

Sistema Basado en Redes Neuronales Generativas Adversarias (GAN) para la  
Obtención de Sismogramas Acústicos a Partir de Sismogramas Elásticos

Danilo Ramírez Gómez

Trabajo de Grado para Optar el Título de Ingeniero Electrónico.

Director

Sergio Alberto Abreo Carrillo

Doctor en Ingeniería Electrónica

Co-Director

Óscar Mauricio Reyes Torres

Doctor en Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2022

### **Dedicatoria**

Primeramente, a Nury y Esteban, mis padres, quienes han sido los pilares de mi vida y que, a través de su acompañamiento han formado la persona que soy hoy en día. A mi madre, que me guió, ayudó y aconsejó en los momentos más complicados de esta gran historia.

A Senaida, mi tía y Elisa María, mi abuela, quienes nunca me han desamparado, y al igual que mis padres, estuvieron presentes en cada día de mi formación.

A mis amigos, Andrea, Duván, Fabián, Helmunt, Juan Diego, Rafael y Sebastián, que fueron una constante guía en todo el pregrado, sin ellos esta historia fuese diferente; por sus grandes enseñanzas, buenos deseos y grandes acompañamientos. A todos ustedes dedico esto.

### **Agradecimientos**

Mi profundo y más grande agradecimiento a los profesores Óscar Reyes y Sergio Abreo por su constante guía y aprendizaje profesional en el desarrollo de mi proyecto de grado; quienes, a través de su experiencia no sólo fueron mis directores, sino también compañeros que aportaron grandes experiencias de vida y me dejan una gran enseñanza de vida: no importa que tanto debas preguntar algo, hazlo sin miedo.

Universidad Industrial de Santander, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, mi alma máter, que junto a sus docentes de gran calidad ayudaron en mi proceso de formación académica.

Al grupo de investigación en Conectividad y Procesamiento de Señales (CPS), y al Instituto Colombiano del Petróleo por ser partícipes en la culminación de este proyecto.

## Tabla de Contenido

<b>Introducción</b>	<b>12</b>
<b>1 Objetivos</b>	<b>15</b>
1.1 Objetivo General	15
1.2 Objetivos Específicos	15
<b>2 Marco de Referencia</b>	<b>16</b>
2.1 Escenario Geofísico	16
2.2 Redes Neuronales Tipo GAN	19
2.2.1 Topología de Red	20
2.2.2 Arquitectura de Red	21
2.3 Redes Neuronales Tipo GAN en la Sísmica	22
2.4 Descripción del Dataset	22
<b>3 Defeiniación del Sistema GAN</b>	<b>23</b>
3.1 Generación del Dataset	23
3.1.1 Datos Crudos: desde Seg-Y a Pandas	25
3.1.2 Vista Previa de las Trazas Generadas	25
3.2 Adecuación del Entorno y Scripts	26

USO DE REDES GAN PARA LA OBTENCIÓN DE SISMOGRAMAS ACÚSTICOS	5
3.2.1 Recursos Computacionales	28
3.3 Definición de los Sistemas Explorados	28
3.4 Diseño de Experimentos	32
3.5 Sistema cGeoGAN Seleccionado	32
3.5.1 Topologías	32
3.5.1.1 Red Generadora	34
3.5.1.1.1 Codificación	35
3.5.1.1.2 Decodificación	36
3.5.1.2 Red Discriminadora	36
3.5.2 Topología Definida	37
<b>4 Evaluación de Modelos Generados</b>	<b>38</b>
4.1 Resultados para Sistema 1 y Selección de Parámetros.	39
4.2 Resultados para Sistema 2 y Selección de Parámetros.	41
4.3 Resultados para Sistema 3 y Selección de Parámetros.	43
4.4 Resumen del sistema cGeoGAN	46
4.5 Comparación en presentación de datos	48
4.6 Costo computacional	49
<b>5 Conclusiones</b>	<b>50</b>
<b>6 Recomendaciones</b>	<b>51</b>

**Referencias Bibliográficas**

**53**

**Apéndice A**

**60**

**Lista de Figuras**

Figura 1	Esquema convencional de una red GAN	20
Figura 2	Flujo de adecuación de entorno y scripts	24
Figura 3	Ejemplo de parejas de trazas sísmicas del dataset.	26
Figura 4	Ejemplo de parejas de sismogramas desde el dataset	27
Figura 5	Arquitectura de una red GAN condicional (cGAN)	31
Figura 6	Topologías de red generadora	35
Figura 7	Topología de red discriminadora	37
Figura 8	Arquitectura cGAN para la generación de trazas acústicas	38
Figura 9	Resultado de las pérdidas respecto a normalización de los datos para sistema 1.	40
Figura 10	Resultados para el sistema 1	41
Figura 11	Resultado de trazas reales y generadas para el sistema 1	42
Figura 12	Identificación de modo de falla de convergencia del sistema 2	43
Figura 13	Métrica MSE para el sistemas cGeoGAN (filtro incluido)	45
Figura 14	Funciones de pérdidas para el sistema cGeoGAN	46
Figura 15	Evolución trazas generadas filtradas	48
Figura 16	Disparos generados con reducción de eventos elásticos	49
Figura 17	Sismogramas luego de entrenar ordenada y pseudo-aleatoriamente	50

**Lista de Tablas**

Tabla 1	Juego de datos Seg	25
Tabla 2	Características de recursos en hardware	28
Tabla 3	Sistemas GAN's explorados	29
Tabla 4	Parámetros de experimentos	33
Tabla 5	Características del sistema cGeoGAN	34
Tabla 6	Parámetros probados en sistema 1	39
Tabla 7	Parámetros probados para el sistema 2	42
Tabla 8	Parámetros probados para el sistema 3	44
Tabla 9	Arquitectura del sistema cGeoGAN definido	47

## **Lista de Apéndices**

**Apéndice A. Guía de instalación del paquete Anaconda.**

## Resumen

**Título:** Sistema basado en redes neuronales generativas adversarias (GAN) para la obtención de sismogramas acústicos a partir de sismogramas elásticos. \*

**Autor:** Danilo Ramírez Gómez. \*\*

**Palabras Clave:** Redes Neuronales, Generative Adversarial Networks (GANs), Sismogramas, Trazas Sísmicas.

**Descripción:** el experimento sísmico se puede definir como el registro de la propagación de ondas sísmicas generadas de forma artificial a través de un medio, que luego pasan a ser registradas en geófonos; las ondas registradas, también conocidas como trazas sísmicas, contienen algunas características del subsuelo como densidad, velocidad, anisotropía, entre otras (Krohn, 1984). Así mismo, este experimento se puede recrear de forma sintética empleando la ecuación de onda para modelar la propagación de la perturbación generada por la fuente y su interacción con el subsuelo, hasta obtener las correspondientes trazas sísmicas (Stein y Wysession, 2009). Debido a que el subsuelo es un medio elástico, las trazas obtenidas del experimento sísmico son esencialmente el resultado de la propagación de ondas elásticas; sin embargo, en muchos casos se desea observar sólo los eventos acústicos, ya que éstos han sido la base para la interpretación de sismogramas (Yilmaz, 2001).

Considerando lo anterior, este trabajo de investigación trata sobre la eliminación de algunos eventos elásticos (por ejemplo, fenómenos superficiales elásticos y ruido *ground roll*) presentes en ondas sísmicas generadas sintéticamente mediante una ecuación de onda elástica (traza elástica), transformándolas a su versión acústica por medio de una red neuronal. Para ello se hizo la implementación de una red neuronal generativa adversaria (GAN) 1-D, adoptando una estrategia de entrenamiento diferencial de las redes discriminadora (D) y generadora (G) (Goodfellow *et al.*, 2014), con el fin de que D sea capaz de aprender y ser selectivo con el conjunto de datos reales para que luego G pueda mejorar a través de la retropropagación del error (Gulrajani *et al.*, 2017). El entrenamiento se hizo por lotes con el fin de optimizar el proceso de aprendizaje y se realizó una comparación del desempeño de la red en cuanto a dos formas de presentación de datos: presentación de datos de forma aleatoria y ordenada. Los criterios de selección fueron basados en la función de costo, métricas de rendimiento para la red discriminadora (D) y generadora (G), y finalmente la calidad de las trazas generadas por el generador de acuerdo a la métrica. El sistema definido está conformado por una topología de red U-Net para el modelo G, para D una red convolucional con conexiones densas que comprenden desde la salida de la última capa convolucional hasta la capa de salida y una arquitectura de entrenamiento condicional para el entrenamiento. El sistema propuesto tiene la capacidad de generar datos a nivel de trazas y sismogramas; a nivel de sismogramas se evidencia la reducción de algunos eventos del dominio elástico.

---

\* Trabajo de Grado.

\*\* Facultad de Ingeniería Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Sergio Alberto Abreo Carrillo; Co-Director: Oscar Mauricio Reyes Torres

## Abstract

**Title:** System based on generative adversarial neural networks (GAN) for obtaining acoustic seismograms from elastic seismograms. \*

**Authors:** Danilo Ramírez Gómez. \*\*

**Keywords:** Neural Networks, Generative Adversarial Networks (GANs), Seismograms, Seismic Traces.

**Description:** the interaction of an energy source with the subsurface layers generates wavefield propagations through the medium. And the geophones are devices in charge of recording the wavefronts that return to the surface (e.g., reflections, refractions, and diving waves). The recorded waves, known as seismic traces, contain information about some subsurface characteristics or events such as density, velocity, anisotropy, among others (Krohn, 1984). The process described above is known as a seismic experiment. The synthetic version of a seismic experiment uses the wave equation and a source to model the wavefield propagation through a synthetic subsurface. And synthetic geophones to obtain the synthetic traces (Stein y Wyssession, 2009). Traces from seismic experiments are a mixture of elastic and acoustic events. Therefore, the elastic wave equation allows better describing the physical phenomenon. However, in many cases, the acoustic events are the desired input of processing stages due to their use for physical characteristics extractions (Yilmaz, 2001).

In the state-of-the-art, deep learning techniques have grown in their capabilities for processing and pattern recognition applications (Chang *et al.*, 2018). Therefore, some neural networks allow the processing and interpretation of seismic data. Some examples are a scalable deep learning platform for identifying geologic features from seismic attributes (Huang *et al.*, 2017) and generating velocity models for FWI from tomographic images using generative adversarial neural networks (GAN) (Arévalo, 2020). This work proposes a system-based GAN model to obtain acoustic seismograms from synthetic elastic seismograms. The system takes advantage of the spatial and temporal information of each trace together with the capacity of the GAN networks to produce patterns with the desired statistical characteristics (i.e., the acoustic seismograms). GAN networks evolve during training through an adversarial confrontation process, which involves two internal networks: the generator (G) and the discriminator (D). The network G extract features from the latent space to create the target data. In contrast, the network D estimates the probability to indicate whether the input data belongs to the desired dataset (Goodfellow *et al.*, 2014). The proposed architecture uses a U-Net network topology for model G; and a convolutional network for model D. Both networks use conditional training. The proposed architecture generates data at the level of traces and seismograms. And there is a reduction of some elastic events on the acoustic signals created.

---

\* Undergrade Work

\*\* Faculty of Engineering Physicomechanics. School of Electrical, Electronic and Telecommunications Engineering. Advisor: Sergio Alberto Abreo Carrillo; Co-Advisor: Oscar Mauricio Reyes Torres

## Introducción

Se han explorado múltiples métodos para obtener información de interés presente en sismogramas acústicos en las diferentes etapas del procesamiento sísmico. Algunos, por ejemplo, están relacionados con métodos de interpolación, como el propuesto por (Spitz, 1991) consistente en interpolación en el dominio  $f - x$ , interpolación de datos sísmicos basado en un filtro de preservación de bordes (Lu *et al.*, 2012), transformación dispersa adaptativa para la reconstrucción de datos sísmicos (Liang *et al.*, 2014). Sin embargo, estos métodos a pesar de ofrecer resultados verídicos para la interpolación y reconstrucción de datos sísmicos, conllevan limitaciones, algunas de estas son el alto costo computacional por el volumen de datos usado, y particularmente en la *FWI* surgen problemas de mínimos locales (Gauthier *et al.*, 1986; Xu *et al.*, 2012).

Recientemente, con la llegada del aprendizaje profundo y aprovechando el campo de reconocimiento de patrones, se han planteado algunas redes neuronales para procesamiento e interpretación de datos sísmicos (Chang *et al.*, 2018), como por ejemplo, identificar características geológicas en atributos sísmicos (Huang *et al.*, 2017) y generación de modelos de velocidad para inversión de onda completa (*FWI*) a partir de imágenes tomográficas usando redes neuronales generativas adversarias (*GAN*) (Arévalo, 2020).

Como es de saber, las trazas obtenidas del experimento sísmico son esencialmente el resultado de la propagación de ondas elásticas (puesto que el subsuelo es un medio elástico) y en

muchos casos se desea observar sólo los eventos acústicos, ya que éstos han sido la base para la interpretación de sismogramas (Yilmaz, 2001; Claerbout y Green, 2010). Algunos trabajos relacionados con la reducción de eventos elásticos y redes neuronales son la supresión de ruido ground roll (Klochikhina *et al.*, 2020; Oliveira *et al.*, 2020), limpieza de atributos sísmicos para datos reales y sintéticos (Li y Wang, 2021), aumento de datos sísmicos (Li *et al.*, 2020), entre otros.

En el presente trabajo de investigación se propone obtener sismogramas acústicos a partir de sismogramas elásticos sintéticos empleando un sistema basado en redes neuronales generativas adversarias (GAN), que está diseñado para aprovechar la información espacial y temporal de los sismogramas y la capacidad de las redes GAN para producir patrones con características estadísticas deseadas (en este caso, los sismogramas acústicos) a partir de los datos fuente (sismogramas elásticos sintéticos). Las redes GAN evolucionan durante el entrenamiento por medio de un proceso de confrontación entre dos redes internas de la GAN: la red generadora (G) y la discriminadora (D); G extraerá características de la distribución de los datos fuente para generar los datos objetivo y D estimará la probabilidad de que los datos generados correspondan realmente a la población de datos deseados (Goodfellow *et al.*, 2014)

En este trabajo se expone la implementación de una red GAN 1-D, para ello, los datos de los sismogramas se toman desde su archivo original (Seg-Y); el cual, mediante el uso de librerías específicas basadas en Python toma el archivo Seg-Y y lo transforma a una estructura de *DataFrame* (Equinor, 2022; NumFOCUS, 2022), que facilita su uso e interacción con las librerías Tensorflow y Keras (TensorFlow, 2022). Se propone un modelo de red GAN 1-D con el propó-

sito de pasar una traza a la vez en el entrenamiento de la red y aprovechar la eficiencia del costo computacional sobre una GPU Tesla K40c y una Tesla V100<sup>1</sup>, pues las operaciones realizadas en una dimensión son más eficientes, en comparación cuando se presentan datos 2-D o inclusive 3-D. El capítulo 2 comprende el marco de referencia, principalmente sobre el escenario geofísico y redes neuronales tipo GAN. En los capítulo 3 y 4 se proponen y comparan tres topologías de redes GAN, y se plantea un análisis preliminar de una cuarta topología. Se escoge una topología y se definen los hiperparámetros para llegar a un modelo final con una arquitectura correcta para la propuesta de ésta investigación. Finalmente se plantean conclusiones y recomendaciones para posibles exploraciones futuras y continuar con la ejecución de este trabajo.

---

<sup>1</sup> Para más información sobre las GPU's usadas, consultar la Tabla 2

## **1. Objetivos**

### **1.1. Objetivo General**

Diseñar un sistema basado en redes neuronales generativas adversarias (GAN) para obtener sismogramas acústicos a partir de sismogramas elásticos.

### **1.2. Objetivos Específicos**

Definir una topología basada en redes GAN adecuada para generar trazas acústicas, a partir de la realización de pruebas con distintas combinaciones de parámetros.

Evaluar el rendimiento del modelo GAN definiendo al menos dos métricas de desempeño desde las planteadas en la librería Keras.

Comparar los resultados obtenidos de la red presentándole de dos formas diferentes los datos en relación con el tipo de sismograma.

## 2. Marco de Referencia

A continuación se hará una introducción sobre algunos temas que conforman el escenario sísmico y algunos subtemas como la sísmica, métodos comunes de procesamiento sísmico y algunas de las limitaciones de métodos tradicionales; además, se presenta la definición de redes neuronales generativas adversarias o GAN (por sus siglas en inglés), y finalmente, un poco de contexto del uso de redes neuronales con la sísmica y el interés particular sobre el tema de investigación.

### 2.1. Escenario Geofísico

La geofísica se define como la ciencia que trata con los procesos y fenómenos físicos que ocurren esencialmente en la tierra. Es decir, mediante la aplicación de los principios de la física se estudia el comportamiento sobre el entorno de la tierra de manera interna y externa. El estudio externo comprende las propiedades físicas del entorno terrestre, algunos estudios se relacionan con el campo magnético terrestre (geomagnetismo), otros con los fenómenos gravitacionales (gravimetría), el océano (oceanología), entre otros. En el estudio interno, como su nombre lo indica se encarga del estudio de las capas que conforman el planeta, algunos de los estudios se relacionan con los terremotos y las propagaciones en el subsuelo de ondas de origen natural (sismología) y de origen artificial (sísmica), la propagación del calor relacionado con el vulcanismo (geotermometría), entre otras (Stein y Wysession, 2009).

Como se mencionó, la sismología y la sísmica tratan el estudio de la propagación de las

ondas (elásticas) al interior de la tierra. Para generar una propagación sobre un medio se debe contar con una fuente generadora de pulsos de energía que puede presentarse de forma natural (por ejemplo, terremotos) o artificiales (a través de explosiones), acompañada de un conjunto de elementos que se encargan del registro de los frentes de ondas como estaciones sismológicas, geófonos o hidrófonos, los cuales permiten capturar las ondas que se reflejan y/o refractan hacia la superficie (Claerbout y Green, 2010). La información sobre la propagación y el registro de la onda (traza sísmica) es contenida en un sismograma. Conociendo el tiempo de origen (cuando las ondas dejan la fuente) y su tiempo de llegada al receptor (sismógrafo, geófono o hidrófono) deja como resultado el tiempo de viaje requerido para atravesar el medio. Este proceso permite conocer la velocidad a la que viajaron, lo que abre paso a conocer algunas propiedades físicas del medio, tales como densidad, velocidad, esfuerzo, anisotropía, viscosidad, entre otras (Stein y Wysession, 2009; Ben-Menahem y Singh, 2012). A este proceso se le conoce como el experimento sísmico (Krohn, 1984). La interpretación de los sismogramas adquiridos tras el experimento sísmico permiten el estudio sobre el interior de la tierra. Con el paso de los años se han explorado múltiples y diversos métodos para el procesamiento de los sismogramas (conjunto de trazas sísmicas), algunos por ejemplo, están relacionados con interpolación en el dominio  $f - x$  (Spitz, 1991), interpolación para la reconstrucción de trazas sísmicas (Liang *et al.*, 2014); otros métodos se relacionan con el mejoramiento de resolución espacial del subsuelo usando la inversión de onda completa (*FWI* por sus siglas en inglés) (Gauthier *et al.*, 1986; Sirgue y Pratt, 2004), entre otros tipos de procesamiento. Según Gauthier *et al.* (1986) y Xu *et al.* (2012), existen algunas limitaciones asociadas a estos métodos tradicionales, entre ellos se encuentran el alto costo computacional por el volumen de

datos usado, y particularmente en la *FWI* surgen problemas de mínimos locales.

Diferentes métodos y estrategias de procesamiento que se han explorado y desarrollado son efectivas cuando reciben datos que sólo reflejan características y propiedades acústicas del medio, pero, como se sabe, los sismogramas son el resultado de propagaciones de onda tipo elásticas (Ben-Menahem y Singh, 2012). En algunos casos, se busca reducir ruido proveniente de las ondas elásticas, como por ejemplo el ruido *ground roll*, dado que el movimiento de tierra se caracteriza por frecuencias relativamente bajas y amplitudes altas, los enfoques comúnmente utilizados para su supresión se basan en filtros de frecuencia-amplitud. Sin embargo, cuando la señal y el ruido comparten los mismos rangos de frecuencia, estos métodos suelen proporcionar también supresión de la información de la señal deseada (Yilmaz, 2001; Li y Wang, 2021). En algunos casos, los métodos comunes deben elegir entre atenuar este tipo de ruido y a la vez parte de la señal ó la conservación de la señal con algo de este tipo de ruido; algunas técnicas comunes son el filtrado *f-k* (Yilmaz, 2001), transformada de radón (Henley, 2003) y la atenuación de ruido de rango de frecuencia mediante la tecnología LIFT (Choo *et al.*, 2004). Es por esto que dentro del algunos tipos de procesamiento de datos sísmicos se intenta mitigar los eventos y características elásticas de los sismogramas con el fin de realizar mejores aproximaciones del subsuelo, ya que, en su mayoría, las herramientas que se usan en la actualidad son efectivas cuando se presentan datos netamente acústicos (Yilmaz, 2001; Claerbout y Green, 2010).

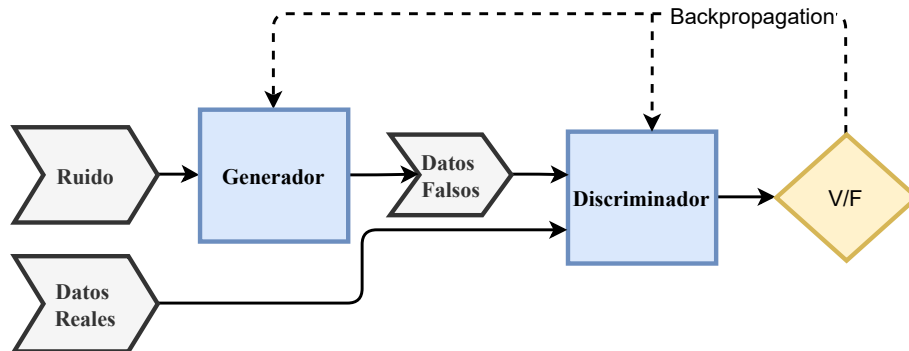
## 2.2. Redes Neuronales Tipo GAN

Este tipo de redes aparecieron por primera vez en el trabajo de Goodfellow *et al.* (2014), donde se describe a la red generativa adversaria ó GAN como un conjunto de dos redes en competencia, una red discriminadora (D) encargada de identificar si su entrada posee o no las características o distribución estadística de un conjunto de datos de referencia, y una red generativa (G) encargada de generar nuevas muestras plausibles que podrían haberse extraído del conjunto de datos originales (Goodfellow *et al.*, 2014). En el proceso de entrenamiento las dos redes entran en una confrontación, la red G mejora cuando D decae y viceversa (proceso conocido como algoritmo de suma cero) (Washburn y Wood, 1995), donde G tratará de generar nuevas muestras plausibles y D decidirá si esta nueva muestra generada es real o falsa. Lo que se busca con la red GAN es que G pueda “engañar” a D para que esa nueva muestra generada sea aceptada como una muestra verdadera. Como punto final, durante el entrenamiento se va guardando un historial de los modelos que permiten ver su evolución (a través de métricas y función de pérdida); el esquema de una red GAN se presenta en la Figura 1.

Para lograr el objetivo de engañar a la red discriminadora (D) se han explorado diferentes configuraciones de arquitectura y topologías de red. Por ejemplo, en el trabajo elaborado por Radford *et al.* (2015), se introdujo la DCGAN (GAN convolucional profunda) que se ha convertido en la base estándar para el desarrollo de otras configuraciones basadas en redes GAN (Brownlee, 2019), Radford y Chintala, mencionan algunas recomendaciones y consideraciones para mantener estable el entrenamiento de la una red GAN, algunas de estas recomendaciones son el uso de nor-

*Figura 1.*

Esquema convencional de una red GAN



malización de los datos, funciones de activación en capas específicas, como por ejemplo ReLu y LeakyRelu, entre otras (Chintala, 2016, 2017). Algunas aplicaciones de las redes GAN son: reconocimiento de imágenes condicionadas a etiquetas de clase (Radford *et al.*, 2015), traducción de imagen a imagen condicionadas a una etiqueta de clase (Isola *et al.*, 2017), traducción de imagen a imagen mediante implementaciones de ciclos, que a diferencia de los trabajos anteriores, en éste no se realiza el emparejamiento entre los dominios de imágenes (Zhu *et al.*, 2017).

### 2.2.1. Topología de Red

La topología de red se refiere a la estructura interna de los modelos de redes neuronales, como las conexiones y configuración de hiperparámetros. Algunos parámetros de las topologías son el tipo de capa (por ejemplo, convolucionales, densas, lstm), número de neuronas por capa, funciones de activación, entre otras (Wilamowski, 2009; Keras, 2022). En este proyecto se exploran aspectos topológicos para el diseño de la red que comprenden: exploración de estructuras de redes, es decir, configuraciones de algunos modelos de red como auto-codificadores (Sohn *et al.*,

2015), U-Net (Ronneberger *et al.*, 2015), convolucionales profundas CNN (Radford *et al.*, 2015). Estructura interna de los modelos como número de capas, funciones de activación como ReLu, LeakyRelu, tanh; y otros parámetros como el *Dropout*, *BatchNormalization*, capas convolucionales y convolucionales transpuestas 1-D (Keras, 2022; Linder-Norén, 2022). Hiperparámetros de la red, como el número de neuronas por capa, tamaño del kernel, paso del kernel (*stride*) para las capas convolucionales, entre otros.

### **2.2.2. Arquitectura de Red**

La arquitectura de red está asociada a la forma de entrenamiento entre los dos modelos de la red, específicamente a esto se refiere el concepto de red GAN (proceso adverso entre los modelos G y D). Existen diferentes tipos de arquitecturas de entrenamiento dentro de las GAN, algunos por ejemplo son: GAN estándar, donde G y D no se acompañan de información adicional, su característica principal es encontrar un equilibrio entre los modelos (Goodfellow *et al.*, 2014), GAN's condicionales (cGAN), que toman como referencia la propuesta por Goodfellow *et al.* (2014) y en las redes G y D los datos los emparejan con una etiqueta de clase con la intención de tener un poco más de control sobre la generación de los datos (Mirza y Osindero, 2014), GAN's cíclicas sin emparejamiento, las cuales permiten traducción o transformación de un dominio A a un dominio B y viceversa (Zhu *et al.*, 2017). La idea general en esta investigación es buscar el tipo de arquitecturas que se adapten al problema planteado; en este caso se realizó una exploración con detalle sobre las cGAN, el detalle de esta tipo de red y la adecuación para el proyecto mismo se puede consultar en la sección 3.3.

### 2.3. Redes Neuronales Tipo GAN en la Sísmica

El uso del *deep learning* ha abierto las puertas a la implementación de nuevos algoritmos orientados al procesamiento de datos sísmicos disminuyendo las limitaciones mencionadas en la sección 2.1, así como también la tarea de reducir incertidumbre en la eliminación de ruido. Por ejemplo, el uso de redes GAN para la interpolación de datos sísmicos (Chang *et al.*, 2018), uso de redes GAN para la construcción rápida y precisa de datos sísmicos muy submuestreados (Siahkoobi *et al.*, 2018), generación de modelos de velocidad para inversión de onda completa mediante el uso de redes GAN (Arévalo, 2020), supresión de ruido específicamente el ruido *ground roll* mediante una red GAN 1-D (Klochikhina *et al.*, 2020), generación de ondas sísmicas reales mediante una red GAN Condicional 1-D (Li *et al.*, 2020), eliminación del ruido aleatorio presente en sismogramas mediante una red de ciclos GAN (Li y Wang, 2021), entre otros.

### 2.4. Descripción del Dataset

Un formato es una estructura pre-establecida para almacenar un conjunto de datos determinados (RAE, 2021, definición 3), a su vez, esta estructura puede estar organizada en una matriz (RAE, 2022, definición 10). Hoy en día, en la informática también se habla de término *dataset*, que es un extranjerismo adaptado a la lengua hispanohablante que se refiere al conjunto de datos.

En este trabajo, el dataset consiste en dos conjuntos de ondas ó trazas sísmicas de tipo elásticas y acústicas, y a su vez contienen información adicional conocida como metadatos. Esta información se encuentra almacenada en formato Seg-Y, que es una norma establecida por la

Sociedad de Exploración Geofísica (SEG) para contener datos de adquisiciones geofísicas (SEG, 2017). Mientras se realiza una adquisición de campo, los geófonos pasan a registrar las ondas (trazas sísmicas) que fueron refractadas hacia la superficie y pasan a almacenarse. El formato Seg-Y permite guardar un conjunto de metadatos que brindan información adicional de la adquisición, como por ejemplo, frecuencia de muestreo, punto de profundidad común (CDP), offset, elevación en X e Y, cantidad de muestras por traza, número de grabación (único para cada geófono), entre otras (SEG, 2017).

### **3. Defeinición del Sistema GAN**

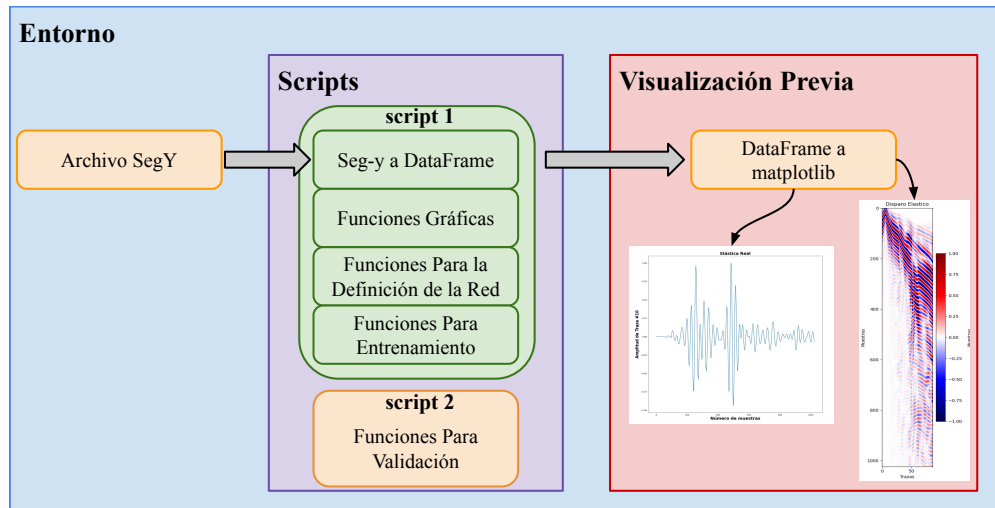
Este capítulo aborda la generación y adaptación del dataset, presentación de sistemas GAN's en el contexto del procesamiento de datos sísmicos, y la serie de experimentos realizados para la elección de topologías y arquitectura adecuada para el objetivo de la investigación. La Figura 2 ilustra el trabajo desarrollado para la adecuación del entorno y scripts de trabajo descrito en las secciones 3.1 y 3.2.

#### **3.1. Generación del Dataset**

Para lograr un entrenamiento con resultados adecuados es necesario un amplio dataset; esto se hace con la intención de evitar al máximo una particularidad en el entrenamiento de redes neuronales (sobre-entrenamiento), que básicamente se refiere a la memorización del conjunto de datos en lugar de realizar una generalización (Brownlee, 2019). Por otra parte, es de gran importancia disponer de una porción del dataset con muestras considerables para realizar la evaluación

Figura 2.

Flujo de adecuación de entorno y scripts



*Nota.* La sección 3.1 explica la generación de los datos Seg-Y. La sección 3.2 hace la claridad sobre la adaptación del entorno, conversión de los datos Seg-Y, pre-visualización y demás scripts.

del modelo (cuando se le presentan a la red muestras que nunca ha visto). En la Tabla 1 se muestran los juegos de datos usados para realizar las pruebas. Un mismo paquete de datos corresponde a la versión de los frentes de ondas elásticas (trazas sísmicas) y su respectiva versión acústica.

El conjunto de datos usado en este proyecto fue elaborado y proporcionado por el grupo de investigación Conectividad y Procesamiento de Señales (CPS). Tanto el conjunto de trazas elásticas, como el respectivo conjunto de trazas acústicas corresponde a datos sintéticos generados mediante la simulación del experimento sísmico empleando modelos de propagación basados en la ecuación de onda elástica y onda acústica (Jeffreys *et al.*, 1999), respectivamente. Cada una de las trazas contiene algunos de los metadatos pertenecientes al formato Seg-Y.

Tabla 1  
*Juego de datos Seg*

<b>Juego de datos: archivos Seg-Y</b>				
		<b>Tipo Traza</b>	<b>Entrenamiento</b>	<b>Validación</b>
Dataset	<b>Dip_low_source</b>	Elásticos	80 %	20 %
27760*	<b>Dip_low_target</b>	Acústicos	80 %	20 %

*Nota.* Los datos tienen un peso de 138 MegaBytes. La separación del conjunto de datos se realiza de forma pseudo-aleatoria. Para garantizar repetibilidad de los experimentos se elige una semilla (`random_state=4` desde `sklearn.model_selection.train_test_split()`).

### ***3.1.1. Datos Crudos: desde Seg-Y a Pandas***

Al disponer el conjunto de los datos en un formato Seg-Y (obtenidos de forma sintética) se elaboró una función que permitiera trabajar desde Python, para ello, fue necesario usar una librería específica para la carga y lectura del archivo Seg-Y llamada `seggio` (Equinor, 2022), posteriormente, mediante la librería `pandas`, el archivo Seg-Y fue transformado en un `DataFrame`.

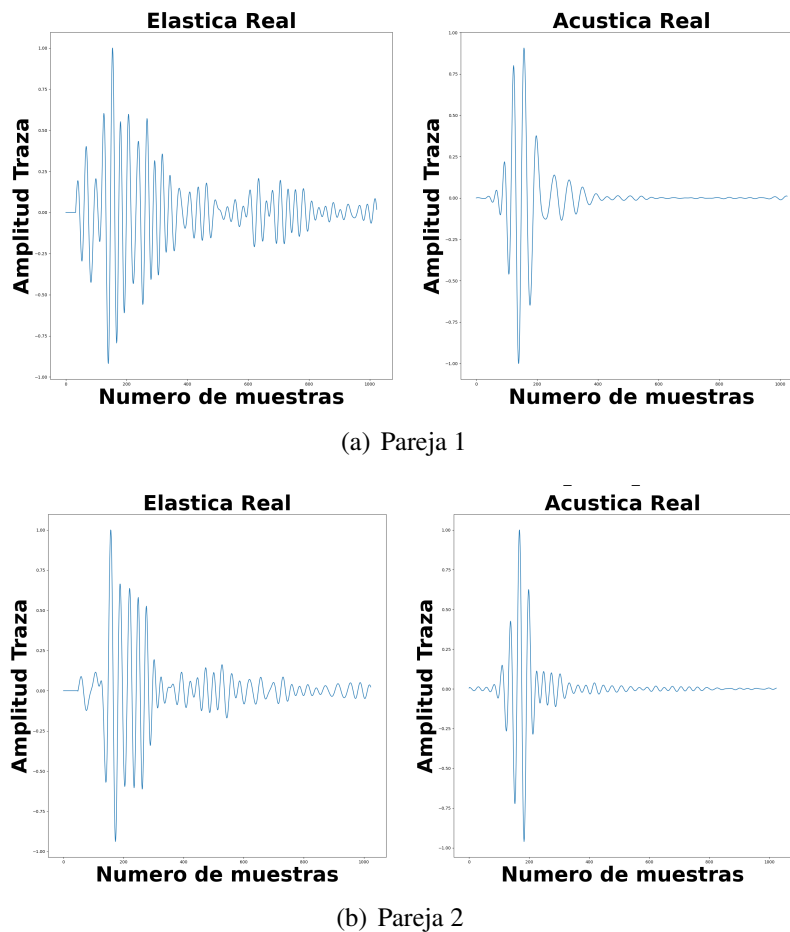
### ***3.1.2. Vista Previa de las Trazas Generadas***

Una vista previa de las trazas sueltas desde el dataset se muestra en la Figura 3, están organizadas en parejas: una traza elástica y su respectiva versión acústica. Adicionalmente, las trazas agrupadas (sismogramas) suelen ser visualizadas mediante alguno de sus metadatos, comúnmente se agrupan mediante `offset común`, `disparo común` o `receptor común`. En la Figura 4 se pueden ver unos ejemplos para `disparo común` y `offset común`.

### 3.2. Adecuación del Entorno y Scripts

Para la elaboración de los experimentos se adecuaron dos entornos: “Anaconda” y “Doc-ker” (Anaconda, 2022; Merkel, 2022). En los cuales se hizo la instalación de los paquetes nece-

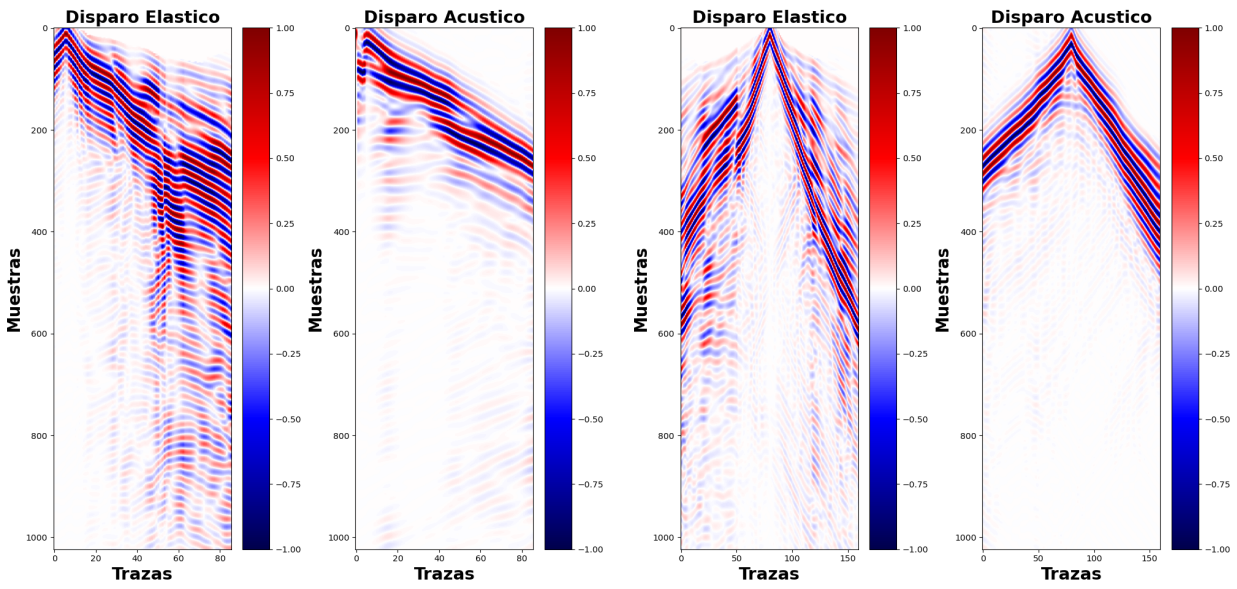
*Figura 3.*  
Ejemplo de parejas de trazas sísmicas del dataset.



*Nota.* Generalmente la amplitud de las trazas elásticas contienen mayor amplitud respecto a las trazas acústicas; en este caso se muestran las trazas normalizadas respecto al máximo absoluto para cada traza.

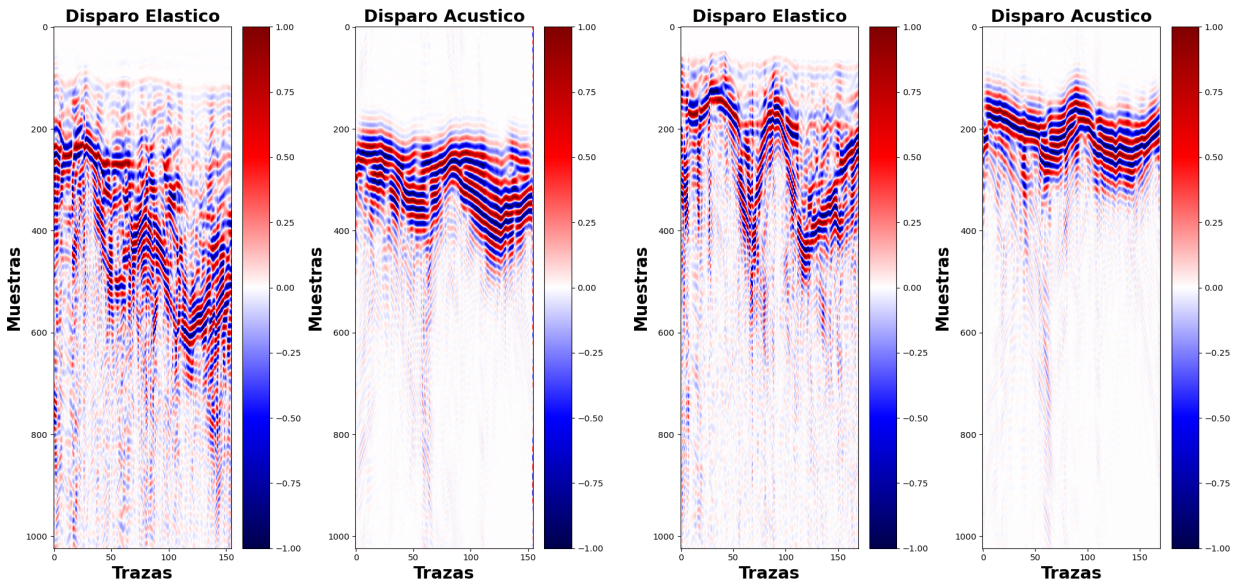
Figura 4.

Ejemplo de parejas de sismogramas desde el dataset



(a) Pareja 1: Disparo Común #3

(b) Pareja 2: Disparo Común #127



(c) Pareja 3: Offset Común #3

(d) Pareja 4: Offset Común #127

Tabla 2

*Características de recursos en hardware*

<b>Especificaciones de GPU's usadas (Clúster CPS)</b>			
<b>Nombre del nodo</b>	<b># GPU's Disponibles</b>	<b>Tipo GPU</b>	<b>Memoria Ram [GB]</b>
Tokyo-00	2	Tesla K40c	12
Kioto	8	Tesla V100	16

sarios para el correcto funcionamiento de los script de programación de la red neuronal definida. El código de programación de la red neuronal fue basado en Python (versión 3.9); otras librerías usadas fueron tensorflow-gpu (versión 2.4.1), keras (incluída ya en tensorflow desde la versión 2.0 en adelante), segyio, pandas, scipy, sklearn, entre otras.

La instalación del paquete Anaconda se puede consultar en el apéndice A que se encuentra en la página 60. Adicional, también se dispone un repositorio web con acceso de lectura, el cual contiene los scripts mencionados en la Figura 2 y otros archivos que complementan el trabajo.

### ***3.2.1. Recursos Computacionales***

Los scripts de entrenamiento y validación del sistema cGeoGAN se ejecutaron en un entorno de Anaconda dentro del clúster computacional del que dispone el grupo de investigación CPS. Las características de los recursos de hardware se pueden consultar en la Tabla 2.

### **3.3. Definición de los Sistemas Explorados**

Es importante aclarar que un sistema GAN es equivalente a una arquitectura de red, para más información ver sección 2.2.2. Los sistemas enumerados en la Tabla 3 se mencionan a

continuación.

El primer sistema estudiado fue el acondicionamiento de una red cGAN 1-D para la generación de ondas sísmicas para aumento de datos (Li *et al.*, 2020), ya que en contraste con el objetivo de este trabajo, el sistema que se está buscando definir tiene gran similitud en el apartado de generación de ondas sísmicas cuando se presentan los datos unidimensionales. En esta propuesta para la red generadora (G) se implementa un topología conocida como U-Net (Ronneberger *et al.*, 2015) y la red discriminadora (D) está formada por una serie de capas convolucionales y una capa completamente conectada (red híbrida); donde los modelos de cada red están configurados para recibir y procesar datos unidimensionales.

El segundo sistema explorado fue el uso de auto-codificadores variacionales condicionales (cVAE) como propuesta del modelo de red generadora (Sohn *et al.*, 2015), según Özdenizci y Erdoğmuş (2021), una cVAE se puede adecuar para sintetizar señales de electroencefalografía (EEG), aprovechando una codificación estocástica y una decodificación determinística para la generación de los datos EEG (para la reconstrucción de señales). En esta implementación se tomó cVAE como

Tabla 3  
*Sistemas GAN's explorados*

<b>Sistemas cGeoGAN</b>			
<b># Sistema</b>	<b>Generador</b>	<b>Discriminador</b>	<b>Arquitectura</b>
01	U-Net	Híbrido	cGAN 1-D
02	Auto-codificador	Hibrido	cGAN 1-D
03	U-Net	Crítico	cGAN Wasserstein 1-D

propuesta de red G 1-D y para la red D se estableció la misma del sistema uno, es decir, la red híbrida; consecuentemente, se planteó una arquitectura cGAN 1-D con los nuevos modelos G y D.

El tercer sistema explorado está fundamentado por una arquitectura de red denominada Wasserstein GAN (WGAN) (Gulrajani *et al.*, 2017). Los modelos de red usados para la red generadora es U-Net y en la red discriminadora un modelo crítico. El modelo crítico, principalmente tiene como base el trabajo de la DCGAN (capas convolucionales) elaborado por Radford *et al.* (2015) y al final está compuesta por una serie de capas densas. El modelo crítico a diferencia de un discriminador de clasificación o predicción para datos real o falsos clasifica la realidad o falsedad de los datos, ya que el modelo crítico busca la relación entre la distribución de datos observados sobre la distribución de datos generados. Otra diferencia entre la WGAN y la DCGAN radica en que se busca una convergencia en las pérdidas del modelo del discriminador; mientras que en la DCGAN se busca un equilibrio en las pérdidas entre los modelos. Otras características de la WGAN son el entrenamiento mas frecuente para el modelo critico que para el generador, para el caso del optimizador se recomienda usar *Root Mean Square Propagation* (RMSProp por sus siglas) (Gulrajani *et al.*, 2017; Arjovsky *et al.*, 2017).

De lo anterior, hay salvedad de que las exploraciones de las arquitecturas estaban ligadas con las topologías que se definían en el estado del arte; en resumen, las arquitectura explorada fue las GAN's condicionales (cGAN) (Mirza y Osindero, 2014) y cGAN Wasserstein (WGAN) (Gulrajani *et al.*, 2017).

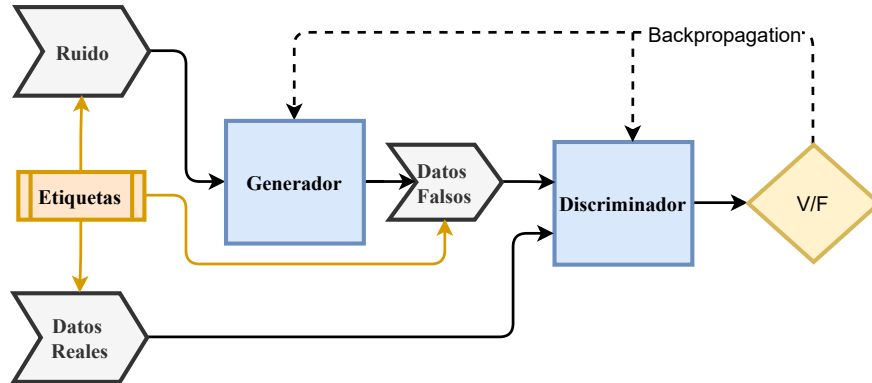
En la literatura, las cGAN han sido propuestas para procesamiento de datos sísmicos, prin-

principalmente basadas en Pix2Pix (Isola *et al.*, 2017), algunas implementaciones son mencionadas en los trabajos elaborados por Arévalo (2020), Klochikhina *et al.* (2020), y Li *et al.* (2020). La principal diferencia entre el primer trabajo y los dos últimos radica en la presentación de los datos, siendo imágenes (2-D) y datos unidimensionales (1-D), respectivamente. La arquitectura cGAN convencional se muestra en la Figura 5.

Según Mirza y Osindero (2014), la información adicional como las etiquetas de clases o datos de otras modalidades han permitido que el aprendizaje en redes generativas puedan generar no solo muestras plausibles a partir del ruido dando alta precisión, sino también tener la posibilidad de tener control o dirigir los datos que el generador pueda crear.

*Figura 5.*

Arquitectura de una red GAN condicional (cGAN)



*Nota.* Representación de una red GAN condicionada a una etiqueta de clase, las etiquetas se emparejan con los datos que van hacia el generador (G) y al discriminador (D).

### 3.4. Diseño de Experimentos

Como se puede observar, las pruebas de algunos de los parámetros de las topologías de la red están condensados en la Tabla 4. La Tabla 3 muestra los sistemas a los cuales se le aplicaron los tests. Es importante mencionar que los parámetros observados en la Tabla 4 están organizados de tal forma que, tras realizar los tests se seleccione el mejor desempeño del parámetro y luego se pase al siguiente parámetro. Adicionalmente, las pruebas sobre los sistemas se hizo de forma secuencial por parámetro para ir fijando un parámetro antes de probar el siguiente.

### 3.5. Sistema cGeoGAN Seleccionado

Tras los experimentos enumerados en la Tabla 4 sobre los sistemas definidos en la Tabla 3 se tomó la decisión de elegir el tercer sistema, las características en resumen del Sistema cGeoGAN están condensadas en la Tabla 5. Para mayor claridad sobre la elección de los parámetros del sistema consultar la sección 4.

#### 3.5.1. Topologías

La topología asociada al modelo del generador (G) está tomado de un esquema *U-NET* (Ronneberger *et al.*, 2015), a su vez, el modelo G está conformado por bloques codificadores y decodificadores (Isola *et al.*, 2017), la conexión se muestra en Figura 6. Por otro lado, la topología asociada al modelo del discriminador (D) está conformado por capas convolucionales profundos seguido de una serie de capas densas (Gulrajani *et al.*, 2017), las conexiones del modelo D se muestran en la Figura 7.

Tabla 4

*Parámetros de experimentos*

<b>Pruebas Realizadas</b>					
<b>Tamaño batch</b>	2220	200	100	20	50
<b>Tamaño Kernel</b>	10	20	5	4	-
<b>Normalización de datos</b>	original	-1,1	tanh	-	-
<b>Valor n_critico</b>	1	2	3	4	5
<b>Tamaño Salida Discriminador</b>	1	4	-	-	-
<b>Función de Costo</b>	binary_crossentropy	mse	Wasserstein	-	-
<b>Optimizador</b>	Adam	SGD	RMSProp	-	-
<b>Numero de Capas*</b>	6	7	5	-	-
<b>Aplicación de filtro</b>	Si	No	-	-	-

*Nota.* \* Parámetro aplicado sobre la red D. El color sobre las casillas es la selección del parámetro que mejor desempeño tuvo. - (Guión) equivale a test no realizado.

A diferencia de algunos trabajos donde usan la *U-NET* como los expuestos por Siahkoohi *et al.* (2018) y Oliveira *et al.* (2019) y los datos de entrada son imágenes de representaciones sísmológicas y/o sísmicas (2-D), en el presente trabajo, los datos que entran al sistema cGeoGAN son directamente extraídos del Seg-Y y a su vez permite un tratamiento directamente sobre las

Tabla 5

*Características del sistema cGeoGAN*

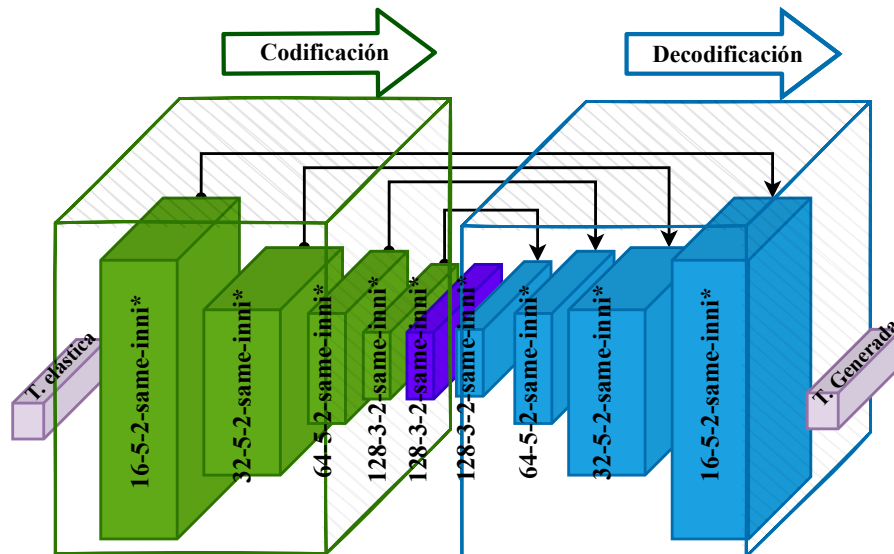
<b>Tamaño batch</b>	<b>Tamaño Kernel</b>	<b>Normalización de datos</b>	<b>Valor n_critico</b>	<b>Tamaño Salida Discriminador</b>
50	4	-1, 1	3	4
<b>Función de Costo</b>	<b>Optimizador</b>	<b>Numero de Capas</b>	<b>Aplicación de filtro</b>	
mse	RMSProp	6	Si	

trazas obtenidas, las trazas están contenidas en una lista unidimensional (1-D); es por eso que los bloques convolucionales tanto en G como D son 1-D. Según Isola *et al.* (2017), la configuración de los hiperparámetros de la red GAN no son estimados fácilmente, si no más bien es un proceso heurístico donde la prueba y error juega un papel importante; no obstante, en este trabajo se tomó como punto de partida las topologías y arquitectura de red presentada por Li *et al.* (2020) y luego se fue afinando a través de los tests definidos en la Tabla 3 para tener una versión final del sistema cGeoGAN. La versión final del sistema cGeoGAN se describe en la Tabla 5 y finalmente, la justificación se expone en la sección 4.

**3.5.1.1. Red Generadora.** El modelo de la red generadora a partir del entrenamiento con trazas elásticas generará versiones acústicas de la misma, es decir, tomando como entrada trazas del dominio elástico, el modelo aprenderá a extraer características ó patrones que permitan la reconstrucción de trazas pertenecientes al dominio acústico. Para lograr que la red generadora logre realizar esta tarea, en el estado del arte se presenta una topología de red denominada codificación - decodificación presentada inicialmente por Isola *et al.* (2017), que se asemeja a un reloj de arena

Figura 6.

Topologías de red generadora



*Nota.* La configuración en las capas indican: #NeuronasPorCapa - #TamañoKernel - #Stride - TipoRelleno - InicializadordePesos(inni), inni es distribución normal con media cero y desviación estándar de 0.02

(ver Figura 6).

**3.5.1.1.1. Codificación.** El generador se encarga de mapear o extraer características a partir de las trazas elásticas mediante la codificación, que a su vez está formado de filtros convolucionales y activaciones no lineales que ayudan en el proceso del mapeo. Cada capa o bloque tiene la tarea de extraer o reconocer algunos patrones característicos de las trazas. Al final de esta etapa se tiene un mapa comprimido con algunas de las características esencialmente de la traza fuente. Cada bloque convolucional está compuesto de un capa convolucional, una capa *BatchNormalization* y activación no lineal ReLu (Keras, 2022).

**3.5.1.1.2. Decodificación.** Luego de la etapa codificadora, los datos ahora hacen parte de un espacio reducido (conocido comúnmente como espacio latente), donde, a partir de éste y junto con la configuración de parámetros e hiperparámetros de las capas, se inicia el proceso de “descompresión” para la generación de trazas acústicas. La configuración de cada bloque está conformada por una capa convolucional transpuesta, una capa *BatchNormalization*, una función de activación *LeakyRelu* y *dropout*, que permiten la reconstrucción de la traza al tamaño original de entrada y con el mayor contenido acústico posible al final del proceso de decodificación. Además, junto a la conexión U-Net permite un posible control o direccionamiento sobre la traza acústica generada, el cual funciona dando una condición sobre los datos que la decodificación está tratando de reconstruir (Ronneberger *et al.*, 2015).

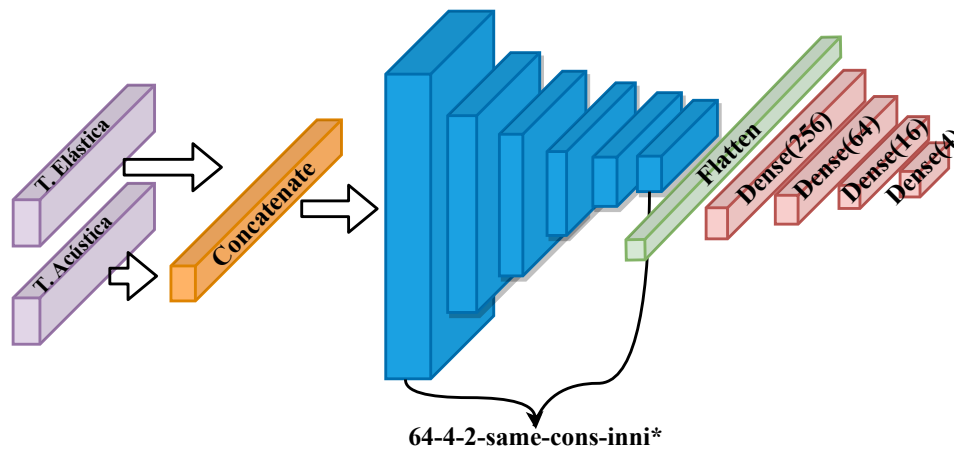
**3.5.1.2. Red Discriminadora.** El modelo de la red discriminadora a partir del entrenamiento tiene la tarea de clasificar y/o identificar si la traza acústica que esté generando la red generadora sea la versión acústica de la elástica presentada, a su vez está acompañada de una etiqueta de clase, la cual le guía a la red discriminadora para identificar si la pareja presentada corresponde a la pareja real o falsa. Mediante la configuración:

$$[(TrazaAcustica, TrazaElastica), Etiqueta(Real/Falsa)].$$

Para lograr esta tarea, la red discriminadora antes de identificar si la pareja corresponde, se entrenará para que sea capaz de reconocer los patrones que hagan parte meramente de las trazas acústicas mediante un dataset de trazas reales acústicas y elásticas reales, para la lograr extraer

Figura 7.

Topología de red discriminadora



*Nota.* T.E. y T.A. corresponde a traza elástica y traza acústica respectivamente. Los parámetros indican: #NeuronasPorCapa - #TamañoKernel - #Stride - TipoRelleno - Constrain - InicializadordePesos(inni), Constrain representa limitador de pesos, inni representa una distribución normal con media 0 y desviación estándar de 0.02.

los patrones se usarán principalmente capas convolucionales 1-D y funciones de activaciones no lineales.

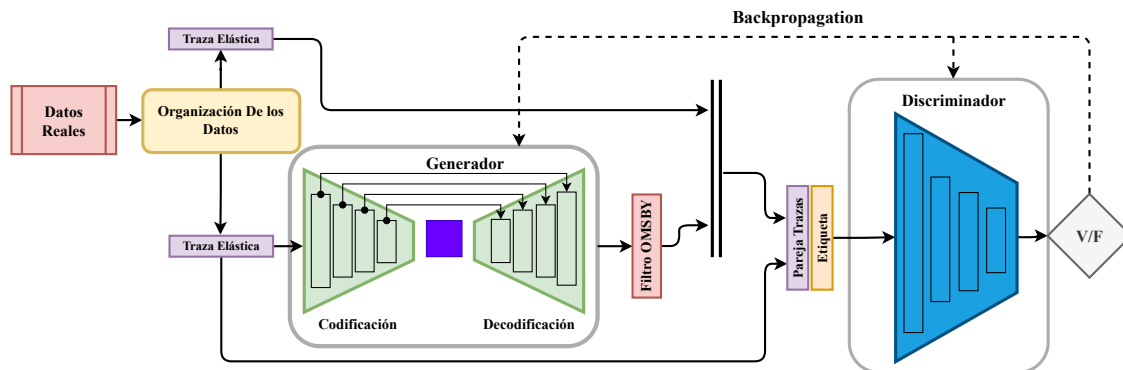
La configuración de los hiperparámetros en las capas de las redes generadora y discriminadora están definidos en el siguiente orden: (Filtros) (TamañoKernel) (Stride) (padding) (InicializadordePesos). La red D está ilustrada en la Figura 7.

### 3.5.2. Topología Definida

La Figura 8 muestra el esquema de la arquitectura propuesta basada en una red cGAN 1-D con etiquetas de clase para clasificar el emparejamiento (trazas elásticas, trazas acústicas) como verdadero o falso. Esta arquitectura se basa en el principio de la WGAN (Arjovsky *et al.*,

Figura 8.

Arquitectura cGAN para la generación de trazas acústicas



*Nota.* La condicionalidad se hace mediante etiquetas de clase, las etiquetas se presentan junto a las trazas que ingresan en parejas al modelo discriminador. D recibe parejas formadas desde conjunto de datos reales y de los generados (cuando G se entrene).

2017); la característica principal de este sistema GAN consiste en entrenar  $n$  veces más al modelo discriminador que al generador con el fin de que D sea capaz de aprender y ser selectivo con el conjunto de datos reales para que luego G pueda mejorar a través de la retropropagación del error (Gulrajani *et al.*, 2017).

#### 4. Evaluación de Modelos Generados

En este capítulo se abordan las pruebas realizadas sobre los sistemas cGeoGAN definidos en la Tabla 3 y la serie de experimentos realizados para obtener el sistema final. Los desempeños de los sistemas fueron evaluados de acuerdo a las funciones de pérdida, métricas y el desempeño con capacidad de generar trazas acústicas, que, en este caso se hizo mediante la métrica error cuadrático medio (mse por sus siglas en inglés) entre la traza acústica deseada y la generada.

Tabla 6

*Parámetros probados en sistema 1*

Normalización de datos	Tamaño batch	Función <sup>x</sup> de costo	Tamaño Salida Discriminador
original/-1 a 1/tanh	2200/200/100/20	binary_crossentropy/mse	1/4

*Nota.* El color resaltado indica la sintonización que presenta mejor desempeño respecto los experimentos realizados.

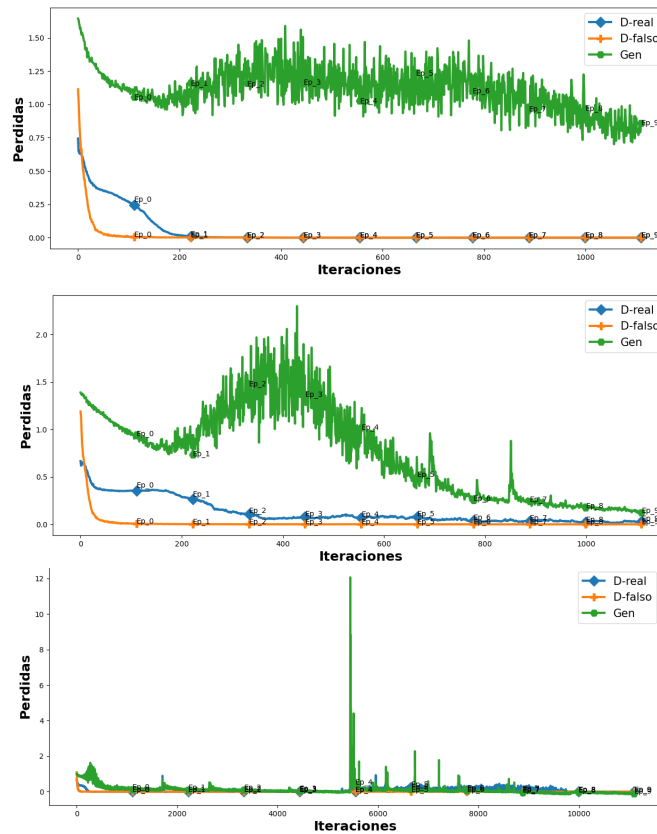
#### 4.1. Resultados para Sistema 1 y Selección de Parámetros.

Para este sistema se aplicaron los parámetros mencionados en la Tabla 6. El primer parámetro fijado fue el de la normalización de los datos. Manteniendo una configuración base, se realizaron tres pruebas que fueron: no normalizar los datos, normalizar de -1 a 1 respecto al máximo absoluto, y normalización logarítmica aplicando tangente hiperbólica (tanh). En la Figura 9 se ve el comportamiento de la función de pérdidas en los modelos; el color verde para las pérdidas del generador (G) y las otras son del discriminador (D): para cuando se presentan trazas reales (azules) y trazas falsas (naranja) generadas desde G. Las pérdidas que disminuyen de mejor forma es la subgráfica del medio para el caso de las pérdidas de G. Consecuentemente, para la elección de los otros parámetros se revisaron las métricas de desempeño y las trazas generadas. Luego de aplicar esta serie de experimentos, se dejaron fijos los parámetros resaltados en la Tabla 6. Cabe aclarar en este punto que la tasa de aprendizaje usada fue tomada como referencia de Li *et al.* (2020) y se ha mantenido en  $10^{-6}$  y  $2 \times 10^{-6}$  para entrenar al discriminador y la GAN respectivamente.

Para concluir esta parte, los parámetros resaltados en la Tabla 6 representan el resultado

Figura 9.

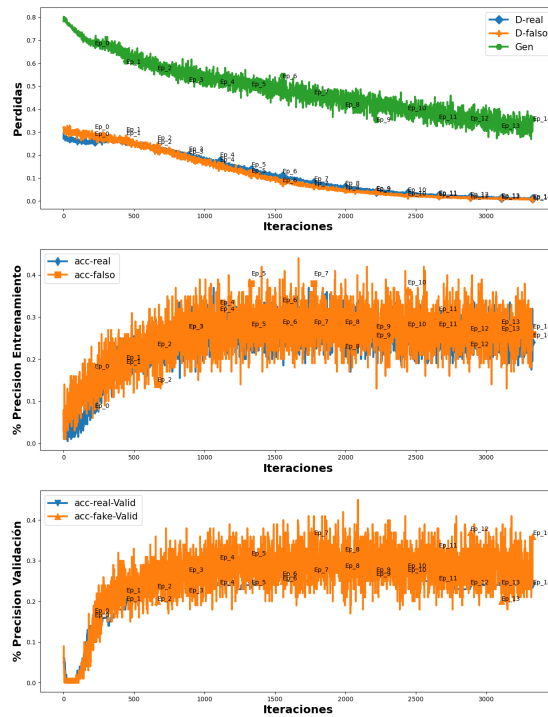
Resultado de las pérdidas respecto a normalización de los datos para sistema 1.



*Nota.* La subgráfica superior representa la no normalización de los datos. La subgráfica del medio representa normalización respecto al máximo absoluto (-1 y 1). La subgráfica inferior representa normalización entre -1 y 1 con amplificación logarítmica tanh.

para la sintonización del sistema, las pérdidas en G y D se pueden visualizar en la Figura 10 (subgráfica superior) y métrica de precisión en las subgráficas inferiores, cada una representa la precisión con datos de entrenamiento y validación respectivamente. La gráfica 11 es el resultado de la traza generada desde el modelo G para la época 1 y 10. Los resultados no brindan la generación de trazas que se busca como propuesta del sistema; pues inicialmente, la generación de las trazas es altamente ruidosa y no se logra apreciar contenido acústico en la traza generada, de hecho, se

Figura 10.  
Resultados para el sistema 1



*Nota.* La subgráfica superior representa las pérdidas para G y D. La subgráfica del medio ilustra la precisión cuando se presentan los datos de entrenamiento y la inferior es la métrica con los datos de validación. Los punteros indican las épocas.

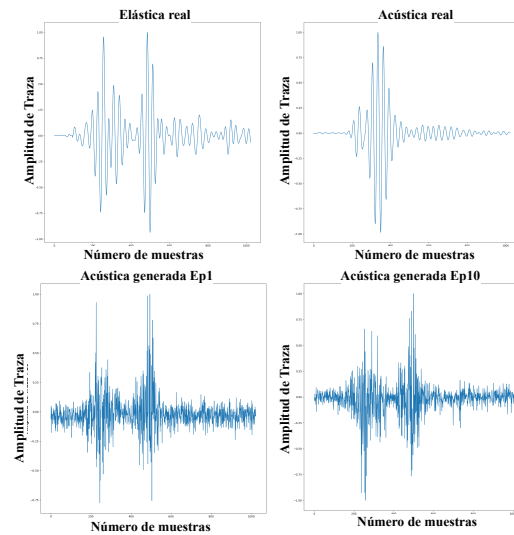
logra ver que el modelo trata de reconocer la segunda envolvente presente en la traza elástica (ver Figura 11); mientras que, la hipótesis tras el sistema planteado es tratar de reducir la información elástica de las trazas fuentes y dejar pasar la mayor cantidad de detalle acústico.

#### 4.2. Resultados para Sistema 2 y Selección de Parámetros.

Para la implementación de este sistema se realizó los experimentos mostrados en la Tabla 7. Este sistema consiste en un auto-codificador para el modelo del generador y siguiendo el modelo híbrido del primer sistema. Los resultados para este sistema a nivel de trazas presentaron fallas,

*Figura 11.*

Resultado de trazas reales y generadas para el sistema 1



*Nota.* La subgráfica superior muestra la pareja de datos reales, y la inferior es el resultado de la traza generada desde el modelo G entrenado para la época 1 y 10 izquierda y derecha respectivamente.

pues en cada experimento realizado, las trazas generadas se tornaban más ruidosas mientras avanzaba en épocas de entrenamiento; de hecho, desde el punto de vista de las pérdidas, en D se logra ver que éstas descienden rápidamente llevando el sistema a un modo de fallo conocido como fallo de convergencia (Brownlee, 2019), donde el modelo del discriminador puede reconocer fácilmente

Tabla 7

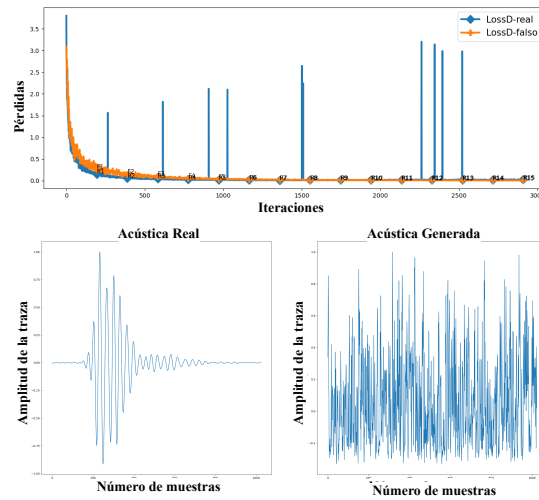
*Parámetros probados para el sistema 2*

Tamaño batch	Tamaño Kernel	Normalización de datos	Función de Costo	Optimizador	Numero de Capas
100/50	10/4	-1, 1	mse	RMSProp	4*

*Nota.* \* Parámetro aplicado sobre la red G

Figura 12.

Identificación de modo de falla de convergencia del sistema 2



*Nota.* La subgráfica superior es un ejemplo de modo de falla de convergencia (pérdidas cercana a cero); se puede evidenciar con los datos de baja calidad generados (subgráfica inferior derecha)

te los datos falsos, pues el generador genera datos basura (ver Figura 12) (Brownlee, 2019). Por los motivos anteriores, no se llevó a cabo una elección de los parámetros de la Tabla 7. De éste experimento se puede afirmar que los auto-codificadores tienen una sintonización diferente para adaptarlo a un problema específico, pues en contraste al sistema U-Net, tienen similitud estructural.

#### 4.3. Resultados para Sistema 3 y Selección de Parámetros.

En este sistema se implementó el concepto de la Wasserstein GAN (WGAN), principalmente consiste en entrenar  $n$  veces más a  $D$  que a  $G$  para lograr mayor afinidad en las pérdidas de  $D$  (Gulrajani *et al.*, 2017). Los experimentos realizados se ilustran en la Tabla 8. La definición original de la WGAN plantea la función de pérdida como *Wasserstein* junto con el optimizador RMSProp, y el sistema planteado tiene como función de costo mse y optimizador RMSProp. Co-

Tabla 8

*Parámetros probados para el sistema 3*

<b>Tamaño batch</b>	<b>Tamaño Kernel</b>	<b>Valor n_critico</b>	<b>Función de pérdida</b>
100/50	10/4	1/2/3/4/5/7	mse/wass_loss
<b>Numero de Capas</b>		<b>Aplicación de filtro</b>	
5/6/7*		Si/No	

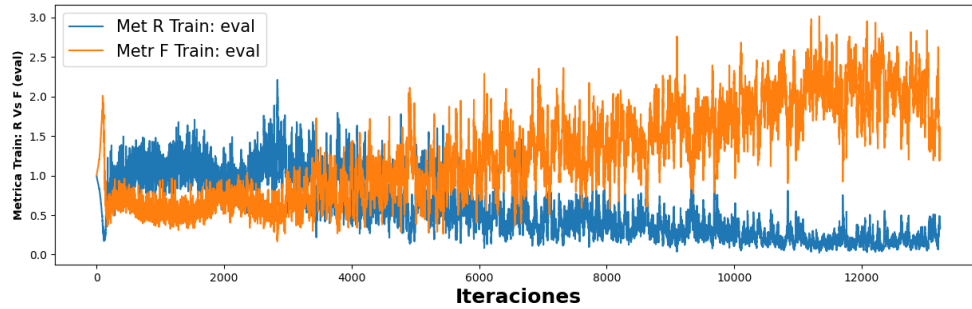
*Nota.* El color rojo indica la selección de los parámetros del sistema 3. El color verde representa una alternativa adicional como solución, \* Este parámetro se aplica sobre la red D

mo alternativa de solución, en la sección de recomendaciones se establece una serie de sugerencias para continuar con la exploración de la función de costo *Wasserstein*, pues en los experimentos mostró un buen desempeño en la generación de los datos, sólo que esa solución requiere de afinación diferente a la propuesta. El sistema definido con función de pérdida mse refleja un buen desempeño de forma global, es decir, desde el punto de vista de la función de pérdida para D y G, desde las métricas planteadas, que en este caso son mse y *BinaryAccuracy* para D y en G sólo se evaluó la métrica mse, y finalmente, la generación tanto de trazas como sismogramas acústicos.

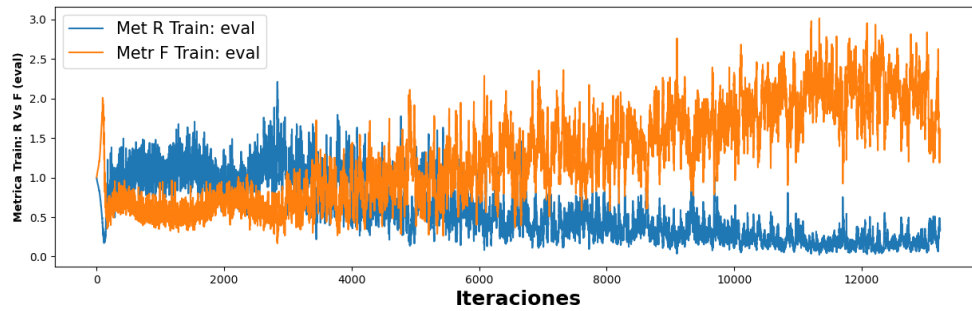
La Figura 14 muestra las pérdidas para la red D y G, la métrica mse en D y G se ilustran en las Figuras 13(a) y 13(b). Finalmente, mediante la aplicación de un filtro ormsby con frecuencias de corte 0, 2, 6, 8 [Hz] (usadas para datos sintéticos de inversión de onda completa) las trazas generadas desde G fueron procesadas dentro del mismo sistema GAN; la Figura 16 (derecha) muestra un mismo sismograma cuando se aplica el filtro ormsby y cuando no.

Figura 13.

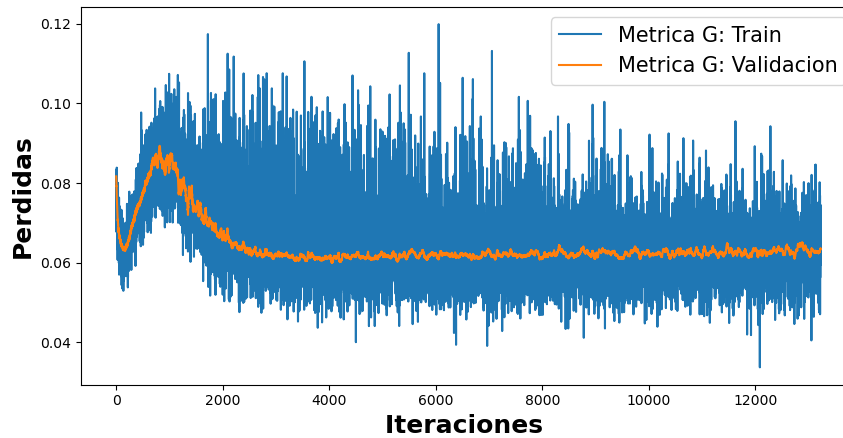
Métrica MSE para el sistemas cGeoGAN (filtro incluido)



(a) Métrica en D: con datos de entrenamiento



(b) Métrica en D: con datos de validación



(c) Métrica en G: acústicos reales Vs generados

#### 4.4. Resumen del sistema cGeoGAN

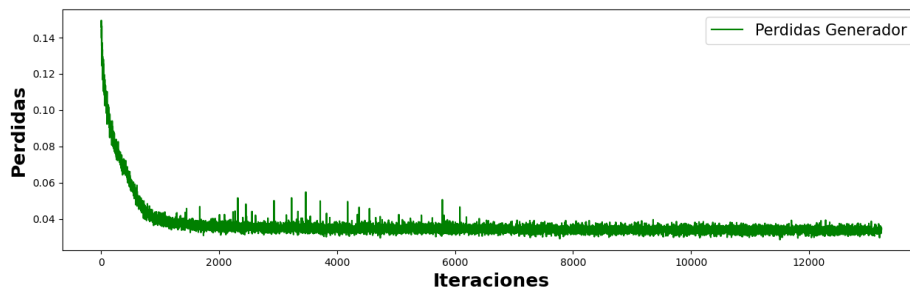
El resumen del sistema seleccionado puede observarse en la Tabla 9. Se toma como base la WGAN y se hacen adaptaciones en los modelos, principalmente porque se aplica la función de costo mse en lugar de la *Wasserstein*.

Figura 14.

Funciones de pérdidas para el sistema cGeoGAN



(a) Perdidas del discriminador



(b) Perdidas del Generador

Como resultado final, se demuestra que el sistema es capaz de generar trazas y a su vez reduce algunos eventos pertenecientes en el dominio elástico. En la Figura 16, el disparo original elástico (subgráfica izquierda) tiene gran amplitud entre las muestras 0-200 y 400-600, y la red tiene la capacidad de reducir estos eventos de forma efectiva. Esto se puede observar en el disparo

Tabla 9

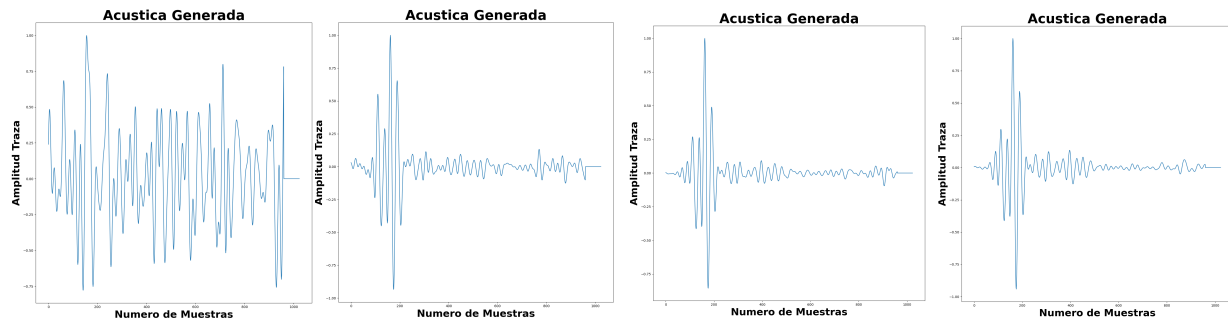
*Arquitectura del sistema cGeoGAN definido*

	Generador U-Net	Discriminador Crítico	Config. en Capas	Config. Modelos
# Capas Codificadores	4	-	16-5-2-same-inni	-
			32-5-2-same-inni	
			64-5-2-same-inni	
			128-3-2-same-inni	
			<b>128-3-2-same-inni*</b>	
# Capas Decodificadores	4	-	128-3-2-same-inni	-
			64-5-2-same-inni	
			32-5-2-same-inni	
			16-5-2-same-inni	
# Capas Convolucionales	-	6	64-4-2-same-inni	Loss: 'mse' Optimizador: 'RMSprop' Lr = $1\mu$
# Capas Densas	-	4	256	-
			64	
			16	
			4	
cGeoGAN	-	-	-	Loss: 'mse' Optimizador: 'RMSprop' Lr = $2\mu$
Otros	batch = 50	n_critico = 3	norm = -1 a 1	filtro = si

*Nota.* En la columna Config. en Capas los valores indican: #NeuronasPorCapa-#TamañoKernel-#Stride-TipoRelleno-InicializadordePesos(inni), inni es distribución normal con media cero y desviación estándar de 0.02. - Indica no aplica. \* Este parámetro indica la configuración del espacio latente luego de la etapa de codificación

*Figura 15.*

Evolución trazas generadas filtradas



*Nota.* De izquierda a derecha son los resultados para la época 1, 10, 20 y 30.

acústico generado (subgráfica derecha en los bordes verdes). Los resultados a nivel de las trazas se visualizan en la Figura 15, y se presenta la evolución para las épocas 1, 10, 20 y 30.

#### 4.5. Comparación en presentación de datos

En esta sección, se realizaron dos experimentos que exploran la respuesta de la red ante dos formas diferentes de presentarle la información de entrada. Los datos en la entrada se presentaron usando agrupaciones por disparo común vs agrupaciones por disparos pseudo-aleatorios (ya que se plantea una semilla para la repetibilidad de los experimentos desde la librería *sklearn*). Los resultados de estas formas de presentación se pueden observar en la Figura 17. La conclusión tras estos experimentos está en que la red aprende mejor de la distribución de datos cuando éstos se entregan de forma aleatoria. Los resultados para los modelos entrenados se evidencian desde los sismogramas, los sismogramas resultante para datos ordenados está presentado en la Figura 17(a) y para datos pseudo-aleatorios en la Figura 17(b). Adicionalmente, revisando otros disparos, de forma

general, hay mejor desempeño (evaluación cuantitativa) en la generación de los sismogramas.

#### 4.6. Costo computacional

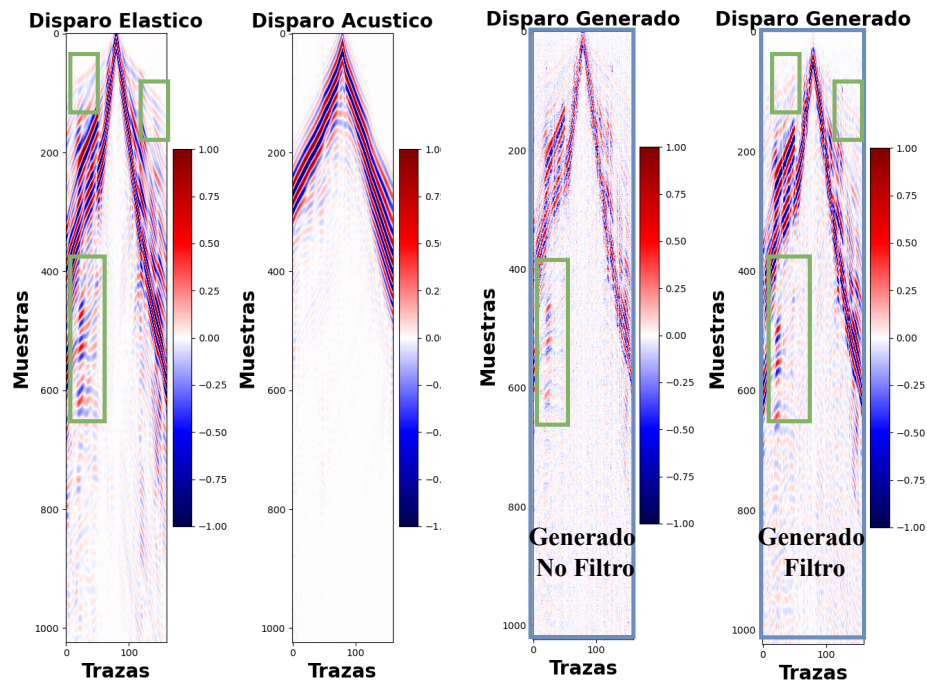
El costo computacional para el sistema cGeoGAN propuesto (resumen del sistema definido en la Tabla 9) es de 187 minutos aproximadamente cuando se ejecuta sobre el nodo *Kioto*<sup>2</sup>; mientras que al probarlo sobre el nodo *Tokyo*<sup>3</sup> la prueba tuvo una duración de 374 minutos; adicio-

<sup>2</sup> Nodo que contiene las GPU's Tesla V100

<sup>3</sup> Nodo que contiene las GPU's Tesla K40c

Figura 16.

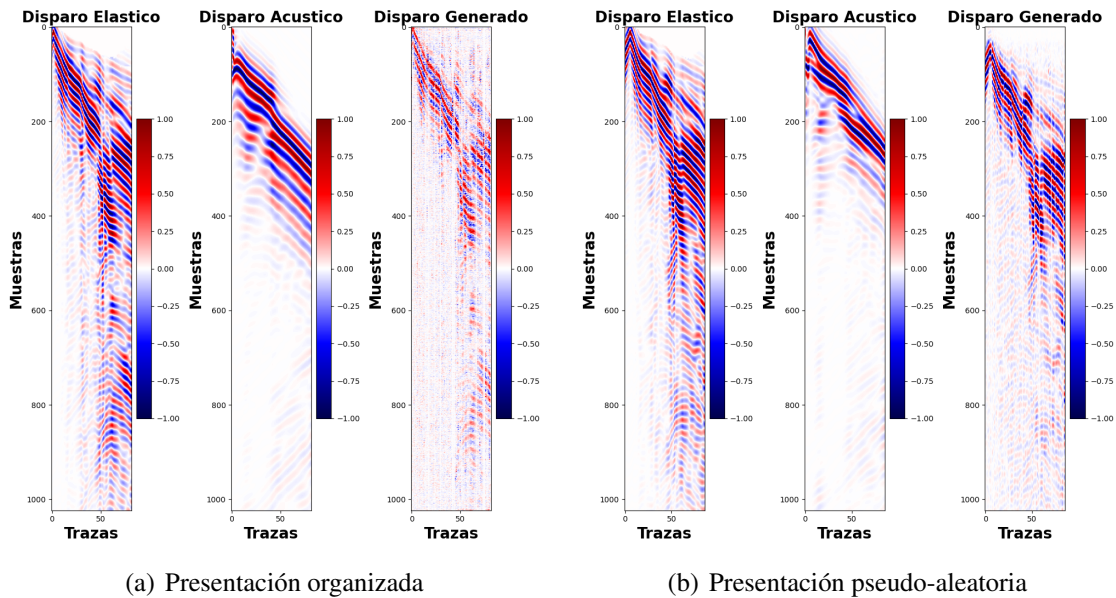
Disparos generados con reducción de eventos elásticos



*Nota.* Disparos de referencia y disparos generados (borde azul). Reducción de eventos elásticos entre las muestras 0 y 200 y 400 y 600 (borde verde).

Figura 17.

Sismogramas luego de entrenar ordenada y pseudo-aleatoriamente



nalmente, el consumo en RAM fue de 9 y 6 GB para Kioto y Tokyo-00, respectivamente.

## 5. Conclusiones

1. Desde el punto de vista de la generación de trazas, el modelo cGeoGAN definido es capaz de comprimir en un espacio latente una porción de información necesaria para luego realizar una reconstrucción efectiva de una nueva traza; mitigando algunos eventos elásticos, como se aprecia en los sismogramas mostrados en la Figura 16, y en la evolución de las trazas generadas a través de las épocas de entrenamiento (ver Figura 15).
2. Luego de obtener las correspondientes trazas y sismogramas es correcto afirmar que no se están eliminando todos los eventos elásticos provenientes de los sismogramas fuente. En la

Figura 16 se logran evidenciar algunos eventos elásticos que son reducidos por la red cGeoGAN (recuadros verdes). El ruido *ground roll* continúa siendo un evento elástico desafiante de eliminar por la red (forma cónica que se observa desde el centro del disparo elástico), ya que logra reconstruirse parcialmente en el nuevo disparo (en lugar de eliminarlo completamente).

3. Al evaluar el rendimiento de la red con la métrica mse se busca que a través de las épocas el error vaya disminuyendo, así como se observa en la Figura 13 (a y b). Cuando se presentan los datos de entrenamiento, se observa que el mse inicia con un error mayor a la unidad, para luego reducirse y situarse debajo de 0.5 (luego de 6000 iteraciones aprox.). Esto indica la buena capacidad de la red para reconocer los datos reales. Por otra parte, en los datos generados, el error empieza a aumentar luego de las 6000 iteraciones, lo que puede conducir al concepto del sobre-entrenamiento de la red. Sin embargo, a pesar del aumento en el error, la generación de las trazas para los datos de validación mejoran cuantitativamente la calidad conforme pasan las épocas.

## 6. Recomendaciones

La primera recomendación es continuar con la exploración y variación de los hiperparámetros del sistema cGeoGAN, con la idea de mejorar la calidad de los sismogramas acústicos generados. Como se evidencia en la Tabla 3, se aplicaron pocos cambios sobre los hiperparámetros de las topologías. Se sugiere continuar con la afinación en la red discriminadora, para reducir el comportamiento oscilatorio que presentan las pérdidas (Ver Anexos).

Como segunda sugerencia de alternativa de solución para obtener sismogramas acústicos a partir de los elásticos, se sugiere replantear el sistema cGeoGAN con la función de pérdida Wasserstein. Algunas sugerencias iniciales son la revisión de la tasa de aprendizaje, ya que los resultados de pruebas preliminares con este hiperparámetro mostró mejoría con respecto a pruebas anteriores (revisando la función de pérdida, tanto en la red discriminadora como en la generadora).

Buscar alguna métrica que permita cuantificar la reconstrucción de los datos, ya sea a nivel de traza o sismograma, esto con la intención de con otr el desempeño de la red; y así mismo, experimentar con otro tipo de datos sísmicos sintéticos, para analizar el comportamiento de la red respecto a los datos generados, la función de pérdida y métricas asociadas.

Se recomienda, en cuestión de función de pérdida, buscar en la literatura otro tipo de expresión para minimizar que se adapte al propósito de este trabajo, ya que una limitación del error cuadrático medio (mse) es opacar los errores pequeños cuando se presentan errores grandes (esto está relacionado con el concepto de la varianza); la finalidad de esta parte consiste en intentar mejorar la propagación del error en el entrenamiento de la red y quizás, encontrar otras posibles soluciones.

### Referencias Bibliográficas

Anaconda, I. (2022). Anaconda documentation. <https://docs.anaconda.com/>.

Arévalo, A. (2020). Generative adversarial networks (gan) to improve spatial resolution in inverted velocity fields. *European Association of Geoscientists amp; Engineers*. <https://doi.org/10.3997/2214-4609.202084007>.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. En Precup, D. & Teh, Y. W., editores, *Proceedings of the 34th International Conference on Machine Learning*, volumen 70 de *Proceedings of Machine Learning Research*, pp. 214–223. PMLR.

Ben-Menahem, A. & Singh, S. J. (2012). *Seismic waves and sources*. Springer Science & Business Media.

Brownlee, J. (2019). *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*. Machine Learning Mastery.

Chang, D., Yang, W., Yong, X., & Li, H. (2018). Generative adversarial networks for seismic data interpolation. En *SEG 2018 Workshop: SEG Maximizing Asset Value Through Artificial Intelligence and Machine Learning, Beijing, China, 17-19 September 2018*, pp. 40–43. Society of Exploration Geophysicists and the Chinese Geophysical Society. <https://doi.org/10.1190/AIML2018-11.1>.

- Chintala, S. (2016). How to train a gan? tips and tricks to make gans work. <https://github.com/soumith/ganhacks>. Accedido el 06 de Junio, 2021.
- Chintala, S. (2017). How to train a gan, nips 2016 workshop on adversarial training. <https://youtu.be/X1mUN6dD8uE>. Accedido el 06 de Junio, 2021.
- Choo, J., Downton, J., & Dewar, J. (2004). Lift: A new and practical approach to noise and multiple attenuation. *First Break*, 22(5). <https://doi.org/10.3997/1365-2397.2004009>.
- Claerbout, J. F. & Green, I. (2010). Basic earth imaging. *Cambridge: Free Software Foundation*.
- Equinor (2022). segyio documentation. <https://seggio.readthedocs.io/en/latest/>. Accedido el 19 de Abril de 2021.
- Gauthier, O., Virieux, J., & Tarantola, A. (1986). Two-dimensional nonlinear inversion of seismic waveforms: Numerical results. *Geophysics*, 51(7):1387–1403. <https://doi.org/10.1190/1.1442188>.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. En *Advances in neural information processing systems*, pp. 2672–2680. <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans.

<https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6->

Paper.pdf.

Henley, D. C. (2003). Coherent noise attenuation in the radial trace domain. *Geophysics*, 68(4):1408–1416. <https://doi.org/10.1190/1.1598134>.

Huang, L., Dong, X., & Clee, T. E. (2017). A scalable deep learning platform for identifying geologic features from seismic attributes. *The Leading Edge*, 36(3):249–256. <https://doi.org/10.1190/tle36030249.1>.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134. <https://doi.org/10.1109/ICCV.2017.244>.

Jeffreys, H., Jeffreys, B., & Swirles, B. (1999). *Methods of mathematical physics*. Cambridge university press.

Keras (2022). Keras. url: <https://keras.io/>.

Klochikhina, E., Crawley, S., Frolov, S., Chemingui, N., & Martin, T. (2020). Leveraging deep learning for seismic image denoising. *First Break*, 38(7):41–48. <https://doi.org/10.1109/TGRS.2021.3057857>.

Krohn, C. E. (1984). Geophone ground coupling. *Geophysics*, 49(6):722–731. <https://doi.org/10.1190/1.1441700>.

- Li, W. & Wang, J. (2021). Residual learning of cycle-gan for seismic data denoising. *IEEE Access*, PP:1–1. <https://doi.org/10.1109/ACCESS.2021.3049479>.
- Li, Y., Ku, B., Zhang, S., Ahn, J.-K., & Ko, H. (2020). Seismic data augmentation based on conditional generative adversarial networks. *Sensors*, 20(23):6850. <https://doi.org/10.3390/s20236850>.
- Liang, J., Ma, J., & Zhang, X. (2014). Seismic data restoration via data-driven tight frame. *Geophysics*, 79(3):V65–V74. <https://doi.org/10.1190/geo2013-0252.1>.
- Linder-Norén, E. (2022). Keras generative adversarial networks (gans). <https://github.com/eriklindernoren/Keras-GAN>. Accedido el 21 de Abril de 2021.
- Lu, Y.-H., Lu, W.-K., & Zhai, Z.-J. (2012). An edge-preserving seismic data interpolation method. *Diqiu Wuli Xuebao*, 55(3):991–997. <https://doi.org/10.6038/j.issn.0001-5733.2012.03.029>.
- Merkel, D. . (2022). Docker: lightweight linux containers for consistent development and deployment. <https://www.docker.com/>. Accedido el 21 de Abril de 2021.
- Mirza, M. & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*. <https://doi.org/10.48550/arXiv.1411.1784>.
- NumFOCUS (2022). pandas. <https://pandas.pydata.org/>. Accedido el 21 de Febrero de 2022.
- Oliveira, D. A. B., Ferreira, R. S., Silva, R., & Brazil, E. V. (2019). Improving seismic da-

ta resolution with deep generative networks. *IEEE Geoscience and Remote Sensing Letters*, 16(12):1929–1933. <https://doi.org/10.1109/LGRS.2019.2913593>.

Oliveira, D. A. B., Semin, D., & Zaytsev, S. (2020). Ground roll suppression using convolutional neural networks. <https://doi.org/10.48550/arXiv.2010.15209>.

Özdenizci, O. & Erdoğmuş, D. (2021). On the use of generative deep neural networks to synthesize artificial multichannel eeg signals. En *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 427–430. IEEE. <https://doi.org/10.1109/NER49283.2021.9441381>.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. <https://doi.org/10.48550/arXiv.1511.06434>.

RAE, R. A. E. (2021). Fomato. En *Diccionario de la lengua española*. Recuperado el 23 de junio de 2021, de <https://dle.rae.es/formato>.

RAE, R. A. E. (2022). Matriz. En *Diccionario de la lengua española*. Recuperado el 01 de marzo de 2022, de <https://dle.rae.es/matriz>.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. En *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).

SEG (2017). Seg-y\_r2.0: Seg-y revision 2.0 data exchange format. Recuperado el 23 de junio de 2021, de <https://library.seg.org/seg-technical-standards>.

- Siahkoohi, A., Kumar, R., & Herrmann, F. (2018). Seismic data reconstruction with generative adversarial networks. En *80th EAGE Conference and Exhibition 2018: Opportunities Presented by the Energy Transition*. European Association of Geoscientists and Engineers, EAGE. <https://doi.org/10.3997/2214-4609.201801393>.
- Sirgue, L. & Pratt, R. G. (2004). Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies. *Geophysics*, 69(1):231–248. <https://doi.org/10.1190/1.1649391>.
- Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28. <https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf>.
- Spitz, S. (1991). Seismic trace interpolation in the fx domain. *Geophysics*, 56(6):785–794. <https://doi.org/10.1190/1.1443096>.
- Stein, S. & Wysession, M. (2009). *An introduction to seismology, earthquakes, and earth structure*. John Wiley & Sons.
- TensorFlow (2022). Por qué tensorflow. <https://www.tensorflow.org/>. Accedido el 21 de Febrero de 2022.
- Washburn, A. & Wood, K. (1995). Two-person zero-sum games for network interdiction. *Operations research*, 43(2):243–251. <https://doi.org/10.1287/opre.43.2.243>.
- Wilamowski, B. M. (2009). Neural network architectures and learning algorithms. *IEEE Industrial Electronics Magazine*, 3(4):56–63. <https://doi.org/10.1109/MIE.2009.934790>.

- Xu, S., Wang, D., Chen, F., Zhang, Y., & Lambare, G. (2012). Full waveform inversion for reflected seismic data. En *74th EAGE Conference and Exhibition incorporating EUROPEC 2012*, pp. cp-293. European Association of Geoscientists & Engineers. <https://doi.org/10.3997/2214-4609.20148725>.
- Yilmaz, Ö. (2001). *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists.
- Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. En *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232. <https://doi.org/10.1109/ICCV.2017.244>.

## Apéndice A

### Guía de instalación del paquete Anaconda.

Guía rápida para instalación de Anaconda en Kioto para ejecutar pruebas.

Ingreso al clúster: `ssh usuario_cps@200.16.119.102 ó 10.1.92.54`

Entrar a la carpeta tmp: `cd tmp/`

Ejecutar: `curl -O https://repo.anaconda.com/archive/Anaconda3-<2021.11>-Linux-x86_64.sh`

No olvidar cambiar <2021.11> por la última versión disponible.

Ejecutar el bash: `bash Anaconda3-<2021.11>-Linux-x86_64.sh`

Activar la instalación: `source ~/.bashrc`

Verificar la instalación: `conda list`

Para la ejecución de pruebas sobre GPUs y más información sobre el proyecto (como scripts y demás), puede consultar el siguiente repositorio:

[https://gitlab.com/CPS\\_Repo/Seismic/semillerogan2020/geogan-elastic2acoustic](https://gitlab.com/CPS_Repo/Seismic/semillerogan2020/geogan-elastic2acoustic)

---

<sup>4</sup> Cuando se esté dentro de las instalaciones de la universidad