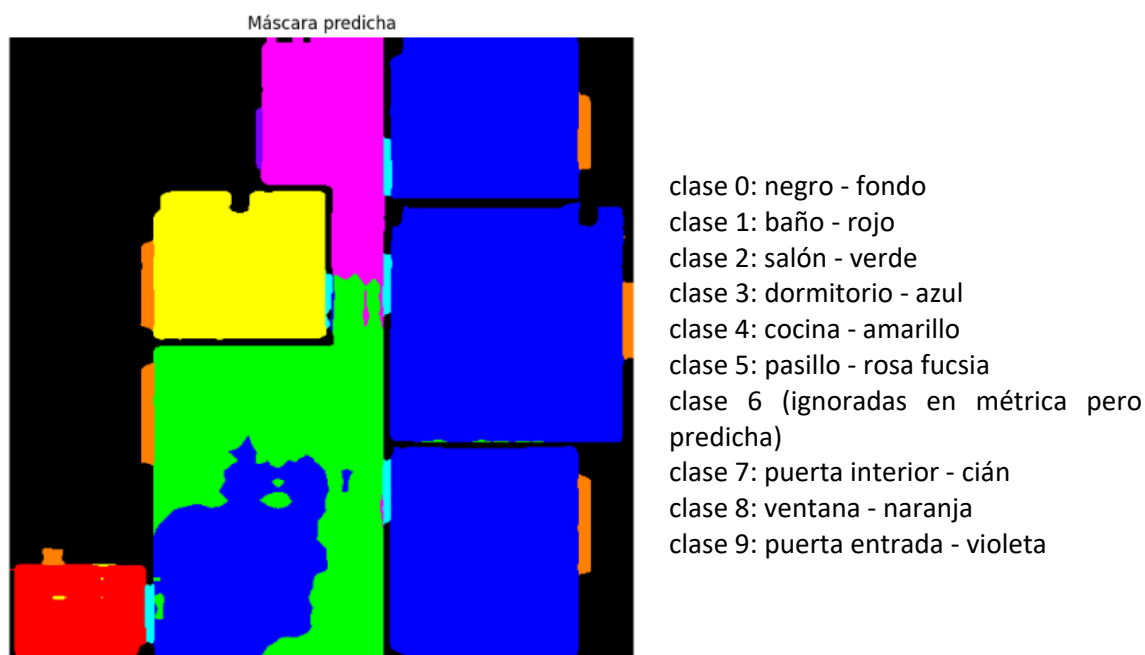


Figura 4.10 Plano propio salida del modelo

Evaluación de resultados gráficos e inferencia con DeepLabV3+

En el caso de **DeepLabV3+**, el rendimiento mejoró de forma significativa respecto a U-Net. La evolución de la pérdida de entrenamiento confirma un **aprendizaje progresivo y estable** a lo largo de las épocas (véase *Figura 4.5*). En validación, la métrica mIoU supera el umbral de 0.80, lo que evidencia un rendimiento global sólido (véase *Figura 4.6*). Las estancias de mayor superficie, como salón, baño o pasillo, se segmentan de manera fiable, mientras que, en los elementos de menor tamaño, como puertas y ventanas, los resultados fueron más desiguales, poniendo de manifiesto la dificultad del modelo para representar con precisión detalles finos en planos complejos.

Los resultados gráficos de **validación y test** muestran una mejora clara en la definición de los límites entre estancias, con segmentaciones más limpias y mayor coherencia espacial (véanse *Figuras 4.7–4.8*). Aunque siguen existiendo fallos en clases muy pequeñas, la consistencia visual es superior a la de U-Net.

En la **inferencia realizada sobre planos no vistos**, DeepLabV3+ fue capaz de generar máscaras con una estructura general bien definida y con un nivel de detalle razonable, lo que refuerza su aplicabilidad práctica frente a U-Net (véanse *Figuras 4.9 y 4.10*). No obstante, se observan aún regiones con ruido en áreas complejas, lo que justifica la posterior adopción de modelos más avanzados como SegFormer.

SEGFORMER

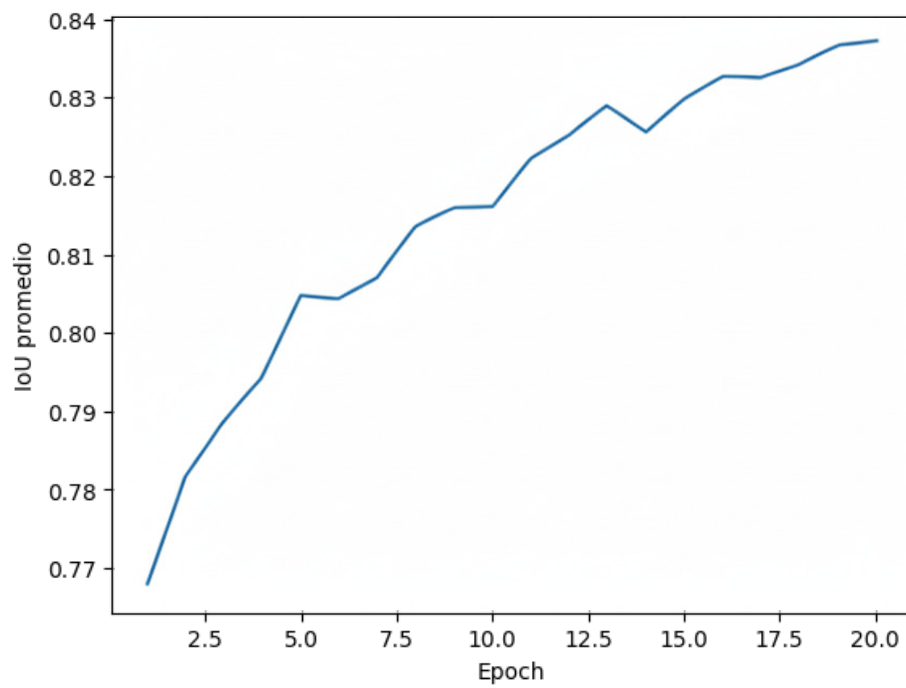
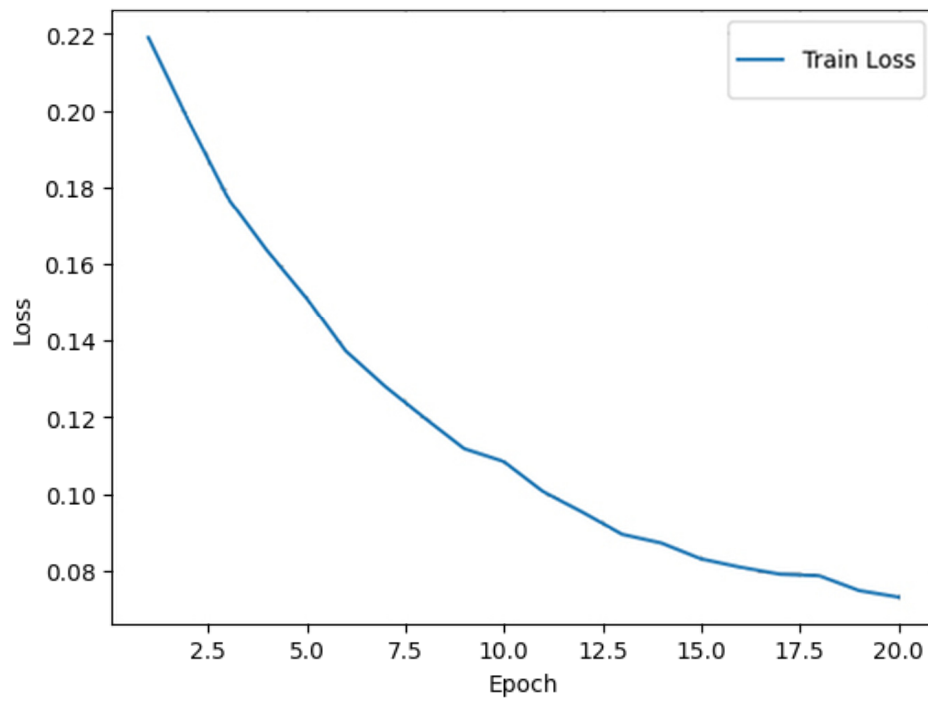
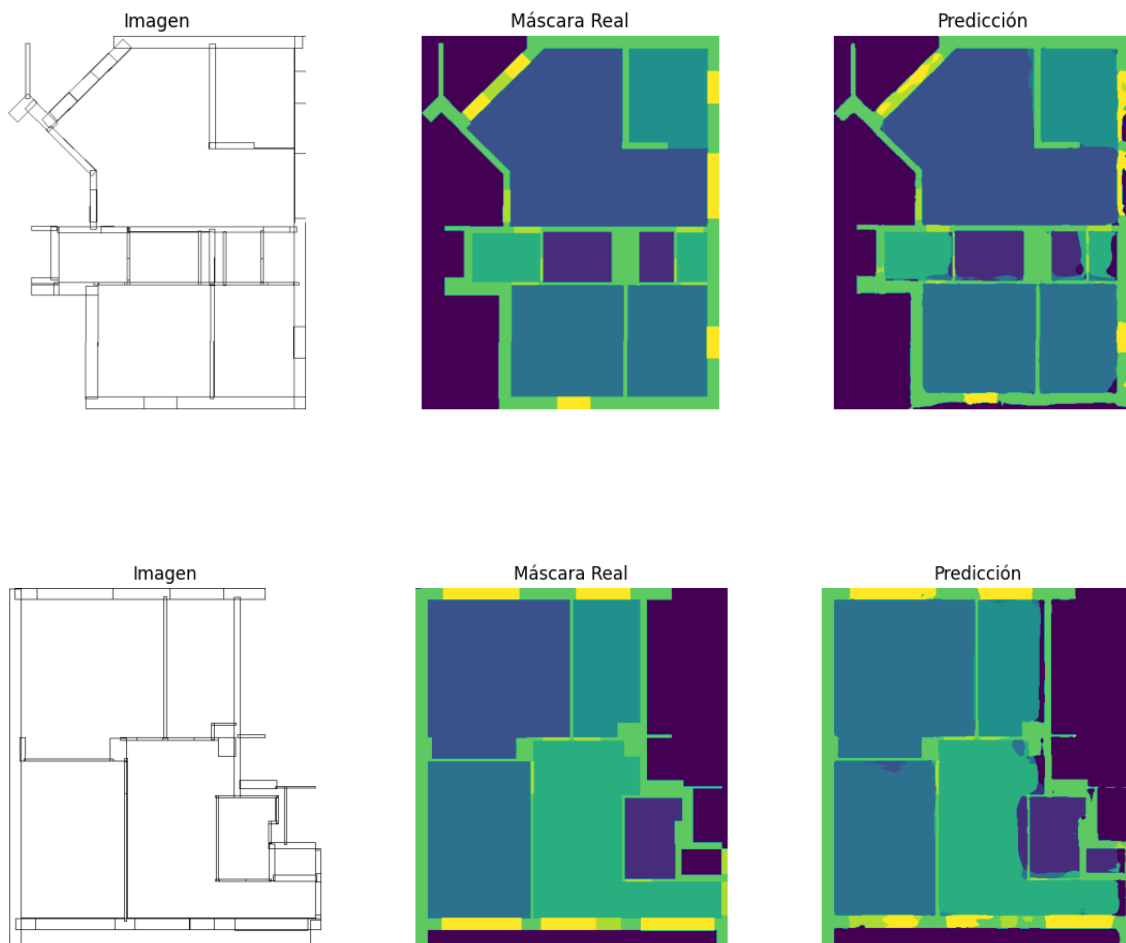


Figura 4.11 Curvas de entrenamiento (loss y mIoU) para SegFormer

	Clase	Epoch 17	Epoch 18	Epoch 19	Epoch 20
0	Fondo	0.9877	0.9877	0.9876	0.9882
1	Baño	0.8995	0.9001	0.8965	0.9019
2	Salón	0.7628	0.7712	0.7844	0.778
3	Habitación	0.8498	0.8537	0.8513	0.8504
4	Cocina	0.8663	0.8617	0.8599	0.8661
5	Distribuidor	0.914	0.9125	0.9144	0.912
6	Estructura	0.903	0.9026	0.9062	0.9068
7	Puerta	0.5842	0.5929	0.5986	0.6009
8	Ventana	0.7228	0.7222	0.7277	0.7273
mIoU	mIoU	0.8322	0.8338	0.8363	0.8368

Figura 4.12 Tablas Mean IoU por clase



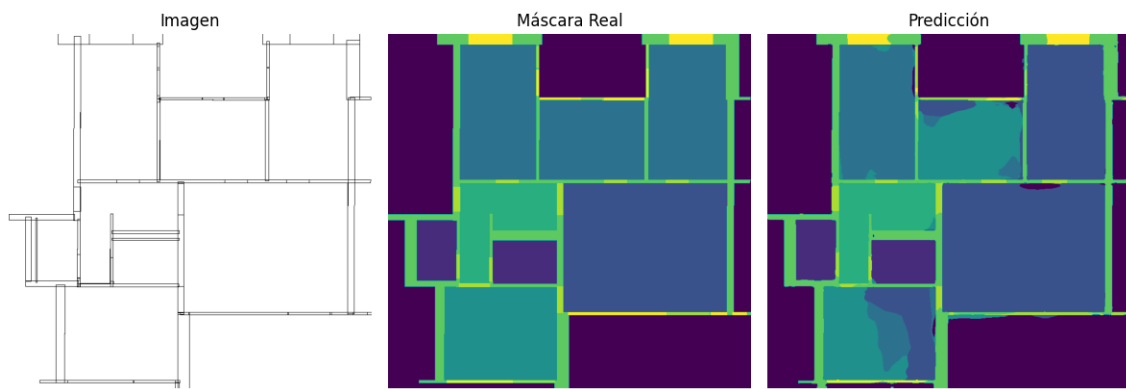


Figura 4.13 Visualización de muestras de validación (entrenado hasta epoch 20)

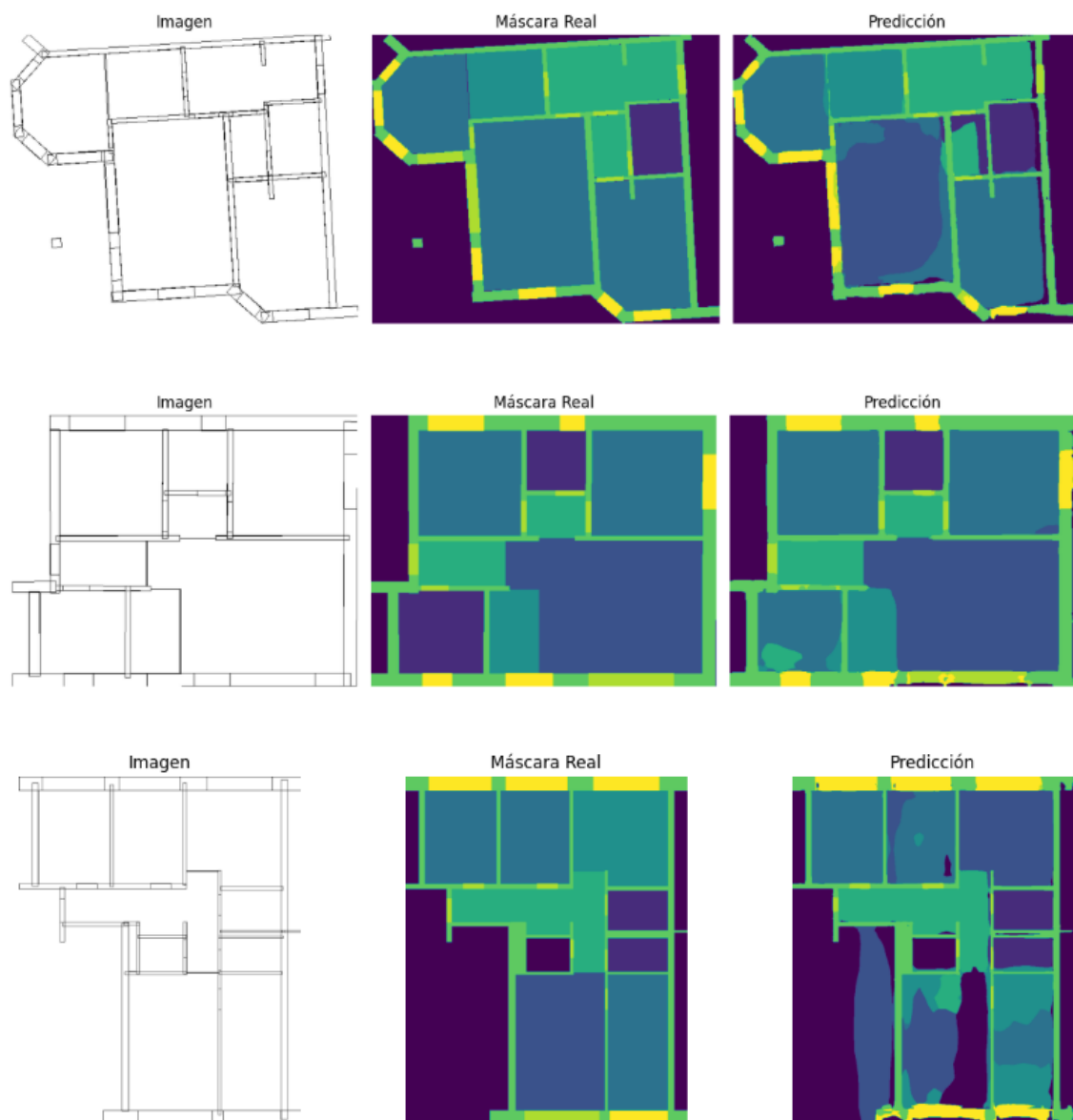


Figura 4.14 Visualización de muestras de test (entrenado hasta epoch 20)

Inferencia plano propio real

Primero se hacen comprobaciones de inferencia con planos de proporciones más o menos cuadradas.

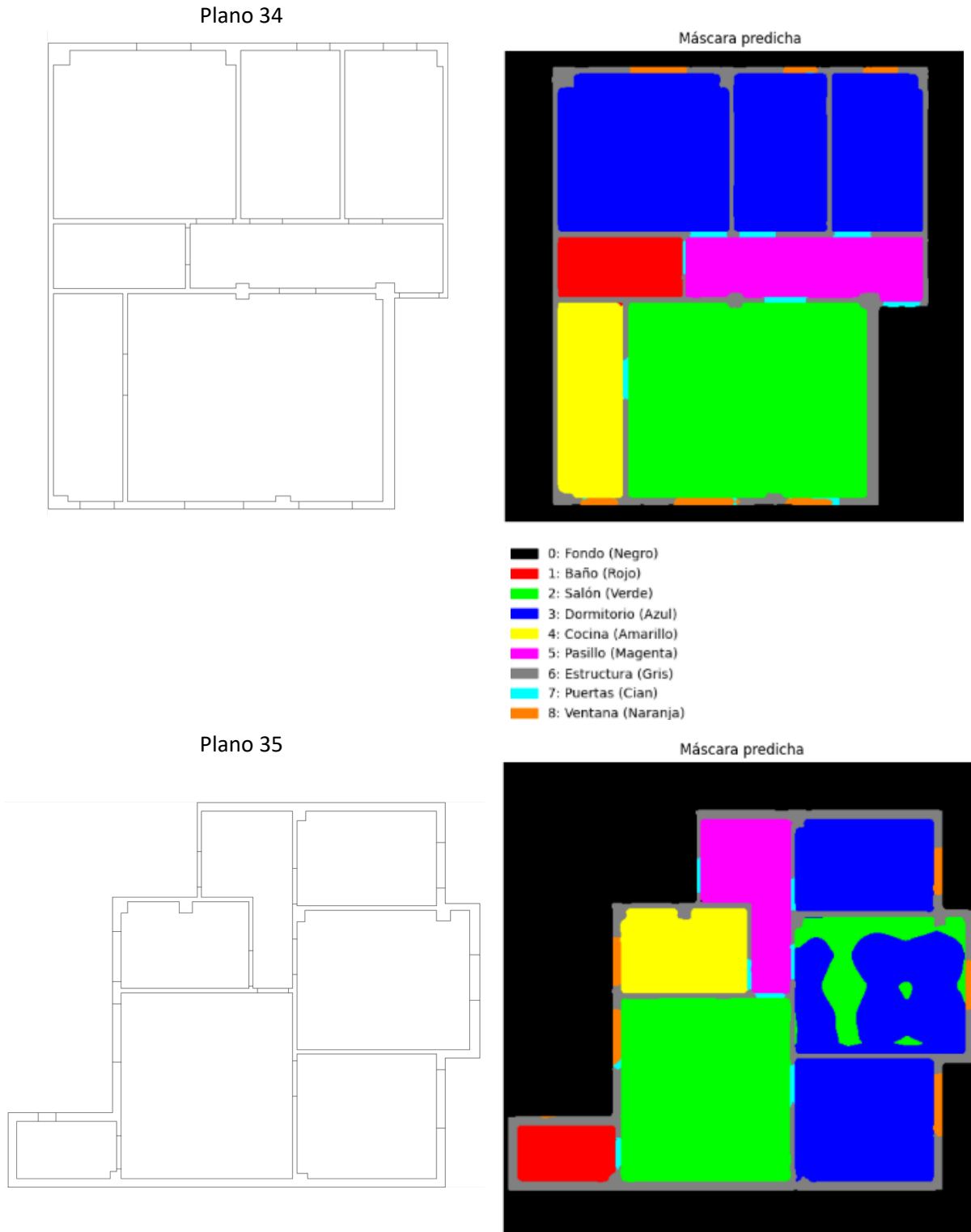
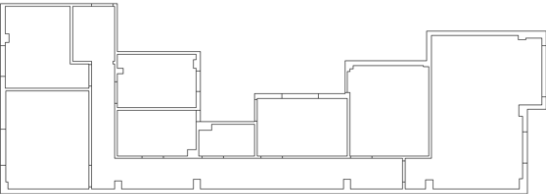


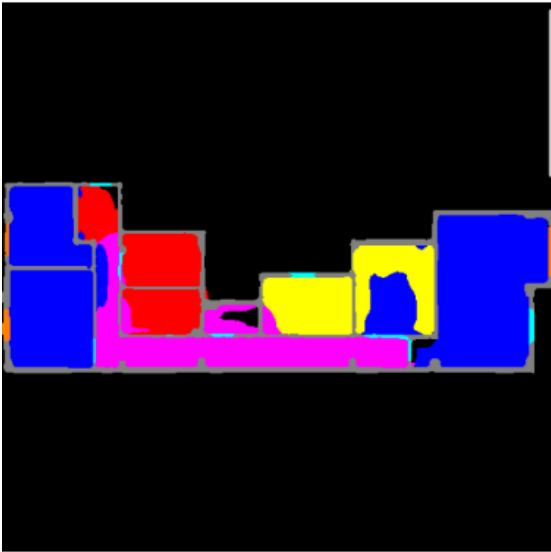
Figura 4.15 Visualización de ejemplos de planos propios reales

Se realizan también comprobaciones con planos de proporciones alargadas para comprobar los resultados.

Plano 36

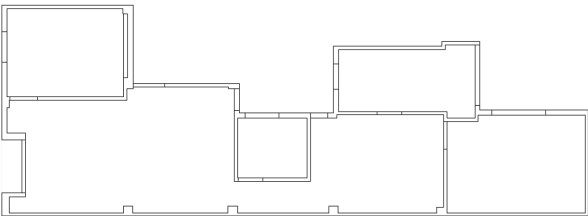


Máscara predicha



- 0: Fondo (Negro)
- 1: Baño (Rojo)
- 2: Salón (Verde)
- 3: Dormitorio (Azul)
- 4: Cocina (Amarillo)
- 5: Pasillo (Magenta)
- 6: Estructura (Gris)
- 7: Puertas (Cian)
- 8: Ventana (Naranja)

Plano 39



Máscara predicha

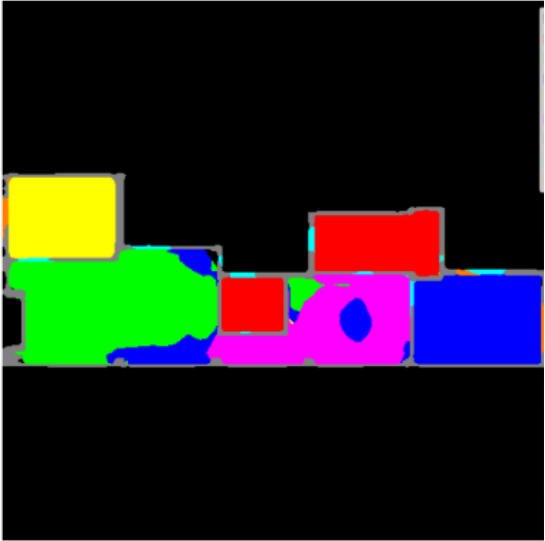


Figura 4.16 Visualización de ejemplos de planos propios reales

Evaluación de resultados e inferencia con SegFormer (B1)

El modelo **SegFormer-B1** se entrenó durante **20 épocas** y se presentan sus **curvas de entrenamiento (loss y mIoU)** junto con tablas de IoU por clase. Estas evidencias muestran una evolución estable del aprendizaje hasta el final del entrenamiento y sirven de base para el análisis cuantitativo y cualitativo que sigue (*véase Figura 4.11*).

Durante el entrenamiento se observó que la **función de pérdida en validación (CrossEntropyLoss)** tendía a estabilizarse a partir de determinadas épocas, mientras que la métrica de **Intersection over Union (IoU)** seguía mostrando mejoras. Esta diferencia se debe a que el loss mide la confianza probabilística del modelo a nivel de píxel, mientras que el IoU evalúa la calidad geométrica de la segmentación final. En la práctica, el modelo puede generar máscaras más precisas y con mejor ajuste a las regiones reales sin necesariamente reducir de forma significativa la pérdida. Por este motivo, en el presente trabajo se considera el **IoU** como la **métrica principal para valorar el rendimiento del modelo**, mientras que la pérdida se emplea como referencia auxiliar durante el entrenamiento."

Rendimiento cuantitativo. SegFormer obtiene los **mejores resultados globales** del conjunto evaluado, con **IoUs > 0,90** en clases de gran superficie como **estructura, salón, cocina y pasillo**, lo que refleja una excelente capacidad para delimitar estancias y captar el contexto global del plano. En **clases pequeñas** (p. ej., **puertas** $\approx 0,58$; **ventanas** $\approx 0,72$) el rendimiento es más modesto, en línea con la dificultad intrínseca de segmentar objetos con muy pocos píxeles (*véase Figura 4.12*).

Validación y test (resultados gráficos). En las **muestras de validación y test** (entrenado hasta la epoch 20), las máscaras predichas muestran **contornos más limpios y continuos** en las estancias principales y la **estructura**, con menor ruido en límites que en los modelos previos; los errores se concentran, sobre todo, en **elementos finos** o de **escasa ocupación** (puertas/ventanas), coherente con las métricas por clase (*véanse Figura 4.13-4.14*).

Inferencia en planos propios: ortogonales vs. alargados. En la inferencia sobre planos no vistos se documentan dos tipologías:

- **Planos de proporción casi cuadrada** (p. ej., **Plano 34 y Plano 35**), donde SegFormer mantiene una **segmentación nítida y coherente** en estancias y estructura (*véase Figura 4.15*).
- **Planos alargados** (p. ej., **Plano 36 y Plano 39**), donde aparecen **degradaciones** visibles: pérdida de continuidad en tabiques delgados, confusión en pasillos estrechos y peor

detección de puertas/ventanas. La causa principal no es el modelo, sino el **preprocesado a 512×512 px con padding**: al reducir resolución por la propia forma del plano de entrada, **las clases pequeñas ocupan muy pocos píxeles** y se vuelven más difíciles de aprender y segmentar de forma consistente. Además, también evidencia una mayor dificultad en la correcta identificación de las estancias (*véase Figura 4.16*).

Conclusión operativa. SegFormer (B1) consolida la **mejor precisión** en clases estructurales y estancias grandes y **generaliza** mejor que los modelos previos; su **mayor dificultad** aparece en **planos alargados** por la limitación de **512×512 px**, ya discutida en el preprocesado, que reduce la representatividad de elementos finos. Una **resolución mayor** (p. ej., 1024×1024) previsiblemente mitigaría esta caída a costa de más coste computacional; esta línea se recoge en la discusión de limitaciones y trabajo futuro.

Postprocesado de las máscaras segmentadas

Durante la inferencia con el modelo SegFormer se observaron pequeñas inconsistencias en algunas máscaras resultantes. Estas incidencias incluían:

- Fragmentación de estancias grandes en varias regiones desconectadas.
- Aparición de regiones residuales muy pequeñas (ruido), especialmente en clases minoritarias como puertas y ventanas.
- Píxeles aislados o mal clasificados que podían distorsionar los cálculos de superficie en fases posteriores.

Para solventar estos problemas, se implementó un proceso de **postprocesado de las máscaras**, consistente en:

1. **Etiquetado de componentes conectados** para cada clase (utilizando conectividad 4).
2. **Filtrado por tamaño mínimo** en píxeles, eliminando las regiones consideradas ruido por debajo de un umbral.
3. **Reasignación de regiones residuales** a la clase mayoritaria circundante, garantizando la coherencia espacial.
4. **Consolidación de estancias** pequeñas que quedaban artificialmente separadas en varias piezas.

Gracias a este procedimiento, las máscaras de salida se limpiaron y homogeneizaron, garantizando que cada estancia quedase representada como una única región coherente. Este

paso es clave para la siguiente fase (Mask2Text), ya que permite calcular áreas y distribuciones de forma estable y sin errores debidos a regiones mal definidas.

Síntesis comparativa de modelos

En conjunto, los resultados reflejan una progresión clara en el rendimiento de los modelos probados. **U-Net presentó limitaciones notables** en su capacidad de generalización, con sobreajuste y un rendimiento inestable en clases de menor tamaño. **DeepLabV3+ supuso una mejora significativa** gracias a su arquitectura multiescala, alcanzando un mejor equilibrio entre clases principales y minoritarias. Finalmente, **SegFormer (B1) consolidó el mejor desempeño global**, con IoUs superiores al 0,90 en estancias principales y un contexto espacial mucho más preciso. Aun así, las puertas y ventanas continuaron siendo las clases más difíciles de segmentar debido a la resolución de 512×512 px, lo que limita su representatividad espacial.

Esta evolución confirma la importancia de arquitecturas con mayor capacidad multiescala y justifica la elección de **SegFormer como modelo final para la fase 1**, sobre el cual se construyen las fases posteriores del pipeline.

Fase 2 – Descripción automática (Mask2Text)

Esta fase tiene como objetivo generar descripciones arquitectónicas a partir de las máscaras segmentadas. Para ello se consolidó un pipeline en dos pasos:

1. **Sistema determinista:** calcula la superficie total y las áreas de cada estancia a partir de la máscara segmentada. En inferencia con planos propios, la escala se ajusta automáticamente en función de los m² introducidos por el usuario, lo que garantiza proporciones realistas.
2. **LLM baseline (Qwen2-7B-Instruct):** transforma esa información estructurada en una descripción fluida en español, manteniendo las cifras exactas y evitando inventar estancias.

Para evaluar este pipeline se empleó el **JSONL generado**, que servía como dataset sintético. Además, se implementó un validador automático que detecta tres tipos de errores:

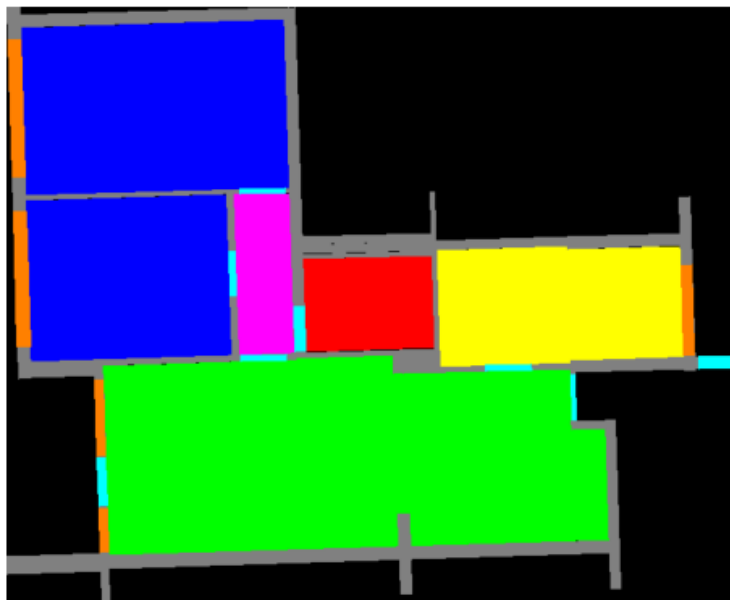
- **Errores en cifras** (m² no reflejados en el texto).
- **Errores en estancias** (espacios omitidos).
- **Estancias inventadas** (mencionadas sin estar en los datos originales).

Los resultados mostraron una elevada fidelidad numérica y de inventario, con desviaciones mínimas en casos de planos complejos o con estancias muy pequeñas. El texto generado fue claro y consistente, aunque limitado en variedad estilística debido a que el prompt estaba diseñado de forma restrictiva para maximizar la precisión.

En conclusión, este pipeline determinista + LLM permitió obtener **descripciones fiables** y directamente utilizables en fases posteriores del sistema, asumiendo la limitación de que la riqueza lingüística quedaba supeditada al diseño del prompt y no a un aprendizaje autónomo.

Evaluación sobre ejemplos del dataset en Fase 2 (con máscaras preprocesadas según fase 1)

🖼 Ejemplo: c67091dea08d68bd8be4810208fb5104.png



📌 Leyenda de clases y colores:

- 0: fondo → negro
- 1: baño → rojo
- 2: salón → verde
- 3: dormitorio → azul
- 4: cocina → amarillo
- 5: distribuidor → magenta
- 6: estructura → gris
- 7: puertas → cian
- 8: ventanas → naranja

◆ Determinista:

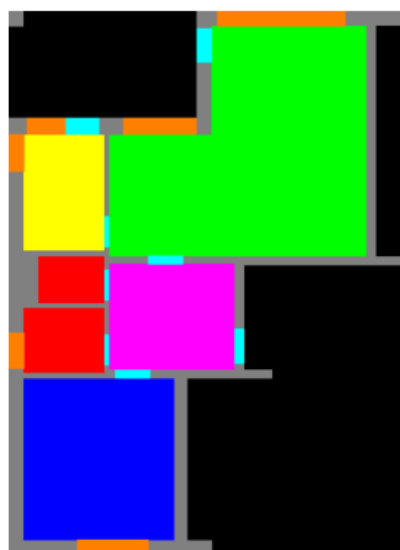
La vivienda tiene 73.5 m². Se divide en un baño de 4.08 m², un salón de 30.55 m², 2 dormitorios de 15.20 m² y 11.08 m², una cocina de 9.54 m² y un distribuidor de 3.06 m².

◆ LLM Baseline:

Esta vivienda es un ejemplo de eficiencia y funcionalidad en un espacio de 73.50 m². Al entrar, se encuentra un amplio y luminoso salón de 30.55 m², perfecto para recibir invitados o disfrutar de un tranquilo momento en familia. A su derecha se ubica el dormitorio principal de 15.20 m², ideal para aquellos que valoran privacidad y comodidad. Por otro lado, el dormitorio más pequeño, con 11.08 m², es perfecto para niños o como estudio. La cocina, de 9.54 m², es mediana y bien equipada, permitiendo una fácil preparación de alimentos. El baño, de 4.08 m², es mediano y cuenta con todas las comodidades necesarias. Finalmente, el distribuidor de 3.06 m², aunque corto, ayuda a optimizar el flujo de movimiento entre las estancias. En resumen, esta vivienda ofrece una distribución inteligente y eficiente que se adapta a las necesidades de cualquier hogar.

Figura 4.17 Visualización de ejemplo de plano del dataset

Ejemplo: 8eac9438e36d64ca394ed02651152fe6.png



- 📌 Leyenda de clases y colores:
- 0: fondo → negro
 - 1: baño → rojo
 - 2: salón → verde
 - 3: dormitorio → azul
 - 4: cocina → amarillo
 - 5: distribuidor → magenta
 - 6: estructura → gris
 - 7: puertas → cian
 - 8: ventanas → naranja

◆ Determinista:

La vivienda tiene 61.76 m². Se divide en 2 baños de 3.12 m² y 1.85 m², un salón de 28.73 m², un dormitorio de 14.56 m², una cocina de 5.58 m² y un distribuidor de 7.92 m².

◆ LLM Baseline:

Esta vivienda es un espacio funcional y eficiente, con una superficie total de 61.76 m². Se destaca por su salón amplio y luminoso de 28.73 m², perfecto para recibir invitados o disfrutar de momentos familiares. El dormitorio, también grande, ofrece un espacio confortable para descansar. La cocina, aunque pequeña, está bien integrada y cuenta con todo lo necesario para preparar comidas deliciosas. Los baños, uno de 3.12 m² y otro de 1.85 m², son prácticos y funcionales. El distribuidor largo de 7.92 m² ayuda a optimizar el flujo de movimiento en la vivienda. En resumen, esta casa es un hogar acogedor y bien diseñado que se adapta a las necesidades de una familia.

Figura 4.18 Visualización de ejemplo de plano del dataset

Conclusión de los ejemplos del dataset en Fase 2

Los ejemplos extraídos del dataset muestran cómo el sistema Mask2Text **cumple adecuadamente con su propósito**. El módulo determinista identifica y calcula con precisión las superficies de cada estancia a partir de la máscara segmentada, mientras que el modelo LLM baseline genera descripciones fluidas y naturales en español.

En los dos casos analizados, las cifras se mantienen fieles a los valores de referencia y las descripciones resultan coherentes y comprensibles (*véanse Figuras 4.17-4.18*). Estos resultados confirman que el pipeline es capaz de transformar la información geométrica de las máscaras en inventarios textuales fiables, listos para ser utilizados en fases posteriores.

Fase 3 – Motor de decisión

La validación del módulo de reglas se centró en garantizar la coherencia interna y la fiabilidad de la salida JSONL generada a partir de las intenciones de reforma. En esta fase, se seleccionaron

ejemplos procedentes del dataset para comprobar el funcionamiento del motor de decisión en casos diversos y no diseñados específicamente para el flujo completo del TFM.

Los resultados obtenidos mostraron que el motor de decisión funcionó de manera estable, robusta y predecible, generando siempre un JSONL estructurado conforme a lo esperado. En el 100 % de los intentos, el archivo de salida incluyó correctamente los campos definidos en el esquema: el identificador del plano procesado, la descripción de las estancias, el cálculo de áreas en metros cuadrados, la acción de reforma propuesta y el bloque de validación de restricciones. Este último elemento reviste especial importancia, ya que permite comprobar de forma automática si la propuesta cumple o no con los criterios geométricos y normativos definidos.

Se verificó igualmente que las acciones de reforma únicamente se ejecutaban cuando las condiciones geométricas y lógicas mínimas estaban presentes. En el caso de la intención *abrir cocina al salón*, la operación se descartaba de manera automática cuando las estancias no eran contiguas, cuando la frontera no correspondía a un tabique interior válido (por ejemplo, al detectarse la proximidad de ventanas que indicaban contacto con fachada), o cuando el segmento de muro seleccionado no alcanzaba la longitud mínima necesaria para alojar un hueco de paso. En estas situaciones, el motor generaba igualmente un JSON válido, pero sin acciones propuestas y con un registro explícito de los errores detectados en el campo de validación.

En términos cuantitativos, el análisis de 1.366 ejemplos de la intención *abrir cocina al salón* mostró que el **100 % de los JSON fueron válidos** a nivel de esquema. Sin embargo, solo un 22,4 % presentaba contigüidad real entre cocina y salón. Dentro de ese subconjunto, **prácticamente el 100 % cumplió también las restricciones geométricas** (tabique interior válido, longitud mínima, ausencia de ventanas en la frontera), lo que equivale a un 22,2 % sobre el total de ejemplos. En otras palabras, **siempre que se detectó contigüidad, el motor de reglas fue capaz de validar correctamente la apertura propuesta.**

Los resultados globales se resumen en la siguiente tabla.

Validez JSON (OPEN) (%)	Contigüidad (GT) (%)	Cumplimiento restricciones (%)	Precisión ubicación (px, l)	Rechazo correcto si no hay contigüidad (%)	Casos (OPEN)
100.0	22.401171	22.254758	61.171014	77.598829	1366

Figura 4.19 Tabla resumen Fase 3

Finalmente, al estratificar los resultados por morfología del plano, se observó que los **planos ortogonales** alcanzaron mejores resultados (25,9 % de contigüidad y 25,8 % de cumplimiento de reglas) frente a los **planos alargados** (18,8 % y 18,6 % respectivamente). Esta diferencia de

unos 7 puntos porcentuales evidencia que la geometría de la vivienda influye en la robustez del motor: los planos alargados concentran la información en franjas estrechas que se ven más afectadas por la reducción a 512×512 píxeles, lo que dificulta la detección correcta de la frontera cocina–salón.

La siguiente tabla muestra los resultados estratificados por forma.

Grupo	Casos (F3)	Validez JSON (% 1)	Contigüidad GT (% 1)	Cumple reglas (% 1)	Precisión ubicación (px, 1)
Alargado	676	100.0	18.79	18.64	61.364
Ortogonales	690	100.0	25.94	25.80	61.034

Figura 4.20 Tabla Fase 4 estratificado por forma

La consistencia en la generación de la salida JSONL constituye, en todo caso, una de las principales aportaciones de esta fase. El carácter determinista garantiza un comportamiento reproducible y aporta trazabilidad al proceso de decisión, incluso en escenarios donde un modelo de lenguaje pudiera producir respuestas ambiguas o inconsistentes.

Conclusión de los resultados cuantitativos en Fase 3

La validación de la Fase 3 demuestra que el motor de reglas genera siempre salidas JSON válidas y reproducibles, garantizando la trazabilidad del proceso. De los 1.366 ejemplos procesados, solo un 22,4 % presentaba contigüidad real entre cocina y salón y un 22,2 % cumplió todas las restricciones geométricas, lo que refleja que el sistema descarta correctamente los casos no viables (véase Figura 4.19).

Al comparar tipologías, los planos ortogonales ofrecieron un rendimiento algo mayor (25,9 % de contigüidad frente al 18,8 % en planos alargados), evidenciando que la discretización a 512×512 penaliza más a las geometrías estrechas (véase Figura 4.20).

En conjunto, los resultados confirman que el enfoque determinista aplicado es robusto, reproducible y técnicamente coherente, si bien la proporción de casos válidos está condicionada por las características del dataset.

Fase 4 – Visualización final de la propuesta

La evaluación de la fase de visualización se centró en la validez geométrica de las aperturas representadas mediante overlays sobre el plano original. El objetivo fue comprobar que las acciones propuestas en la Fase 3 pudieran trasladarse de manera coherente y reproducible al

raster del plano. En total, se generaron **304 casos con apertura aceptada**, sobre los cuales se aplicaron varias métricas para valorar su adecuación geométrica.

Los resultados globales se resumen en la siguiente tabla.

Alineación con frontera (%)	No-intersección con estructura (%)	Legibilidad geométrica (%)	Espesor relativo (1 estable)	Casos con apertura (OPEN)
6.866844	39.82362	100.0	3.34983	304

Figura 4.21 Tabla resumen Fase 4

En ella se observa que la **alineación media con la frontera cocina–salón** fue del **6,9 %**, lo que indica que únicamente una fracción reducida de píxeles del segmento coincidió exactamente con los píxeles de la frontera detectada. Aunque la métrica de alineación con la frontera cocina–salón arrojó un valor medio del 6,9 %, este resultado refleja una limitación de la propia métrica (línea de referencia de un píxel de grosor) más que un fallo del sistema. Visualmente, las aperturas se sitúan en los tabiques correctos y la **legibilidad geométrica fue del 100 %**, lo que valida la coherencia práctica del overlay generado.

La métrica de **no-intersección con estructura** alcanzó un **39,8 %**, reflejando que en muchos casos el trazo del hueco no coincidió exactamente con los píxeles etiquetados como tabiques. Este resultado no implica que el sistema propusiera huecos en posiciones incorrectas, sino que revela la dificultad de capturar con precisión subpíxel la geometría del muro en una imagen discretizada. Visualmente, las aperturas generadas se situaban en la frontera cocina–salón, pero el chequeo pixelar penaliza cualquier desviación mínima.

Por otro lado, la **legibilidad geométrica** de las aperturas fue del **100 %**, garantizando que todos los huecos se representaron como un único trazo válido y sin fragmentaciones. El **espesor relativo** se mantuvo estable en torno a **3,35 píxeles**, confirmando la consistencia de la representación entre ejemplos.

Al estratificar los resultados según la forma de los planos, se observó una diferencia atribuible a la proporción geométrica: los **planos alargados** alcanzaron un **8,3 % de alineación con la frontera**, frente al **5,8 % en los planos ortogonales**. Este comportamiento se debe a que los planos más alargados se ven más afectados por la discretización al normalizarse a 512×512, lo que genera ligeros desfases en la localización de las fronteras. En contraste, tanto la legibilidad como el espesor relativo del trazo se mantuvieron prácticamente constantes entre ambos grupos.

La siguiente tabla muestra el desglose estratificado por forma.

Grupo	Casos con apertura (F4)	Alineación frontera (% , t)	No-intersección estructura (% , t)	Legibilidad geométrica (% , t)	Espesor relativo (t estable)
Alargado	126	8.34	42.68	100.0	3.387
Ortogonales	178	5.83	37.80	100.0	3.323

Figura 4.22 Tabla Fase 4 estratificado por forma

En conjunto, la Fase 4 demostró que el sistema es capaz de **producir overlays claros y coherentes**, con una geometría reproducible y estable. No obstante, los resultados también subrayan que la resolución de entrada influye directamente en la precisión geométrica, lo que abre la vía a explorar en el futuro estrategias de normalización adaptativa o resoluciones mayores para mejorar la alineación de las aperturas con las fronteras detectadas.

Conclusión de los resultados cuantitativos en Fase 4

La evaluación de la Fase 4 demuestra que el sistema puede representar gráficamente las aperturas propuestas de manera coherente y reproducible. Aunque la métrica de alineación con la frontera cocina–salón ofrece valores bajos debido a la discretización por píxeles (*véase Figura 4.21*), las aperturas generadas se sitúan en los tabiques correctos y mantienen una legibilidad geométrica del 100 %. Los valores de espesor relativo confirman la estabilidad de la representación y las diferencias observadas entre planos ortogonales y alargados son consistentes con la influencia de la proporción de cada plano en la métrica (*véase Figura 4.22*). En conjunto, los resultados validan la viabilidad del enfoque de visualización adoptado, garantizando que las propuestas de reforma no solo se generen en formato estructurado, sino que también puedan comunicarse de forma clara sobre el plano original.

4.2 Resultados cualitativos

A continuación, se presenta un **ejemplo completo del pipeline**, desde la entrada inicial hasta la propuesta de reforma final. Este ejemplo ilustra cómo las distintas fases del sistema trabajan de manera sinérgica para transformar un plano en una propuesta inteligente.

FASE 1 - Plano segmentado

Imagen de entrada: Se utiliza un plano de vivienda en formato rasterizado. El modelo de segmentación lo procesa y es capaz de reconocer y separar las estancias (salón, cocina, pasillo), la estructura de las paredes y otros elementos como puertas y ventanas.

Máscara segmentada: El resultado es una imagen que representa la "comprensión" del plano por parte de la IA.

Máscara refinada: Posteriormente, se aplica un proceso de post-procesado sobre la máscara para corregir pequeñas incoherencias y mejorar la homogeneidad en las áreas de cada clase.

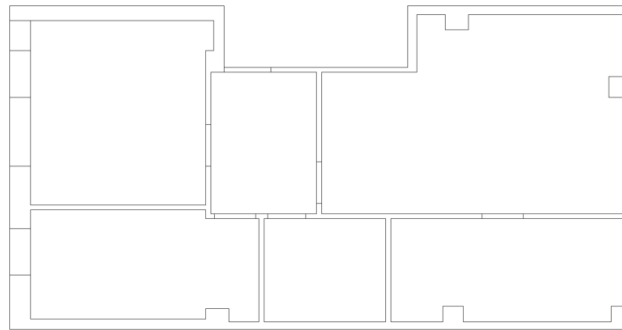


Figura 4.23 Plano propio de entrada

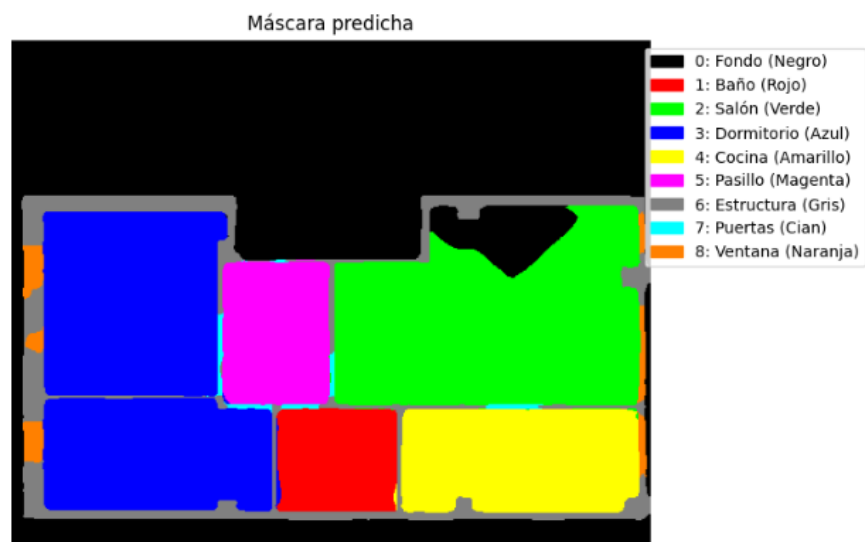


Figura 4.24 Plano segmentado en Fase 1



Figura 4.25 Máscara refinada en Fase 1

Conclusiones del comportamiento en Fase 1

El modelo de segmentación demuestra una capacidad sólida para identificar correctamente las distintas estancias y elementos constructivos de un plano real (véase *Figura 4.24*). La refinación posterior de la máscara permite mejorar la definición de los contornos y reducir errores en clases críticas como puertas y ventanas véase (*Figura 4.25*). En conjunto, esta fase asegura una base geométrica fiable sobre la que se construyen las fases siguientes del pipeline.

FASE 2 - Descripción generada

Texto de salida: El modelo **Mask2Text** toma la máscara de segmentación como entrada y la traduce a una descripción textual estructurada y legible. El usuario debe introducir los m2 totales de la vivienda para que el modelo calcule los m2 de cada estancia.

Introduce los m² construidos de la vivienda (ej: 88.43): 69,2



- 📌 Leyenda de clases y colores:
- 0: fondo → negro
 - 1: baño → rojo
 - 2: salón → verde
 - 3: dormitorio → azul
 - 4: cocina → amarillo
 - 5: distribuidor → magenta
 - 6: estructura → gris
 - 7: puertas → cian
 - 8: ventanas → naranja

◆ Determinista:

La vivienda tiene 69.2 m². Se divide en un baño de 5.32 m², un salón de 23.01 m², 2 dormitorios de 13.77 m² y 10.41 m², una cocina de 10.33 m² y un distribuidor de 6.36 m².

◆ LLM Baseline:

Esta vivienda es un espacio funcional y eficiente, con una superficie total de 69.20 m². El salón, con 23.01 m², es el corazón de la casa, ofreciendo amplitud y luminosidad gracias a su tamaño generoso. Los dormitorios son de diferentes tamaños, con el dormitorio_1 de 13.77 m², un espacio mediano que se adapta a una variedad de necesidades, y el dormitorio_2 de 10.41 m², más pequeño pero aún confortable. La cocina, de 10.33 m², es grande y práctica, diseñada para aquellos que disfrutan de la gastronomía. El baño, con 5.32 m², es de tamaño grande y cuenta con todas las comodidades necesarias. Finalmente, el distribuidor de 6.36 m², aunque de tamaño medio, es esencial para mantener la fluidez del tráfico de la casa. En resumen, esta vivienda ofrece una distribución equilibrada y espacios bien definidos para satisfacer las necesidades de cualquier hogar.

Figura 4.26 Texto descriptivo de la vivienda – Salida de Fase 2

Conclusiones del comportamiento en Fase 2

El sistema logra combinar de manera coherente el cálculo determinista de superficies con la generación de un texto más expresivo gracias al LLM. En este ejemplo, los valores numéricos obtenidos a partir de la segmentación se trasladan con precisión al discurso, y el modelo de lenguaje añade matices descriptivos que aportan naturalidad y una visión más cercana al usuario final. La inferencia sobre un plano propio confirma que el módulo es capaz de traducir datos puramente geométricos en una narrativa comprensible, equilibrando exactitud técnica y claridad comunicativa (véase Figura 4.26).

FASE 3 - Motor de decisión con acción de reforma

Intención: El sistema recibe una intención de reforma predefinida, como “abrir cocina al salón”.

Salida JSON: El motor de decisión procesa esta intención y, tras validar que la cocina y el salón son contiguos, detecta la pared que los separa y genera un JSON con la información geométrica necesaria. Además, muestra un debug visual que ayuda a interpretar si ha funcionado correctamente.

Verificación normativa (RAG): En esta fase se han integrado reglas duras y blandas mediante un sistema de retrieval-augmented generation (RAG). Las reglas duras verifican automáticamente el cumplimiento de la normativa DC-09 de Valencia, garantizando que la superficie resultante del espacio integrado (cocina + salón) supera los mínimos exigidos. Paralelamente, se incorporan recomendaciones “blandas” procedentes del CTE (por ejemplo, ventilación y paso útil), que no bloquean la propuesta pero sí ofrecen pautas adicionales para su revisión.



Figura 4.27 Debug visual – Comprobación de salida de Fase 3

🚩 Verificación normativa (DC-09)
- Cocina: 10.33 m²
- Salón: 23.01 m²
- Integrado: 33.34 m² | Umbral: 18.0 m²
✅ Cumple

Figura 4.28 Comprobación de normativa - Salida de Fase 3

Conclusiones del comportamiento en Fase 3

En la aplicación sobre un plano propio, el sistema identificó correctamente la pared separadora (véase Figura 4.27), generó el JSON de salida con la acción de reforma y verificó el cumplimiento normativo. La superficie integrada (33,34 m²) superó holgadamente el umbral de 18 m² establecido en la DC-09 (véase Figura 4.28), validando la propuesta como viable tanto desde un punto de vista geométrico como regulatorio.

FASE 4 - Visualización final de la propuesta

Plano original vs plano reformado: El sistema superpone un overlay gráfico sobre el plano original, mostrando la pared a demoler con una línea verde semitransparente. Esta representación visual permite identificar de un vistazo el lugar exacto de la intervención sin perder la legibilidad del plano base.

Etiqueta gráfica: Se añade una etiqueta textual que resume el cambio: “Tabique a demoler”. Esta anotación refuerza la claridad del resultado y facilita la interpretación de la acción propuesta.

Descripción final: El sistema genera una descripción narrativa mediante un LLM (Qwen2-7B-Instruct), que no solo explica la reforma realizada, sino que además incorpora recomendaciones “blandas” de carácter normativo. De este modo, la descripción incluye tanto la justificación espacial de la apertura (continuidad, luz natural, mejora de la circulación) como sugerencias alineadas con el CTE y la DC-09, tales como la verificación de la ventilación o la comprobación de las condiciones de accesibilidad.



Propuesta de reforma: La reforma se centra en la integración de la cocina y el salón, creando un espacio abierto y luminoso. Al unir estas áreas, se alcanza una superficie total de 33.34 m², superando el mínimo requerido de 18.00 m² establecido en la normativa local. Esta acción permite una mayor continuidad espacial, aumenta la cantidad de luz natural gracias a la eliminación de una pared no estructural, y mejora la circulación del aire. Es recomendable incorporar sistemas de ventilación y/o extracción en la zona de la cocina para mantener la calidad del aire y evitar posibles malos olores o acumulaciones de humo. Además, es importante asegurarse de que se mantiene un buen flujo de aire en todo el espacio, garantizando así una correcta circulación del aire y optimizando la sensación de amplitud y confort del hogar.

Figura 4.29 Plano final de propuesta de reforma + descripción textual - Salida de Fase 4

Conclusiones del comportamiento en Fase 4

La fase de visualización integra en un mismo resultado el plano reformado y la explicación normativa. El overlay aporta un recurso gráfico claro y directo, mientras que la narrativa generada combina viabilidad técnica con pautas de diseño responsables (véase Figura 4.29). El resultado final es una propuesta de reforma comprensible, fundamentada y comunicable tanto a usuarios no especializados como a profesionales del sector.

4.3 Explicabilidad del modelo

Más allá de medir métricas cuantitativas o mostrar ejemplos cualitativos, resulta de interés analizar cómo el modelo SegFormer **llega a identificar las distintas estancias de una vivienda**. El objetivo de este apartado no es evaluar su precisión numérica, sino **intentar descifrar el comportamiento interno del modelo**, observando qué patrones utiliza para clasificar los espacios.

Para ello, se han realizado pruebas de inferencia controladas en las que se utilizan planos de vivienda y se generan versiones modificadas con variaciones en el tamaño de ciertas estancias. De este modo se estudia cómo influyen esas modificaciones en la segmentación resultante y qué pistas parecen guiar las decisiones del modelo.

El estudio parte del **Plano 25**, un plano propio real utilizado como caso de referencia, al que se aplicarán modificaciones en las dimensiones de sus estancias con el fin de comparar el comportamiento del modelo en distintas configuraciones.

4.3.1 Ejemplo de plano de inferencia: Plano 25

En la Figura 4.30 se muestra el **plano original de entrada** y en la Figura 4.31 la **máscara segmentada predicha** por el modelo SegFormer. Conviene subrayar que, en el proceso de

inferencia, el modelo recibe únicamente el plano rasterizado en blanco y negro: **no dispone de etiquetas textuales ni de cotas de dimensiones**. Los nombres de estancias y medidas que se incluyen en la figura del plano original se han añadido únicamente con fines explicativos para este análisis.

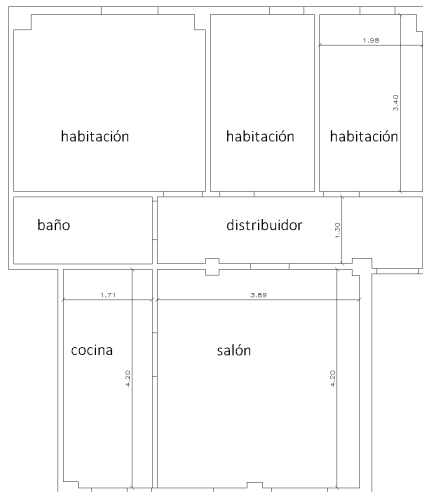


Figura 4.30 Plano 25

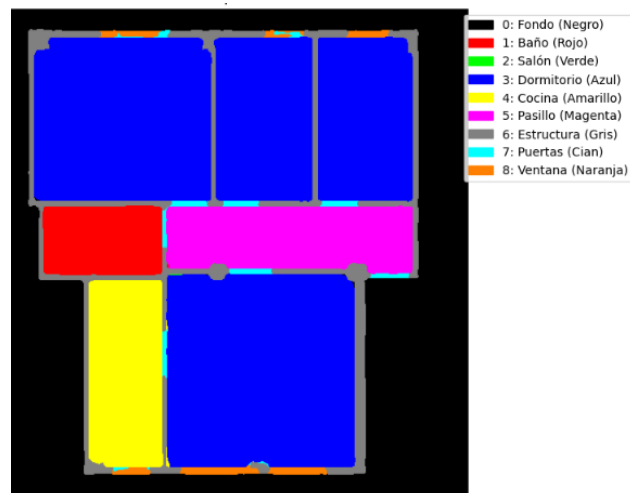


Figura 4.31 Segmentación Plano 25

El resultado muestra que **la mayoría de espacios se identificaron correctamente**: la cocina, el baño, el distribuidor y los tres dormitorios aparecen segmentados de forma coherente con la realidad. Sin embargo, **el salón fue clasificado erróneamente como dormitorio**, pese a ser la estancia de mayor tamaño en la vivienda.

Variación 1

En esta variación se elimina el tabique que separa el pasillo del salón.

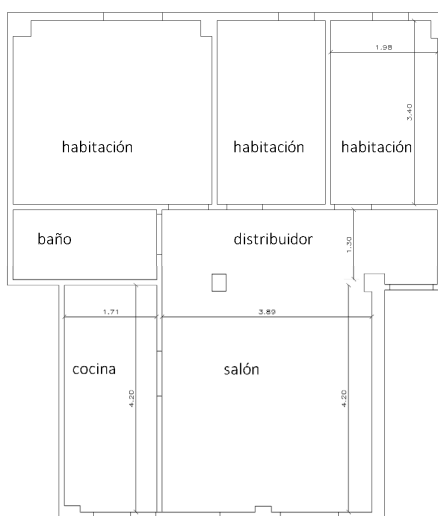


Figura 4.32 Plano 25a

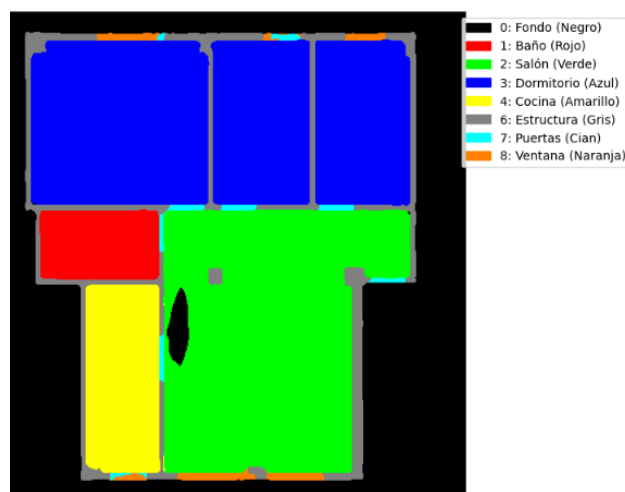


Figura 4.33 Segmentación Plano 25a

El resultado muestra que la mayoría de estancias principales **se mantuvieron correctamente identificadas**: la cocina, el baño y los dormitorios aparecen segmentados de forma coherente con la realidad. Sin embargo, al eliminar el tabique, **el pasillo dejó de reconocerse** como estancia independiente y toda la zona fue clasificada como salón (véase Figura 4.33).

Variación 2

En esta variación se amplía el salón 60 cm hacia la izquierda, la cocina también se amplía 20 cm.

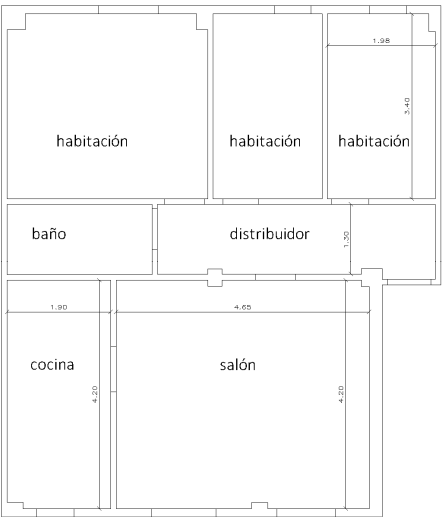


Figura 4.34 Plano 25b

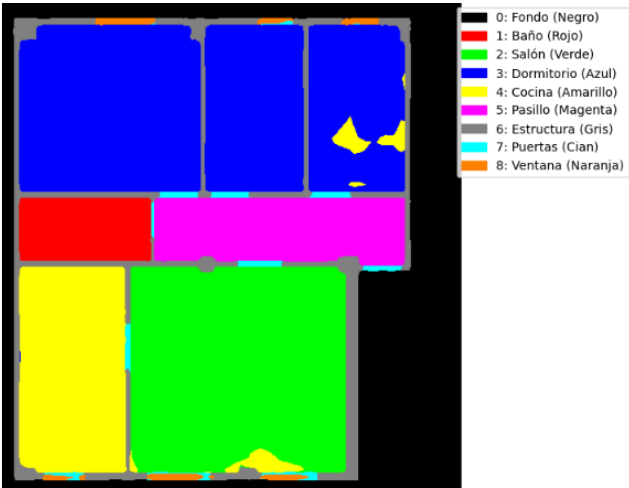


Figura 4.35 Segmentación Plano 25b

El resultado muestra que **todas las estancias se identificaron correctamente**: la cocina, el baño, el distribuidor, los dormitorios y el salón fueron segmentados de forma coherente con la realidad (véase Figura 4.35).

Variación 3

En esta variación la cocina se amplía 70 cm más y se reduce el salón a su dimensión original.

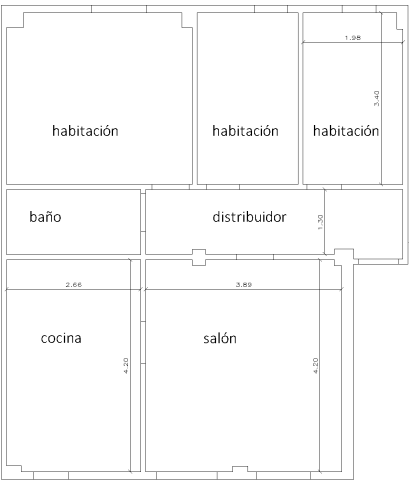


Figura 4.36 Plano 25c

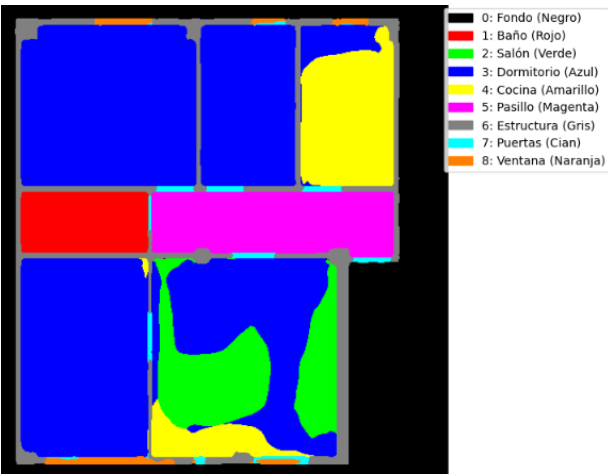


Figura 4.37 Segmentación Plano 25c

El resultado muestra una **segmentación más inestable** que en las pruebas previas. Aunque el baño, el distribuidor y dos de los dormitorios se identificaron correctamente, el salón presentó una segmentación confusa y la cocina dejó de reconocerse como tal, siendo clasificada como dormitorio. Además, una de las habitaciones apareció mayoritariamente segmentada como cocina (véase Figura 4.37).

4.3.2 Observaciones del comportamiento

El comportamiento del modelo SegFormer frente al Plano 25 y sus variaciones revela varios aspectos clave:

- **Sensibilidad a la geometría:** cuando un tabique desaparece, también lo hace la estancia asociada. El modelo interpreta la continuidad espacial como un único recinto, lo que demuestra que la **delimitación física** es su principal criterio de segmentación.
- **Dependencia de proporciones:** los espacios son clasificados en función de sus dimensiones relativas. Cocinas alargadas y pequeñas suelen reconocerse bien, mientras que salones más reducidos tienden a confundirse con dormitorios amplios.
- **Estabilidad limitada:** el sistema mantiene predicciones coherentes mientras las estancias permanecen dentro de los rangos habituales aprendidos. Sin embargo, cuando una cocina se amplía más allá de lo común, o un salón pierde protagonismo, surgen errores notables en la clasificación.
- **Influencia de las dimensiones de las estancias:** cuando se amplía una estancia que el modelo tiene aprendida como “pequeña” hasta alcanzar dimensiones habituales de otros recintos, la red puede confundir las categorías y reasignar etiquetas. En lugar de mantener la estancia ampliada con su uso original, el modelo tiende a “buscar” otro espacio más reducido al que asignarle esa función.

Un ejemplo claro se observa con la cocina: al aumentar su superficie hasta tamaños próximos a los de un dormitorio, deja de reconocerse como cocina y una de las habitaciones más pequeñas pasa a clasificarse como tal.

En conjunto, las pruebas muestran que el modelo **no necesita mobiliario para segmentar**, sino que se guía por geometría, proporciones y delimitaciones. Su fiabilidad depende de que las viviendas mantengan configuraciones cercanas a las del dataset de entrenamiento, lo que explica tanto los aciertos consistentes como las confusiones observadas en escenarios menos habituales.

4.3.3 Conclusiones

El análisis de explicabilidad confirma que el SegFormer basa sus predicciones en la geometría y las proporciones de los recintos, con los tabiques como elemento clave para diferenciar estancias. Lo más significativo es que el modelo es capaz de reconocer espacios **sin necesidad de mobiliario ni elementos interiores**, apoyándose únicamente en las formas, perímetros y tamaños relativos de cada recinto.

Esta estrategia le permite segmentar con precisión viviendas convencionales, pero también explica las confusiones detectadas en configuraciones menos habituales, como cocinas ampliadas que se asemejan a dormitorios o salones reducidos que pierden protagonismo frente a otras habitaciones. En esos casos, el modelo tiende a reasignar etiquetas, trasladando funciones de unas estancias a otras.

En definitiva, el SegFormer actúa como un **clasificador geométrico condicionado por los patrones estadísticos del dataset**, con buen rendimiento en tipologías estándar y limitaciones claras al enfrentarse a distribuciones menos frecuentes.

5. Discusión

El desarrollo de este trabajo ha permitido demostrar la viabilidad técnica de un *pipeline* basado en inteligencia artificial para automatizar, de manera parcial y asistida, el proceso de propuesta de reformas arquitectónicas. Los resultados obtenidos, tanto a nivel cuantitativo como cualitativo, validan el enfoque metodológico adoptado y abren una serie de reflexiones cruciales sobre las fortalezas, limitaciones y el futuro de esta tecnología en la práctica profesional.

No obstante, los resultados cuantitativos evidencian **ciertas limitaciones** derivadas de la discretización a 512×512 píxeles: únicamente un 22,4 % de los ejemplos del dataset presentaron contigüidad real entre cocina y salón en la Fase 3, y en la Fase 4 la alineación media con la frontera se situó en torno al 6,9 %. Estos valores, aunque bajos, reflejan más una **limitación metodológica** en la representación pixelar que un fallo conceptual del motor de reglas, pero subrayan la necesidad de explorar estrategias de normalización adaptativa o mayor resolución en fases futuras.

5.1 Fortalezas del sistema propuesto

El enfoque metodológico y la arquitectura del sistema presentan varias fortalezas significativas que lo distinguen de otros enfoques académicos y comerciales:

- **Pipeline completo e integrado:** A diferencia de la mayoría de los trabajos de investigación que se centran en una única fase (por ejemplo, solo segmentación o solo generación de *layouts*), el presente proyecto ha logrado construir un **flujo de trabajo end-to-end**. La integración de la visión por computador, el procesamiento del lenguaje natural y un motor de reglas en un único sistema es una contribución metodológica clave. Esta sinergia demuestra que la combinación de diferentes tecnologías de IA puede resolver un problema complejo de manera holística, sin necesidad de intervención manual entre fases.

El pipeline desarrollado demuestra viabilidad técnica, pero **no puede considerarse todavía un sistema aplicable directamente en el ámbito profesional**. Requiere supervisión y ajustes adicionales para cumplir con normativas y garantizar precisión geométrica.

- **Sencillez en la preparación de los planos:** Una ventaja destacada del sistema es que los planos empleados en el pipeline carecen de mobiliario y se reducen a perímetros y espesores de muros. Este grafismo depurado y uniforme simplifica enormemente la

preparación de los planos para su uso en el modelo, reduciendo la necesidad de preprocesamiento y facilitando su implementación práctica en flujos de trabajo reales.

- **Validación normativa integrada mediante RAG:** Una de las contribuciones más destacadas es la incorporación de un mecanismo de *retrieval-augmented generation* (RAG) que permite consultar directamente la normativa vigente, en particular el **CTE** y la **DC-09 de Valencia**, a partir de sus documentos oficiales en PDF. Gracias a este módulo, el sistema no solo comprueba reglas geométricas básicas (contigüidad, dimensiones de huecos o clases de tabiques), sino que también valida el cumplimiento de **reglas duras** como las superficies mínimas exigidas por la DC-09. Adicionalmente, en la Fase 4 se incluyen recomendaciones **blandas** relacionadas con el CTE (ventilación, paso útil, accesibilidad) que enriquecen la propuesta final. Esta combinación dota al sistema de una mayor **robustez y realismo profesional**, reduciendo la distancia entre un prototipo académico y una herramienta aplicable en la práctica arquitectónica.
- **Modularidad de la arquitectura:** El diseño del sistema en **módulos independientes** es una fortaleza fundamental para su escalabilidad y sostenibilidad. Cada fase, desde la segmentación hasta la visualización, puede ser optimizada o incluso reemplazada por modelos más avanzados sin comprometer la integridad del *pipeline* completo. Por ejemplo, si en el futuro se desarrolla un modelo de segmentación de planos con mayor precisión, se podría integrar fácilmente en la Fase 1 sin tener que reentrenar los modelos de las Fases 2, 3 y 4. Esto reduce significativamente los costes de desarrollo y facilita la mejora continua.
- **Aplicabilidad práctica:** El sistema constituye una base sólida para su aplicación en el mundo profesional. Al generar propuestas preliminares de manera **rápida y reproducible**, podría funcionar como una herramienta de pre-análisis en estudios de arquitectura, reduciendo el tiempo de las primeras reuniones con clientes y permitiendo una exploración más ágil de las opciones de diseño. Además, su potencial en plataformas inmobiliarias es considerable, al poder ofrecer a los usuarios una visualización instantánea de las posibilidades de reforma de una vivienda.
- **Uso de datasets públicos:** El haber trabajado principalmente con *datasets* de acceso público, garantiza que la metodología y los resultados sean **reproducibles y verificables** por otros investigadores. Esto no solo contribuye a la transparencia del trabajo, sino que también permite a la comunidad académica construir sobre las bases establecidas en este proyecto.

5.2 Limitaciones del trabajo

Pese a sus fortalezas, el sistema presenta una serie de limitaciones inherentes a la naturaleza de un prototipo de investigación:

- **Número limitado de intenciones de reforma:** Con el objetivo de validar la viabilidad del *pipeline* completo, la lógica del motor de decisión se restringió a una única acción de reforma (abrir cocina al salón). Si bien esto permitió un análisis en profundidad de la cadena de valor, limita la diversidad de propuestas que el sistema puede generar. Un sistema profesional debería ser capaz de procesar un conjunto mucho más amplio de intenciones, como "añadir un baño", "reubicar un tabique" o "redistribuir las zonas de día y noche".

El hecho de que solo se haya implementado una intención de reforma (abrir cocina al salón) pone de manifiesto tanto la complejidad del problema como la necesidad de modularidad. Cada nueva intención exige reglas específicas y comprobaciones geométricas particulares.

- **Dependencia de la segmentación:** Como se ha demostrado en la metodología, los errores en la Fase 1 tienen un **efecto cascada** que se propaga por todo el sistema. Con la introducción del módulo determinista esta dependencia se vuelve aún más crítica, ya que cualquier fallo en la segmentación afecta directamente al cálculo de superficies y, por tanto, a la descripción textual y a la propuesta de reforma. Una delimitación incorrecta de paredes o estancias no solo degrada el texto generado, sino que también conduce a incoherencias geométricas en las fases posteriores.

A esta limitación se suma la resolución de entrada fijada en 512×512 píxeles por motivos de eficiencia computacional, que compromete el rendimiento en clases pequeñas como puertas y ventanas. Un aumento a 1024×1024 píxeles podría mejorar sustancialmente los resultados, constituyendo una línea de trabajo futura a explorar.

- **Limitación del corpus textual:** Aunque se resolvió el problema del dataset sintético descartando el entrenamiento directo de un encoder-decoder, el nuevo sistema sigue generando descripciones con una **expresividad limitada**. El módulo determinista garantiza consistencia numérica, pero produce salidas muy estructuradas y homogéneas, lo que condiciona al LLM a un rango reducido de variación lingüística. Para alcanzar una narrativa más natural y rica sería necesario disponer de un corpus real de descripciones arquitectónicas anotadas por profesionales.

- **Visualización esquemática:** La Fase 4 se centra en una **visualización simple y esquemática** que es un *overlay* 2D. Aunque útil para el propósito de este trabajo, no alcanza la riqueza expresiva y la complejidad de las visualizaciones generadas por herramientas profesionales como AutoCAD o Revit, que ofrecen modelos 3D, texturas y detalles constructivos.

5.3 Comparación con procesos manuales

El contraste entre el *pipeline* automático y el proceso manual tradicional es clave para entender el valor del sistema desarrollado.

En la práctica profesional, la elaboración de una propuesta de reforma es un proceso artesanal que implica:

1. **Análisis manual** del plano (identificación de elementos, toma de medidas).
2. **Deliberación** sobre las opciones de diseño.
3. **Generación de alternativas** mediante dibujo digital o a mano.

El sistema desarrollado, a pesar de sus limitaciones, ofrece una **propuesta preliminar en cuestión de minutos**. Esto representa una ventaja significativa en términos de **rapidez** y **reproducibilidad**. Permite que un arquitecto explore un gran número de ideas de forma rápida, reduciendo la carga de trabajo en la fase inicial del proyecto y facilitando la comunicación con el cliente. No obstante, es fundamental destacar que el sistema **no reemplaza la necesidad de supervisión humana**. La propuesta final debe ser validada por un profesional para garantizar su cumplimiento normativo, su viabilidad estructural completa y su adaptabilidad a las necesidades específicas del cliente. El sistema se presenta como una **herramienta de asistencia**, no como un sustituto del juicio y la experiencia del arquitecto.

Comparándolo con procesos manuales se evidencia que, si bien el tiempo de propuesta se reduce drásticamente, la calidad de las soluciones aún no alcanza la flexibilidad y riqueza de la práctica arquitectónica tradicional.

5.4 Líneas de investigación futura

El presente trabajo sienta las bases para una agenda de investigación prometedora. Las siguientes líneas representan una continuación natural del proyecto, con el objetivo de llevar el sistema de un prototipo a una herramienta más completa y robusta:

- **Ampliación de las intenciones de reforma:** Expandir el conjunto de reglas del motor de decisión para incluir acciones más complejas y comunes, como "añadir un baño", "ampliar una habitación uniendo dos estancias", "redistribuir las zonas de día y noche" o "crear un espacio de oficina".
- **Incorporación de normativas arquitectónicas:** Implementar un sistema de validación de reglas que incorpore normativas de construcción. Esto podría lograrse mediante la creación de un motor de reglas más complejo o la integración de un **sistema RAG** que consulte bases de datos legales y de construcción para verificar la viabilidad de la reforma.
- **Mejora del *dataset* textual:** Para superar las limitaciones del *dataset* sintético, se podría colaborar con estudios de arquitectura para generar un corpus de descripciones reales y anotadas. Esto mejoraría la naturalidad del lenguaje del sistema y su capacidad para describir detalles más complejos.
- **Refinamiento del motor de decisión:** Evolucionar desde un sistema puramente basado en reglas a uno que utilice un modelo de razonamiento basado en IA, como una **red neuronal de grafo (GNN)** que analice las relaciones espaciales entre las estancias y aprenda las reglas de diseño a partir de ejemplos.
- **Visualización avanzada:** Explorar la posibilidad de conectar el *pipeline* con motores de renderizado 3D o con *frameworks* de **BIM** (Building Information Modeling). Esto permitiría generar propuestas visuales más ricas y detalladas, incluyendo texturas, mobiliario y la integración con flujos de trabajo profesionales.
- **Comparación con modelos LLM generales:** Analizar de forma exhaustiva las capacidades actuales de los **grandes modelos de lenguaje multimodales (LLMs)** como GPT-5, Claude 4 o Gemini 2.5 en el dominio de la reforma arquitectónica. Esta investigación futura permitiría determinar si el enfoque de *pipeline* especializado sigue siendo superior en términos de precisión y salida estructurada, o si los LLMs, con el entrenamiento adecuado, pueden asumir la totalidad de la tarea.

5.5 Contraste con Modelos de Lenguaje Generalistas (LLMs)

La aparición de grandes modelos de lenguaje multimodales como GPT-5, Claude 4 o Gemini 2.5 ha planteado la cuestión de si un *pipeline* especializado como el desarrollado en este TFM sigue siendo necesario. Para responder a esta pregunta, se realizó un experimento cualitativo,

sometiendo a un LLM líder, como es GPT-5, a la misma tarea de reforma de planos. Para poder comparar los resultados de forma rigurosa, se utilizó el **mismo plano propio de entrada** que se ha mostrado en los resultados cualitativos y que recorría todo el *pipeline* especializado.

si te doy un plano me harías una propuesta de reforma?

Claro, Víctor 👍.

Si me pasas un plano puedo analizarlo y proponerte opciones de reforma. Para hacerlo bien necesitaría:

1. 📐 El plano en imagen o PDF (aunque lo ideal es que sea lo más claro posible, con estancias y cotas si las tienes).
2. 🏠 Tu objetivo: por ejemplo, si buscas ampliar el salón, abrir la cocina, ganar un dormitorio, mejorar la luz natural, redistribuir zona de día/noche, etc.
3. 🚧 Limitaciones o condiciones: metros que no se pueden tocar, si quieres mantener baños en su sitio, si hay muros de carga que ya conoces, etc.
4. 🧠 Estilo o prioridades: más espacio abierto, más almacenaje, accesibilidad, eficiencia energética, etc.

Con eso puedo proponerte varias alternativas:

- Esquema explicativo con los tabiques a modificar.
- Justificación funcional de cada cambio.
- Posibles pros y contras de cada opción.

Figura 5.1 Conversación inicial con GPT-5

La conversación inicial demostró que el LLM tiene una gran capacidad para entender el lenguaje natural y entablar un diálogo conversacional. El modelo es capaz de pedir información relevante para el proyecto y de organizar las ideas de reforma de manera lógica en formato de texto.



abrir cocina al salón, hazmelo de forma gráfica dibujando sobre el plano con marcas y una descripción

Figura 5.2 Conversación con GPT-5

Sin embargo, al enfrentar la primera tarea crucial —la **representación gráfica** de una reforma sobre un plano— el LLM mostró su primera limitación fundamental. En lugar de modificar el plano original proporcionado, el modelo generó una imagen genérica y esquemática que no se corresponde con el plano real. Esta "alucinación" gráfica, aunque plausible, demuestra su incapacidad para manipular o editar una imagen con la precisión requerida en el dominio arquitectónico.

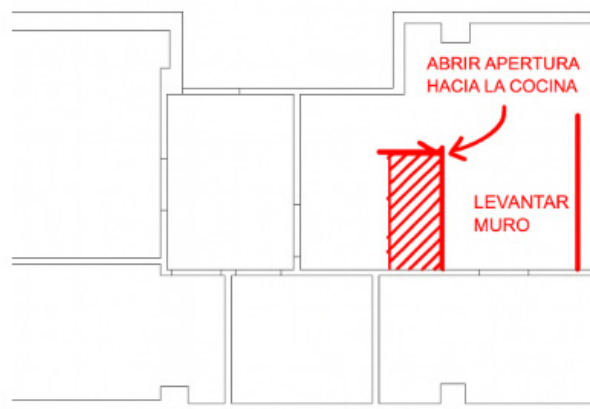


Figura 5.3 Plano de reforma propuesto por GPT-5

La limitación más significativa, y la principal justificación de este TFM, se reveló al solicitarle que **identificara y etiquetara las estancias del plano**. El LLM no pudo hacerlo de forma autónoma. Su respuesta fue pedirle al usuario que proporcionara las etiquetas, evidenciando que no había realizado una **segmentación semántica** de la imagen. El modelo no "vio" ni "entendió" el plano a nivel de píxel, sino que solo "razonó" sobre él a nivel textual, un proceso insuficiente para una tarea que depende de la precisión geométrica.

puedes marcarme en el plano qué es cada estancia?

Perfecto 🙌, lo hacemos. Para marcarte cada estancia necesito que me confirmes algunas cosas antes, porque en el plano en bruto no aparecen etiquetas:

1. 🚪 **Puerta de entrada:** ¿Dónde está la entrada principal? (Así puedo orientar el resto).
2. 🍳 **Cocina:** ¿Es la que está en la parte derecha, donde hicimos la apertura al salón?
3. 🚽 **Baño:** ¿Dónde está situado?
4. 🛏 **Dormitorios:** ¿Cuántos hay y cuáles serían?
5. 🛋 **Salón-comedor:** ¿Dónde lo prefieres ubicar?

Con esos datos puedo marcar directamente en el plano **salón, cocina, baño, dormitorios, distribuidor** y cualquier otra estancia.

Sí, puedo hacer una **propuesta lógica** de distribución basándome en la forma del plano, la posición de puertas y proporciones. Evidentemente, sin saber usos reales ni instalaciones es una hipótesis, pero bastante aproximada.

 Observando tu plano:

- La **entrada** parece estar en la parte inferior central (hay dos accesos pequeños que parecen puertas, típicos en recibidores).
- La **pieza más grande y rectangular (parte derecha)** encaja como **salón-comedor**.
- La estancia justo al lado (arriba del salón, a la derecha del centro) sería la **cocina** (donde antes marcamos la apertura).
- En la **parte inferior** hay tres estancias más pequeñas, que encajan como **dormitorios** (dos medianos y uno más pequeño).
- En el **centro** queda un espacio pequeño y cerrado, que probablemente es el **baño**.
- A la izquierda hay otra estancia amplia, que puede funcionar como **dormitorio principal** o incluso un **segundo salón**.

 Distribución probable:

- **Derecha** → Salón-comedor.
- **Centro-derecha** → Cocina.
- **Centro** → Baño.
- **Inferior** (tres habitaciones) → Dormitorios.
- **Izquierda** → Dormitorio principal / Estudio.



Figura 5.4 Conversación con GPT-5

La última respuesta del LLM es la prueba definitiva de esta limitación. El modelo usa su poderosa capacidad de razonamiento para **hipotetizar** sobre la distribución del plano basándose en una descripción textual (“la pieza más grande”, “tres estancias más pequeñas”). Es una deducción lógica, pero **no es un análisis visual**. El modelo deduce lo que *debería* ser cada estancia, pero no realiza un correcto análisis gráfico del plano.

Finalmente, cuando se le pide que lo dibuje, genera una imagen donde las estancias no se corresponden con la realidad y donde tampoco se detectan las ventanas ni las puertas. De esta manera, se confirma que **la segmentación no ha sido correcta** y, por tanto, **la propuesta del LLM no puede ser realista**.

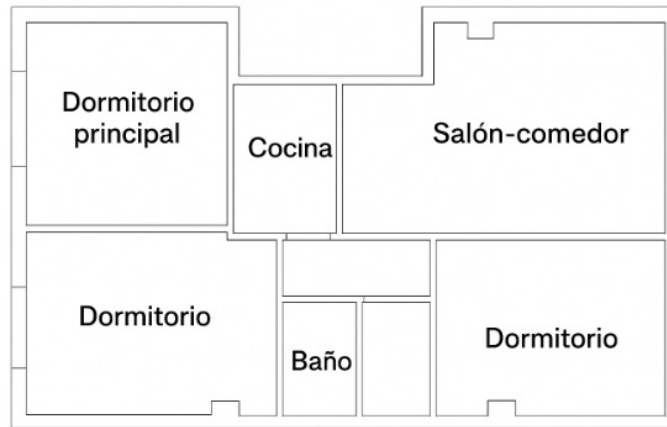


Figura 5.5 Plano segmentado propuesto por GPT-5

En conclusión, este experimento cualitativo demuestra que, a pesar de sus impresionantes capacidades de lenguaje y razonamiento, los LLMs generalistas carecen de las habilidades fundamentales para la tarea de reforma arquitectónica:

1. **Segmentación semántica precisa:** No pueden identificar y clasificar cada píxel de un plano para comprender su distribución real (véase Figura 5.5).
2. **Salida gráfica y estructurada:** No pueden generar una salida que sea a la vez visualmente precisa y, al mismo tiempo, un archivo estructurado y analizable como el JSON de este TFM (véase Figura 5.3).

Este contraste refuerza el valor de un *pipeline* especializado. Al ser modular y estar optimizado para tareas específicas (segmentación, generación de descripciones, motor de reglas), el sistema desarrollado en este TFM ofrece una solución **más confiable, precisa y reproducible** que la proporcionada por un LLM generalista.

El pipeline diseñado no compite directamente con LLM multimodales de propósito general. Estos últimos pueden describir un plano en términos generales, pero carecen de precisión normativa y de interpretación geométrica robusta. El presente trabajo cubre ese vacío, aunque con un alcance limitado.

6. Conclusiones

Este Trabajo de Fin de Máster ha logrado con éxito el objetivo de desarrollar un **sistema experimental y multimodal** que integra distintas técnicas de inteligencia artificial para abordar uno de los desafíos más persistentes en el sector de la arquitectura: la generación de propuestas de reforma a partir de planos no estructurados. El *pipeline* desarrollado ha demostrado que es posible automatizar de manera precisa y coherente una serie de tareas que, tradicionalmente, dependen del análisis y el juicio manual de un profesional.

Los principales logros alcanzados, que en conjunto validan la metodología propuesta, pueden resumirse en los siguientes puntos clave:

Logros Fundamentales del Pipeline

1. **Validez y simplicidad del dataset empleado:** Uno de los resultados más relevantes del trabajo es haber demostrado que el dataset preparado constituye una base sólida para entrenar modelos de segmentación de planos arquitectónicos. El grafismo simplificado, centrado únicamente en perímetros y espesores de muros, se ha mostrado suficiente para que el modelo aprenda a identificar estancias y elementos constructivos principales sin necesidad de información adicional como mobiliario. Esta simplicidad no solo facilita la tarea de segmentación, sino que también reduce de manera significativa el preprocesamiento requerido, lo que convierte al pipeline en un sistema más ágil, reproducible y fácil de implementar en la práctica profesional.
2. **Segmentación semántica precisa como base:** Se ha establecido un **punto de partida sólido** mediante un modelo de segmentación de última generación (SegFormer), logrando **IoUs elevadas** (superiores a 0.90) en la identificación de estancias principales y elementos constructivos. Este logro es crucial, ya que dota al sistema de una **comprensión visual a nivel de píxel**, una capacidad de análisis que distingue el enfoque de este TFM de los modelos de lenguaje generalistas, que no pueden realizar una interpretación tan granular.
3. **El puente semántico de la imagen al texto:** Para superar la ausencia de datasets de descripciones arquitectónicas reales, se diseñó una solución híbrida basada en un **módulo determinista y un modelo de lenguaje**. El componente determinista calcula las superficies exactas y el número de estancias a partir de la segmentación de la Fase 1, generando una representación estructurada fiable. A partir de esta información, un modelo de lenguaje (LLM) genera descripciones naturales y legibles para el usuario. Este

enfoque permite mantener la precisión geométrica y, al mismo tiempo, enriquecer la salida textual con una narrativa más fluida, sin necesidad de entrenar un encoder-decoder completo sobre un corpus sintético de baja calidad.

4. **Validación normativa integrada:** mediante un sistema de *retrieval-augmented generation (RAG)* sobre documentos oficiales (CTE y DC-09), el pipeline incorpora tanto reglas duras (p. ej., superficies mínimas de estancias según DC-09) como reglas blandas (recomendaciones del CTE sobre ventilación y accesibilidad). Este aspecto supone un avance notable respecto a trabajos previos, acercando el prototipo a la práctica profesional.

La validación normativa constituye una de las principales aportaciones diferenciales de este trabajo, abriendo la puerta a un sistema que no solo genera propuestas geométricamente coherentes, sino también viables desde el punto de vista legal.

5. **Un motor de decisión lógico y fiable:** Se ha implementado un módulo de reglas deterministas **capaz de interpretar intenciones de reforma** y validarlas geométricamente sobre el plano. Este motor no se limita a generar texto, sino que traduce una intención concreta —por ejemplo, abrir la cocina al salón— en acciones estructuradas dentro de un archivo JSON. Además, descarta automáticamente propuestas inviables (como la ausencia de contigüidad entre estancias o la interferencia con ventanas), aportando una capa de fiabilidad crítica al sistema. Esta validación lógica asegura que las salidas no sean únicamente narrativas, sino también consistentes desde un punto de vista geométrico y arquitectónico.
6. **Visualización como herramienta de comunicación:** La fase final del *pipeline* ha probado que es posible **traducir los datos técnicos en una visualización y narrativa accesibles**. La superposición de gráficos sobre el plano original no es solo una función estética, sino una **herramienta de comunicación** fundamental que permite a los profesionales y a los clientes comprender de un vistazo la propuesta de reforma.

A nivel cuantitativo, la validación confirmó que el sistema opera de forma **coherente y reproducible**, aunque condicionado por la geometría de los planos y la resolución de entrada. Los 1.366 ejemplos evaluados en la intención *abrir cocina al salón* mostraron que solo en un 22 % de los casos se daban las condiciones para ejecutar la apertura, y en los 304 ejemplos aceptados la alineación media con la frontera se redujo a un 6,9 %. Estos resultados ponen de relieve la **solidez del motor** para descartar escenarios no viables y, al mismo tiempo, señalan el **margen de mejora en la precisión geométrica de las propuestas**.

El valor del enfoque especializado

En un panorama tecnológico donde los modelos de IA se vuelven cada vez más generalistas, este TFM aporta una **sólida justificación a la relevancia de los *pipelines* especializados**. La demostración realizada en la discusión, al contrastar las capacidades del sistema con un LLM líder, ha evidenciado que el enfoque modular y optimizado para una tarea específica es **claramente superior** en términos de precisión, fiabilidad y capacidad de generar una salida estructurada y accionable. Mientras que un LLM puede razonar y generar texto plausible sobre un plano, el sistema desarrollado en este trabajo realmente lo "entiende" y actúa sobre él con una lógica de dominio específica.

Hacia el futuro de la arquitectura inteligente

Este proyecto, aunque experimental, sienta las bases para el desarrollo de herramientas de asistencia mucho más avanzadas en el sector de la arquitectura. Las líneas de investigación futuras se centran en escalar el sistema para:

- **Ampliar la variedad de reformas**, permitiendo una mayor exploración del diseño.
- **Adaptar la integración de normativas arquitectónicas** para garantizar que las propuestas sean legalmente viables y se adecúen a las particularidades de cada Comunidad Autónoma.
- **Refinar la visualización**, conectando el sistema con entornos 3D y flujos de trabajo profesionales de **BIM**.

En conclusión, este Trabajo de Fin de Máster no es solo la culminación de un proyecto, sino el **punto de partida** para una nueva generación de herramientas inteligentes que no buscan reemplazar al arquitecto, sino **potenciar su creatividad y eficiencia**, liberándolo de las tareas más repetitivas para que pueda concentrarse en la innovación y el diseño de espacios que mejoran la calidad de vida de las personas.

7. Líneas futuras

El presente trabajo sienta una base sólida para el desarrollo de herramientas de asistencia mucho más sofisticadas en el campo de la arquitectura. El sistema, en su estado actual, es un prototipo experimental que demuestra la viabilidad de un *pipeline* multimodal. Sin embargo, las siguientes líneas de investigación representan un camino claro y estructurado para evolucionar el sistema hacia una herramienta de uso profesional, resolviendo las limitaciones identificadas y ampliando su funcionalidad.

Ampliación de intenciones de reforma

Una de las limitaciones más evidentes del sistema actual es su reducido conjunto de intenciones de reforma. La primera y más importante línea de trabajo es ampliar el motor de decisión para incluir un catálogo más diverso de acciones. Esto implicaría definir nuevas reglas de coherencia para tareas más complejas, como la **creación de un nuevo baño**, la **ampliación de dormitorios** mediante la integración de un pasillo o el **cambio de la distribución de la zona de día y noche**. Este enfoque no solo aumentaría la utilidad del sistema, sino que también permitiría al arquitecto explorar una gama más amplia de soluciones de diseño en las etapas iniciales de un proyecto.

Generación automática de planos reformados

La fase de visualización actual se basa en la superposición de un *overlay* gráfico. El siguiente paso lógico es evolucionar esta fase para **producir un nuevo plano vectorial completo** con la distribución actualizada. Esto implicaría una serie de pasos técnicos más complejos, como la eliminación de los vectores del tabique a demoler y la generación de nuevos polígonos para las estancias unificadas. El resultado final sería un archivo de salida (.DXF o .DWG) que podría ser editado directamente en un *software* profesional como AutoCAD o Revit, integrando el sistema en un flujo de trabajo de diseño profesional.

Adaptación normativa por comunidades autónomas

Aunque se ha integrado el CTE y la DC-09 de Valencia, un reto futuro consiste en extender el sistema a las distintas normativas autonómicas, de manera que el motor de decisión pueda adaptarse al contexto regulatorio específico de cada territorio.

En conjunto, estas líneas de investigación futura consolidan el camino hacia un asistente arquitectónico integral, capaz de combinar precisión geométrica, adecuación normativa y personalización en la interacción con el usuario.

Entrenamiento con datasets reales de reformas

Las descripciones y propuestas del sistema están actualmente limitadas por el uso de un *dataset* sintético. La calidad y naturalidad del lenguaje, así como la coherencia de las propuestas, podrían mejorarse significativamente al entrenar el sistema con **casos reales de reformas de viviendas**. La recopilación de datos de estudios de arquitectura o de plataformas inmobiliarias que documenten cambios de distribución permitiría al sistema aprender patrones de diseño auténticos, lo que resultaría en propuestas más realistas y creativas que las generadas por un motor de reglas fijo.

Interactividad con el usuario

Finalmente, una línea de investigación clave es la evolución del sistema hacia un modelo interactivo. En lugar de procesar una intención de reforma predefinida, el sistema podría entablar un **diálogo con el usuario** para entender sus preferencias, prioridades y limitaciones. Mediante un *chatbot* o una interfaz conversacional, la IA podría hacer preguntas como: "¿Prefieres un espacio abierto o estancias separadas?", "¿Es más importante la luz natural o el aislamiento acústico?" o "¿Cuántos dormitorios necesitas?". Esto convertiría la herramienta en un verdadero **asistente de diseño personalizado**, capaz de ofrecer soluciones que se adapten de forma más precisa a las necesidades y al estilo de vida de cada cliente.

8. Bibliografía

1. Fundamentos teóricos y académicos

Chaillou, S. (2020). *Generating Residential Floor Plans with a Latent-Space GAN*. [Tesis de Máster]. École Polytechnique Fédérale de Lausanne (EPFL).

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). *Encoder-decoder with atrous separable convolution for semantic image segmentation*. En *European Conference on Computer Vision (ECCV)*.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Yianilos, P. (2022). *Masked-attention Mask Transformer for Universal Image Segmentation*. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, R., Huang, Z., Liu, S., Zhang, Z., Xu, G., & Chu, X. (2020). *Graph2Plan: Learning Floorplan Generation from Layout Graphs*. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8339–8348).

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., & Koltun, V. (2011). *Interactive Furniture Layout Using Interior Design Guidelines*. *ACM Transactions on Graphics (TOG)*, 30(4), 87.

Nauata, N., et al. (2020). *House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation*. En *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. En *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 234–241).

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*. En *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 12077–12090.

Zhou, J., et al. (2019). *DeepFloorPlan: Automatic Generation of Floorplans from a Single RGB Image*. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

2. Modelos multimodales y aplicaciones del estado del arte

Li, J., Li, D., Xiong, C., & Hoi, S. C. (2022). *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. *arXiv preprint arXiv:2201.12086*.

Li, J., et al. (2023). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. *arXiv preprint arXiv:2301.12597*.

Microsoft. (2023). *Kosmos-1: A Large Language Model for Multimodal Reasoning*. Recuperado de <https://www.microsoft.com/en-us/research/publication/kosmos-1-a-large-language-model-for-multimodal-reasoning/>

DeepMind. (2022). *Flamingo: A Visual Language Model for Few-Shot Learning*. Recuperado de <https://www.deepmind.com/publications/flamingo-a-visual-language-model-for-few-shot-learning>

Radford, A., et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. En *International Conference on Machine Learning (ICML)*.

Sánchez, F. C., & Rueda, D. E. M. (2022). *A Review of Artificial Intelligence in Enhancing Architectural Design Efficiency*. *Applied Sciences*, 12(3), 1476.

3. Datasets de planos

Kalervo, A., et al. (2019). *CubiCasa5K: A Dataset and an Improved Multi-task Model for Floorplan Image Analysis*. *arXiv preprint arXiv:1908.05155*.

van Engelenburg, L., Giger, D., Kneubuehler, T., & Mathis, C. (2024). *Modified Swiss Dwellings: A Large-Scale Dataset of Annotated Floor Plans for Architectural Research*. *Scientific Data*, 11, 123.

Wu, Q., et al. (2019). *Data-Driven Interior Plan Generation for Residential Buildings (RPLAN)*. *ACM Transactions on Graphics*, 38(6), 1–12.

4. Aplicaciones comerciales

Maket.ai. (2024). *AI-Powered Architectural Design*. Recuperado de <https://maket.ai>

Architectures. (2024). *IA para Arquitectura: Diseña edificios con IA*. Recuperado de <https://architectures.com/es/>

MagicPlan. (2024). *MagicPlan* [Software]. Recuperado de <https://www.magicplan.app/>

Planner 5D. (2024). *Planner 5D* [Software]. Recuperado de <https://planner5d.com/>

5. Herramientas de desarrollo

Hugging Face. (2024). *Transformers* [Librería de software]. Recuperado de <https://huggingface.co/docs/transformers>

Hugging Face. (2024). *SegFormer* [Modelo pre-entrenado]. Recuperado de https://huggingface.co/docs/transformers/model_doc/segformer

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. En *Proceedings of the 9th Python in Science Conference (SciPy 2010)*.

OpenAI. (2024). *ChatGPT* [Software]. Recuperado de <https://chat.openai.com>

Google DeepMind. (2024). *Gemini* [Software]. Recuperado de <https://deepmind.google>

6. Normativa

Ministerio de Transportes, Movilidad y Agenda Urbana. (2019). *Código Técnico de la Edificación (CTE)*. Recuperado de <https://www.codigotecnico.org>

Ajuntament de València. (2009). *Ordenança d'Edificació DC-09*. Recuperado de <https://www.valencia.es>