SISMOAI: DETECCIÓN, INTERPRETACIÓN Y REPORTE DE FALLAS MEDIANTE IMÁGENES SÍSMICAS

Francisco Acosta[1], Alyce Enriquez[1], Dayana Gómez[1], Michael Reinoso[1], Alan Zambrano[1], Christian Mejia[1] and Victor Collaguazo[1]

[1] *Universidad Central del Ecuador, Facultad de Ingeniería en Geología, Minas, Petróleos y Ambiental, Quito, Ecuador*

[1] {faacosta, adenriquezi, degomeze1, mjreinoso, ajzambranob1,cimejia,vfcollaguazo}@uce.edu.ec

Reception date of the manuscript:
Acceptance date of the manuscript:
Publication date:

Abstract— La identificación de fallas sísmicas es vital para caracterizar reservorios de hidrocarburos, pero la interpretación manual es cognitivamente exhaustiva y subjetiva. Este estudio busca automatizar la detección de fallas estructurales y generar reportes técnicos mediante inteligencia artificial. Metodológicamente, se depuró un conjunto de datos de la Universidad de Harvard, resultando en 1,050 imágenes sísmicas con máscaras binarias dilatadas y validadas. La visión computacional se abordó con la arquitectura YOLOv8-seg, aplicando aprendizaje por transferencia y ajuste fino durante 80 épocas. Las inferencias gráficas se acoplaron con Modelos de Lenguaje Grande (GPT-4o y Gemini) mediante ingeniería de prompts para la interpretación tectono-estratigráfica y redacción documental. El modelo alcanzó una precisión de 0.72, exhaustividad de 0.702 y mAP50 de 0.715, delineando geometrías complejas sin sobreajuste estadístico. Se concluye que integrar redes convolucionales con agentes generativos en una plataforma web reduce los tiempos de procesamiento, brindando a la industria exploratoria una herramienta objetiva que mitiga la incertidumbre geocientífica.

Keywords—Fallas, Inteligencia Artificial, Machine Learning, Reporte.

1. INTRODUCCIÓN

En distintas ramas de las geociencias que se centran en la caracterización del subsuelo, como son la exploración de hidrocarburos y minerales, la geotecnia, la gestión de riesgos geológicos y el almacenamiento geológico de CO_2, es crucial interpretar fallas [1, 2]. En particular, en la exploración hidrocarburífera, las fallas regulan la migración, el aprisionamiento y la acumulación de fluidos. Por consiguiente, identificarlas adecuadamente disminuye la incertidumbre vinculada al análisis estructural de los reservorios y a la determinación de prospectos exploratorios [1, 3].

Tradicionalmente, el proceso de interpretación sísmica de fallas incluye una serie de pasos que abarca la exploración minuciosa de líneas sísmicas, la evaluación de atributos estructurales como la curvatura y orientación, la identificación manual de discontinuidades, el mapeo estructural y la elaboración de modelos geológicos tridimensionales [3]. A pesar de que este método ha mostrado ser confiable durante decenios, tiene limitaciones significativas vinculadas con el alto tiempo de procesamiento, el esfuerzo necesario y la subjetividad inherente al juicio del intérprete, sobre todo cuando se maneja una cantidad considerable de información sísmica. En estas situaciones, la integración y validación de diversas interpretaciones aumenta de manera significativa los gastos y el tiempo requerido para el análisis [4, 5].

El aumento exponencial de los datos sísmicos ha promovido la creación de métodos fundamentados en inteligencia artificial que pueden automatizar algunos segmentos del proceso de interpretación sin poner en riesgo la calidad de los resultados. En particular, el aprendizaje profundo ha mostrado una notable capacidad para detectar patrones complejos en imágenes y realizar tareas de segmentación y detección con precisión y eficiencia computacional alta [6, 7]. En el contexto de estas técnicas, las redes neuronales convolucionales han logrado desempeños destacados en la detección automática de fallas geológicas en imágenes sísmicas. Esto ha permitido una disminución considerable en los periodos de interpretación y ha dado hallazgos más consistentes y reproducibles [5].

En este marco, el propósito general de esta investigación es emplear métodos de inteligencia artificial para identificar e interpretar automáticamente las fallas en las líneas sísmicas. Para lograr este objetivo, se establecen tres metas específicas: (i) reunir un conjunto de imágenes sísmicas con la clasificación y etiquetado adecuados de fallas geológicas; (ii) poner en funcionamiento un modelo de aprendizaje profundo para detectar, segmentar e interpretar automáticamente las fallas; y (iii) crear una aplicación web pública que haga posible la localización automática de fallas y la elaboración de informes técnicos.

La metodología que se propone consta de cuatro fases fundamentales. En primer lugar, se creó y refinó un conjunto de datos tomando como base el *Harvard Seismic Dataset* [8]. Después, se utilizó un modelo de visión por computador, fundamentado en la arquitectura YOLOv8-Seg. El modelo que se eligió para el modelo de visión fue la arquitectura YOLOv8-Seg, ésta se basa en la necesidad de segmentar imágenes y se que logre un equilibrio entre eficiencia computacional y precisión elevada. Para especializar la red en el reconocimiento y segmentación de fallas geológicas, se usaron el aprendizaje por transferencia (*transfer learning*) y el ajuste fino (*fine tuning*) [9]. Se incorporaron modelos de lenguaje de gran escala (LLM), en particular Gemini y ChatGPT, en la tercera fase con el objetivo de generar automáticamente informes técnicos de interpretación estructural [10, 11]. Se integró todo el flujo de procesamiento en una aplicación web creada en Streamlit, lo que posibilitó la automatización de la detección, comprensión y documentación de los resultados.

Los resultados obtenidos brindan una herramienta que tiene como objetivo reducir el tiempo y el esfuerzo necesarios para interpretar de forma estructural la información sísmica; además, contribuyen a minimizar la subjetividad del proceso mediante la estandarización de los criterios de análisis. La aplicación creada, que recibe imágenes sísmicas como entrada, detecta y segmenta automáticamente las fallas existentes y produce un informe técnico que se puede exportar en formatos PDF y DOCX. Este estudio sugiere un proceso de trabajo que combina visión por computadora y modelos lingüísticos para respaldar el proceso de toma de decisiones del geocientífico. El sistema reduce la subjetividad requiere evidencia experimental que compare distintos intérpretes humanos. Por lo tanto, el uso de la inteligencia artificial en este trabajo busca apoyar y complementar el criterio del intérprete, más no reemplazarlo.

2. ESTADO DEL ARTE

La interpretación de fallas en imágenes sísmicas ha tenido un gran avance en las últimas décadas gracias a la inteligencia artificial, pasando de ser un proceso manual a ser uno automatizado. Las investigaciones expuestas en la Tabla 1 y 2 muestran cómo la interpretación sísmica ha evolucionado desde el uso de métodos estadísticos tradicionales hacia arquitecturas avanzadas de redes neuronales.

TABLA 1: EVOLUCIÓN DE LOS MÉTODOS PARA LA DETECCIÓN E INTERPRETACIÓN DE FALLAS SÍSMICAS.

Año	Autor(es)	Datos	Método	Resultados principales
1995	Bahorich y Farmer [12]	Sintético 3D y datos de campo	Semiautomático	Introduce el atributo de coherencia para resaltar discontinuidades sísmicas.
1999	Gersztenkorn y Marfurt [13]	Sísmica 3D (Cuenca Anadarko)	Semiautomático	Desarrolla la medida de coherencia basada en eigenestructura para mejorar la resolución estructural.
2002	Randen et al. [14]	Volúmenes sísmicos 3D del Mar del Norte	Semiautomático	Aplica orientación 3D y atributos de buzamiento y azimut para caracterizar zonas estructuralmente complejas.
2005	Pedersen et al. [15]	Volumen sísmico marino 3D	Semiautomático	Introduce Ant Tracking para la extracción automática de redes de fallas.
2007	Donias et al. [16]	Sísmica 3D	Semiautomático	Propone atributos direccionales robustos para mejorar la continuidad de las fallas.
2008	Atencio et al. [17]	Sísmica 2D y 3D	Semiautomático	Implementa visualización OVS para facilitar la interpretación estructural.
2009	Vargas et al. [18]	Datos sísmicos de campo	Manual	Destaca la importancia de reducir la subjetividad durante la interpretación estructural.
2009	Hale [19]	Sísmica 3D	Semiautomático	Introduce Fault Slips para estimar desplazamientos y continuidad de fallas.
2014	Hale [20]	Sísmica 3D	Semiautomático	Desarrolla Fault Likelihood mediante filtros orientados para detectar discontinuidades.
2017	Kolyukhin et al. [21]	Sísmica 3D	Semiautomático	Modela probabilísticamente zonas de falla facilitando posteriores procesos automáticos.
2018	Chopra y Marfurt [3]	Sísmica 3D	Manual	Resume las mejores prácticas para interpretación estructural basada en atributos sísmicos.
2019	Wu et al. [4]	Datos sintéticos y sísmica real	Automático	Implementa CNN para detectar fallas automáticamente reduciendo la intervención humana.
2019	Alaudah et al. [5]	Volúmenes sísmicos	Automático	Demuestra la eficacia de redes convolucionales profundas para segmentación automática de fallas.
2020	Wu et al. [22]	Sísmica 3D	Automático	Utiliza Deep Learning para mejorar la continuidad espacial de las fallas detectadas.
2021	An et al. [23]	Imágenes de líneas sísmicas	Automático	Obtiene resultados comparables a intérpretes expertos mediante CNN profundas.
2021	Di et al. [24]	Sísmica sintética y real	Automático	Integra aprendizaje profundo para segmentar fallas complejas en presencia de ruido.
2022	Li et al. [25]	Sísmica real	Automático	Emplea 2.5D Attention U-Net para incrementar la continuidad espacial de las fallas.
2022	OpenAI [26]	Modelos de lenguaje	Automático	Populariza los modelos conversacionales aplicados a documentación científica y técnica.

TABLA 2: EVOLUCIÓN DE LOS MÉTODOS PARA LA DETECCIÓN E INTERPRETACIÓN DE FALLAS SÍSMICAS.

Año	Autor(es)	Datos	Método	Resultados principales
2023	Corbetta [27]	Sísmica	Automático	Demuestra la utilidad de U-Net para reducir la subjetividad en interpretación sísmica.
2023	Jocher et al. [9]	Segmentación de imágenes	Automático	Presenta YOLOv8-Seg para detección y segmentación de objetos en tiempo real.
2024	Di et al. [28]	Sísmica 3D	Automático	Mejora la detección de fallas mediante arquitecturas profundas híbridas.
2024	Wang et al. [29]	Sísmica	Automático	Incrementa la precisión en la segmentación utilizando mecanismos de atención.
2024	Liu et al. [30]	Sísmica	Automático	Propone modelos robustos frente al ruido y estructuras complejas.
2024	Zhao et al. [31]	Sísmica 3D	Automático	Optimiza la extracción automática de fallas mediante aprendizaje profundo.
2025	Zhang et al. [32]	Sísmica 3D	Automático	SEU-Net incrementa la robustez frente al ruido gaussiano.
2025	Chen et al. [33]	Sísmica	Automático	Integra atención multiescala para mejorar la delimitación de fallas pequeñas.
2025	Sun et al. [34]	Sísmica	Automático	Mejora la segmentación mediante Transformers híbridos.
2025	Xu et al. [35]	Sísmica	Automático	Combina CNN y Transformers para interpretar estructuras complejas.

Aunque en los últimos treinta años se han realizado avances significativos en la detección de fallas sísmicas, el dilema de la interpretación estructural no ha sido resuelto completamente con ninguna de las aproximaciones presentadas. Los métodos manuales iniciales dependen casi por completo de la experiencia del intérprete, lo cual genera un alto grado de subjetividad y requiere tiempos prolongados para el análisis, en particular cuando se trata de grandes cantidades de datos sísmicos. Los métodos semiautomáticos que se fundamentan en atributos sísmicos, como la curvatura, la coherencia y la orientación estructural, han mejorado notablemente la detección de discontinuidades; no obstante, su rendimiento sigue dependiendo de factores como el ruido, la calidad de imagen, la correcta elección de parámetros y el hecho de que el experto necesita hacer una validación e interpretación manual. Más recientemente, se ha evidenciado que los métodos automáticos fundamentados en aprendizaje profundo son más capaces de identificar patrones complejos y segmentar fallas con gran exactitud. Sin embargo, estos modelos tienen limitaciones vinculadas con la falta de conjuntos de datos de calidad superior, la poca capacidad para generalizar entre distintos estilos tectónicos y cuencas, la sensibilidad ante dominios geológicos no incluidos en el entrenamiento y el escaso entendimiento que se tiene de sus decisiones. La mayoría de los trabajos revisados solo incluyen la detección o segmentación de las fallas, sin añadir un análisis geológico para contextualizar las estructuras detectadas ni una explicación técnica que haga más sencilla su utilización en la exploración de hidrocarburos. Estas limitaciones demuestran que la automatización de la interpretación sísmica todavía supone un reto sin resolver. Las investigaciones futuras deberían estar dirigidas a sistemas capaces de incorporar datos geológicos, estratigráficos y estructurales en el proceso de interpretación, en lugar de simplemente aumentar la exactitud de los modelos de visión por computadora. La posibilidad de detectar fallos, clasificarlos, comprender su contexto tectónico, reconocer estructuras relacionadas y elaborar informes técnicos con estándares geológicos sólidos se lograría al fusionar modelos de segmentación y modelos de lenguaje a gran escala (LLM), así como bases de conocimiento geocientífico y sistemas para interactuar con el experto.

La detección y la segmentación automática de fallas sísmicas y otras estructuras geológicas han sido realizadas con frecuencia mediante arquitecturas recientes, entre las que se encuentran: SegFormer, DeepLabV3+, Mask R-CNN, U-Net, Attention U-Net y modelos híbridos CNN-Transformer. Aunque U-Net y sus variantes tienen una precisión muy alta en la segmentación semántica, necesitan fases adicionales para separar las fallas individuales y normalmente tienen tiempos de inferencia más largos [36, 37]. Mask R-CNN tiene la capacidad de segmentar por instancias, lo que posibilita el reconocimiento de objetos individuales; sin embargo, al tener una arquitectura en dos etapas, aumenta el costo computacional [38].

DeepLabV3+, en cambio, mejora la extracción de características a través de convoluciones dilatadas, y SegFormer incluye mecanismos Transformer que capturan relaciones espaciales de amplio alcance. Por su parte, las arquitecturas híbridas CNN-Transformer proporcionan una representación más sólida del contexto geológico; sin embargo, necesitan un volumen más alto de datos y recursos computacionales para ser entrenadas [39, 40].

Para este estudio se eligió YOLOv8-seg, ya que incorpora detección y segmentación de instancias en una arquitectura de una sola etapa (single-stage), lo cual permite un balance positivo entre la rapidez de inferencia, la eficacia computacional y la precisión [9]. En aplicaciones web interactivas, donde la respuesta en tiempo real es un requisito fundamental, estas características son particularmente relevantes. De igual manera, la segmentación por instancias posibilita el reconocimiento y etiquetado individual de cada falla geológica, lo que permite al intérprete editarlas después y crear automáticamente informes técnicos.

A pesar de que arquitecturas híbridas CNN-Transformer, DeepLabV3+, Mask R-CNN o SegFormer pueden lograr rendimientos similares o incluso mejores en algunos conjuntos de datos, por lo general requieren una infraestructura computacional más exigente o un tiempo de procesamiento más alto [38, 39, 40]. Por ende, la arquitectura utilizada constituye una respuesta apropiada para incorporar en una única plataforma enfocada en la investigación de hidrocarburos las funciones de interpretación asistida, detección automática y generación de reportes.

3. METODOLOGÍA

La metodología, en primer lugar, incluye la recolección y depuración de un conjunto de datos (dataset) representativo de imágenes sísmicas que se obtiene de la Universidad de Harvard. Estas imágenes pasan por un exhaustivo preprocesamiento en línea y fuera de línea, cuyo objetivo es normalizar la resolución mediante el marco de trabajo Ultralytics, suavizar los valores atípicos y transformar las máscaras binarias de los errores a polígonos sólidos. Seguidamente, los pares de imágenes y sus anotaciones son validados visualmente de manera rigurosa por geólogos expertos, para garantizar que las máscaras poligonales se correspondan geométricamente con las trazas reales de las fallas y eliminar eventuales equivocaciones.

Después, se entrena un modelo que utiliza la arquitectura de redes neuronales convolucionales profundas YOLOv8-seg. Este algoritmo emplea métodos de aprendizaje por transferencia (fine-tuning) para detectar y segmentar automáticamente las fallas estructurales. Después de crear las máscaras de segmentación, se incorpora un módulo basado en Modelos de Lenguaje Grande (LLMs) multimodales, como Gemini y GPT-4o. Este sistema, a través de la puesta en práctica de ingeniería de prompts, genera una interpretación automatizada tectono-estratigráfica al procesar los resultados visuales que el modelo de segmentación produce.

En última instancia, todo este proceso de trabajo se concreta en una aplicación web interactiva creada en Streamlit. Esta plataforma posibilita que los usuarios suban secciones sísmicas, vean y modifiquen la imagen interpretada y descarguen un informe técnico geológico integral en PDF o Word automáticamente. La Figura 1 a continuación describe cada una de estas fases.

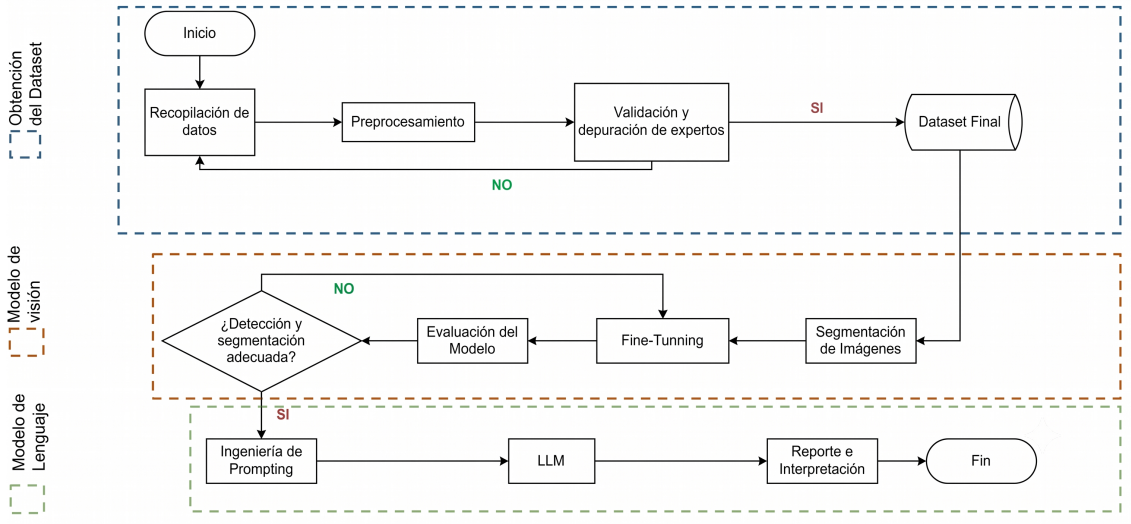


Figura. 1: Flujograma general de las etapas principales de la metodología

3.1. Obtención de Dataset

La sección muestra cómo se conformó el dataset de imágenes sísmicas, con la recopilación de los datos crudos, el respectivo preprocesamiento y la validación visual por parte de expertos geólogos para contar con información estandarizada y limpia.

3.1.1. Recopilación de datos

Las imágenes sísmicas fueron recopiladas a partir del repositorio de datos A Gigabyte Interpreted Seismic Dataset for Automatic Fault Recognition [1], elaborado por la Universidad de Harvard [8]. Este dataset base se presenta en formato de matrices tridimensionales con extensiones de archivo .npy y .npz, lo que permite guardar todo el volumen de información conservando la calidad de los datos. Esta estructura está definida por 1803 líneas transversales o crosslines, 1537 muestras de profundidad y 3174 líneas inlines. Cada uno de estos archivos de matrices tiene un tamaño aproximado de 4.02 GB y tiene resoluciones de 3174 x 1537 píxeles. Se necesita la librería NumPy en la plataforma Python para visualizar las imágenes en estas matrices. Además del conjunto anteriormente mencionado, se decidió recopilar 50 imágenes sísmicas adicionales extraídas de artículos científicos y trabajos de titulación, con el fin de tener un grupo de datos totalmente independiente para poner a prueba el modelo más adelante. A diferencia de los datos provenientes del dataset de Harvard, estas nuevas imágenes se obtuvieron directamente en formatos gráficos comunes como PNG y JPEG.

3.1.2. Preprocesamiento de dataset

El propósito principal de esta fase es preparar los datos originales obtenidos del conjunto de datos Thebe, perteneciente a la Universidad de Harvard, para que el modelo de visión por computadora se separa el flujo de trabajo en dos etapas para realizar un traspaso exitoso de las matrices originales a un formato que sea compatible con el aprendizaje profundo: una fase de preprocesamiento fuera de línea y otra en línea. Primero, el preprocesamiento fuera de línea se lleva a cabo una

[1] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YBYGBK

sola vez en el dataset para definir y preparar con precisión las variables de entrada y salida del modelo. Las variables de entrada, identificadas con la letra X, son las imágenes sísmicas originales que se han transformado a escala de grises. El acondicionamiento de estas imágenes iniciales empieza con una modificación estadística de la amplitud de la señal sísmica, que se reduce a un intervalo específico de ±3 desviaciones estándar en relación con la media. Luego, se lleva a cabo un escalado lineal para convertir esta matriz numérica en una imagen digital. Al convertir los datos sísmicos al formato PNG, las matrices de amplitud se convierten en imágenes rasterizadas a 8 bits en escala de grises. Para disminuir el impacto de valores atípicos y mantener el rango dinámico geológico relevante, las amplitudes se recortan inicialmente al intervalo de ±3 desviaciones estándar con respecto a la media. Después, se normalizan y escalan los valores linealmente entre 0 y 255, otorgándole a cada píxel un valor de intensidad que guarda proporción con la amplitud sísmica. La imagen final se redimensiona manteniendo su proporción hasta llegar a 1024 píxeles en el lado más largo. Se guarda en formato PNG, que tiene una compresión sin pérdidas y mantiene la información radiométrica, lo que permite entrenar modelos de visión por computadora sin inconvenientes (Figura 2).

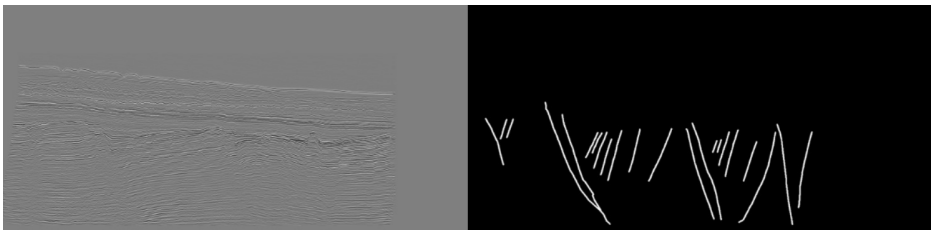


Figura. 2: Sísmica y máscara de falla, extraida de dataset.

Las interpretaciones originales que brindan los geólogos expertos se presentan como máscaras de fallas binarias extremadamente delgadas, a menudo restringidas a uno o dos píxeles de ancho. Ya que el modelo necesita ejemplos con un área física que se pueda medir para la tarea de segmentación, se lleva a cabo un procedimiento de engrosamiento usando una operación morfológica de dilatación (Figura 3). Cuando se aplica un kernel de 5×5 píxeles a la matriz, la falla original se transforma en una engrosada, lo que proporciona a la estructura el área bidimensional requerida para el entrenamiento. Cuando las fallas presentan este cuerpo bidimensional definido, el flujo computacional comienza a extraer sus límites externos mediante el algoritmo de Suzuki y Abe [41]. A causa de que los contornos resultantes pueden tener una densidad de puntos demasiado alta, estos polígonos se simplifican utilizando un método de aproximación poligonal para que su procesamiento en la memoria sea más eficiente. En este bloque de refinamiento también se lleva a cabo un paso fundamental llamado eliminación de manchas diminutas. Es frecuente que surjan errores visuales que no se corresponden con estructuras geológicas auténticas, ya sea en las anotaciones manuales originales o después de las dilataciones morfológicas. El algoritmo elimina las áreas aisladas con un tamaño menor a 60 píxeles cuadrados para filtrar el ruido digital, asegurándose de que la red neuronal se concentre solamente en errores válidos.

Figura. 3: Engrosamiento y manchas detectadas en preprocesamiento

La conversión de máscara a polígono se lleva a cabo automáticamente como la última etapa del procesamiento fuera de línea (Figura 4). Los contornos de las máscaras creadas por los geólogos se transforman al formato estructural requerido por la red neuronal, a través de la ejecución de un script en Python llamado yolo_prepare.py, evitando así que se tengan que hacer anotaciones manuales adicionales. En este procedimiento, las coordenadas absolutas de los vértices se convierten en valores relativos entre cero y uno. Así se crean las etiquetas estructuradas, que incluyen la clase correspondiente (por ejemplo, clase 0 o clase 1 para indicar la falla) y sus respectivas coordenadas normalizadas.

Figura. 4: Conversión de máscara a polígono.

La etapa del preprocesamiento en línea, que es realizada de manera automática y dinámica por el marco Ultralytics mientras se cargan los lotes de datos durante la fase de entrenamiento, es la segunda gran fase del acondicionamiento. Aunque las imágenes fueron normalizadas a 1024 píxeles en un principio, para funcionar adecuadamente la arquitectura escogida requiere un tamaño fijo de tensor. Cada imagen es adaptada a un tamaño estándar de 640 píxeles (imgsz = 640) durante este proceso. Se aplica la técnica de letterbox para adaptar la sección sísmica rectangular a un contenedor cuadrado de 640 × 640 píxeles sin distorsionar los rasgos geométricos de la estratigrafía. Para beneficiarse del aprendizaje por transferencia, es necesario realizar este ajuste dimensional. Cuando se introducen imágenes con una resolución precisa de 640 píxeles, el sistema se ajusta a los pesos preentrenados del modelo base que se adquirieron del conjunto de datos COCO. Estos fueron calibrados para trabajar a esa resolución. Todo este proceso asegura que los datos se encuentren limpios, estandarizados y geométricamente en consonancia con la estructura del subsuelo al llegar al modelo de inteligencia artificial (Figura 5).

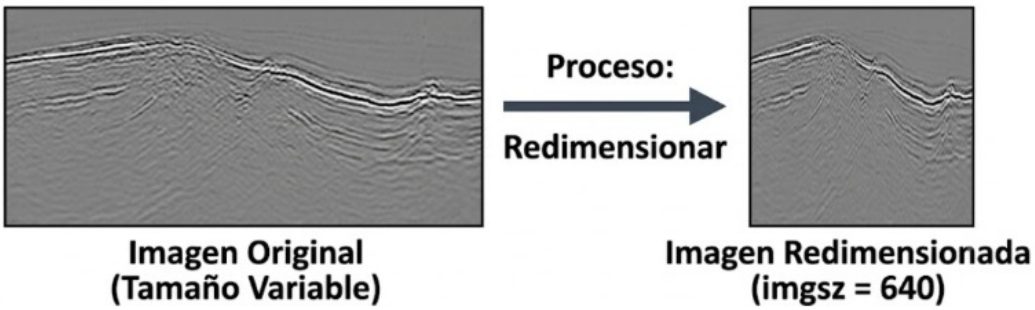


Figura. 5: Redimensionamiento de imágenes a 640 x 640 px.

3.1.3. Validación y Depuración de Expertos

Una vez finalizado el preprocesamiento del conjunto de datos, se realiza la fase de validación y depuración por expertos. En esta etapa, el sistema genera automáticamente pares de datos compuestos por una imagen sísmica de entrada y su correspondiente etiqueta de salida. El geólogo inspecciona visualmente una muestra representativa de los pares generados para verificar que las etiquetas coincidan con las estructuras geológicas observadas en las imágenes sísmicas. Para ello, superpone los polígonos vectoriales generados automáticamente sobre las secciones sísmicas correspondientes. La validación del conjunto de datos se desarrolla en tres etapas. Primero, se verifica la calidad radiométrica de las imágenes para asegurar que la conversión a 8 bits conserve el contraste de los reflectores y que el recorte estadístico no elimine información relevante. Segundo, se evalúa la precisión geométrica de las etiquetas para comprobar que los polígonos representen correctamente las fallas geológicas sin desplazamientos ni deformaciones. Finalmente, se revisa el filtrado espacial para confirmar que los umbrales eliminen únicamente ruido y artefactos, preservando las estructuras geológicas válidas (Figura 6).

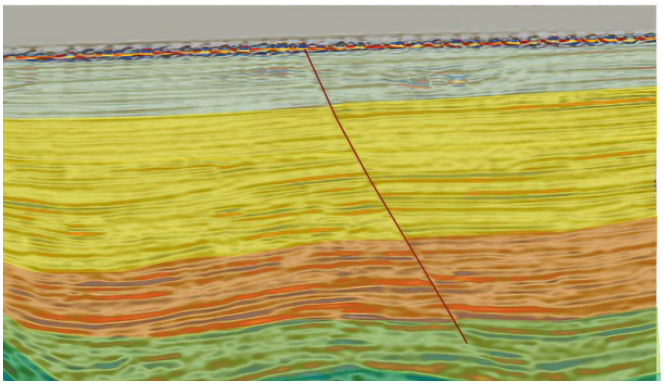


Figura. 6: Interpretación, validación y depuración de fallas en imágenes sísmicas por parte de expertos.

3.1.4. Dataset Final

Tras acabar los pasos de validación, estandarización y preprocesamiento, se estableció el conjunto de datos final que es utilizado para evaluar y entrenar el modelo de segmentación de fallas geológicas. El conjunto de datos está compuesto por 1,050 imágenes sísmicas bidimensionales de alta calidad. Este conjunto incluye 1000 imágenes obtenidas del *Thebe Dataset* de la Universidad de Harvard, las cuales fueron depuradas, normalizadas y anotadas antes de ser guardadas en formato JPG con el fin de equilibrar la eficiencia del almacenamiento y la calidad visual. Además, se añadieron 50 imágenes adquiridas de repositorios especializados y artículos científicos que fueron empleadas únicamente para examinar la habilidad del modelo para generalizar sobre contextos geológicos no empleados a lo largo del entrenamiento. Los parámetros del modelo se ajustaron usando el conjunto de entrenamiento, mientras que los conjuntos de validación y prueba hicieron posible la supervisión del proceso de aprendizaje y la evaluación del rendimiento final en datos no empleados durante el

entrenamiento. Se produjeron automáticamente las anotaciones de las fallas geológicas a través de un script creado en Python, que convirtió las máscaras de segmentación en archivos de texto con coordenadas poligonales normalizadas entre 0 y 1. Cada archivo mantiene el mismo nombre que la imagen que le corresponde e incorpora el identificador de la clase vinculada a los fallos geológicos, además de los vértices que describen el contorno de cada estructura. Esto asegura una correspondencia unívoca entre las etiquetas y las imágenes. Al final, se consolidaron las 1050 imágenes y sus archivos de anotación correspondientes en un único repositorio almacenado en Google Drive [2] , con el fin de facilitar la entrada a los pares imagen–etiqueta durante el procesamiento. El tamaño del conjunto completo de datos es de aproximadamente 1.52 GB, lo que constituye un repositorio estandarizado y preparado para ser empleado en las fases de entrenamiento, validación y evaluación del modelo.

3.1.5. Segmentación de Imágenes – Modelo YOLOv8s-seg

El algoritmo YOLOv8-seg, basado en redes neuronales convolucionales profundas de la familia YOLO ("You Only Look Once"), fue introducido por Jocher et al. (2023) [9] para abordar de manera conjunta la detección de objetos y la segmentación de instancias en tiempo real. Su propuesta se centra en extender el paradigma clásico de detección en una sola pasada (one-stage) con una rama de segmentación paralela, evitando así arquitecturas de dos etapas como Mask R-CNN que requieren procesamiento adicional independiente por cada objeto detectado. A diferencia de los enfoques de segmentación semántica pixel a pixel, YOLOv8-seg trata cada falla como una instancia individual la cual está delimitada por una caja y máscara propias, lo que permite identificar identificar, contar y diferencia estructuras contiguas sin fusionarlas en un único mapa binario. Una de las principales fortalezas de YOLOv8-seg es su rama de segmentación de tipo YOLACT, la cual genera para cada instancia detectada un conjunto de prototipos de máscara junto con sus respectivos coeficientes; la máscara final se obtiene combinando ambos mediante una multiplicación y suma ponderada, lo que permite producir máscaras de alta resolución sin el costo computacional de una rama de segmentación exclusiva por objeto. Gracias a este diseño, el modelo ha demostrado una gran capacidad para segmentar estructuras alargadas y de forma irregular en tiempo real, lo que favorece su adopción en diversas aplicaciones de visión por computadora, incluyendo las geociencias. En este trabajo se emplea YOLOv8s-seg con backbone CSPDarknet (modulo C2f) preentrenado en el conjunto de datos COCO siguiendo el esquema de transfer learning. La arquitectura se compone de tres bloques principales como se muestra en la Figura 7:.

• Backbone (CSPDarknet con módulos C2f): Extraer las características de la imagen de entrada mediante una red convolucional con conexiones tipo cross-stage partial (CSP), que reduce el costo computacional sin sacrificar la capacidad representativa. YOLOv8 sustituye el módulo C3 usando versiones previas por el módulo C2f (cross-stage partial bottleneck con dos convoluciones), lo que mejora el flujo de gradientes y la precisión sin aumentar tiempo de inferencia.

• Neck (Feature Pyramid Network/PAN): Combina los mapas de características de distintas escalas extraídos por el backbone, integrando información semántica de alto nivel con detalles espacial de bajo nivel, favoreciendo la detección de estructuras de distinto tamaño, es un paso clave en este estudio porque las fallas geológicas varían considerablemente en longitud y extensión vertical dentro de una misma sección.

• Head desacoplada y sin anclas(anchor-free): YOLOv8-seg emplea una cabeza de detección desacoplada y sin anclas, dividida en ramas separadas de clasificación y regresión de caja donde se añade una rama paralela de segmentación de mascara en cada escala de detección. Esto contrasta con versiones anteriores de YOLO basadas en anclas predefinidas y simplifica el ajuste del modelo a la geometría de las fallas, que no se ajusta bien a proporciones de caja fija.

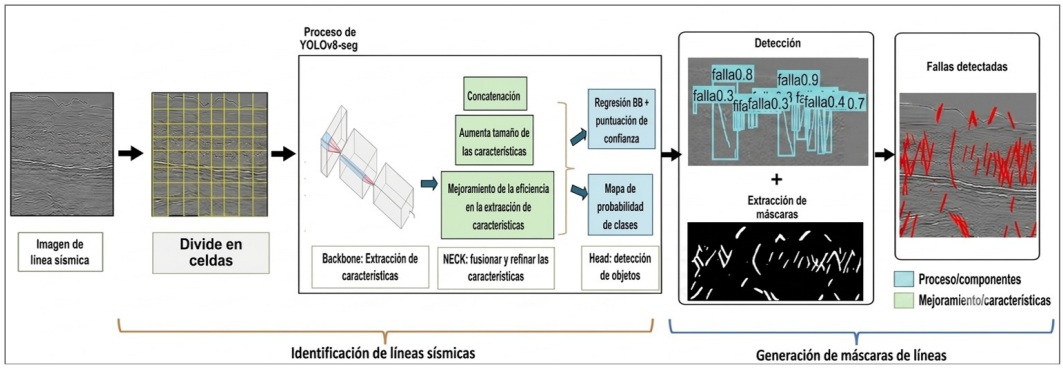


Figura. 7: Arquitectura de solución del modelo YOLOv8-seg empleado para la segmentación automática de falla.

4. EXPERIMENTACIÓN

La presente sección muestra la configuración de los algoritmos utilizados en el modelo de visión por computadora y la evaluación de la precisión durante el entrenamiento, además, se indica la ingeniería de prompting en el modelo de lenguaje para los reportes automatizados. El objetivo es automatizar la detección de fallas geológicas en imágenes sísmicas, para lo cual, se realizará el fine-tuning en el modelo de visión por computadora YOLOv8-seg, para utilizar un modelo de lenguaje

[2] https://drive.google.com/drive/folders/1rTNNP4PqPm-SbM6t7Wj_c3k1R6z85ZkA?usp=drive_link

natural como Gemini, ChatGPT, Copilot y DeepSeek en redacción de reportes automáticos. La hipótesis asume que el uso CNN e ingeniería de prompting reduce tiempos y subjetividad en la delimitación de fallas en imágenes sísmicas.

4.1. Plataforma computacional

El entrenamiento y la validación de la red neuronal de aprendizaje profundo se ejecutaron utilizando como entorno de desarrollo integrado el software Visual Studio Code. Esta elección metodológica se fundamentó en su capacidad para facilitar el procesamiento de datos pesados y la gestión de los scripts, priorizando en todo momento la eficiencia del trabajo computacional durante la estructuración del proyecto SismoAI.

A nivel de hardware, el entorno computacional local empleado para el diseño, desarrollo y las pruebas del modelo estuvo compuesto por el sistema operativo Linux Ubuntu, el cual proporciona estabilidad para flujos de trabajo de aprendizaje automático. El equipo integró el procesador AMD Ryzen 5 7235HS operando a la frecuencia de 3.20 GHz, acompañado de la unidad de procesamiento gráfico NVIDIA RTX 3050 equipada con 6 GB de memoria de video dedicada. Estas características técnicas resultaron fundamentales para la optimización del cálculo tensorial requerido por las redes convolucionales, permitiendo gestionar de manera eficiente el procesamiento masivo de los lotes del dataset sísmico. A diferencia de las estaciones de trabajo de alto rendimiento que utilizan aceleración por hardware a gran escala, la programación del código fuente, la estructuración de la interfaz interactiva y las pruebas preliminares locales se ejecutaron adaptándose meticulosamente a este sistema provisto de gráficos integrados NVIDIA. Esto obligó a optimizar al máximo la eficiencia de los algoritmos de carga y transformación matemática, incluso en las fases donde se operó sin el soporte nativo de procesamiento paralelo CUDA.

Finalmente, en este mismo entorno y utilizando las dependencias descritas, se programó y ejecutó de forma exitosa el script algorítmico denominado yolo_prepare.py. Este código informático fue el responsable directo de llevar a cabo el proceso automatizado de conversión, transformando las máscaras binarias de segmentación delineadas previamente en coordenadas poligonales relativas y estandarizadas en el rango continuo de 0 a 1, cumpliendo así con los estrictos requerimientos de entrada que exige el modelo de visión para iniciar su entrenamiento a lo largo de las 80 épocas.

4.1.1. Entorno de ejecución y parámetros de reproducibilidad

Para poder hacer reproducible este producto tanto los experimentos de segmentación y preprocesamiento, el flujo de trabajo se configuro de las siguientes maneras con especificaciones de software y hardware, por ejemplo:

TABLA 3: CONFIGURACIÓN DEL ENTORNO DE DESARROLLO Y DEL CONJUNTO DE DATOS UTILIZADOS EN ESTE ESTUDIO.

Elemento	**Características**
Lenguaje	Python 3.10.12.
Framework de Inteligencia Artificial	Ultralytics v8.1.0 (YOLOv8-seg).
Librerías principales	PyTorch 2.2.0+cu121, NumPy 1.26.4, OpenCV 4.9.0 y Streamlit 1.31.0.
Control de estocasticidad	Semilla aleatoria global (`seed = 42`) aplicada en PyTorch, CUDA y NumPy, garantizando la reproducibilidad en la partición del conjunto de datos y la inicialización de la red neuronal.
Configuración del dataset	1050 imágenes sísmicas en formato JPG con sus correspondientes archivos de anotación, distribuidas en 800 imágenes para entrenamiento, 100 para validación, 100 para prueba y 50 imágenes externas para evaluar la capacidad de generalización del modelo.

4.2. Modelo de visión

4.2.1. Entrenamiento del modelo de segmentación automática de fallas

Se llevó a cabo el entrenamiento utilizando un esquema de segmentación de instancias, que tenía en cuenta una sola clase de interés, la cual se correspondía con la falla geológica; todo lo demás de la imagen fue definido como fondo (background). Se elaboró, de antemano, un archivo de configuración .yaml que detalla la estructura de las carpetas, las rutas del conjunto de datos y la codificación de las clases. En esta configuración, la clase 0 representó el fondo y la clase 1 a la falla geológica. Después, el conjunto de datos fue dividido en tres subconjuntos: entrenamiento (train), validación (val) y prueba (test). Estos estaban compuestos por 800 imágenes para el primer grupo, 100 para el segundo y otros 100 para el último. Para ajustar los parámetros del modelo se empleó el conjunto de entrenamiento, y para supervisar el proceso de aprendizaje, se utilizó el conjunto de validación. El conjunto de prueba se utilizó para evaluar la actuación final del modelo.

TABLA 4: DISTRIBUCIÓN Y CANTIDAD DE IMÁGENES PARA EL DATASET.

Conjunto	Número de imágenes	Número de máscaras	Porcentaje
Train	400	400	80%
Val	50	50	10%
Test	50	50	10%
Total	**500**	**500**	**100%**

Los hiperparámetros del modelo fueron determinados antes de comenzar el entrenamiento. Estos incluyen la tasa de aprendizaje (learning rate), el tamaño del lote, la cantidad de épocas y las dimensiones de entrada de las imágenes. La Tabla 3 muestra la configuración de estos hiperparámetros.

TABLA 5: HIPERPARÁMETROS UTILIZADOS PARA EL ENTRENAMIENTO DEL MODELO YOLOV8-SEG.

Hiperparámetro	Descripción	Valor
image_size	Tamaño de imagen	640 × 640 px
channels	Número de canales de color	2 (8 bits)
epochs	Iteraciones del conjunto de datos	80
optimizer	Algoritmo de optimización	auto
lr0	Tasa de aprendizaje inicial	0.01
momentum	Momentum para el optimizador	0.937
weight_decay	Decaimiento de peso para regularización L2	0.0005
warmup_epochs	Épocas de calentamiento para el learning rate	3
warmup_momentum	Momentum durante el calentamiento	0.8
warmup_bias_lr	Tasa de aprendizaje para el bias durante el calentamiento	0.0
box	Peso de la pérdida del *bounding box*	7.5
cls	Peso de la pérdida de clasificación	0.5
cls_pw	Peso de la pérdida de clasificación positiva	0.0
iou	Umbral de IoU para la supresión no máxima (NMS)	0.7
hsv_h	Aumento de datos: variación del tono (*hue*)	0.015
hsv_s	Aumento de datos: variación de la saturación (*saturation*)	0.7
hsv_v	Aumento de datos: variación del valor (*value*)	0.4
translate	Aumento de datos: traslación de la imagen	0.1
scale	Aumento de datos: escala de la imagen	0.5
fliplr	Aumento de datos: volteo horizontal	0.5
mosaic	Aumento de datos: uso de mosaico	1.0

Fueron cinco fases fundamentales constituyeron el proceso de entrenamiento. Primero, se llevó a cabo la inicialización de parámetros, en la que cada capa de la red neuronal proporcionó valores preliminares a los sesgos y pesos. Estos valores sirvieron de base para el proceso de aprendizaje. Se llevó a cabo la propagación hacia adelante (forward propagation). En esta fase, las imágenes del conjunto de entrenamiento fueron introducidas en la red YOLOv8-seg y pasaron por sus distintas capas convolucionales. En consecuencia, el modelo detectó propiedades significativas de las imágenes sísmicas, como los bordes, las discontinuidades, las variaciones en la amplitud y los patrones estructurales relacionados con las fallas geológicas. Utilizando estos datos, elaboró las proyecciones de ubicación, categorización y segmentación de los objetos identificados. Luego, se realizó el cálculo de la función de pérdida (loss function), comparando las predicciones del modelo con las anotaciones de referencia que hicieron los especialistas. Esta función calculó el error vinculado a la segmentación, clasificación y detección, ofreciendo una medida del rendimiento obtenido en cada iteración. Seguidamente, se utilizó el descenso del gradiente (gradient descent), un procedimiento de optimización que sirve para reducir la función de pérdida. Para lograrlo, se estableció el rumbo en que el error disminuye más rápidamente, lo cual posibilitó enfocar la corrección de los parámetros de la red hacia una solución más exacta. La retropropagación (backpropagation) fue puesta en práctica en la etapa posterior; a través de este proceso, el error calculado se difundió desde la capa de salida hacia las capas internas de la red. Este proceso posibilitó el cálculo de los gradientes de cada uno de los pesos y sesgos, determinando así la aportación que hace cada parámetro al error global. La actualización de parámetros se llevó a cabo; en esta etapa, el optimizador cambió los sesgos y pesos por medio del uso de la tasa de aprendizaje establecida y los gradientes adquiridos. El modelo disminuyó de manera gradual el valor de la función de pérdida y aumentó su habilidad para identificar y dividir las fallas geológicas.

Cada conjunto de imágenes y cada periodo fijado experimentaron la repetición de las cinco etapas. Asimismo, el modelo fue evaluado con el conjunto de validación al término de cada época, a fin de supervisar cómo progresaba el aprendizaje y comprobar su capacidad para generalizar.

4.2.2. Evaluacion del modelo

La Figura 8 muestra el análisis de las curvas de error y la estabilidad del entrenamiento del modelo a lo largo de las 80 épocas, evidenciando que no existen grandes fluctuaciones ni sobreajuste.

Al evaluar la primera curva o pérdida de caja mediante la métrica CIoU, se cuantifica el error de localización espacial considerando la superposición y la distancia entre los centros. Esta gráfica muestra un inicio con un error alto de 4.5, una caída brusca hasta 2.0 en las primeras 10 épocas, seguida de un descenso constante hasta 1.5 en la época 40, para converger de manera suave en 1.0. Este comportamiento indica una excelente eficacia del algoritmo para ubicar las fallas en la posición correcta.

La segunda curva, titulada pérdida de segmentación evalúa la exactitud de los contornos a nivel de píxel frente a las máscaras reales trazadas por los geólogos expertos. Inició con el valor más alto de 6.2, pero logró un descenso pronunciado hasta 3.0 en la época 10 y una reducción continua hasta 2.0 en la época 40, estabilizándose finalmente en 1.2. Esto demuestra que el modelo muestra una mejora consistente y las máscaras predichas se alinean con gran precisión a las anotaciones geológicas reales.

Finalmente, la tercera curva, titulada pérdida de distribución focal, es la encargada de refinar la precisión de los bordes prediciendo distribuciones de probabilidad en lugar de coordenadas exactas. Esta gráfica cayó rápidamente desde 3.4 hasta 1.7 en las primeras 10 épocas, mantuvo un descenso gradual hasta 1.3, para converger en 1.0. Esta rápida convergencia inicial demuestra que el modelo aprende pronto a afinar las ubicaciones exactas de los límites estructurales.

En el análisis comparativo de la Figura 7, las tres métricas logran una estabilización completa alrededor de las épocas 60 y 70, momento en el que las mejoras son mínimas. La pérdida de segmentación se mantiene como la más alta debido a su dificultad inherente, mientras que la pérdida de distribución focal converge más rápido. En conjunto, las gráficas confirman el éxito del entrenamiento indicando que el modelo SismoAI está listo para localizar fallas, segmentar contornos precisos y refinar bordes de detección con alta precisión.

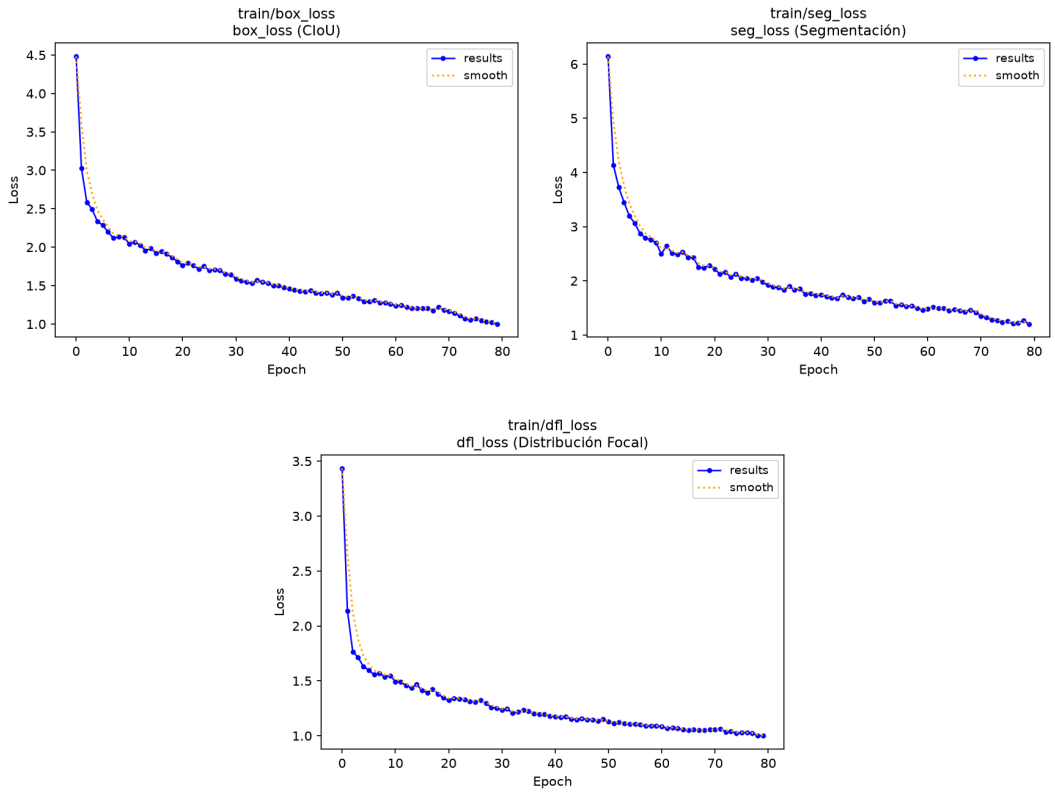


Figura. 8: Curvas de aprendizaje del entrenamiento de YOLOv8-seg.

La curva de exactitud (Figura 9) revela la dinámica de aprendizaje del modelo SismoAI a lo largo de las 80 épocas de entrenamiento. Inicialmente, el algoritmo parte de un 40% de rendimiento y experimenta un aprendizaje acelerado hasta la época 20. Se observa una convergencia clara entre el 72% y el 76% a partir de la época 40, momento en el que las curvas se estabilizan de forma definitiva. Un aspecto técnico fundamental es la brecha mínima de apenas 3% a 4% entre la precisión de entrenamiento y la de validación. Esta proximidad confirma una excelente capacidad de generalización, descartando un sobreajuste severo, ya que el modelo no memoriza los datos, sino que aprende patrones estructurales generalizables en las imágenes sísmicas. Cabe destacar que la parada anticipada podría haberse activado en la época 50 para optimizar recursos

computacionales sin sacrificar el rendimiento, aunque completar el ciclo configurado resultó en un entrenamiento validado.

Por su parte, la matriz de confusión desglosa el comportamiento de la clasificación binaria. El sistema logró identificar correctamente 295 verdaderos positivos para la clase falla y 299 verdaderos negativos para la clase background. No obstante, cometió 105 falsos positivos y 101 falsos negativos, resultando en una exactitud global del 74.25%. Al analizar las métricas derivadas, la sensibilidad o recall para la falla se sitúa en un 74.5%. Esto implica que el valor predictivo positivo también se mantiene en niveles similares, garantizando que cuando el modelo predice una discontinuidad, existe una alta probabilidad de que sea real. La notable simetría en la distribución de los errores demuestra la ausencia de sesgos, indicando que el modelo está estadísticamente bien calibrado.

Desde la perspectiva de las geociencias, la interpretación de estos errores es crítica. Aunque el sistema está equilibrado, los 101 falsos negativos representan fallas geológicas reales omitidas. En la exploración sísmica, ignorar una discontinuidad estructural altera drásticamente la interpretación del subsuelo, afectando la estimación de reservorios o la evaluación de riesgos geomecánicos. En conclusión, el modelo demuestra una base sólida, pero para aplicaciones profesionales es imperativo superar el 75% de exactitud. Esto se lograría optimizando el umbral de decisión para priorizar la sensibilidad, aplicando aumentos de datos o escalando a una arquitectura más robusta que minimice los falsos negativos.

La causa principal atribuida a los errores geofísicos en el modelo se atribuye principalmente a la representación de los reflectores sísmicos en las imagenes introducidas en el modelo, ya que los pixeles no alcanzan a capturar los reflectores como un programa especializado, aun asi el modelo cumple con un error aceptable, en la interfáz gráfica de la aplicaciá+ón generada, el usuario controla estos errores del sistema para mayor exactitud en la interpretacion de las fallas.

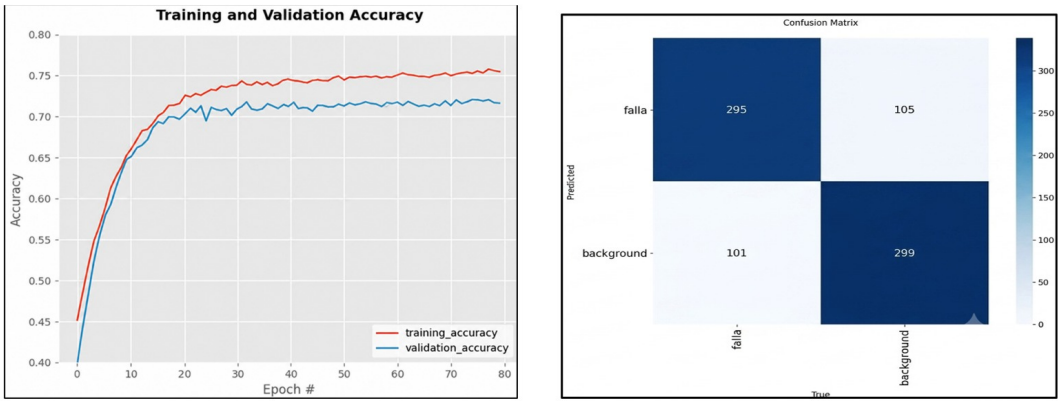


Figura. 9: A la izquierda de la imagen, la curva de aprendizaje entre la validación y el entrenamiento y a la derecha, la matriz de confusión.

Tras finalizar la detección y segmentación de las fallas mediante visión por computadora (Figura 10), el flujo de trabajo avanza hacia la ingeniería de prompting. En esta etapa, el sistema integra Modelos de Lenguaje Grande como GPT-4o y Gemini para traducir los resultados visuales en un documento técnico comprensible. Mediante un prompt estructurado, se acoplan las máscaras espaciales generadas por la red convolucional con la capacidad de razonamiento del lenguaje natural. De este modo, la inteligencia artificial se convierte en un asistente analítico capaz de redactar automáticamente la interpretación geológica y el reporte final.

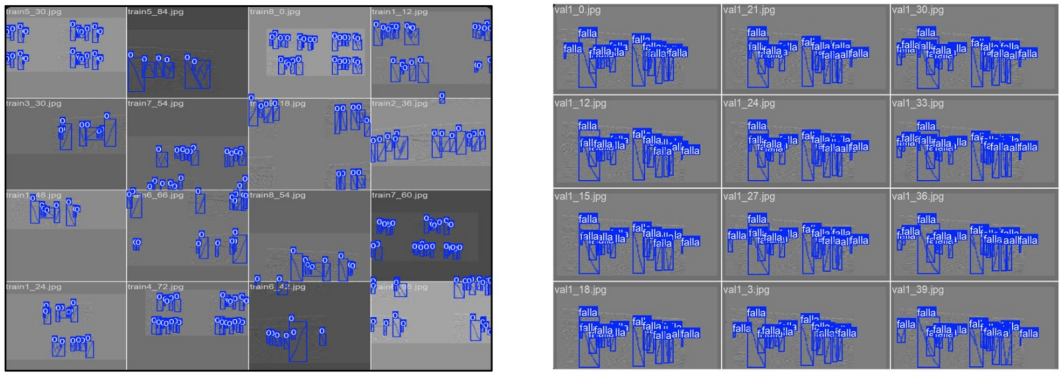


Figura. 10: La imagen a la izquierda representa los cuadros de detección del entrenamiento y la derecha de validación.

4.3. Modelo de Lenguaje

La interpretación tectónica y estratigráfica en conjunto con la redacción automatizada del reporte se fundamenta en el uso de Modelos de Lenguaje Multimodal (GPT-4o y Gemini). Al ser accedidos por medio de interfaces de aplicaciones (APIs) que son comerciales que suelen funcionar bajo un esquema de caja negra, por medio, de la derivación matemática que se detallan en sus arquitecturas internas como: mecanismos de autoatención, DiT o decodificadores, VAE, etc.

4.3.1. Ingeniería del Prompting

Para la generación de interpretaciones geológicas a partir de las fallas segmentadas, se empleó el modelo de lenguaje multimodal GPT-4o desarrollado por OpenAI y Gemini desarrollado por Google. Mediante un prompt estructurado, el sistema recibe tanto la imagen segmentada como el texto descriptivo introducido por el usuario, generando una interpretación geológica integral. Esta aproximación permite combinar la segmentación automática con capacidades avanzadas de razonamiento contextual y generación de lenguaje natural. Con el fin de garantizar coherencia interpretativa y reducir la variabilidad en las respuestas generadas, el prompt fue diseñado siguiendo una estructura formal basada en principios de prompt engineering siguiendo la fórmula propuesta. Esta estructura organiza explícitamente los componentes fundamentales de la interacción con el modelo siguiendo la estructura recomendada para prompts profesionales RTF creada en 2023, se evidencian resumidas en la siguiente formula y más explicitas en la tabla consiguiente.

$$\text{Prompt} = \text{Rol} + \text{Tarea} + \text{Entradas} + \text{Estructura} + \text{Reglas de estilo} \tag{1}$$

TABLA 6: ESTRUCTURA DEL PROMPT EMPLEADO PARA LA INTERPRETACIÓN GEOLÓGICA Y GENERACIÓN AUTOMÁTICA DE REPORTES.

Elemento	**Descripción**
Rol	Geólogo estructural y estratigráfico con más de 25 años de experiencia en interpretación sísmica de reflexión para exploración de hidrocarburos.
Tarea	Interpretar la imagen sísmica preservando su geometría, resolución y escala originales. Las fallas detectadas pueden eliminarse, extenderse, agregarse o reclasificarse (normales, inversas, lístricas, sintéticas, antitéticas, ciegas, entre otras), diferenciando visualmente las principales (línea continua) de las secundarias (línea discontinua). Identificar únicamente, cuando exista evidencia clara, elementos geológicos como *rollovers*, horst, graben, discordancias, *onlap*, *downlap*, *toplap*, canales y pliegues. Generar además un segundo panel con pseudo-facies basado en continuidad, amplitud, frecuencia y geometría de los reflectores. Incorporar una leyenda discreta sin agregar elementos gráficos adicionales y producir como salida un archivo JSON estructurado con las fallas, sismofacies e interpretación. El contexto integra dinámicamente métricas del modelo (precisión, *recall* y F1-score), interpretación previa generada por IA e información del proyecto proporcionada por el usuario, sin inventar datos faltantes.
Entradas	Nombre de imagen sísmica y su formato (.jpg, .png).
Estructura	El reporte generado se organiza en diez secciones: (1) Información del proyecto, (2) Resumen ejecutivo, (3) Figura principal y objetivo del estudio, (4) Contexto geológico, (5) Análisis estructural, (6) Interpretación estratigráfica, (7) Implicaciones para la exploración de hidrocarburos, (8) Conclusiones, (9) Recomendaciones y (10) Bibliografía.
Reglas de estilo	Redacción técnica y sobria, priorizando el texto sobre las tablas, escrita íntegramente en español, evitando frases vacías y manteniendo un estándar de calidad equivalente al de publicaciones científicas como *AAPG Bulletin* e *Interpretation*.

Una vez obtenido el prompt inicial con la estructura especificada en las indicaciones se lo implementa en un modelo de lenguaje LLM con el fin de obtener un reporte.

4.3.2. Implementación de LLM

Luego de tener el prompt inicial estructurado se implementa en el modelo de lenguaje LLM, donde se evaluaron 4 modelos de lenguaje de inteligencia artificial para la generacion de reportes, siendo escogidos al final GPT y Gemini basándose principalmente en 4 parametros de la siguiente tabla.

TABLA 7: COMPARACIÓN DE MODELOS DE INTELIGENCIA ARTIFICIAL UTILIZADOS PARA LA GENERACIÓN DE REPORTES.

Parámetro	**Gemini IA**	**ChatGPT**	**Copilot**	**DeepSeek**
Capacidad de razonamiento	✓	✓	✓	✓
Tiempo de respuesta	✓	✓	✓	✓
Redacción técnica	✓	✓	×	✓
Gratuidad	✓	✓	✓	×

La arquitectura fundamental de estos modelos de lenguaje GPT y Gemini, las cuales sustentan el procesamiento y la generación de las respuestas interpretativas, se basa en la arquitectura correspondiente al Transformer, introducido por

Vaswani (2017) [42] en el artículo “Attention Is All You Need”, el cual propuso un esquema encoder–decoder fundamentado en mecanismos de atención y autoatención. En su variante decoder-only, el procesamiento comienza con la normalización y tokenización del texto de entrada frecuentemente mediante sub-palabras, donde cada token es convertido en un índice numérico dentro de un vocabulario discreto. Éstos índices son transformados en vectores menos densos a través de una matriz de embeddings. A cada embedding léxico se le suma un embedding posicional, generalmente basado en funciones seno-coseno, que codifica la posición del token en la secuencia y permite preservar el orden secuencial. El resultado es una secuencia de vectores en un espacio latente continuo que atraviesa múltiples capas de masked self-attention, donde cada token atiende únicamente a los anteriores durante la generación autoregresiva del texto. Para incorporar la información visual, el sistema emplea un decoder donde la imagen correspondiente al mapa de fallas previamente generado por YOLOv8s-seg se divide en parches de tamaño fijo, los cuales son proyectados linealmente al espacio de embeddings visuales y enriquecidos con codificación posicional antes de pasar por capas de autoatención. Estos embeddings visuales se proyectan al mismo espacio latente que los embeddings lingüísticos y se integran como tokens dentro de la secuencia procesada por el decoder. De esta manera, el modelo aplica autoatención conjunta sobre información textual y visual, permitiendo que el proceso autorregresivo genere interpretaciones geológicas condicionadas tanto por los patrones espaciales segmentados como por el contexto semántico del prompt, integrando así análisis cuantitativo y razonamiento lingüístico para la generación de la interpretación geológica.

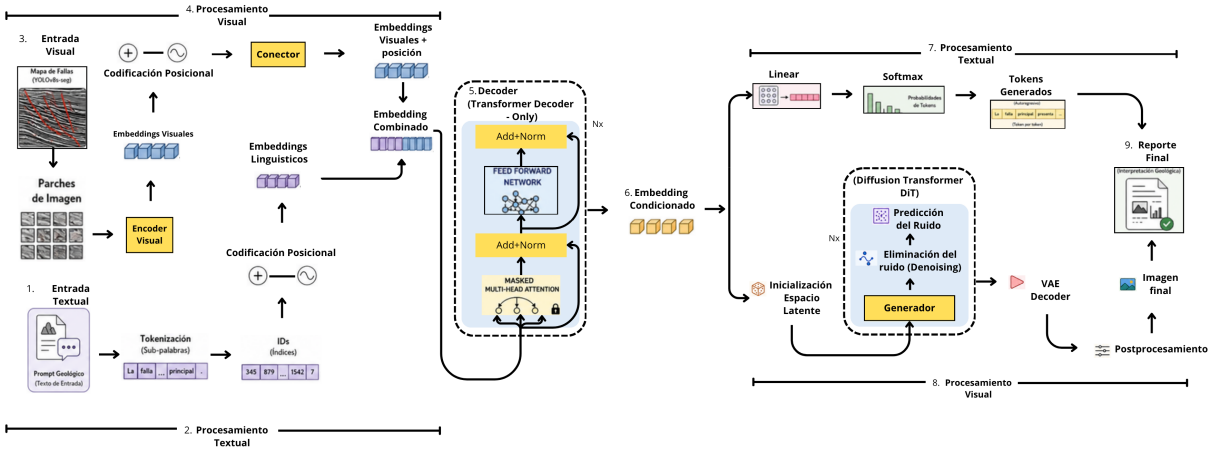


Figura. 11: Diagrama de decoder en los modelos de lenguaje usados para la generacion del reporte.

La arquitectura se usó para la generación del reporte automatizado (Figura 11), con dos entradas: el prompt explicado anteriormente como entrada textual y la imagen interpretada con máscaras como entrada visual.

La entrada de prompt se divide en tokens que a la vez se convierten en embeddings donde se le agrega posición con funciones de seno y coseno como determina la arquitectura, estos se combinan con los embeddings para generar embeddings de posición, de igual forma, para la entrada visual se generan parches que se transforman a embeddings visuales y por medio de un conector se logran sumar ambas entradas ya transformadas en un solo formato para crear un embedding combinado, después entra al decoder donde por medio de los mecanismos de atención la arquitectura encuentra patrones y relaciones agregando adición y normalización para pasar por una red neuronal y finalmente dar como producto un embedding condicionado que cuenta con contexto, así se generan dos fuentes de salida.

Por un lado, el embedding condicionado que pasa por un proceso de linealización y una función de softmax, misma que redistribuye la probabilidad de todas las palabras del vocabulario y escoge la de mayor probabilidad para generar de forma secuencial el texto del reporte.

Mientras que, para la segunda salida se inicia el proceso de espacio latente que parte del embedding condicionado, para entrar a otro decoder adicional donde se modelan las relaciones espaciales por medio de matrices y se refina las características produciendo un ruido en la imagen latente, después se elimina este ruido (Denoising) con cada iteración, para pasar al VAE decoder donde se convierte la representación en una imagen RGB reconstruyendo (bordes, texturas, colores, etc), en el postprocesamiento se aplican ajustes finales (escala, normalización y conversión para el formato de salida), así al final se obtiene la imagen con las especificaciones del prompt de entrada.

Por ello, en la última etapa se integran ambas salidas y se genera el reporte automatizado como producto final.

4.3.3. Evaluación de LLM

Para evaluar la calidad de los informes creados por los modelos de lenguaje, un geólogo con vasta experiencia en interpretación sísmica para la exploración de hidrocarburos revisó las interpretaciones producidas. La evaluación consistió en cotejar los informes generados de manera automática con la interpretación llevada a cabo por el experto, comprobando que las estructuras geológicas se identificaran correctamente, que el contexto estratigráfico fuera coherente, que la terminología técnica se empleara adecuadamente y que las conclusiones fueran consistentes. Este método se basa en la

estrategia de evaluación humana que se sugiere para modelos de lenguaje extensos, en los que el criterio de los especialistas es el sistema más apropiado para evaluar la calidad de tareas complejas y con un dominio específico [43, 10].

Como la finalidad principal de esta investigación era crear una herramienta para ayudar en la interpretación, y no un sistema de decisión totalmente autónomo, no se calcularon métricas cuantitativas como el coeficiente Kappa o los índices de similitud semántica. Por otro lado, la validación se basó en una revisión cualitativa efectuada por el experto, que concluyó que los informes tienen un grado de coherencia técnica apropiado para funcionar como respaldo durante la interpretación, siempre bajo la supervisión y juicio del geólogo a cargo.

Se llevó a cabo una estrategia de ingeniería de instrucciones para el modelo, que se basa en la creación de restricciones explícitas y en la estructuración de las instrucciones, con el objetivo de disminuir la probabilidad de alucinaciones y generar información sin fundamento. El sistema solo recibe la imagen sísmica analizada, las fallas que se han segmentado con anterioridad y los datos suministrados por el usuario, evitando así hacer deducciones sobre información que no existe y distinguiendo observaciones objetivas de interpretaciones geológicas.

El modelo fue entrenado para emplear un lenguaje técnico y mantener una estructura de reporte en consonancia con las publicaciones científicas, además de seguir las sugerencias recientes sobre el diseño de prompts para modelos multimodales [44, 45]. Antes de generar el informe final, existe la opción de que el usuario modifique manualmente la interpretación, lo cual incluye un sistema de validación humana (human-in-the-loop). Este sistema es muy aconsejado para aplicaciones críticas que se basan en inteligencia artificial [46, 47].

5. RESULTADOS

Como producto final se obtuvo una imagen interpretada por medio de máscaras detectando las fallas o lineamientos estructurales, en función de ello se optó por aplicar el modelo mediante la creación de una página web, diseñada para cumplir con las bases fundamentales de la interpretación sísmica estructural, tiene como características ser de libre acceso, facilidad de uso con la implementación de una interfaz interactiva que permite al usuario obtener un reporte automatizado completo, sirviendo como base de apoyo para la interpretación sísmica profesional. La plataforma Sismo IA presenta la página donde se debe cargar la imagen sísmica, para que el algoritmo realiza la detección e interpretación de las fallas, obteniendo una imagen interpretada, después por medio del prompt o de la interpretación automática obtener el reporte automatizado estructura en formato técnico, en esta plataforma también se permite exportar la imagen y el reporte dependiendo del formato de preferencia, editable en Word o fijo en PDF.

5.1. Aplicación Web

Streamlit[3] es una herramienta de código abierto que utiliza Python para crear rápidamente aplicaciones web interactivas, evitando el uso de tecnologías web convencionales como HTML, CSS y JavaScript. Su objetivo principal es simplificar el proceso de generación de interfaces gráficas para iniciativas en ciencia de datos, inteligencia artificial, machine learning y visualización de datos, haciendo posible convertir scripts en Python en aplicaciones web operativas con un número reducido de líneas de código. Esto ahorra considerablemente tiempo en el desarrollo y ayuda a exponer hallazgos técnicos de manera clara y participativa. Las aplicaciones creadas con Streamlit pueden operar en un entorno local o en un servidor, y se pueden abrir directamente a través de un navegador web, permitiendo que el usuario interactúe con elementos como gráficos, tablas, imágenes, formularios y modelos en tiempo real. Su mecanismo de funcionamiento consiste en ejecutar un script en Python que automáticamente construye la interfaz web, actualizando la información cada vez que el usuario realiza alguna interacción con la aplicación. Gracias a su integración con bibliotecas como NumPy, Pandas, Matplotlib, Plotly y TensorFlow, Streamlit se ha convertido en una herramienta ampliamente reconocida para el desarrollo de aplicaciones científicas y de análisis de datos.

Se creó una aplicación web interactiva usando Streamlit (Anexos, Figura 12), que incorpora todo el proceso de trabajo para la interpretación automática de fallas sísmicas. La plataforma brinda la posibilidad de cargar imágenes sísmicas en los formatos JPG, PNG y TIFF, realizar la detección y segmentación automática de fallas a través de un modelo de visión entrenado con anterioridad y modificar el grado de confianza de las predicciones. Además, el usuario tiene la posibilidad de revisar y modificar los resultados, eliminando o añadiendo nuevas fallas según su criterio geológico, para así conseguir una interpretación final ajustada. Con base en esta información, la aplicación produce automáticamente un informe técnico estructurado a través de modelos de lenguaje de gran escala, el cual es exportable en formatos PDF o Word.Debido a que la plataforma incorpora diversas herramientas y opciones de configuración, su funcionamiento se describe detalladamente en forma de manual de usuario en los Anexos.

5.2. Generación de reporte automatizado

La aplicación y en adición, la detección e interpretación de fallas, crea de manera automática un informe técnico asistido por inteligencia artificial (Anexos, Figura 11)). El usuario tiene la posibilidad de elegir al proveedor del modelo (Gemini o ChatGPT) e introducir información fundamental acerca del proyecto, como el país, nombre, orientación, cuenca y escala de la sección sísmica, con el fin de proporcionar contexto al análisis. El reporte consta de diez partes técnicas como son:

[3] https://streamlit.io/

información del proyecto, resumen ejecutivo, figura principal, contexto geológico, análisis estructural, interpretación estratigráfica, implicaciones para la exploración de hidrocarburos, conclusiones y recomendaciones. Se mantiene un formato profesional y se puede descargar en PDF o Word.

6. CONCLUSIONES

La construcción del conjunto de datos permitió contar con una base de información adecuada para el desarrollo del modelo. Para ello, se recopilaron, depuraron y validaron 1050 imágenes sísmicas, las cuales fueron sometidas a un proceso de preprocesamiento que incluyó la conversión al formato YOLO, el redimensionamiento de las imágenes, el engrosamiento de las fallas y la eliminación de ruido. Fue posible mejorar la calidad de los datos de entrada y facilitar la identificación de las características geológicas necesarias para el entrenamiento del modelo.

Por otra parte, la implementación del modelo YOLOv8-Seg, junto con la estrategia de Transfer Learning (Fine-Tuning), permitió adaptar un modelo previamente entrenado a la detección de fallas geológicas en imágenes sísmicas. Gracias a esta metodología, se redujo el tiempo de entrenamiento y se obtuvo una localización más precisa de las fallas, alcanzando predicciones con altos niveles de confianza. En consecuencia, se demostró que el aprendizaje profundo constituye una alternativa eficiente para apoyar la automatización de la interpretación geofísica.

Asimismo, la incorporación de modelos de lenguaje como GPT y Gemini permitió complementar el proceso de detección automática mediante la interpretación de las fallas identificadas y la generación de reportes técnicos. De esta manera, el sistema no solo identifica las estructuras geológicas presentes en las imágenes sísmicas, sino que también presenta los resultados en un formato más comprensible para el usuario, facilitando el análisis y reduciendo el tiempo necesario para elaborar informes técnicos.

Finalmente, la integración de todo el flujo de trabajo en una aplicación web desarrollada en Streamlit permitió reunir en una sola plataforma la carga de imágenes, la detección automática de fallas, la interpretación asistida y la generación de reportes en formatos .doc y .pdf. En conjunto, la metodología propuesta constituye una herramienta de apoyo para especialistas en geociencias, ya que agiliza el análisis de información sísmica, disminuye la subjetividad durante la interpretación y contribuye a la toma de decisiones en las etapas de exploración y evaluación de yacimientos, sin reemplazar el criterio del intérprete.

Como trabajos futuros se propone ampliar el conjunto de datos incorporando un mayor número de imágenes sísmicas reales provenientes de diferentes cuencas geológicas y condiciones de adquisición, con el fin de mejorar la capacidad de generalización del modelo. Asimismo, se plantea evaluar arquitecturas más recientes basadas en Transformers y modelos multimodales, integrar información adicional como atributos sísmicos y datos de pozos, y desarrollar un sistema multiagente más avanzado que permita realizar interpretaciones geológicas más completas, generar reportes personalizados y brindar soporte en la toma de decisiones durante las etapas de exploración y evaluación de yacimientos.

DISPONIBILIDAD DE DATOS Y CÓDIGO

Los datos y código que soportan los hallazgos del presente estudio están libremente disponibles en el siguiente URL: `https://drive.google.com/drive/folders/1rTNNP4PqPm-SbM6t7Wj_c3k1R6z85ZkA?usp=drive_link`

REFERENCIAS

[1] H. Fossen, *Structural Geology*, 2nd ed. Cambridge: Cambridge University Press, 2016.

[2] B. Alaei and K. Petersen, "Artificial intelligence applications in structural geology: A review," *Earth-Science Reviews*, vol. 244, p. 104514, 2023.

[3] S. Chopra and K. J. Marfurt, *Stratigraphic and Structural Interpretation Using Seismic Attributes*. Tulsa: Society of Exploration Geophysicists, 2018.

[4] X. Wu, L. Liang, Y. Shi, and S. Fomel, "Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation," *Geophysics*, vol. 84, no. 3, pp. IM35–IM45, 2019.

[5] Y. Alaudah, M. Michałowicz, M. Alfarraj, and G. AlRegib, "A machine-learning benchmark for facies classification and fault interpretation," *Interpretation*, vol. 7, no. 3, pp. SE175–SE187, 2019.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[8] Y. An, J. Guo, Q. Ye, C. Childs, J. Walsh, and R. Dong, "A gigabyte interpreted seismic dataset for automatic fault recognition," *Data in Brief*, vol. 38, p. 107219, 2021. [Online]. Available: https://doi.org/10.1016/j.dib.2021.107219

[9] G. Jocher, A. Chaurasia, J. Qiu *et al.*, "Ultralytics yolov8," https://github.com/ultralytics/ultralytics, 2023, accessed: 2026-07-12.

[10] J. Achiam, S. Adler, S. Agarwal *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[11] Gemini Team, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2024.

[12] M. Bahorich and S. Farmer, "3-d seismic discontinuity for faults and stratigraphic features: The coherence cube," *The Leading Edge*, vol. 14, no. 10, pp. 1053–1058, 1995.

[13] A. Gersztenkorn and K. J. Marfurt, "Eigenstructure-based coherence computations as an aid to 3-d structural and stratigraphic mapping," *Geophysics*, vol. 64, no. 5, pp. 1468–1479, 1999.

[14] T. Randen, L. Sønneland *et al.*, "Three-dimensional texture attributes for seismic data analysis," in *SEG Technical Program Expanded Abstracts*, 2002, pp. 668–671.

[15] S. I. Pedersen, T. Randen, L. Sønneland *et al.*, "Automatic fault extraction using artificial ants," in *SEG Technical Program Expanded Abstracts*, 2005, pp. 566–569.

[16] M. Donias, C. David, Y. Berthoumieu, O. Lavialle, S. Guillon, and N. Keskes, "Edge-preserving filtering for seismic images based on an anisotropic diffusion," *IEEE International Conference on Image Processing*, 2007.

[17] M. R. Atencio, S. Periale, and G. Zamora Valcarce, "Interpretación sísmica estructural de la zona de falla de huincul," *VII Congreso de Exploración y Desarrollo de Hidrocarburos*, 2008.

[18] J. A. Vargas Parra, M. T. Duarte, C. A. Pérez Solano, G. Y. Ojeda, and W. M. Agudelo, "Procesamiento de datos sísmicos 2d en el área de apiay-ariari (llanos orientales)," *CT&F-Ciencia, Tecnología y Futuro*, vol. 3, no. 5, pp. 25–42, 2009.

[19] D. Hale, "Structure-oriented smoothing and semblance," *Center for Wave Phenomena Report*, 2009.

[20] D. Hale, "Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images," *Geophysics*, vol. 78, no. 2, pp. O33–O43, 2013.

[21] D. R. Kolyukhin, V. V. Lisitsa, M. I. Protasov, D. Qu, and G. V. Reshetova, "Seismic attribute analysis for fault detection: A comparison," *Journal of Geophysics and Engineering*, 2017.

[22] X. Wu, H. Di, and G. AlRegib, "Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation," *Geophysics*, vol. 85, no. 3, pp. WA27–WA39, 2020.

[23] Y. An, J. Guo, Q. Ye, C. Childs, J. Walsh, and R. Dong, "Seismic fault detection based on a multi-scale attention mechanism," *Geophysics*, vol. 86, no. 4, 2021.

[24] H. Di, Z. Wang, and G. AlRegib, "Deep learning for fault detection and structural interpretation in seismic images," *Interpretation*, vol. 9, no. 2, pp. SF1–SF14, 2021.

[25] S. Li *et al.*, "Automatic geological fault identification from seismic data using 2.5d channel attention u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[26] OpenAI, "Chatgpt," https://chat.openai.com, 2022.

[27] F. Corbetta, "Deep learning for automated seismic fault detection," *Technical Scientific Reports*, 2023.

[28] H. Di *et al.*, "Deep learning advances for automatic seismic fault interpretation," *Interpretation*, 2024.

[29] Y. Wang *et al.*, "Attention-based deep neural network for seismic fault segmentation," *IEEE Access*, 2024.

[30] X. Liu *et al.*, "Robust seismic fault detection using deep convolutional neural networks," *Geophysics*, 2024.

[31] H. Zhao *et al.*, "Automatic seismic fault interpretation based on multi-scale feature fusion," *Interpretation*, 2024.

[32] Y. Zhang *et al.*, "Fault identification of seismic data based on seu-net approach," *ResearchGate Scientific Papers*, 2025.

[33] Y. Chen *et al.*, "Multi-scale attention network for automatic seismic fault detection," *Geophysics*, 2025.

[34] J. Sun *et al.*, "Hybrid transformer-cnn architecture for seismic fault segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[35] Z. Xu *et al.*, "Deep hybrid networks for seismic structural interpretation," *Interpretation*, 2025.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *MICCAI*, 2015.

[37] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of ICCV*, 2017.

[39] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *ECCV*, 2018.

[40] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *NeurIPS*, 2021.

[41] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS 2017)*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[43] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, 2022.

[44] J. White *et al.*, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[45] OpenAI, "Prompt engineering guide," 2024, https://platform.openai.com/docs/guides/prompt-engineering.

[46] S. Amershi *et al.*, "Guidelines for human-ai interaction," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2019.

[47] S. Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Books, 2021.

[48] K. Abhishek, "Advancing semantic segmentation: Enhanced unet algorithm with attention mechanism and deformable convolution," *Journal of Visual Communication and Image Representation*, vol. 102, pp. 103–115, 2026.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

ANEXOS

Anexo A. Manual de la aplicación

Este anexo tiene la finalidad de guiar al usuario de forma práctica a través de la interfaz de la plataforma, mostrando las funciones de la misma e indicando los pasos a seguir.
Segmentación y etiquetación de fallas: la plataforma permite cargar una imagen en varios formatos (png, jpg y TIFF), cambia la imagen a escala de grises y permite ajustar el nivel de confianza mínima por medio de in slider interactivo entre 0.05 - 0.90. Entre menor sea el valor, más fallas identificará (Figura 12). La plataforma devolverá una imagen con fallas etiquetadas de izquierda a derecha sirviendo después como identificadores para todo el reporte.

Figura. 12: Pasos para segmentación y etiquetación de fallas.

Aumento y eliminación de la falla etiquetada: después de obtener la imagen sísmica con las fallas identificadas, la plataforma permite agregar fallas por medio de clics con dos puntos sobre la imagen (Figura 13) o eliminarlas donde se escoge la falla etiquetada que se desea eliminar (Figura 14). Se obtiene una imagen final interpretada con un conjunto de fallas detectadas y editadas (Figura 15). El aumento o eliminación de fallas es según el criterio del usuario.

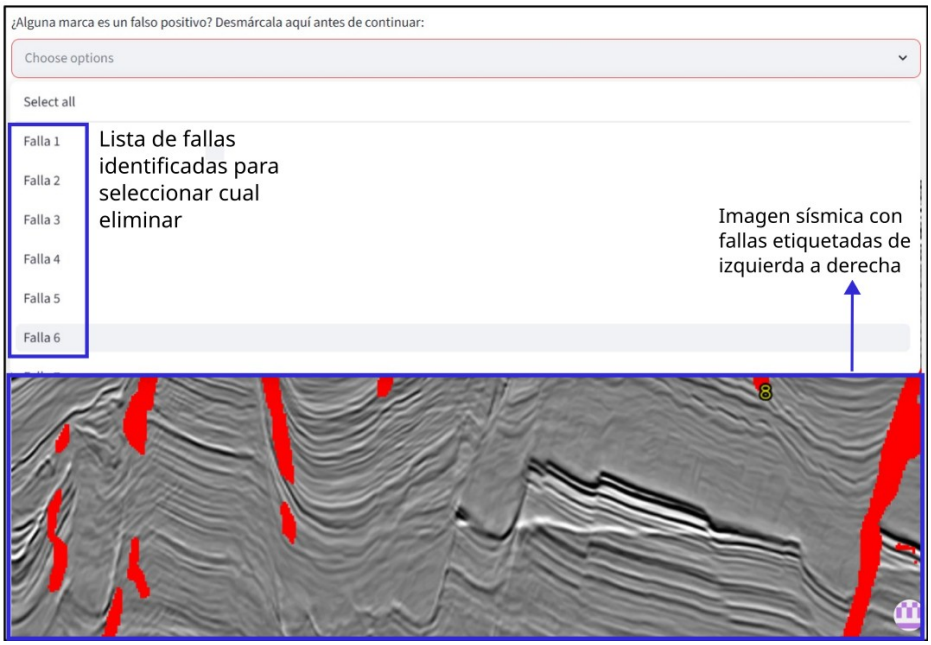


Figura. 13: Sección de eliminar fallas.

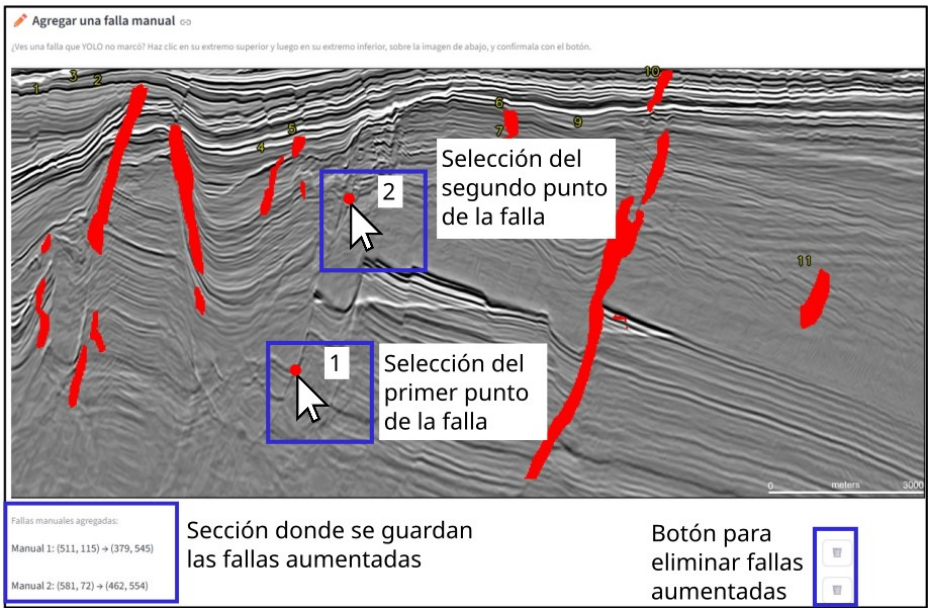


Figura. 14: Sección de agregar fallas.

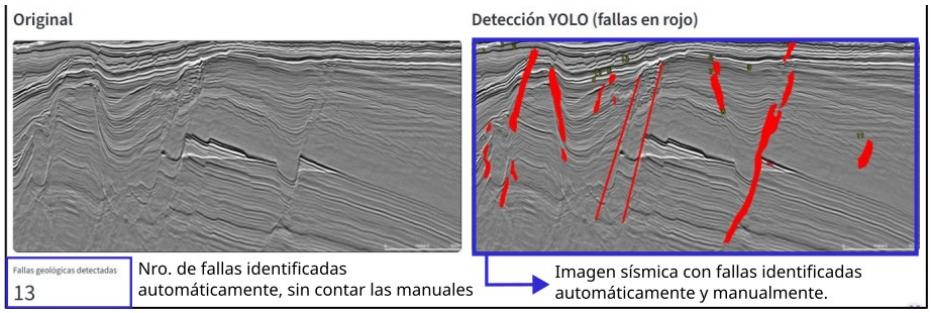


Figura. 15: Resultado de agregar fallas en la imagen interpretada final.

Interpretación con IA de sismofacies: con la imagen interpretada final, la plataforma permite realizar la interpretación de facies sísmicas con dos opciones a escoger, según el objetivo y criterio del usuario (Figura 16). La primera opción es con un prompt estructurado sugerido que puede se copiar en alguno de los modelos como de lenguaje ChatGPT o Gemini, junto con la imagen interpretada para obtener una interpretación de alta calidad (Figura 17). La segunda opción es automática con IA integrada dentro de la plataforma web, sin embargo, se advierte que el modelo tiene limitaciones técnicas donde las sismofacies interpretadas se deben tomar como referencia y ser sometida bajo el criterio del usuario. (Figura 18).

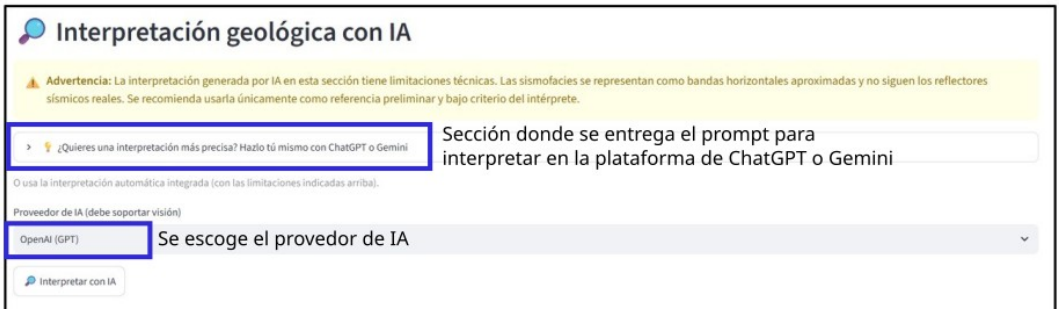


Figura. 16: Opciones para la interpretación de sismofacies.

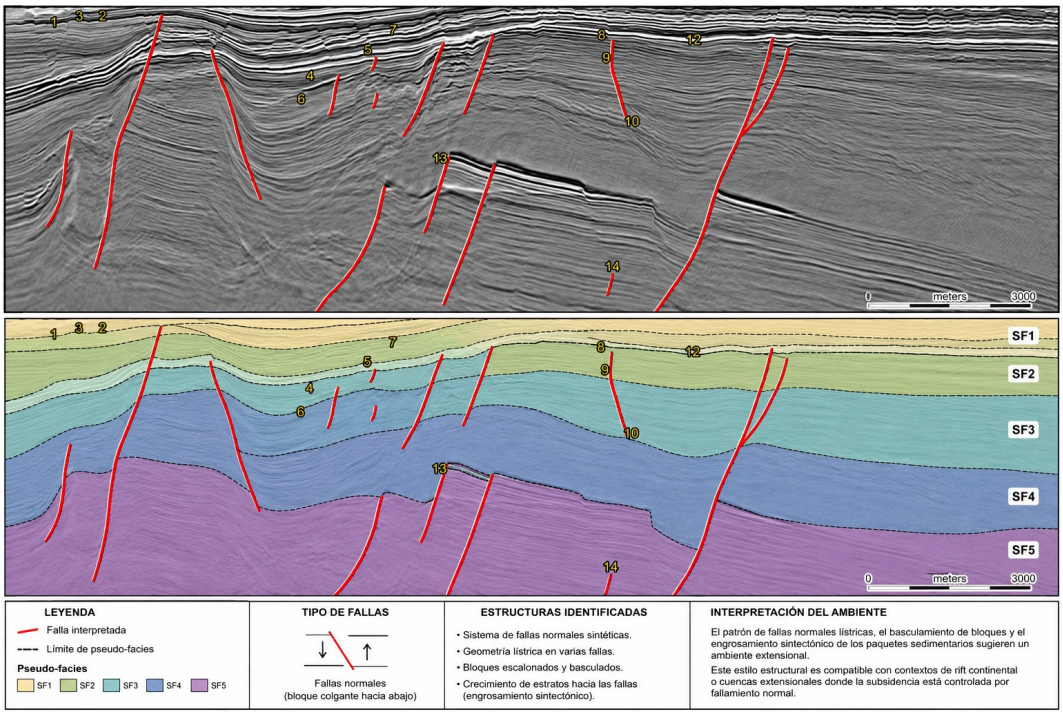


Figura. 17: Resultado de interpretación de sismofacies con prompt.

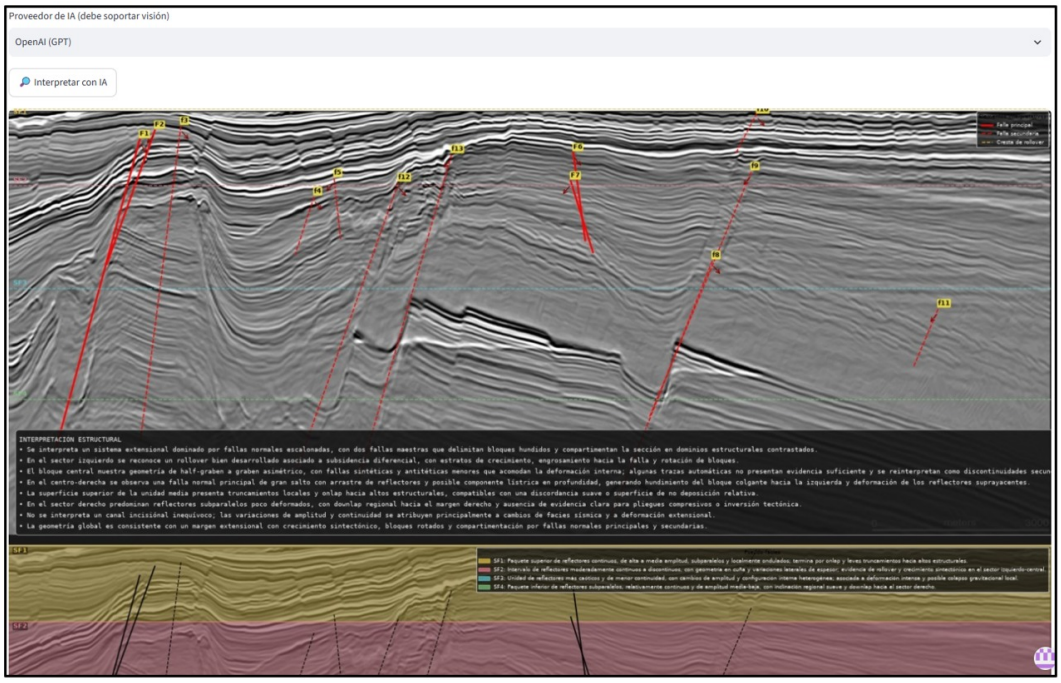


Figura. 18: Interpretación de sismofacies automática desde la página web

Reporte automatizado: una vez obtenida las imágenes con la interpretación de sismofacies con cualquiera de las anteriores opciones, se puede generar el reporte automático con IA, pudiendo escoger entre dos proveedores (ChatGPT o Gemini), es necesario llenar la información preliminar de la sección sísmica como: nombre del proyecto, orientación, escala, cuenca, país, entre otros (Figura 19). Los campos de información son opcionales y dependen de la disponibilidad de información. Se obtiene un reporte con la información puesta en la plataforma en formato .pdf (Figura 20)

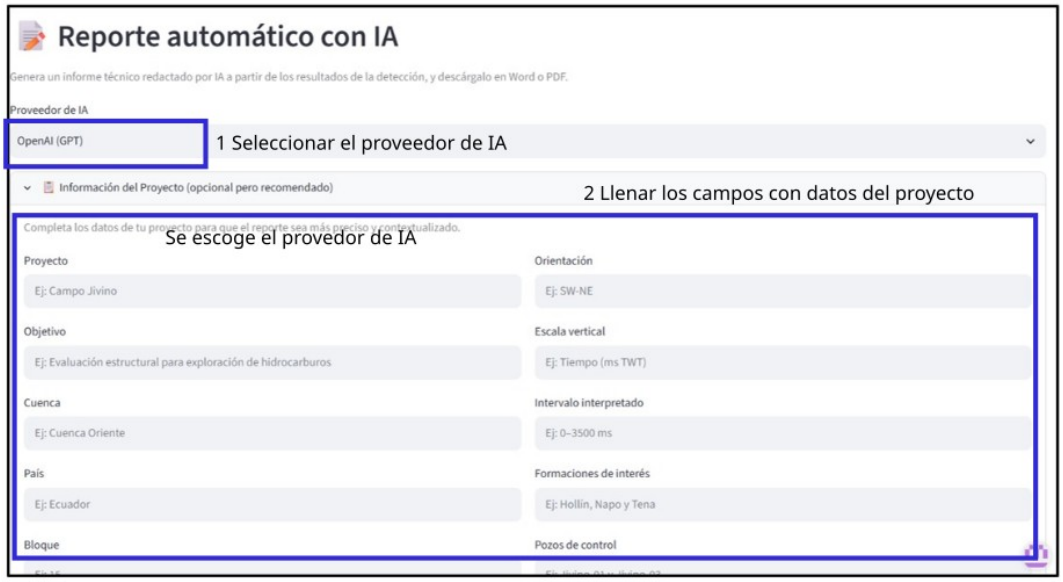
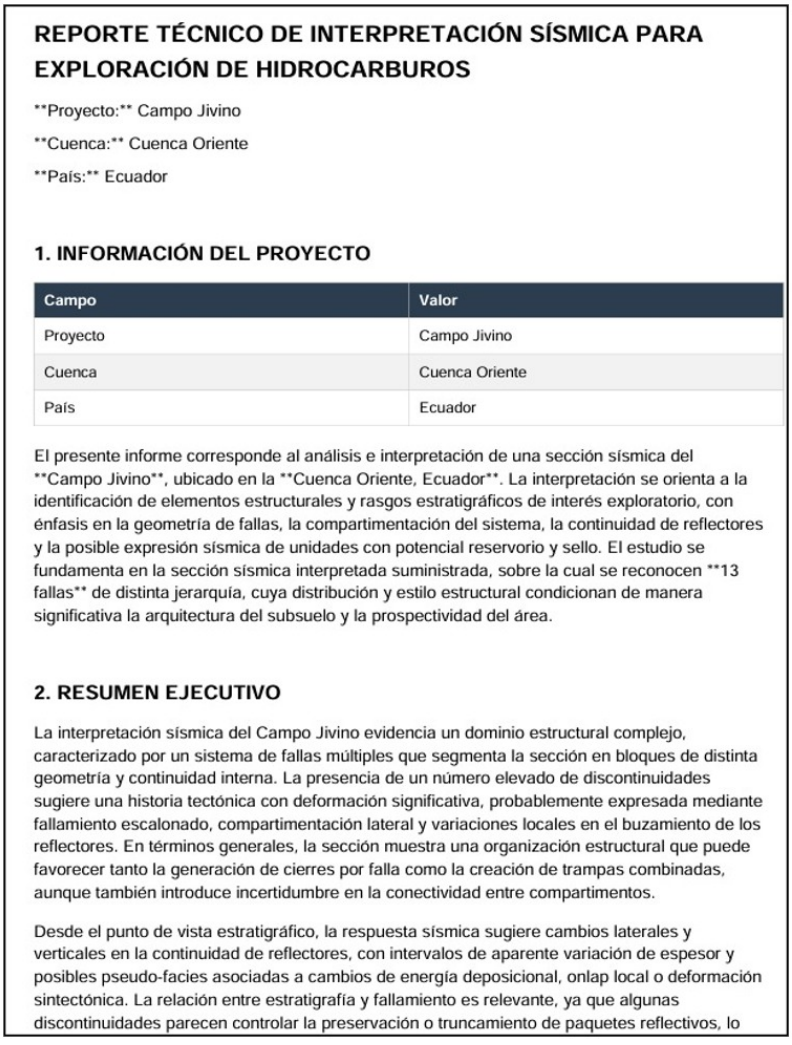


Figura. 19: Sección de información preliminar de la imagen sísmica solicitada al usuario.



REPORTE TÉCNICO DE INTERPRETACIÓN SÍSMICA PARA EXPLORACIÓN DE HIDROCARBUROS

Proyecto: Campo Jivino

Cuenca: Cuenca Oriente

País: Ecuador

1. INFORMACIÓN DEL PROYECTO

Campo	Valor
Proyecto	Campo Jivino
Cuenca	Cuenca Oriente
País	Ecuador

El presente informe corresponde al análisis e interpretación de una sección sísmica del **Campo Jivino**, ubicado en la **Cuenca Oriente, Ecuador**. La interpretación se orienta a la identificación de elementos estructurales y rasgos estratigráficos de interés exploratorio, con énfasis en la geometría de fallas, la compartimentación del sistema, la continuidad de reflectores y la posible expresión sísmica de unidades con potencial reservorio y sello. El estudio se fundamenta en la sección sísmica interpretada suministrada, sobre la cual se reconocen **13 fallas** de distinta jerarquía, cuya distribución y estilo estructural condicionan de manera significativa la arquitectura del subsuelo y la prospectividad del área.

2. RESUMEN EJECUTIVO

La interpretación sísmica del Campo Jivino evidencia un dominio estructural complejo, caracterizado por un sistema de fallas múltiples que segmenta la sección en bloques de distinta geometría y continuidad interna. La presencia de un número elevado de discontinuidades sugiere una historia tectónica con deformación significativa, probablemente expresada mediante fallamiento escalonado, compartimentación lateral y variaciones locales en el buzamiento de los reflectores. En términos generales, la sección muestra una organización estructural que puede favorecer tanto la generación de cierres por falla como la creación de trampas combinadas, aunque también introduce incertidumbre en la conectividad entre compartimentos.

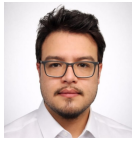
Desde el punto de vista estratigráfico, la respuesta sísmica sugiere cambios laterales y verticales en la continuidad de reflectores, con intervalos de aparente variación de espesor y posibles pseudo-facies asociadas a cambios de energía deposicional, onlap local o deformación sintectónica. La relación entre estratigrafía y fallamiento es relevante, ya que algunas discontinuidades parecen controlar la preservación o truncamiento de paquetes reflectivos, lo

Figura. 20: Ejemplo de reporte técnico generado por la opción con prompt en la página web de SismoIA.

SOBRE LOS AUTORES

Francisco Acosta
Estudiante de décimo semestre de Ingeniería en Geología de la Universidad Central del Ecuador. Sus principales áreas de investigación comprenden la inteligencia artificial aplicada a las geociencias, interpretación sísmica, exploración de hidrocarburos y aprendizaje profundo. Desarrolla investigaciones sobre visión por computador y generación automática de reportes técnicos.

Dayana Gómez
Estudiante de Ingeniería en Geología de la Universidad Central del Ecuador. Ha participado en proyectos de investigación geológica, cartografía y evaluación de riesgos. Posee conocimientos en inteligencia artificial, aprendizaje automático, aprendizaje profundo, redes neuronales convolucionales, visión por computador y procesamiento de lenguaje natural.



Alan Zambrano
Estudiante de décimo semestre de Ingeniería en Geología de la Universidad Central del Ecuador y vicepresidente del SEG-UCE Student Chapter. Posee conocimientos en inteligencia artificial, machine learning, aprendizaje profundo, redes neuronales convolucionales, visión por computador, modelos de lenguaje y desarrollo de aplicaciones web para las geociencias.

Victor Collaguazo
Ingeniero Geólogo con experiencia en geología, petrofísica, geofísica e inteligencia artificial aplicada a las geociencias. Docente de la Universidad Central del Ecuador e investigador en caracterización de yacimientos, integración de datos del subsuelo y aplicación de tecnologías digitales para la exploración y desarrollo de hidrocarburos.

Alyce Enríquez
Estudiante de décimo semestre de Ingeniería en Geología de la Universidad Central del Ecuador. Sus áreas de interés incluyen geología, geofísica, interpretación sísmica y SIG. Posee conocimientos en inteligencia artificial, aprendizaje profundo, redes neuronales convolucionales, visión por computador y procesamiento de lenguaje natural. Actualmente es presidenta del SEG-UCE Student Chapter.

Michael Reinoso
Estudiante de Ingeniería en Geología de la Universidad Central del Ecuador. Cuenta con experiencia en exploración petrolera e interpretación geológica. Posee conocimientos en inteligencia artificial, machine learning, aprendizaje profundo, redes neuronales convolucionales, procesamiento de lenguaje natural y análisis de datos con Python, Petrel y QGIS.

Christian Mejía
Docente e investigador de la Universidad Central del Ecuador. Sus líneas de investigación comprenden inteligencia artificial aplicada a las geociencias, aprendizaje automático, visión por computador, procesamiento de lenguaje natural y análisis de imágenes. Ha dirigido proyectos de automatización e innovación mediante inteligencia artificial.