

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE CIENCIA
Departamento de Matemática y Ciencia de la Computación



**Generación de secuencias de réplicas sísmicas a partir de la
ocurrencia de terremotos en Chile utilizando redes adversariales
generativas**

Isidora Jara Muñoz

Profesor guía:

Dr. Francisco Plaza Vega

Comisión evaluadora:

Dra. Orietta Nicolis

Dr. Victor Hugo Salinas

**Tesis para optar al título de
Ingeniera Estadística**

**Santiago - Chile
2025**

AGRADECIMIENTOS

Quiero comenzar agradeciendo profundamente a mi familia por su apoyo fundamental a lo largo de todo mi camino educacional y de vida. En particular, a mi mamá Verónica y a mi hermano Matías, por estar siempre conmigo cuando más los necesito: para reír, llorar, darme un abrazo o darme palabras de aliento en los momentos en que sentí que el mundo se venía abajo. Las palabras nunca serán suficientes para expresar cuánto los amo ni para agradecerles todo lo que han hecho por mí.

Extiendo mis agradecimientos a mis tías Fabiola, Soledad y Jacqueline; a mis tíos Alejandro, Gonzalo y Carlos; y a mis primos Florencia y Diego, quienes han estado presentes en los momentos más importantes de mi vida, entregándome siempre su apoyo incondicional.

Agradezco a mi profesor guía, Dr. Francisco Plaza-Vega, por mostrarme lo lindo que es el Deep Learning y por todo el apoyo -y los cafés- durante el desarrollo de este trabajo. También al Dr. Victor Hugo Salinas y a la Dra. Orietta Nicolis, por ser parte de este proyecto y aportar con comentarios que permitieron mejorar y sacar adelante esta tesis.

De igual forma, agradezco a todos los profesores de la carrera por formarme y acompañarme durante estos años. En especial, a la Dra. Claudia Matus Correa, por darme el ánimo que necesitaba para continuar cuando sentía que no iba a poder.

Quiero dar un reconocimiento especial a mi “otra familia” de Puente Alto: la tía Fabi, el tío Andrés, el Lucas, el Benja y el Vale. En particular, a Fabio (el payo payo), por confiar en mi potencial, por mostrarme lo linda que puede ser la estadística, por ayudarme cada vez que tuve dudas académicas y por darme siempre los ánimos que necesitaba para seguir adelante.

Agradezco a mi pareja, Bruno, por su apoyo constante durante este proceso, por prestarme su computador para acelerar los códigos y, sobre todo, por motivarme siempre a confiar en mí y en lo que sé <3.

Gracias también a mis amigos, quienes han sido un pilar en mi vida y en esta etapa universitaria: a Néstor (mi primer amigo en la universidad), Alan (gracias por enseñarme inferencia y series de tiempo), Xami, Tivi, Anto, Álvaro, Pablo, Sofi Valderrama, Sofi Duque y muchos otros que de una u otra forma fueron parte de este proceso. En especial, a Valentina (la Vale), quien me acompaña desde el colegio y ha estado a mi lado a pesar de que nuestras carreras son completamente

distintas. Gracias por escucharme siempre y aguantar mis conversaciones sobre la tesis aunque no entiendas nada de lo que digo.

Quizás se me quede alguien en el tintero, pero le agradezco a todos quienes, de alguna manera, formaron parte de mi vida universitaria y de este camino.

Finalmente, agradezco a mis gatos (Larry y Pitufo) y a mis perritas (Siri y Jommy) por acompañarme en largas jornadas de estudio con su compañía incondicional. A One Direction, cuya música me dio ánimo en los momentos más difíciles de este proceso, y como bien dicen ellos: “You and me got a whole lot of history.”

RESUMEN

Chile, al encontrarse en el límite de la subducción entre la placa de Nazca y la placa Sudamericana, es uno de los países con mayor actividad sísmica del mundo, donde los terremotos y sus réplicas representan un riesgo constante para la población. Si bien existen avances en el monitoreo y modelamiento estadístico de la sismicidad, la predicción de réplicas sigue siendo un desafío debido a la complejidad de los procesos geológicos involucrados. En este contexto, la presente tesis propone el uso de modelos generativos adversariales (GAN) para la generación de secuencias de réplicas sísmicas a partir de eventos principales, con el objetivo de predecir sus magnitudes y explorar los patrones espacio-temporales asociados.

La metodología consideró la recopilación de datos del Centro Sismológico Nacional (2000–2024) y, dada la limitada cantidad de secuencias en Chile, se amplió el análisis al Anillo de Fuego del Pacífico utilizando información del USGS. Tras un análisis descriptivo y exploratorio de los datos, se implementaron tres variantes de modelos GAN: el estándar (cGAN), un modelo con capa Transformer (TGAN) y un modelo con función de costo de Wasserstein (WGAN). La evaluación del desempeño se realizó mediante métricas de error (RMSE, MAE, MAPE) y el índice de persistencia (PI).

Los resultados muestran que los modelos son capaces de capturar las características generales de las réplicas, destacando la variante con Transformer por su mayor estabilidad y capacidad de representación. Estos hallazgos evidencian el potencial de las redes generativas como herramientas complementarias al modelamiento estadístico tradicional, ofreciendo una vía innovadora para la simulación de secuencias sísmicas y el fortalecimiento de sistemas de gestión del riesgo. Finalmente, se plantean trabajos futuros siguiendo esta línea de investigación que incluyen la incorporación del modelo ETAS como input del modelo, el ajuste de las arquitecturas neuronales, el uso de ventanas temporales adaptativas y la extensión del análisis a distintas regiones del Anillo de Fuego.

Keywords: Sismos, Réplicas sísmicas, Modelos generativos adversariales, Aprendizaje profundo, Predicción sísmica, Anillo de Fuego del Pacífico.

TABLA DE CONTENIDOS

1	Introducción	1
1.1	Formulación del problema	2
2	Objetivos de la investigación	4
2.1	Objetivo general	4
2.2	Objetivos específicos	4
3	Marco teórico	5
3.1	Conceptos sismológicos	5
3.1.1	Modelo ETAS	7
3.2	Conceptos estadísticos	7
3.2.1	Procesos temporales	7
3.2.2	Procesos espacio-temporales	8
3.3	Conceptos de aprendizaje automático	8
3.3.1	Algoritmo de agrupación	9
3.3.2	Modelos Generativos Adversariales (GAN)	10
3.3.3	Métricas de desempeño	12
4	Metodología	14
5	Resultados	16
5.1	Análisis descriptivo	16
5.2	Análisis exploratorio de los datos	24

TABLA DE CONTENIDOS

5.3	Análisis de patrones sísmicos	36
5.4	Resultados del modelo	43
5.4.1	Modelo cGAN estándar (cGAN)	45
5.4.2	Modelo GAN con capa Transformer (TGAN)	53
5.4.3	Modelo GAN con capa Wasserstein (WGAN)	61
6	Conclusiones	70
6.1	Trabajos futuros	71
	ANEXO	72
A	Anexo	73
A.1	Dependencia temporal en terremotos de Chile	73
A.2	Código	83
A.2.1	cGAN	83
A.2.2	TGAN	86
A.2.3	WGAN	90
	Referencias bibliográficas	94

Capítulo 1

Introducción

Los sismos son fenómenos naturales de gran importancia debido a su capacidad de causar daños significativos y poner en riesgo la vida humana. La magnitud y frecuencia de los terremotos varían considerablemente en todo el mundo, y algunos de los eventos más devastadores son recordados hasta el día de hoy. A nivel mundial, algunos de los sismos más destacables incluyen el terremoto de Sumatra en 2004 en Indonesia, con una magnitud de 9.3 que provocó un devastador tsunami en el océano Índico [40], y el terremoto de Tohoku en 2011 en Japón, con una magnitud de 9.1 que también generó un tsunami y afectó gravemente a la región, desencadenando la crisis nuclear de Fukushima [11].

Es de amplio conocimiento que Chile es uno de los países más sísmicos del mundo, ya que se encuentra en el límite de la subducción entre la placa tectónica de Nazca y la placa Sudamericana [21]. Esta subducción genera una acumulación de energía que se libera de forma repentina en eventos sísmicos, como los terremotos. En nuestro país se han registrado algunos de los terremotos más grandes del mundo, por ejemplo el terremoto de Valdivia en 1960, con una magnitud de 9.5 siendo uno de los sismos con mayor magnitud registrados a nivel mundial [9]. Más recientemente, el terremoto de Constitución en 2010 tuvo una magnitud de 8.8, causando daños significativos y desencadenando un tsunami devastador para la región [37]. La actividad sísmica en Chile representa un riesgo significativo para la población, motivo por el cual distintos gobiernos han impulsado políticas enfocadas en reducir los daños y proteger a la ciudadanía. Entre estas medidas se incluye la modificación del Plan Regulador Metropolitano de Santiago (PRMS), con la creación de normas de construcción antisísmica [13]; el establecimiento de reglamentaciones que permiten a los edificios resistir movimientos telúricos [12]; la realización de simulacros en establecimientos educacionales, promoviendo la cultura preventiva y el autocuidado [38]; así como la asignación de recursos para

reforzar hospitales, carreteras y puentes, garantizando su operatividad tras sismos severos [6], entre otras iniciativas.

1.1. Formulación del problema

Tomando en cuenta la capacidad destructiva de los eventos sismológicos de gran intensidad, es que cobra una gran relevancia el avanzar en la búsqueda de modelos predictivos que puedan entregar información relevante para la toma de decisiones, estableciendo alertas oportunas hacia la población. Aunque un evento principal puede ser desastroso, las réplicas asociadas a ese evento pueden aumentar la magnitud de la catástrofe [36], por lo que se hace necesario que estos modelos predictivos mencionados anteriormente consideren la información de las réplicas, o de algún modo se pueda obtener información de las secuencias sísmicas generadas por los eventos principales para establecer protocolos de seguridad, evacuación, entre otras políticas públicas asociadas a la ocurrencia de estos eventos.

Sin embargo, la predicción de terremotos sigue siendo un gran desafío, a pesar de los esfuerzos por identificar patrones y señales precursoras de actividad sísmica. La complejidad y la naturaleza impredecible de los procesos geológicos, sumadas a la falta de tecnologías avanzadas, hacen que la predicción exacta de terremotos continúe siendo una tarea sumamente difícil [16]. Aunque ha habido avances en el monitoreo sísmico y en el estudio de los patrones históricos de actividad sísmica, los terremotos son fenómenos inherentemente caóticos, con una gran variabilidad e incertidumbre, lo que complica aún más la predicción precisa [19].

A pesar de esto, se han realizado varios trabajos utilizando técnicas de machine learning y deep learning contribuyendo en problemas complejos como la estimación, clasificación y predicción de terremotos. Por ejemplo, en [26] se muestra cómo los modelos de machine learning utilizan características de señales sísmicas para estimar magnitudes en tiempo real de forma más rápida, asimismo, [31] explora el uso de algoritmos como Support Vector Machines (SVM) y Random Forests para clasificar señales sísmicas en categorías como eventos tectónicos y no tectónicos, mejorando la detección temprana y la interpretación de datos, por último, [3] destaca el uso de redes neuronales profundas, como Convolutional Neural Networks (CNNs), para analizar señales electromagnéticas y predecir magnitudes mediante la identificación de patrones relacionados con terremotos, por mencionar algunos. Estos avances demuestran cómo el machine learning y el deep learning están mejorando la velocidad y precisión en el análisis sísmico, al tiempo que abren nuevas oportunidades para desarrollar sistemas de alerta temprana más eficaces.

Sin embargo, en muchas ocasiones son las réplicas asociadas las que incrementan la magnitud de la catástrofe, por lo que es de suma importancia estudiarlas.

Este proyecto de tesis busca desarrollar un modelo generativo que permita generar y predecir las magnitudes de las réplicas sísmicas tras un evento principal. En particular, se busca emplear técnicas de redes neuronales generativas adversariales (GAN), que han mostrado un gran potencial para la generación de secuencias de datos complejos.

Capítulo 2

Objetivos de la investigación

2.1. Objetivo general

Predecir la magnitud de las réplicas sísmicas en Chile ocasionadas por un evento principal mediante la generación de secuencias sísmicas utilizando modelos generativos de aprendizaje profundo con información histórica disponible.

2.2. Objetivos específicos

1. Recopilar la información sísmica disponible de Chile a través de repositorios y bases de datos apropiadas.
2. Estudiar la ocurrencia de sismos y su relación con las réplicas mediante una descripción estadística de la información disponible.
3. Diseñar un marco de trabajo apropiado para la generación de réplicas sísmicas provocadas por terremotos utilizando un modelo de redes generativas adversariales (*Generative Adversarial Network*, GAN).
4. Implementar el marco de trabajo para la generación de réplicas.
5. Analizar el impacto de los terremotos en sus réplicas mediante la evaluación de los resultados obtenidos por el modelo.

Capítulo 3

Marco teórico

En este capítulo, se presentan los antecedentes teóricos necesarios para abordar la predicción de la magnitud de las réplicas sísmicas ocasionadas por un evento principal. En primer lugar, se estudiarán conceptos sismológicos fundamentales para este trabajo, luego se describirán los conceptos estadísticos a utilizar para realizar análisis más profundos sobre los datos. Finalmente se revisarán los conceptos de aprendizaje automático (*Machine Learning*, ML) y aprendizaje profundo (*Deep Learning*, DL) enfatizando en los modelos de Redes Generativas Adversariales (del inglés *Generative Adversarial Networks*, GAN) que serán utilizados para realizar la predicción de la magnitud de las réplicas.

3.1. Conceptos sismológicos

A continuación, se definirán los conceptos de evento principal [5] y réplica[5, 36].

- **Evento principal:** Corresponde al proceso de liberación súbita de energía, generando ondas elásticas y su posterior propagación por el interior de la Tierra. Al llegar a la superficie de la Tierra, estas ondas producen movimiento y vibración del suelo. En Chile, se usa el término Terremoto para un sismo que genera daños estructurales, esto es, que sea reportado con Intensidad en la Escala de Mercalli Modificada¹ con grado VII o superior.
- **Réplica:** Una réplica corresponde a un sismo de magnitud menor que ocurre en el entorno

¹Escala de doce grados que mide la intensidad observada en un lugar específico dado los efectos que produce el sismo.

o interior del área de ruptura² en un período posterior al evento principal. La ocurrencia de las réplicas en general depende de la magnitud del sismo principal y se puede extender por varios días, semanas, meses o años.

En sismología, un evento principal está asociado a una serie de réplicas [36]. Esto sucede por los procesos físicos que ocurren durante y después de la ruptura de una falla tectónica. El evento principal libera una gran cantidad de energía acumulada en la corteza terrestre a lo largo de la falla tectónica, lo que provoca un desplazamiento que altera el equilibrio de tensiones en la región afectada [21, 39]. Luego del evento principal, la corteza terrestre entra en un proceso de ajuste que genera una serie de eventos sísmicos secundarios denominados réplicas. Estas réplicas son, generalmente, de menor magnitud y ocurren en las proximidades de la zona de ruptura.

El comportamiento de las réplicas sigue ciertos patrones temporales y espaciales. De forma espacial, sabemos que las réplicas tienden a estar distribuidas alrededor de la zona de ocurrencia del evento principal, aunque la distancia puede variar de acuerdo a la magnitud del terremoto inicial. Por su parte, de forma temporal, sabemos que la disminución de la actividad de las réplicas está representada por la Ley de Utsu-Omori [44], la cual establece que, en una determinada región, el número de réplicas decae de manera exponencial. El decaimiento del número de réplicas en general se puede describir como

$$N(t) = \frac{k}{(c + t)^p} \quad (3.1)$$

Donde $N(t)$ es el número de eventos registrados luego de la ocurrencia del evento principal durante un tiempo t , k , c y p dependen de las características del evento principal y de la región de ocurrencia.

Los estudios de [44], junto con investigaciones más recientes como [27], han analizado cómo la aplicación de esta ley varía con la profundidad en la corteza terrestre. Se ha observado que, en la corteza superior, los mecanismos de fractura y la relajación de tensiones favorecen un ajuste más preciso de la dinámica de réplicas al modelo empírico de Omori. Sin embargo, la variación del parámetro c con la profundidad, indica que las condiciones de estrés y los procesos de fractura no son homogéneos en toda la corteza.

²Lugar donde ocurre el deslizamiento durante un terremoto, el cual libera la energía acumulada en forma de ondas sísmicas.[22]

3.1.1. Modelo ETAS

El modelo ETAS (*Epidemic-Type Aftershock Sequence*) es una herramienta muy utilizada en el modelamiento de la predicción de réplicas posteriores a un evento principal. Este modelo describe la ocurrencia de terremotos como un proceso autorregresivo en el tiempo y espacio, considerando que estos eventos principales generan réplicas que pueden generar más réplicas, creando una cadena de sismos. Se utilizan funciones de intensidad condicional que dependen del tiempo, ubicación y magnitud de los eventos previos. Este modelo describe el decaimiento de las réplicas después de un terremoto siguiendo la Ley de Utsu-Omori y es capaz de ajustarse a regiones con características sísmicas distintas [29].

3.2. Conceptos estadísticos

3.2.1. Procesos temporales

Se define un proceso estocástico. Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, sea $\mathcal{T} \subset \mathbb{R}$ para cada $t \in \mathcal{T}$ la función

$$\begin{aligned} X(t, \cdot) &: \Omega \rightarrow \mathbb{R} \\ \omega &\rightsquigarrow X(t, \omega) \end{aligned}$$

una variable aleatoria y el conjunto $\mathbf{X} = \{X(t, \cdot) : t \in \mathcal{T}\}$ una colección de variables aleatorias definidas en $(\Omega, \mathcal{F}, \mathbb{P})$ corresponde a un proceso estocástico.

Un proceso temporal se define mediante un proceso estocástico, así, sea $Y(t)$ un proceso estocástico escrito como $\mathbf{Y} = \{Y(t) : t \in \mathcal{T}\}$ donde t corresponde a una indexación de Y o a una realización en el tiempo de Y , donde $\mathcal{T} \subset \mathbb{R}$ [8].

De esta manera, las series de tiempo corresponden a procesos temporales con tiempo discreto, donde el conjunto \mathcal{T} está compuesto por $\{0, 1, \dots, T\}$. Es importante mencionar que las series de tiempo cuentan con su función de media y de autocovarianza, las cuales se definen por [30]

- $\mu_t = \mathbb{E}(Y_t) = \mu < \infty$.
- $\gamma(h) = \text{Cov}(Y_t, Y_{t+h})$.

También tenemos a las series de tiempo irregulares, en ellas los datos no se registran en intervalos regulares o simplemente son incompletos. Esto ocurre generalmente cuando hablamos

de datos que son aleatorios (como los sismos) y el hecho de no observarlos regularmente puede ser problemáticos en el análisis de series temporales, pues dificulta la detección de patrones o la predicción de las observaciones.

3.2.2. Procesos espacio-temporales

Un proceso espacial corresponde a un campo aleatorio indexado por un conjunto contenido en \mathbb{R}^d con $d > 1$. Se denota un proceso espacial por $Z(\mathbf{s})$, donde $\mathbf{s} \in \mathcal{D} \subset \mathbb{R}^d$ o escrito como $\mathbf{Z} = \{Z(\mathbf{s}), \mathbf{s} \in \mathcal{D} \subset \mathbb{R}^d\}$. En la práctica, $Z(\mathbf{s})$ es una función aleatoria indexada por \mathbf{s} perteneciente al conjunto \mathcal{D} .

Los procesos espacio-temporales corresponden a procesos que tienen una componente temporal y otra espacial. Este proceso es denotado por $Z(\mathbf{s}, t)$ donde $(\mathbf{s}, t) \in \mathcal{D} \times \mathcal{T} \subset \mathbb{R}^d \times \mathbb{R}$. También puede ser escrito como $\mathbf{Z} = \{Z(\mathbf{s}, t) : (\mathbf{s}, t) \in \mathcal{D} \times \mathcal{T}\}$.

Según [25] desde un punto de vista matemático se puede decir que los procesos espacio-temporales son campos aleatorios en \mathbb{R}^{d+1} , lo cual permite entender este proceso de forma análoga a los procesos espaciales junto con sus propiedades y características. Más aún, se puede considerar un proceso espacial como un caso particular de un proceso espacio-temporal considerando un tiempo en particular y fijo.

Adicionalmente, un proceso puntual es un modelo probabilístico que describe la ocurrencia de eventos en puntos específicos dentro de un dominio, como el tiempo, el espacio o ambos. Matemáticamente, un proceso puntual se define en un espacio medible $(\mathbb{S}, \mathcal{B})$, donde \mathbb{S} es el espacio de ocurrencia de los puntos y \mathcal{B} es un σ -álgebra sobre \mathbb{S} . Este proceso está representado por una medida aleatoria \mathcal{N} la cual describe una colección aleatoria de puntos $\{t_i\}$ en \mathbb{S} , donde cada punto indica la localización (temporal, espacial o espacio-temporal) de un evento. Esta formalización permite modelar y analizar fenómenos en los que los eventos ocurren en ubicaciones específicas de manera aleatoria [7].

3.3. Conceptos de aprendizaje automático

Los modelos de aprendizaje automático son algoritmos que permiten a las computadoras aprender patrones a partir de datos, sin necesidad de una programación explícita para cada tarea. Se utilizan en tareas como predicción, clasificación y la identificación de estructuras ocultas en los datos. Según su enfoque, pueden dividirse en tres tipos principales: aprendizaje supervisado (que

utiliza datos etiquetados), aprendizaje no supervisado (que trabaja sin etiquetas) y aprendizaje por refuerzo (que optimiza decisiones mediante prueba y error).

3.3.1. Algoritmo de agrupación

El algoritmo K-means es una técnica de agrupamiento no supervisado ampliamente utilizada en análisis de datos, cuya lógica se basa en minimizar la distancia entre observaciones dentro de un mismo grupo y maximizar la distancia entre grupos distintos. Los clusters que se obtienen son internamente homogéneos pero heterogéneos entre sí [10]. Su objetivo principal es minimizar la variabilidad dentro de los grupos, expresada como la suma de los errores cuadráticos intracluster (*within-cluster sum of squares*, WCSS), generando agrupaciones compactas y bien separadas.

El funcionamiento del algoritmo se puede describir en los siguientes pasos:

1. **Inicialización:** Se eligen aleatoriamente k observaciones del conjunto de datos como los centroides iniciales. Para mejorar esta selección y evitar converger a soluciones no óptimas, existen variantes como K-means++, que utiliza una estrategia de inicialización probabilística que elige los centros con mayor impacto en la calidad de la agrupación. Esto permite que el algoritmo converja más rápido y con mejores resultados en comparación con la inicialización aleatoria tradicional.[2].
2. **Asignación:** Cada observación se asigna al cluster cuyo centroide esté más cercano, usualmente utilizando la distancia euclidiana como métrica.
3. **Actualización:** Se recalculan los centroides de cada cluster como la media de las observaciones asignadas a él.
4. **Repetición:** Se repiten los pasos 2 y 3 hasta que se cumpla algún criterio de parada, como que no haya cambios en las asignaciones o que la variación total intracluster no disminuya significativamente.

El algoritmo busca minimizar la siguiente función objetivo:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (3.2)$$

donde C_i representa el conjunto de puntos asignados al cluster i y μ_i su centroide correspondiente.

Una de las principales limitaciones del algoritmo es que requiere definir el número de clusters k de antemano. Para abordar este problema, se suele utilizar el método del codo (*elbow method*), que consiste en ejecutar el algoritmo para varios valores de k y graficar la WCSS para cada uno. El valor óptimo de k se identifica visualmente como aquel en el que la reducción de la WCSS comienza a estabilizarse, formando una especie de “codo” en la curva [20].

En el contexto del análisis sísmico, K-means ha demostrado ser una herramienta útil para identificar zonas con patrones similares de actividad, lo cual facilita la segmentación geográfica según características como la magnitud, profundidad o frecuencia de los eventos sísmicos [10]. Esta agrupación permite realizar análisis exploratorios más precisos sobre fenómenos geológicos complejos.

3.3.2. Modelos Generativos Adversariales (GAN)

El aprendizaje profundo (*Deep Learning*) es una rama del aprendizaje de máquinas (*Machine Learning*) que utiliza arquitecturas de redes neuronales profundas para modelar relaciones complejas en los datos. Entre sus arquitecturas más relevantes se encuentran el perceptrón multicapa (*Multilayer Perceptron*, MLP), útil para clasificación de datos con estructuras no lineales; las redes neuronales convolucionales (*Convolutional Neural Networks*, CNNs), empleadas en procesamiento de imágenes; y las redes recurrentes (*Recurrent Neural Networks*, RNNs), utilizadas en series temporales o datos con dependencia secuencial [18].

En este contexto surgen las Redes Generativas Adversariales (*Generative Adversarial Networks*, GANs), introducidas por [14] en el 2014. Estas redes permiten generar datos sintéticos que imitan la distribución de datos reales, lo que las convierte en una herramienta poderosa para tareas como generación de imágenes, síntesis de audio, aumento de datos y simulación de eventos complejos.

Arquitectura GAN

La arquitectura GAN consta de dos redes neuronales profundas que compiten entre sí:

1. **Generador (G):** recibe como entrada una variable aleatoria z proveniente de una distribución latente (por ejemplo, una distribución normal) y genera una muestra sintética $G(z)$ que intenta imitar los datos reales.

2. **Discriminador (D):** recibe como entrada una muestra (real o generada) y entrega una probabilidad de que dicha muestra provenga del conjunto real de datos.

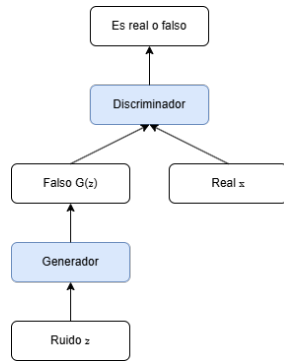


Figura 3.1: Traducción de la arquitectura de los modelos GAN, extraída de [47].

El entrenamiento de un modelo GAN puede describirse como el minmax entre el generador y el discriminador. El discriminador intenta maximizar la probabilidad de distinguir correctamente entre datos reales y generados, mientras que el generador intenta minimizarla, es decir, engañar al discriminador:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.3)$$

Donde, G es el generador, D es el discriminador, $x \sim p_{\text{data}}(x)$ son las muestras reales que provienen de la distribución de datos reales, $z \sim p_z(z)$ es el ruido o vector latente proveniente de una distribución conocida, $D(x)$ es la probabilidad que el discriminador asigna a que una muestra x sea real y $D(G(z))$ es la probabilidad que el discriminador asigna a una muestra generada [14].

Durante el entrenamiento, el discriminador entrega retroalimentación al generador a través de sus gradientes, lo que permite a este último mejorar en cada iteración la calidad de las muestras generadas. Este proceso continúa hasta que el generador produce muestras tan realistas que el discriminador no puede diferenciarlas de las reales (idealmente, $D(G(z)) = 0,5$).

El entrenamiento de modelos GAN puede ser inestable debido a problemas como el colapso de modo (generación limitada de muestras), la alta sensibilidad a hiperparámetros, y el desequilibrio entre el generador y el discriminador. Para abordar estos desafíos, se han desarrollado diversas variantes, entre las que destacan:

- **Wasserstein GAN (WGAN):** mejora la estabilidad del aprendizaje al utilizar la distancia de

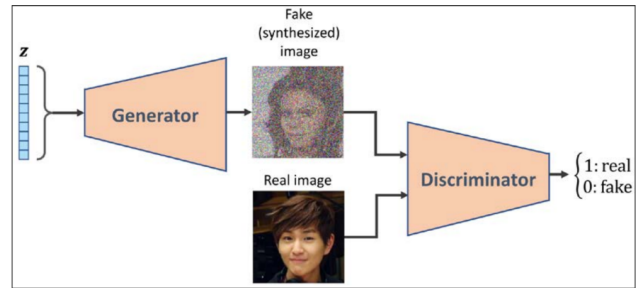


Figura 3.2: Representación de los modelos GANs obtenida de [35]

Wasserstein. Este enfoque reduce el problema de colapso de modo y proporciona curvas de aprendizaje más interpretables [1]. La clave de WGAN es reemplazar la función de pérdida tradicional de GANs con una basada en la distancia de Wasserstein, permitiendo una mejor retroalimentación para el generador y facilitando la convergencia del modelo.

- **Conditional GAN (cGAN):** su principio fundamental es que, en lugar de generar datos de manera completamente aleatoria, el modelo recibe una variable condicional que guía el proceso de generación. Esto posibilita, por ejemplo, la creación de imágenes de dígitos específicos en MNIST o el etiquetado de imágenes basado en características aprendidas [23].

Arquitectura Transformer

Los Transformers se han consolidado como una de las arquitecturas más influyentes en el campo del aprendizaje profundo. Originalmente introducidos en el contexto del procesamiento de lenguaje natural [45], su principal fortaleza radica en el mecanismo de autoatención (self-attention), que permite capturar dependencias de largo alcance en las secuencias sin necesidad de un procesamiento estrictamente secuencial, como ocurre en las redes recurrentes.

En el contexto de la generación de secuencias sísmicas, el uso de capas Transformer resulta particularmente relevante, dado que las réplicas no se producen de manera uniforme ni lineal, sino que exhiben patrones temporales complejos con dependencias a distintos horizontes de tiempo. Incorporar una capa de este tipo en la arquitectura del generador permite modelar mejor estas relaciones, otorgando mayor capacidad de representación y mejorando la calidad de las secuencias sintéticas frente a arquitecturas convencionales basadas únicamente en convoluciones o perceptrones multicapa.

Como se mencionó anteriormente, los modelos de Deep Learning están mejorando la velocidad y precisión en el análisis sísmico, por lo que la generación de réplicas utilizando los modelos GAN puede contribuir a la creación de un modelo que logre predecir las magnitudes de las réplicas después de un evento principal.

3.3.3. Métricas de desempeño

A continuación, se presentan las métricas de desempeño a utilizar para evaluar el rendimiento del modelo.

1. **Raíz del error cuadrático medio (RMSE):** Corresponde a la diferencia al cuadrado entre el

valor predicho por el modelo y el valor real. Se define como

$$RMSE = \sqrt{\frac{\sum_{t=1}^N (\hat{Y}_t - Y_t)^2}{N}}$$

2. **Índice de persistencia (PI):** El índice de persistencia (PI) es utilizado para evaluar el rendimiento de modelos neuronales frente a predicciones persistentes [32], permite cuantificar si el modelo mejora o empeora respecto a una estrategia base. Se define como

$$PI = 1 - \frac{\sum_{i=1}^N (Y_t - \hat{Y}_t)^2}{\sum_{i=1}^N (Y_t - Y_{t-L})^2}$$

Donde Y_{t-L} corresponde a la descarga en el tiempo $t - L$.

3. **Error porcentual absoluto medio (MAPE)** Es una medida de error relativa que permite comparar la precisión de las predicciones realizadas por el modelo. Se define como

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{\hat{Y}_t - Y_t}{\hat{Y}_t} \right|$$

4. **Error medio absoluto (MAE)** Corresponde al promedio del error absoluto en las predicciones realizadas por el modelo. Se define como

$$MAE = \frac{\sum_{t=1}^N |\hat{Y}_t - Y_t|}{N}$$

Capítulo 4

Metodología

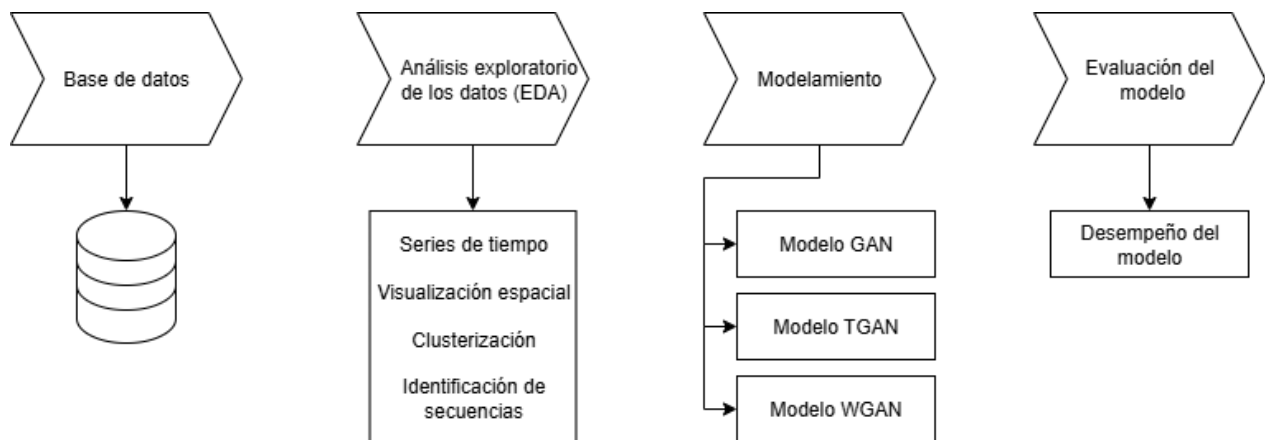


Figura 4.1: Esquema metodológico

En primer lugar se realiza una revisión bibliográfica en relación a los eventos principales y sus réplicas, la cual permite contextualizar los análisis posteriores. En paralelo, se recopila la información disponible a partir del repositorio de acceso libre del Centro Sismológico Nacional [5], considerando los sismos ocurridos en Chile entre los años 2000 y 2024.

Una vez obtenida la base de datos sismológica, se realiza un análisis exploratorio utilizando diversas técnicas de ciencia de datos. Se incluyen visualizaciones temporales (series de tiempo) y espaciales, que permiten describir, identificar patrones y tendencias respecto a la ocurrencia de sismos. Se emplean estadísticas descriptivas para obtener distribuciones de la magnitud y profundidad de los eventos. Además, se aplican algunas técnicas de clusterización, como K-Means [10], que ayudan a establecer agrupaciones de eventos principales y sus respectivas réplicas e

identificar los grupos significativos en función de su ubicación geográfica, magnitud y tiempo. Utilizando toda esta información, se busca identificar las secuencias espacio-temporales de réplicas asociadas a cada evento principal. Dado el número limitado de secuencias identificadas en el territorio chileno, se amplía el alcance geográfico de la base de datos al Anillo de Fuego del Pacífico. Para ello, se realiza una extracción adicional de eventos sísmicos utilizando la información disponible en el repositorio de libre acceso del United States Geological Survey (USGS) [42]. Todo el análisis exploratorio es implementado en el entorno de programación RStudio [34].

Con las secuencias de réplicas identificadas, se procede a realizar el entrenamiento del modelo generativo adversarial (**GAN**), donde se considerarán las ubicaciones de las réplicas y sus magnitudes, entregando las secuencias de réplicas sísmicas

Adicionalmente, se entrenan dos variantes del modelo: la primera incorpora una capa basada en Transformer, y la segunda implementa la función de costo de Wasserstein. De esta manera, el modelamiento con los tres enfoques, GAN estándar, GAN con capa Transformer (TGAN) y GAN con capa Wasserstein (WGAN), se lleva a cabo en el entorno de programación Python [33].

Para finalizar, se evalúa el desempeño de los modelos obtenidos, utilizando diversas métricas de desempeño, como el error cuadrático medio (MSE), el índice de persistencia (PI) [32], el error porcentual absoluto medio (MAPE) y error absoluto medio (MAE). Se discuten los resultados obtenidos, se presentan las conclusiones derivadas de este trabajo y se plantean posibles líneas de investigación futura en esta temática.

Capítulo 5

Resultados

En este capítulo se presentan los hallazgos obtenidos a partir del análisis de la ocurrencia de sismos y su relación con las réplicas. Utilizando un enfoque estadístico y exploratorio, se describen las características fundamentales de los eventos sísmicos registrados en Chile, permitiendo una mejor comprensión de sus patrones espaciales y temporales. Además, se introduce el desarrollo del marco de trabajo basado en redes generativas adversariales (GAN) para la simulación y generación de réplicas sísmicas, donde el modelo busca capturar la dinámica de las réplicas provocadas por terremotos, proporcionando una herramienta innovadora para el estudio de la actividad sísmica. La implementación de este enfoque permitirá evaluar el impacto de los terremotos en sus réplicas y analizar la capacidad del modelo para reproducir las características observadas en los datos reales. Con ello, se espera aportar nuevos conocimientos a la sismología y mejorar la predicción de patrones sísmicos en el país.

5.1. Análisis descriptivo

Antes de profundizar en el análisis de los eventos sísmicos, es fundamental examinar la estructura de los datos mediante técnicas de análisis descriptivo. En esta sección se presentan medidas estadísticas clave como medias, desviaciones estándar y distribuciones, complementadas con visualizaciones como histogramas y diagramas de caja, que permiten evaluar la calidad y características fundamentales de la base de datos.

La base de datos contiene la información de los sismos registrados en Chile entre los años 2000 y 2024. Cuenta con aproximadamente 138.000 observaciones y 6 variables, las cuales son:

- **Fecha.UTC:** Fecha en UTC (Tiempo Universal Coordinado) en la que se registró el sismo expresadas en una zona horaria estándar sin ajustes por usos horarios locales ni cambios por horario de verano. Se encuentra de la forma “Año-Mes-Día”.
- **Hora.UTC:** Hora en UTC (Tiempo Universal Coordinado) en la que se registró el sismo expresadas en una zona horaria estándar sin ajustes por usos horarios locales ni cambios por horario de verano. Se encuentra de la forma “Hora:Minutos:Segundos”.
- **Latitud:** Corresponde a un sistema de referencia para definir la localización en un punto en la Tierra. La latitud proporciona la localización de un lugar al norte o al sur del Ecuador, y se expresa con medidas angulares que van desde 0° en el Ecuador hasta 90° en los polos (latitud norte /latitud sur)
- **Longitud:** Corresponde a un sistema de referencia para definir la localización en un punto en la Tierra. La longitud representa la localización de un lugar al este o al oeste de una línea norte-sur denominada “meridiano de referencia” (Greenwich), que se mide en ángulos que van de 0° en el meridiano de origen a 180° en la línea internacional de cambio de fecha.
- **Profundidad:** La profundidad de un sismo expresa la distancia entre el hipocentro y la superficie (epicentro). Está medida en kilómetros (Km).
- **Magnitud:** La magnitud de un sismo mide su tamaño y está relacionada con la energía liberada en forma de ondas elásticas. Se determina a partir de la amplitud máxima del movimiento sísmico, ajustada por la distancia y profundidad, o mediante el espectro de amplitud en frecuencia. También puede estimarse a partir de las deformaciones en la corteza terrestre. Está medida en Mw.

Se presenta una tabla resumen con las principales estadísticas descriptivas de las variables cuantitativas, la cantidad de missings y de outliers.

Tabla 5.1: Estadísticas descriptivas.

Variable	N°Obs.	Prom.	SD	Min.	Max.	Asimetría	Curtosis	1°Cuartil	Mediana	3°Cuartil	Missings	%Mis.	Outliers	%Out.
Fecha.UTC	138.137	-	-	-	-	-	-	-	-	-	0,00	0,00 %	0,00	0,00 %
Hora.UTC	138.137	-	-	-	-	-	-	-	-	-	0,00	0,00 %	0,00	0,00 %
Latitud	138.137	-28,02	6,09	-54,84	-17,00	-0,17	-0,91	-32,97	-29,44	-21,98	0,00	0,00 %	103	0,07 %
Longitud	138.137	-70,38	1,66	-75,00	-66,03	0,07	-0,62	-71,67	-70,48	-68,98	0,00	0,00 %	0,00	0,00 %
Profundidad	138.137	78,55	60,73	0,00	407,00	1,06	0,71	31,00	60,00	113,00	0,00	0,00 %	3.423	2,48 %
Magnitud	138.137	3,23	0,65	1,00	8,80	0,87	1,37	2,800	3,10	3,600	0,00	0,00 %	2.918	2,11 %

De la tabla 5.1 se puede concluir que las variables Fecha.UTC y Hora.UTC no presentan valores outliers, lo que indica que la información temporal se encuentra correctamente registrada para todos los sismos. Además, ninguna variable posee valores faltantes, por lo que la información se encuentra completa.

Latitud

Por su parte, la latitud tiene un valor medio de -28,02 con una desviación estándar de 6,09, lo que indica una dispersión moderada. El rango va desde -54,84 hasta -17,00, con una asimetría ligeramente negativa (-0,17) y una curtosis de -0,91, sugiriendo una distribución algo achatada. Se identifican 103 outliers, lo que representa un 0,07 % del total, indicando que existen algunos valores extremos pero en proporción reducida.

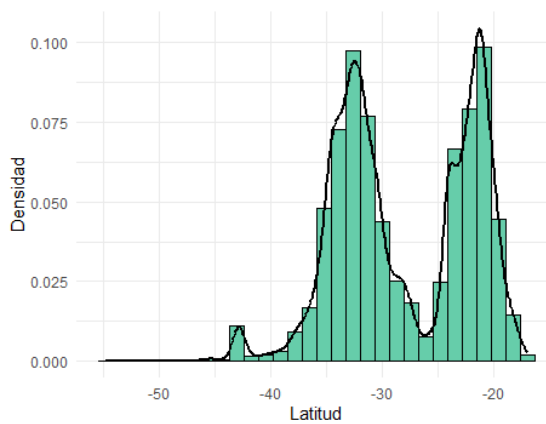


Figura 5.1: Histograma Latitud.

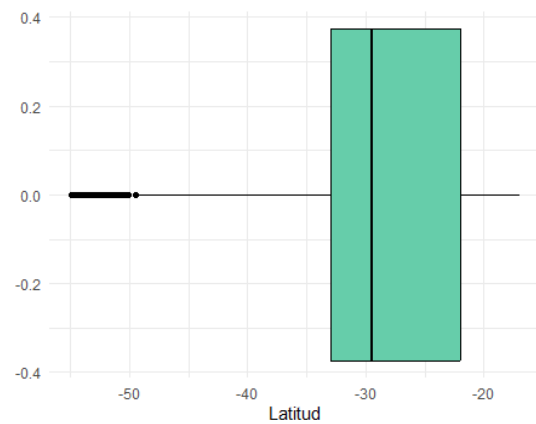


Figura 5.2: Boxplot Latitud.

A partir del histograma de la latitud, se observa que la distribución de esta variable presenta un comportamiento trimodal, lo que indica la existencia de tres zonas geográficas con alta concentración de actividad sísmica. Esta distribución sugiere que los sismos no se distribuyen de manera uniforme a lo largo del territorio chileno, sino que tienden a agruparse en regiones específicas. El gráfico revela que dos de estas zonas presentan una mayor densidad de eventos sísmicos en comparación con la tercera, lo que podría estar relacionado con características geológicas particulares o con la ubicación de fallas tectónicas activas. Esta evidencia respalda la hipótesis de la posible formación de clusters sísmicos en el país.

Dado el rango de valores de latitud observados, es razonable asumir que estas tres agrupaciones corresponden a las zonas norte, centro y sur de Chile, lo que permite establecer una clasificación geográfica que será útil para análisis posteriores.

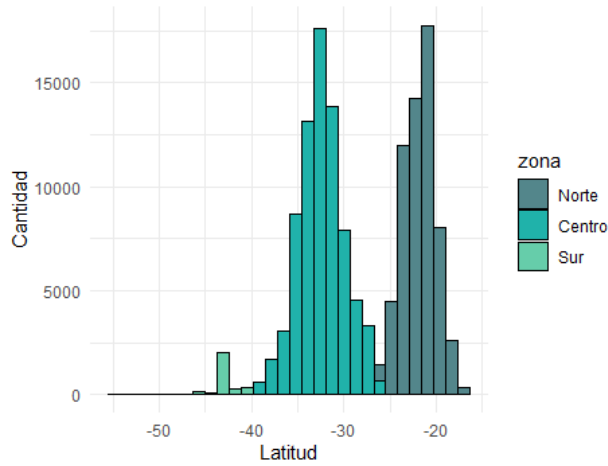


Figura 5.3: Latitudes de Chile divididas por zona.

Estas latitudes serán utilizadas a lo largo de toda la tesis como criterio para dividir el territorio chileno en zonas sísmicas: norte, centro y sur. Esta clasificación geográfica permitirá realizar un análisis más estructurado y preciso de la distribución y comportamiento de los eventos sísmicos, facilitando la identificación de patrones regionales y la formación de posibles clusters sísmicos.

Longitud

La longitud presenta un valor medio de $-70,38$ y una dispersión baja ($1,66$), concentrándose en un rango relativamente estrecho (de $-75,00$ a $-66,03$). La asimetría y curtosis apuntan a una distribución casi simétrica y ligeramente plana, sin presencia de outliers.

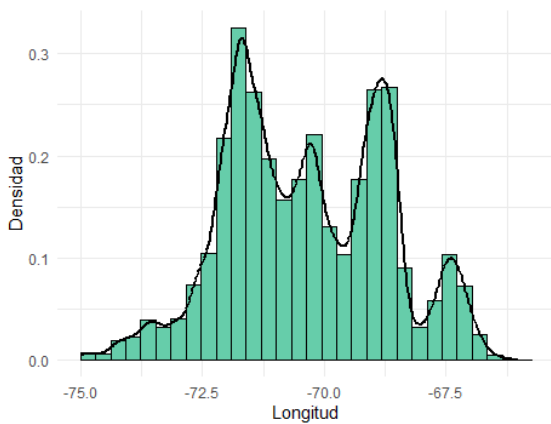


Figura 5.4: Histograma Longitud.

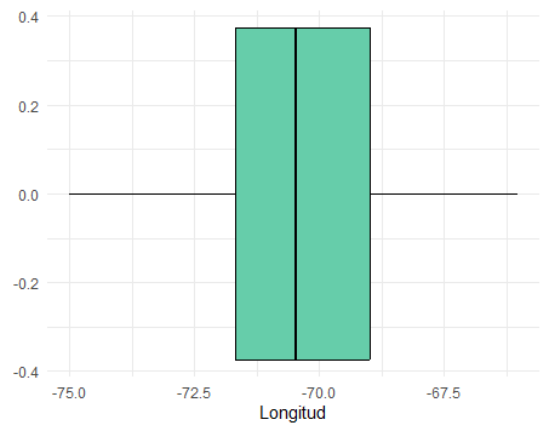


Figura 5.5: Boxplot Longitud.

La figura 5.4 presenta una distribución con múltiples modas, lo que sugiere la presencia de diversas concentraciones geográficas de eventos sísmicos a lo largo de la longitud del país. Se observa que la mayoría de los valores se agrupan entre aproximadamente -75 y -66 grados. Por otro lado, la figura 5.5 confirma la ausencia de valores atípicos y evidencia la baja dispersión de los datos, con una mediana cercana a -70, lo que refuerza la idea de una distribución concentrada.

Profundidad

La profundidad muestra un promedio de 78,55 Km con una alta dispersión (60,73 Km) y un rango amplio (0,00 Km a 407,00 Km), lo que evidencia una variabilidad considerable en la profundidad de los sismos. La asimetría positiva indica que existen eventos con profundidades significativamente mayores que el promedio. La presencia de 3.423 outliers (2,48 %) resalta la existencia de casos extremos, lo que podría estar relacionado con sismos que son inusualmente profundos.

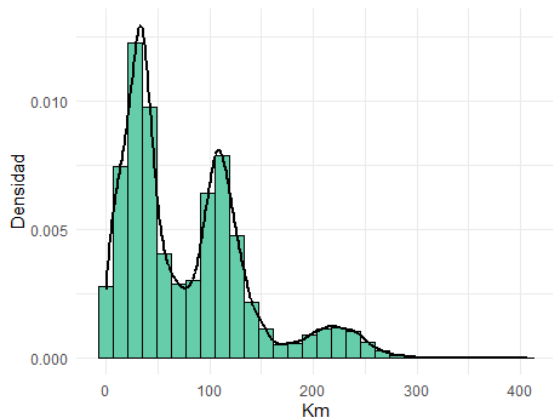


Figura 5.6: Histograma Profundidad.

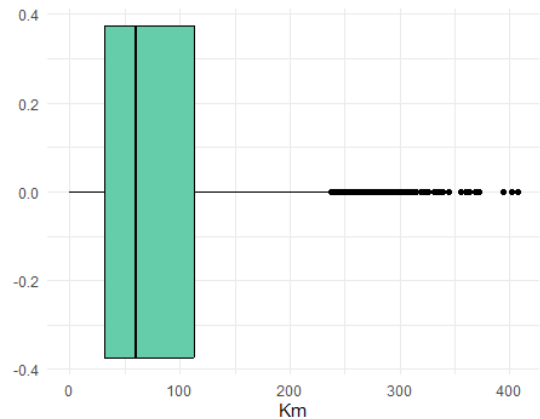


Figura 5.7: Boxplot Profundidad.

Se identifican al menos tres grupos bien definidos en la distribución de la profundidad de los eventos sísmicos, lo que indica la presencia de distintos niveles de actividad sísmica en el subsuelo chileno. La mayor concentración de sismos se registra en profundidades menores a 60 km, lo que corresponde a eventos más superficiales. Un segundo grupo significativo se encuentra alrededor de los 100 km, mientras que un tercer grupo, menos frecuente, se localiza cerca de los 200 km de profundidad.

Esta distribución sugiere la presencia de distintos regímenes tectónicos y estructuras geológicas a lo largo del territorio, posiblemente vinculados con el proceso de subducción de la placa de Nazca bajo la placa Sudamericana, así como con la compleja interacción de fallas activas presentes en la región. Por ello, se realiza un histograma de profundidad restringido a eventos

sísmicos con una profundidad máxima de 60 km.

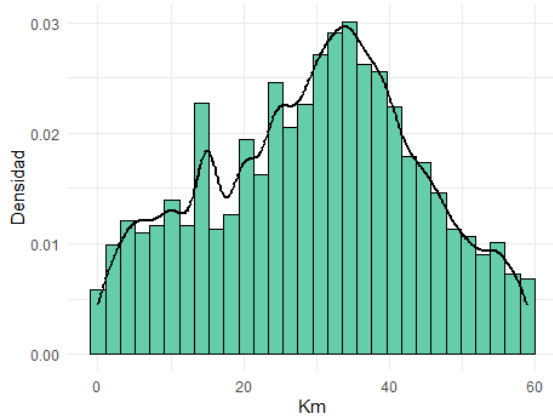


Figura 5.8: Histograma Profundidad menor a 60 Km.

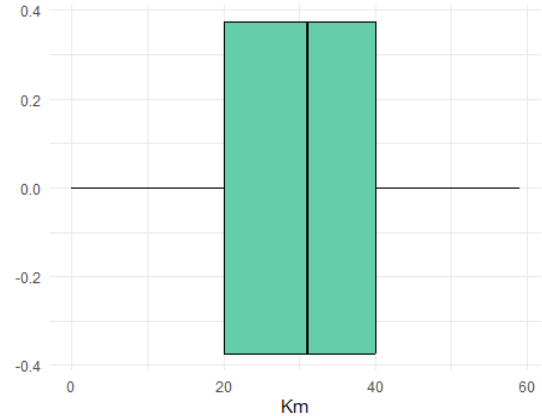


Figura 5.9: Boxplot Profundidad menor a 60 Km.

El histograma de profundidades revela una mayor concentración de sismos entre los 30 y 45 km, lo que indica que una parte significativa de los eventos sísmicos ocurre en niveles relativamente superficiales. La distribución observada no es simétrica, lo que sugiere una posible influencia de estructuras geológicas superficiales en la generación de estos eventos.

La mediana de la distribución se encuentra cercana a los 30 km, lo que refuerza la predominancia de sismos en ese rango. Cabe destacar que no se observan valores atípicos, lo que sugiere una distribución relativamente homogénea dentro del rango analizado.

Magnitud

Finalmente, la Magnitud de los sismos tiene un promedio de 3,23 Mw con una desviación estándar de 0,65 Mw, lo que refleja que la mayoría de los eventos se concentran en una magnitud moderada. Con valores mínimos de 1,00 Mw y máximos de 8,80 Mw, se observa una asimetría positiva ($0,87 \text{ Mw}^3$) y una curtosis de $1,37 \text{ Mw}^4$, sugiriendo la presencia de sismos de baja frecuencia pero con magnitudes excepcionalmente altas. La existencia de 2.918 outliers (2,11 %) indica que, aunque la mayoría de los eventos se sitúan en rangos moderados, hay un número relevante de sismos que sobresalen por su alta magnitud.

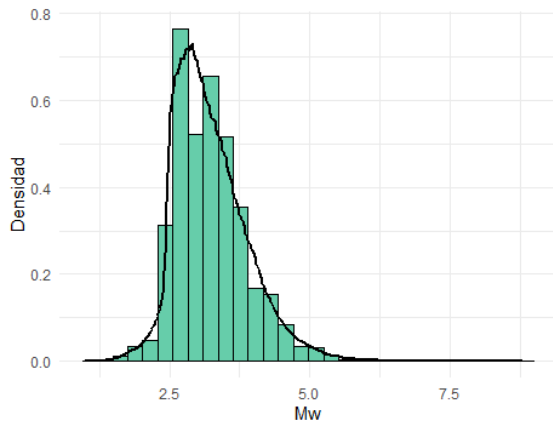


Figura 5.10: Histograma Magnitud.

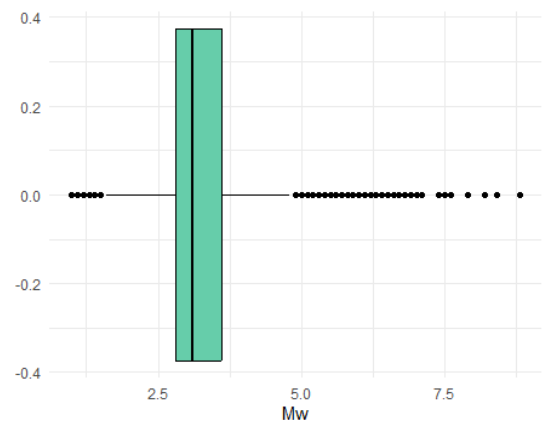


Figura 5.11: Boxplot Magnitud.

En la figura 5.10, se observó que la distribución de la magnitud de los sismos presenta una forma asimétrica y sesgada hacia la derecha, lo que sugiere que podría aproximarse a una distribución Gamma o a una Weibull. Por esta razón, se procedió a verificar si la magnitud posee alguna de estas distribuciones, obteniendo en ambos casos que la magnitud no posee ninguna de las dos distribuciones. A continuación se presentan gráficas complementarias (histograma, densidad, QQ-plot y CDF empírica vs teórica), las cuales nos permiten evaluar visualmente qué tan bien se ajusta cada distribución a los datos observados.

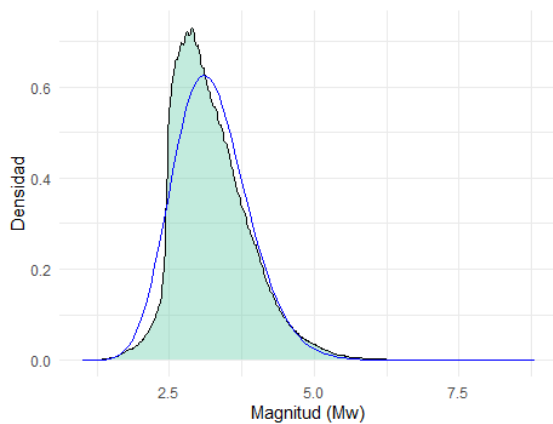


Figura 5.12: Densidad Gamma.

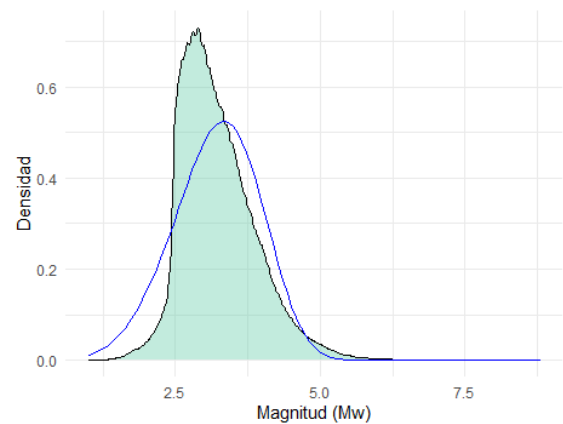


Figura 5.13: Densidad Weibull.

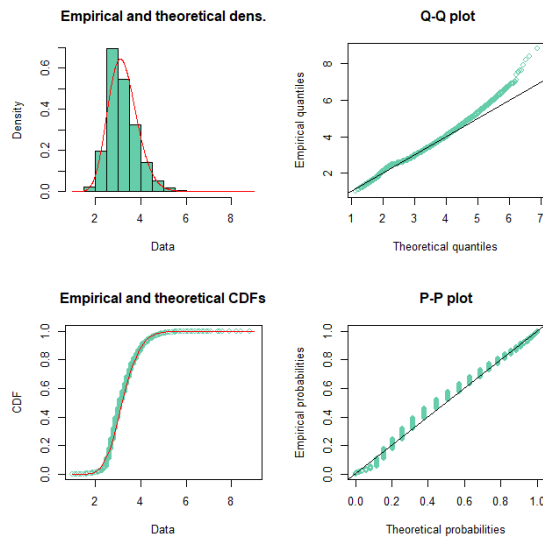


Figura 5.14: Gráficas Gamma.

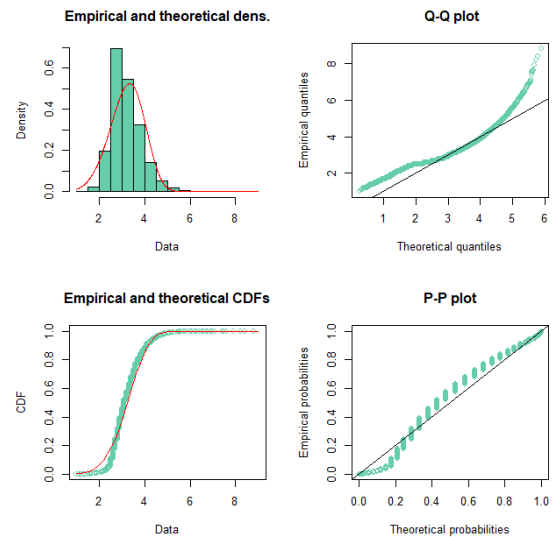


Figura 5.15: Gráficas Weibull.

La distribución Gamma presenta un ajuste razonable a la forma de los datos, aunque se observa una ligera desviación en los extremos, también, muestra que los cuantiles empíricos se desvían de la línea de referencia principalmente en los extremos, lo que indica cierta discrepancia en las colas. La CDF teórica sigue bastante bien a la empírica, lo que sugiere un buen ajuste en términos de probabilidad acumulada y , aunque hay una leve separación entre probabilidades empíricas y teóricas, el patrón sigue relativamente cerca de la diagonal.

Por su parte, el ajuste de la Weibull no sigue tan de cerca la densidad empírica como lo hace la Gamma, mostrando diferencias visibles en la cola derecha, los cuantiles empíricos se desvían significativamente de la línea teórica, sobre todo en los valores bajos, lo que indica un ajuste deficiente. Hay una mayor discrepancia entre la función de distribución empírica y la teórica comparado con la Gamma. Por último, se observa una mayor dispersión respecto a la diagonal, lo que refuerza la conclusión de un ajuste inferior.

Por lo tanto, se puede concluir que la distribución Gamma ajusta mejor a los datos que la distribución Weibull. Aunque ninguna es perfecta, la Gamma logra capturar de mejor manera la forma general de la distribución como la probabilidad acumulada.

La base de datos presenta una calidad elevada en cuanto a la integridad de las variables temporales y geográficas, lo que permite un análisis robusto de la actividad sísmica. La distribución de la latitud y la longitud confirma la concentración geográfica de los eventos, centrándose solo en el territorio nacional. Mientras que la profundidad y la magnitud revelan la heterogeneidad y la presencia de valores extremos. Este análisis es fundamental para comprender tanto la variabilidad

como los patrones atípicos en la sismicidad, lo que puede ser de gran utilidad para estudios en sismología y, en particular, para esta tesis.

5.2. Análisis exploratorio de los datos

El análisis exploratorio de datos permite identificar relaciones y patrones ocultos en la información sísmica. Mediante técnicas como el estudio de correlaciones, la distribución geográfica de los eventos y la evolución temporal de los sismos, esta sección profundiza en la interacción entre variables y en la detección de posibles tendencias espaciales y temporales relevantes para la tesis.

En la matriz de correlaciones se aprecia una fuerte correlación positiva entre Latitud y Longitud (0,732) y una correlación aún más fuerte entre Longitud y Profundidad (0,811), lo que sugiere patrones espaciales significativos en los datos; también se observa una correlación positiva moderada entre Latitud y Profundidad (0,472). Sin embargo, la Magnitud muestra correlaciones muy débiles con la Latitud (-0,006), la Longitud (0,022) y la Profundidad (0,123), lo que indica que la magnitud de los eventos sísmicos parece ser independiente de la ubicación geográfica y la profundidad en este conjunto de datos.

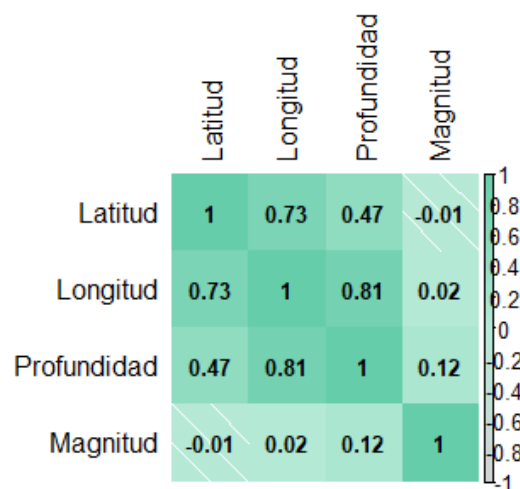


Figura 5.16: Gráfico de correlaciones

Luego de analizar las relaciones entre las variables sísmicas, es importante examinar la evolución temporal de los eventos registrados. Se presenta un análisis de la cantidad de sismos por año, lo que permite identificar posibles tendencias en la actividad sísmica a lo largo del tiempo.

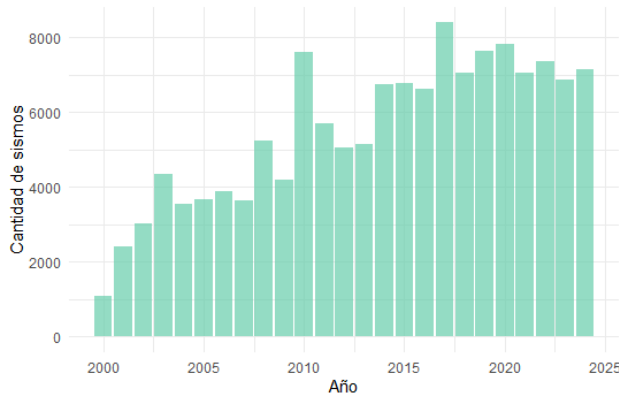


Figura 5.17: Cantidad de sismos por año.

La figura 5.17 muestra una leve tendencia al aumento en la cantidad de sismos registrados desde el año 2000. Sin embargo, este incremento podría estar influenciado por la incorporación de nuevas tecnologías y la mejora en los instrumentos de detección sísmica. El año 2017 destaca como el periodo con mayor número de eventos, alcanzando un total de 8.400 sismos. Además, los años 2010, 2017, 2018, 2019, 2020, 2021, 2022 y 2024 registran más de 7.000 sismos cada uno.

Dado el interés en comprender no solo la frecuencia, sino también la magnitud de los sismos, se realiza una categorización de los eventos según su magnitud. Esta clasificación permitirá identificar qué rangos de magnitud concentran la mayor cantidad de sismos, facilitando así un análisis más detallado del comportamiento sísmico en el país.

$$\text{Magnitud Categorizada} = \begin{cases} \text{Menor a 3} & \text{si Magnitud} < 3, \\ \text{Entre 3 y 4} & \text{si } 3 \leq \text{Magnitud} < 4, \\ \text{Entre 4 y 5} & \text{si } 4 \leq \text{Magnitud} < 5, \\ \text{Entre 5 y 6} & \text{si } 5 \leq \text{Magnitud} < 6, \\ \text{Entre 6 y 7} & \text{si } 6 \leq \text{Magnitud} < 7, \\ \text{Mayor a 7} & \text{si Magnitud} \geq 7 \end{cases}$$

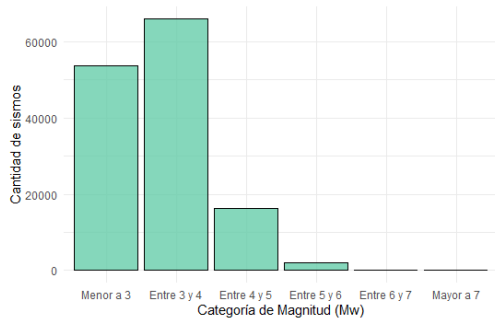


Figura 5.18: Cantidad de sismos por magnitud.

La figura 5.18 muestra la distribución de los sismos según su magnitud. Se observa que 53.658 sismos presentan magnitudes menores a 3 Mw, 65.955 sismos tienen magnitudes entre 3 y 4 Mw, 16.311 sismos se encuentran entre 4 y 5 Mw, 2.007 sismos tienen magnitudes entre 5 y 6 Mw, 179 sismos se ubican entre 6 y 7 Mw, 12 sismos presentan magnitudes iguales o superiores a 7 Mw.

Esta distribución evidencia que la gran mayoría de los eventos sísmicos registrados corresponden a sismos de baja magnitud, mientras que los eventos de mayor magnitud son considerablemente menos frecuentes.

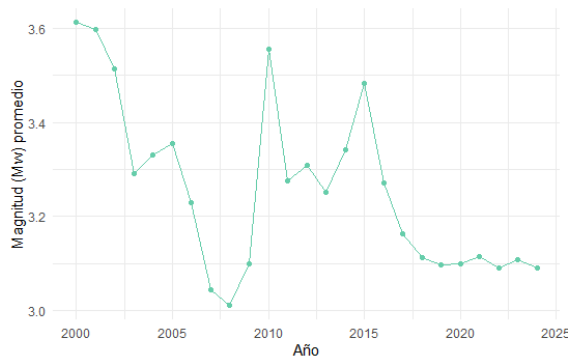


Figura 5.19: Magnitud promedio.

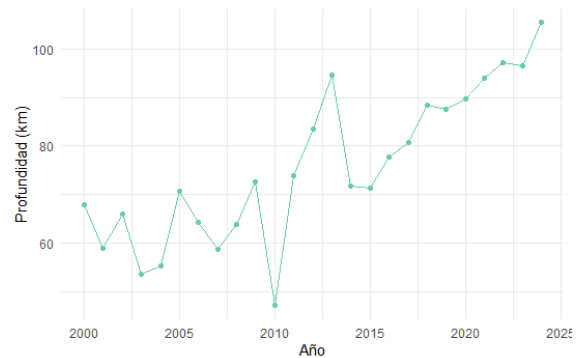


Figura 5.20: Profundidad promedio.

Las figuras 5.19 y 5.20 permiten analizar la evolución temporal de la magnitud y profundidad promedio de los sismos registrados entre los años 2000 y 2024. En cuanto a la magnitud, se observa que en ningún año el promedio supera los 3,7 Mw. El año 2000 destaca por presentar la mayor magnitud promedio, mientras que el año 2010 también resalta con un valor promedio de aproximadamente 3,5 Mw.

En relación con la profundidad, se aprecia una tendencia creciente a lo largo del tiempo. El año 2010, correspondiente al terremoto de mayor impacto en el período analizado, presenta la menor profundidad promedio, cercana a los 35 km. En contraste, el año 2024 registra la mayor profundidad promedio, superando los 100 km.

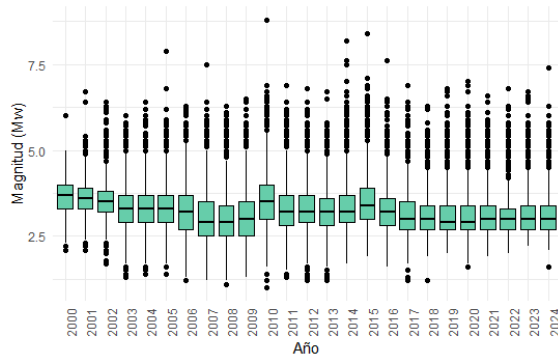


Figura 5.21: Boxplot de la magnitud por año.

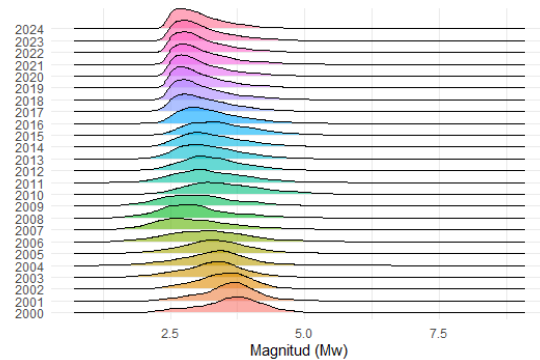


Figura 5.22: Densidad de la magnitud por año.

La figura 5.21 muestra que, en general, las medianas de las magnitudes sísmicas se sitúan en torno a los 3,5 Mw. Además, se observa una alta presencia de valores atípicos (outliers) en la mayoría de los años. En total, se identifican 9 sismos con magnitudes superiores a 7 Mw, los cuales son clasificados como terremotos o eventos principales.

Por otro lado, la figura 5.22 permite visualizar que, para todos los años analizados, existe una mayor concentración de eventos con magnitudes cercanas a los 3,5 Mw. Sin embargo, esta representación no permite distinguir claramente los sismos con magnitudes superiores a 5 Mw, debido a su baja frecuencia relativa.

Para abordar esta limitación, la figura 5.23 presenta una categorización de los sismos según su magnitud. En ella se observa que todos los años, con excepción del 2000, registran eventos con magnitudes menores a 6 Mw, lo que refuerza la predominancia de sismos de baja magnitud en el conjunto de datos.

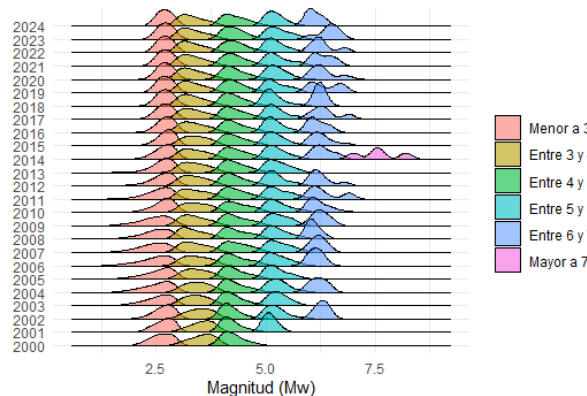


Figura 5.23: Densidad de la categoría de magnitud por año.

En el año 2014 se registraron varios sismos con magnitudes superiores a 7 Mw. No obstante, según la figura 5.21, se observa que al menos en otros seis años también ocurrieron eventos sísmicos que superaron esta magnitud. La diferencia radica en que, durante 2014, se registraron cuatro sismos con magnitudes mayores a 7 Mw, lo que incrementa la densidad de estos eventos, haciéndolos más visibles en la representación gráfica.

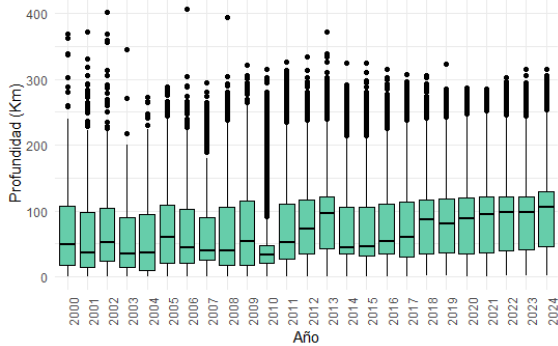


Figura 5.24: Boxplot de la profundidad por año.

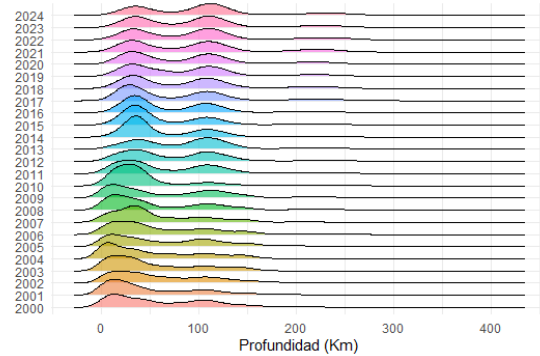


Figura 5.25: Densidad de la profundidad por año.

La figura 5.24 revela la presencia de numerosos valores atípicos en la profundidad de los sismos, algunos de los cuales superan los 400 km. Además, se observa que la mayoría de los años presentan cajas con rangos intercuartílicos similares, con excepción del año 2010, cuya caja es considerablemente más estrecha, lo que indica una menor variabilidad en las profundidades registradas durante ese período.

Por su parte, la figura 5.25 permite visualizar con mayor detalle la distribución de las profundidades a lo largo del tiempo. En ella se aprecia que las densidades tienden a concentrarse en torno a los 50 km y, en menor medida, cerca de los 100 km, lo que sugiere la existencia de dos zonas de acumulación de eventos sísmicos a diferentes niveles de profundidad.

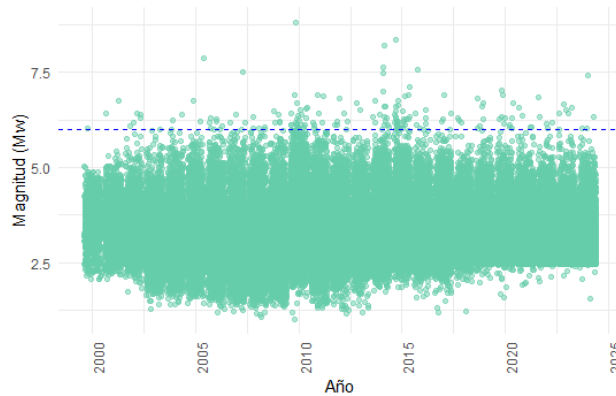


Figura 5.26: Sismos mayores a 6 Mw.

En la figura 5.26 se observa la distribución de las magnitudes de los sismos registrados. La mayoría de estos eventos se concentran entre los 2,5 Mw y 5 Mw. Se identifican 9 sismos con magnitudes iguales o superiores a 7 Mw, y un total de 191 sismos con magnitudes mayores o iguales a 6 Mw, los cuales están delimitados por una línea azul en la figura. Para el entrenamiento del modelo, se considerarán como eventos principales aquellos sismos cuya magnitud supere los 6 Mw.

Dado el interés en analizar el comportamiento sísmico según la ubicación geográfica, se ha creado la variable zona, que clasifica los sismos en tres categorías: Norte, Centro y Sur, dependiendo de la latitud en la que se registraron. Esta división se realizó con base en los valores de latitud presentados en la figura 5.3, donde

$$\text{Zona} = \begin{cases} \text{Norte} & \text{si Latitud} > -26, \\ \text{Centro} & \text{si } -26 \geq \text{Latitud} \geq -40, \\ \text{Sur} & \text{si Latitud} < -40 \end{cases}$$

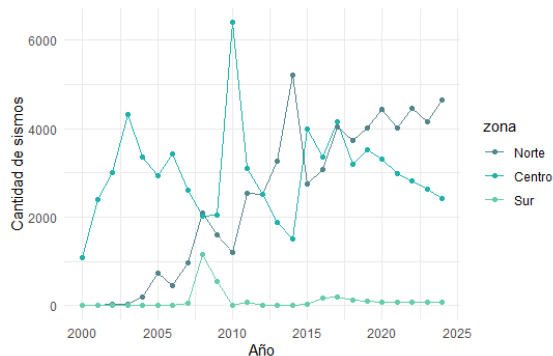


Figura 5.27: Cantidad de sismos por zona.

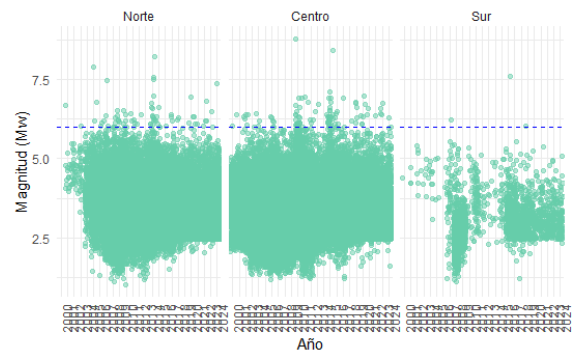


Figura 5.28: Sismos por zona mayores a 6 Mw.

De la figura 5.27 se desprende que la zona con mayor actividad sísmica registrada es la zona centro, seguida por la zona norte y, finalmente, la zona sur. El mayor número de sismos registrados en la zona norte se produjo en el año 2014, con un total de 5.222. En la zona centro, el año con mayor actividad fue 2010, con 6.395 sismos, mientras que en la zona sur el máximo se registró en 2008, con 1.147 eventos.

Por otro lado, la figura 5.28 permite visualizar con mayor claridad la distribución de sismos de gran magnitud en cada zona. Se observa que tanto en la zona norte como en la zona centro existen múltiples registros de sismos con magnitudes superiores a 6 Mw, umbral definido previamente. En contraste, en la zona sur solo se identifican dos eventos que superan dicha magnitud. Esta diferencia podría ser un indicio de la cantidad de clusters sísmicos que es posible identificar en cada región.

Tabla 5.2: Cantidad de sismos.

Zona	Total	Menores a 4 Mw	Mayores a 4 Mw
Norte	60.219	53.694	6.525
Centro	75.002	66.663	8.339
Sur	2.916	2.719	197

De la tabla 5.2 se observa que la mayoría de los sismos registrados en cada una de las zonas presentan magnitudes inferiores a 4 Mw. La zona centro destaca por concentrar la mayor cantidad de eventos sísmicos, tanto de magnitudes menores como mayores a 4 Mw. En contraste, la zona sur es la que registra el menor número de sismos en ambos rangos de magnitud. A partir de estos datos, se puede concluir que la zona central de Chile es la más sísmicamente activa del país.

Con el objetivo de comprender mejor el comportamiento espacial de la actividad sísmica, a continuación se presenta un análisis basado en las ubicaciones geográficas de los eventos registrados. Este enfoque permite identificar patrones de distribución y zonas de mayor concentración sísmica.

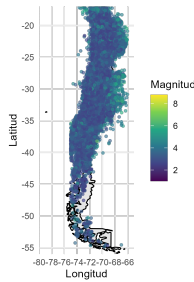


Figura 5.29: Distribución espacial de los sismos.

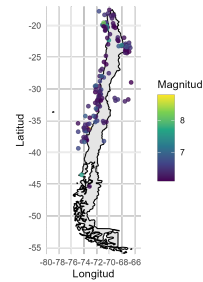


Figura 5.30: Distribución espacial de sismos mayores a 6 Mw.

En la figura 5.29 se observa que los sismos se distribuyen a lo largo de todo el territorio nacional. Sin embargo, su frecuencia disminuye notablemente a partir de la latitud 45° sur. Además, según la escala de colores utilizada en el gráfico, la mayoría de estos eventos sísmicos presentan magnitudes entre 3 Mw y 5 Mw. Al aplicar un filtro para considerar únicamente los sismos de magnitud igual o superior a 6 Mw, como se muestra en la figura 5.30, se evidencia una reducción significativa en la cantidad de eventos. Estos sismos de mayor magnitud se concentran principalmente en la zona costera del norte y centro del país. En particular, en el norte también se identifica un grupo de eventos localizados en las cercanías de la cordillera de los Andes. Por otro lado, en la zona sur solo se registran dos sismos de magnitud mayor a 6 Mw, lo que refuerza la menor actividad sísmica en esa región.

Con el fin de reforzar las observaciones realizadas a partir de la distribución geoespacial de los sismos, resulta de interés repetir el análisis, esta vez segmentando los eventos según las distintas zonas del país. Esta aproximación permitirá confirmar si efectivamente las regiones norte y centro concentran la mayor cantidad de sismos, especialmente aquellos de mayor magnitud, y si la actividad sísmica en la zona sur es considerablemente menor. Al clasificar los eventos por zonas geográficas, se busca validar de manera más precisa los patrones espaciales previamente identificados.

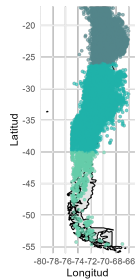


Figura 5.31: Distribución espacial de los sismos por zona.

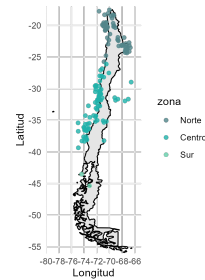


Figura 5.32: Distribución espacial de sismos mayores a 6 Mw por zona.

Al realizar el análisis segmentado por zonas geográficas, se confirma lo observado anteriormente. Tal como se muestra en las figuras 5.31 y 5.32, las zonas norte y centro del país concentran la mayor parte de los sismos, especialmente aquellos de magnitud igual o superior a 6 Mw. En contraste, la zona sur presenta una actividad sísmica considerablemente menor, registrando solo unos pocos eventos de alta magnitud. Estos resultados refuerzan la idea de que la actividad sísmica en Chile no solo es más frecuente, sino también más intensa en las regiones ubicadas más al norte, particularmente en sectores costeros y cercanos a la cordillera de los Andes.

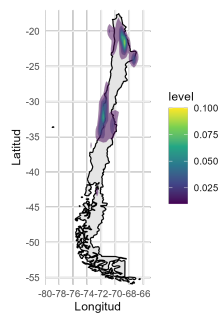


Figura 5.33: Mapa de calor.

La figura 5.33 muestra claramente que la mayor densidad de eventos sísmicos se concentra en la zona costera de la región centro y en sectores cercanos a la cordillera en la zona norte. Esta visualización permite identificar con mayor precisión las áreas con mayor recurrencia de sismos.

K-means

Con el objetivo de identificar patrones espaciales más definidos en la distribución de los sismos, se aplicó el algoritmo de agrupamiento K-means sobre las coordenadas geográficas de

los eventos registrados. Esta técnica permite segmentar el conjunto de datos en grupos o clusters que comparten características similares, en este caso, su proximidad geográfica. A través de este enfoque, es posible detectar zonas con alta concentración de actividad sísmica y analizar si estas coinciden con las divisiones territoriales previamente consideradas.

Como primer paso, se determinó el número óptimo de clusters utilizando el método del codo (elbow method), el cual permite identificar el punto a partir del cual agregar más grupos no mejora significativamente la compactación de los datos. A partir de este análisis, se definió el número adecuado de agrupaciones, lo que permitió generar una segmentación más representativa de la actividad sísmica. A continuación, se presentan los resultados del agrupamiento y su interpretación en el contexto del comportamiento sísmico observado.

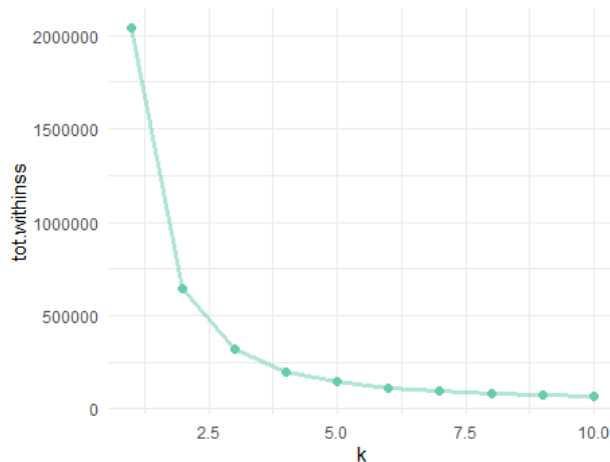


Figura 5.34: Método del codo.

En la figura 5.34 se observa la gráfica del método del codo, la cual permite identificar el número óptimo de clusters para aplicar el algoritmo K-means. En este caso, el “codo” se forma claramente en el valor 2, lo que indica que a partir de ese punto la mejora en la compactación de los datos al aumentar el número de grupos es marginal. Por esta razón, se decidió utilizar K-means con $k = 2$, permitiendo así una segmentación eficiente y representativa de los patrones espaciales presentes en los datos sísmicos.

Adicionalmente, se consideró una segunda configuración del algoritmo K-means utilizando $k = 3$, con el objetivo de explorar una segmentación que refleje de manera más directa las agrupaciones naturales observadas a lo largo de la latitud del país. Esta elección se basa en la división geográfica tradicional de Chile en zonas norte, centro y sur, lo que permite comparar los resultados del agrupamiento automático con las divisiones territoriales previamente utilizadas en el

análisis. Esta aproximación busca evaluar si los patrones sísmicos identificados mediante K-means coinciden con las zonas geográficas comúnmente reconocidas.

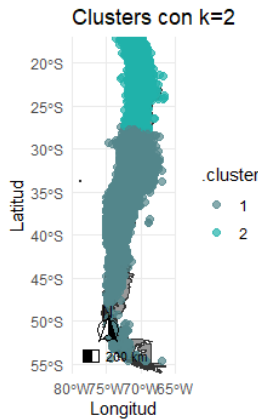


Figura 5.35: Cluster $k = 2$.

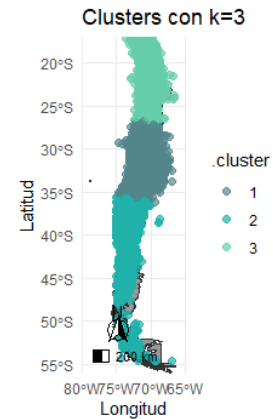


Figura 5.36: Cluster $k = 3$.

En la figura 5.35, se observa que el cluster 1 abarca principalmente la zona centro-sur del país, mientras que el cluster 2 corresponde a la zona norte. Es notable que el cluster 1 agrupa tanto la zona central como la sur, y que, según las gráficas de densidad (verificar si efectivamente representan densidad), la distribución espacial del cluster 2 se concentra más cerca de la cordillera. En contraste, el cluster 1 muestra una distribución más homogénea a lo largo del territorio.

Por otro lado, en la figura 5.36, se distingue una segmentación más detallada: el cluster 1 representa la zona centro, el cluster 2 la zona norte, y el cluster 3 la zona sur. En comparación con la figura anterior, se añade un nuevo cluster que permite diferenciar la zona sur de la central. Además, los límites de estos clusters coinciden de manera bastante precisa con las divisiones previamente establecidas según la latitud.

Series de tiempo

Con el propósito de analizar la evolución temporal de la actividad sísmica, se desarrolló un estudio basado en series de tiempo. Este enfoque permite observar cómo varía la frecuencia y magnitud de los sismos a lo largo del tiempo, identificar posibles tendencias, patrones recurrentes, y detectar periodos de mayor o menor actividad. A través de este análisis, se busca complementar la perspectiva espacial previamente abordada, incorporando la dimensión temporal como un elemento clave para comprender el comportamiento sísmico en el territorio.

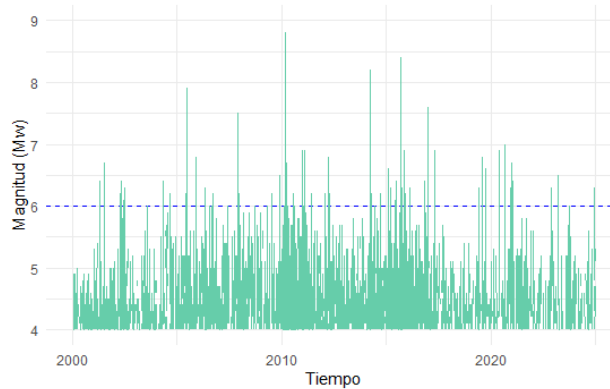


Figura 5.37: Sismos en el tiempo.

En la figura 5.37 se observa la ocurrencia de múltiples sismos con magnitudes superiores a 6 Mw. No se aprecia una estacionalidad clara en los datos, aunque sí es posible identificar con mayor nitidez las réplicas asociadas a eventos principales. Es importante tener en cuenta que esta figura incluye todos los sismos registrados en el país, lo que implica que pueden estar representados eventos de distintas regiones. Esta superposición puede influir en el análisis e interpretación de los resultados.

A partir de esta observación, se procede a un análisis más detallado de cada terremoto. Para este propósito, se consideraron como eventos principales aquellos sismos cuya magnitud es igual o superior a 7 Mw. Además, con el fin de enfocar el análisis en una misma zona geográfica, se aplicó un filtro por latitud, seleccionando únicamente los sismos que ocurrieron dentro de una misma región.

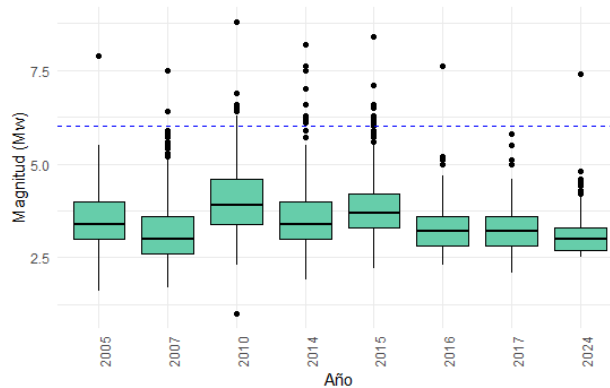


Figura 5.38: Magnitud de los terremotos y sus réplicas.

En la figura 5.38 se puede observar que, desde el año 2000, han ocurrido ocho terremotos

en distintas regiones del país. En cada caso, el punto con mayor magnitud representa el evento principal, mientras que los demás puntos del diagrama de caja y bigotes corresponden a las réplicas asociadas a ese terremoto. La línea azul en el gráfico indica el umbral de 6 Mw, lo que permite visualizar fácilmente qué eventos superan esta magnitud.

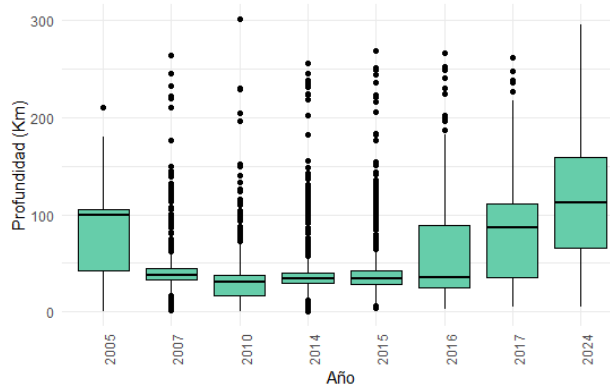


Figura 5.39: Profundidad de los terremotos y sus réplicas.

En la figura 5.39 se observa que la mayoría de los eventos sísmicos presenta valores atípicos (outliers) en cuanto a la profundidad.

5.3. Análisis de patrones sísmicos

Una vez completado el análisis exploratorio de la base de datos de sismos en Chile, se procedió a transformar la información a una estructura de secuencias sísmicas. Este paso es esencial para el entrenamiento del modelo generativo adversarial (GAN) [14]. El procedimiento consistió en identificar, dentro del conjunto de datos, los eventos principales, definidos como aquellos con magnitud superior a 6 Mw, y agrupar en torno a ellos las posibles réplicas. Estas réplicas debían encontrarse dentro de una ventana temporal de dos semanas y, espacialmente, ubicarse dentro de las 17 regiones sísmicas de Chile, las cuales están definidas en [24].

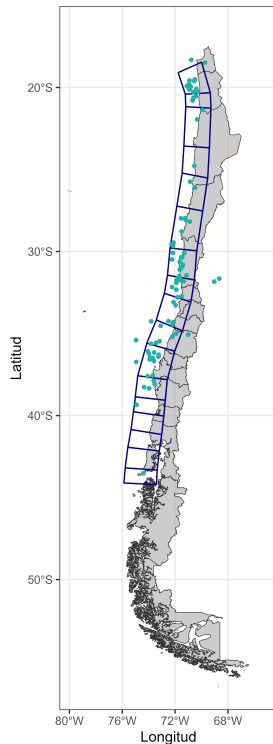


Figura 5.40: Zonas sísmicas y eventos principales.

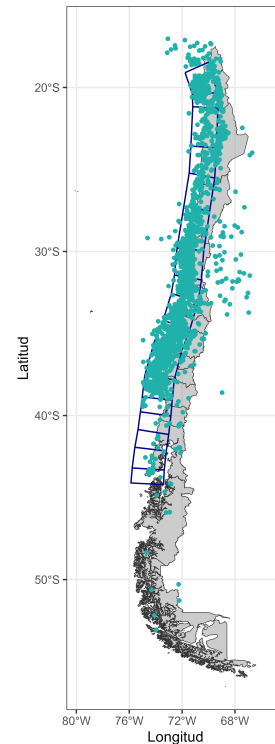


Figura 5.41: Zonas sísmicas y sismos con magnitudes mayores o iguales a 4,5 Mw.

En la figura 5.40 se presentan las 17 zonas sísmicas del país junto con los sismos ocurridos en dichas zonas cuya magnitud fue igual o superior a 6 Mw. Por otro lado, la figura 5.41 muestra las mismas 17 zonas sísmicas, incorporando todos los eventos sísmicos con magnitudes iguales o superiores a 4,5 Mw.

No obstante, al aplicar esta metodología exclusivamente sobre los datos de sismos ocurridos en Chile, se constató una cantidad limitada de secuencias (191 en total), lo que resultaba insuficiente para entrenar adecuadamente el modelo. Considerando que los modelos generativos requieren un volumen considerable de datos para lograr una fase de aprendizaje eficaz, se adoptó la decisión metodológica de ampliar la base de datos incorporando información sísmica proveniente de todo el Anillo de Fuego del Pacífico. Esta región concentra la mayor actividad sísmica mundial [17], lo que permite incrementar significativamente el número de secuencias útiles para el entrenamiento.

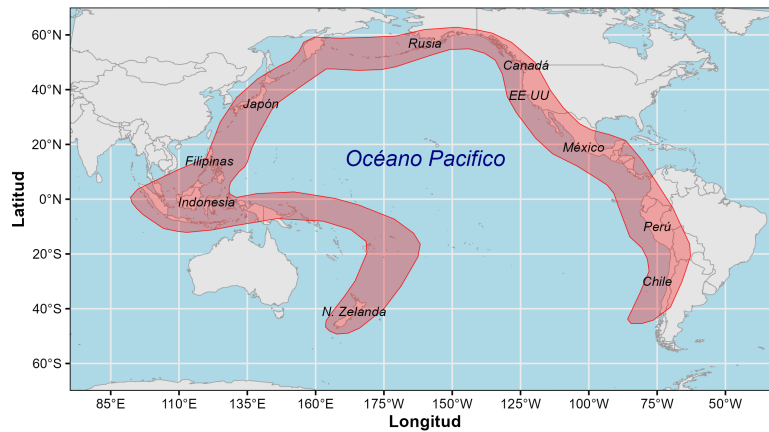


Figura 5.42: Anillo de Fuego.

En la figura 5.42 se muestra el Anillo de Fuego del Pacífico, que abarca países como Chile, Perú, México, Estados Unidos, Japón, Filipinas, Indonesia, entre otros. Estas regiones son ampliamente reconocidas por ser la zona sísmica y volcánica más activa del planeta [17].

La decisión de ampliar la base de datos al Anillo de Fuego del Pacífico se fundamenta en que este cinturón tectónico concentra la mayoría de los megaterremotos asociados a zonas de subducción a nivel global [4],[41]. La dinámica de subducción que caracteriza tanto a Chile como al resto del Anillo permite mantener coherencia en los mecanismos físicos de generación de grandes sismos y réplicas. Asimismo, la variabilidad global de las rupturas en megaterremotos presenta patrones estructurales y cinemáticos comunes, lo que respalda el uso de catálogos ampliados para robustecer la muestra de secuencias sísmicas [4]. Por otra parte, leyes empíricas ampliamente validadas, como la de Omori-Utsu para la tasa de decaimiento de réplicas, han sido comprobadas en distintos contextos de subducción alrededor del mundo [43]. En consecuencia, la ampliación al Anillo de Fuego no solo incrementa el tamaño muestral disponible, sino que también se sustenta en evidencia científica que demuestra que las secuencias sísmicas en zonas de subducción comparten propiedades fundamentales, facilitando así el entrenamiento de modelos predictivos más robustos.

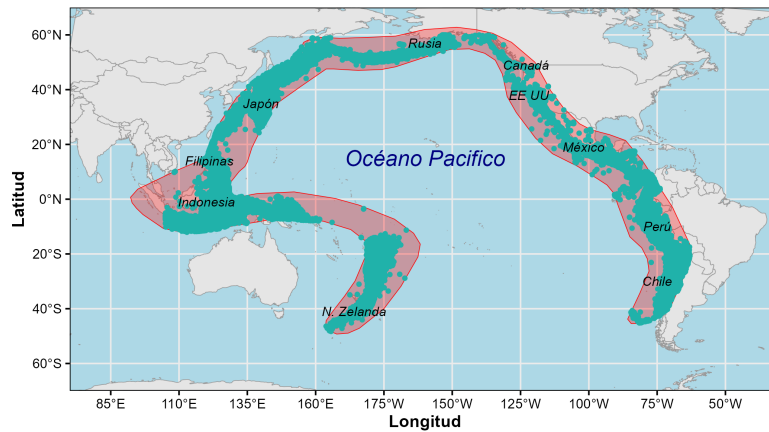


Figura 5.43: Sismos en el anillo de fuego.

En la figura 5.43 se visualizan todos los sismos ocurridos en el Anillo de Fuego del Pacífico entre los años 2000 y 2024, con magnitudes superiores a 4 Mw. Se aprecia una distribución amplia de eventos sísmicos a lo largo de toda la región, lo que confirma su elevada actividad tectónica. Al ampliar la base de datos para abarcar la totalidad del Anillo de Fuego, se incrementó significativamente el volumen de información disponible para el análisis. Se pasó de una base inicial de aproximadamente 138.000 observaciones centradas en el territorio nacional a un conjunto de aproximadamente 1.900.000 registros sísmicos. Asimismo, el número de secuencias identificadas aumentó de 191 a 2.179.

Es relevante destacar que esta expansión de la base de datos tiene un propósito exclusivamente técnico: optimizar el desempeño del modelo durante su fase de entrenamiento. En ningún caso se busca extrapolar los resultados obtenidos en el Anillo de Fuego al contexto chileno. Por el contrario, los datos correspondientes a Chile se reservan estrictamente para la etapa de evaluación del modelo.

Con el fin de estructurar secuencias sísmicas que nutran el proceso de entrenamiento del modelo, fue necesario adaptar la base de datos del Anillo de Fuego.

Sabemos que un sismo es un proceso de generación y propagación de ondas a través del interior de la Tierra, las cuales producen movimiento y vibración del suelo. Para que esta vibración sea perceptible, la energía liberada debe ser suficiente para generar un desplazamiento significativo del terreno. Considerando esta información, se puede afirmar que existe una vibración constante del suelo, aunque en muchos casos la energía liberada no es suficiente para que estos movimientos sean perceptibles por las personas o incluso detectables por los sismógrafos. Bajo esta lógica, es razonable suponer que los sismos podrían estar ocurriendo de manera equidistante en el tiempo,

pero que solo aquellos con suficiente energía son registrados.

Con base en este argumento, se propone asumir una distribución temporal equidistante para los sismos. Para ello, es necesario primero identificar la relación temporal entre los eventos sísmicos registrados. En este contexto, se realiza un análisis para los terremotos ocurridos en Chile, con el objetivo de entender el comportamiento temporal que hay entre un evento principal y sus réplicas. Este análisis se encuentra en el Anexo A.1.

A continuación se presenta el paso a paso para obtener las secuencias de réplicas en el anillo de fuego:

1. **Selección de eventos principales:** Se identifican los sismos cuya magnitud sea igual o superior a 6 Mw. Estos se consideran como eventos principales dentro del análisis.
2. **Aplicación de filtros temporales:** Una vez definidos los eventos principales, se establecen filtros temporales para delimitar las secuencias de réplicas asociadas. Dichos filtros consisten en:
 - Búsqueda de sismos cada 90 minutos.
 - Repetición de la búsqueda durante un período de dos semanas posteriores al evento principal.
3. **Definición del filtro espacial:** Se traza un radio de 100 kilómetros desde las coordenadas (latitud y longitud) del evento principal. Este radio delimita el área en la que se buscarán las réplicas.
4. **Identificación de réplicas:** Para cada evento principal, se localizan los sismos ocurridos dentro del radio de 100 km, siguiendo el intervalo de 90 minutos durante dos semanas. Estos eventos se consideran como réplicas del evento principal.
5. **Simulación en ausencia de réplicas:** En caso de que no se registre ningún sismo dentro de un intervalo de 90 minutos, se incorpora un evento sintético con una magnitud de 2.5 Mw, a modo de sustitución para mantener la continuidad de la secuencia.

Los criterios aplicados para la construcción de secuencias sísmicas son coherentes con los utilizados en estudios previos sobre agrupamiento de réplicas sísmicas [46][28].

La elección de una ventana de 90 minutos busca tener un equilibrio entre la representatividad y la comparabilidad de las secuencias. En términos prácticos, un intervalo

reducido permite construir series de longitud controlada, evitando la heterogeneidad excesiva que se generaría al considerar ventanas muy extensas.

Una vez completadas las etapas descritas, se obtiene la estructura final de la base de datos, adecuada para su implementación en el modelo generativo adversarial (GAN). El resultado de este proceso consiste en dos conjuntos de datos: el primero contiene información completa sobre los eventos principales, totalizando 2.179 registros; el segundo agrupa las magnitudes de las réplicas asociadas a cada evento principal, con un total de 224 réplicas por evento.

A continuación, se presentan visualizaciones del evento principal junto con las secuencias de réplicas correspondientes a los sismos de mayor magnitud registrados en el Anillo de Fuego del Pacífico.

Las figuras 5.44, 5.45, 5.46 y 5.47 presentan una selección de cuatro secuencias sísmicas representativas del Anillo de Fuego del Pacífico. Cada una incluye tanto el evento principal como las réplicas asociadas, proporcionando ejemplos concretos de los patrones que se pretende capturar y modelar mediante la arquitectura GAN. Estas visualizaciones, derivadas directamente de la construcción de la nueva base de datos, también ilustran de manera clara la estructura que conforma el input del modelo generativo.

Terremoto: Perú 2001

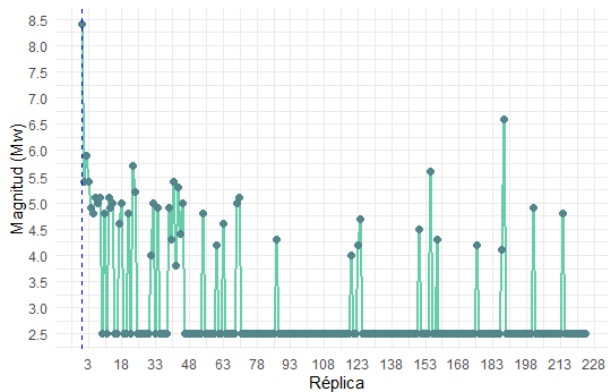


Figura 5.44: Terremoto Perú 2001.

El terremoto ocurrido en Perú el 23 de junio de 2001 alcanzó una magnitud de 8,4 Mw. En la figura 5.44 se distingue el evento principal, seguido por la secuencia de réplicas asociadas a este sismo.

Terremoto: Chile (Concepción) 2010

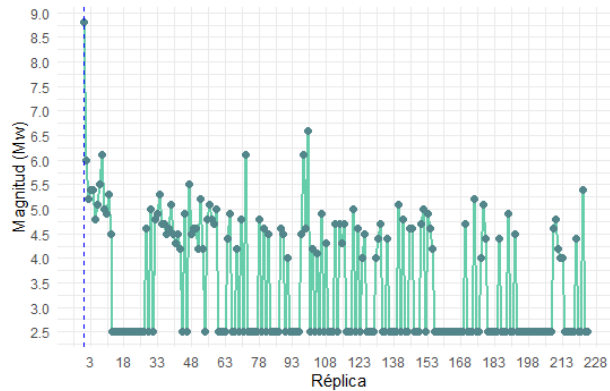


Figura 5.45: Terremoto Chile (Concepción) 2010.

El terremoto ocurrido en Chile el 27 de febrero de 2010 alcanzó una magnitud de 8,8 Mw. En la figura 5.45 se distingue el evento principal, seguido por la secuencia de réplicas asociadas a este sismo.

Terremoto: Japón 2011

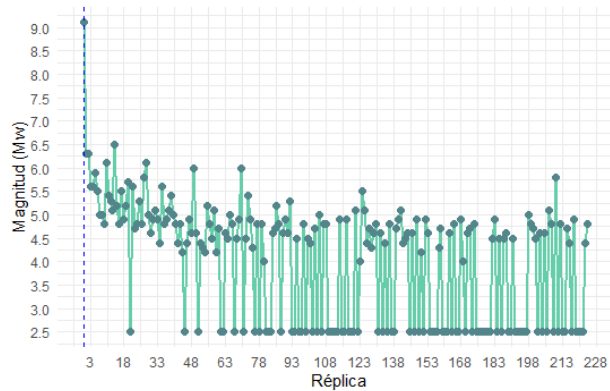


Figura 5.46: Terremoto Japón 2011.

El terremoto ocurrido en Japón el 11 de marzo de 2011 alcanzó una magnitud de 9,1 Mw. En la figura 5.46 se distingue el evento principal, seguido por la secuencia de réplicas asociadas a este sismo.

Terremoto: Chile (Iquique) 2014

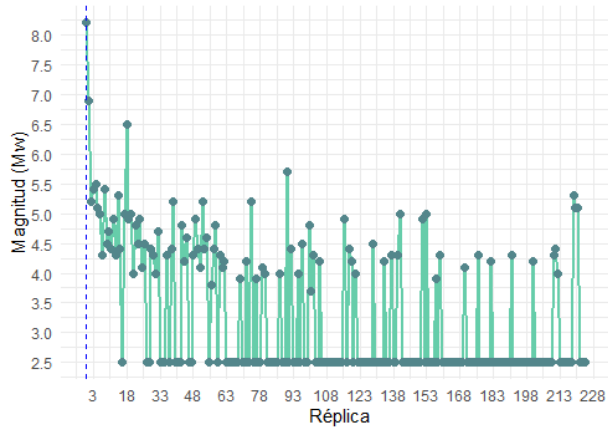


Figura 5.47: Terremoto Chile (Iquique) 2014.

El terremoto ocurrido en Chile el 01 de abril de 2014 alcanzó una magnitud de 8,2 Mw. En la figura 5.47 se distingue el evento principal, seguido por la secuencia de réplicas asociadas a este sismo.

5.4. Resultados del modelo

En esta sección se presentan los resultados obtenidos a partir de tres arquitecturas generativas implementadas con el objetivo de modelar secuencias de réplicas sísmicas para Chile. En primer lugar, se utilizó un GAN estándar [15], compuesto por un generador y un discriminador basados en redes densas, el cual constituye la referencia inicial para evaluar la capacidad del modelo de reproducir patrones temporales y magnitudes de las secuencias. Posteriormente, se incorporó una variante en la que el generador integra una capa Transformer [45], con el propósito de capturar dependencias de largo alcance en las secuencias, dada la capacidad demostrada de este tipo de arquitecturas para modelar relaciones temporales complejas. Finalmente, se probó un GAN con función de costo Wasserstein [1], el cual reemplaza la divergencia de Jensen–Shannon por la distancia de Wasserstein como métrica de entrenamiento, buscando una mayor estabilidad en la optimización y una mejora en la calidad de las secuencias generadas. La comparación de estas tres configuraciones permite evaluar no solo la capacidad de cada modelo para reproducir las características estadísticas de las réplicas, sino también la robustez de su entrenamiento y su potencial aplicación en escenarios predictivos.

La elección de los tres modelos propuestos responde a la necesidad de abordar la generación de secuencias sísmicas desde perspectivas complementarias. El cGAN se emplea como punto de partida al ser un modelo generativo clásico que permite condicionar la síntesis de datos en variables sísmológicas clave, ofreciendo una línea base de comparación. El TGAN, por su parte, se justifica en su capacidad para modelar dependencias de largo alcance y relaciones complejas en secuencias temporales, aprovechando mecanismos de atención que resultan adecuados para capturar patrones no lineales en la ocurrencia de réplicas. Finalmente, el WGAN se incorpora debido a sus ventajas en términos de estabilidad y calidad de la generación, mitigando problemas comunes como el colapso de modos mediante la distancia de Wasserstein y la penalización de gradiente.

El modelo generativo utilizado, basado en arquitecturas GAN con capas tipo Transformer o Wasserstein, produce la secuencia completa de eventos sísmicos (incluyendo las 224 réplicas) en una única salida, sin recurrir a generación autoregresiva ni retroalimentación paso a paso. Esta estrategia permite capturar dependencias globales y patrones estructurales de la secuencia, aunque puede implicar una menor precisión en los eventos más tardíos, dado que el modelo no ajusta dinámicamente su salida según la evolución temporal. En consecuencia, aunque se logra una representación coherente del comportamiento general de las réplicas, es esperable que la calidad de las predicciones decaiga hacia los últimos sismos de cada secuencia.

Los tres modelos seleccionados cubren distintas dimensiones metodológicas: un modelo base (cGAN), una variante orientada a dependencias temporales complejas (TGAN) y una variante enfocada en la estabilidad de entrenamiento (WGAN). A continuación, se presentan sus resultados, mostrando la arquitectura de cada modelo, las métricas generales de desempeño en el conjunto de prueba y el análisis detallado de las secuencias generadas para tres terremotos ocurridos en Chile con magnitudes superiores a 7 Mw, que fueron identificados dentro del conjunto de testeo. Esta presentación permite comparar la capacidad de cada modelo para reproducir patrones y dependencias de las secuencias sísmicas.

- **Terremoto Tarapacá 2005:** El 13 de junio de 2005 se registró un terremoto con una magnitud de 7,8 Mw, correspondiente a uno de los eventos sísmicos de mayor intensidad ocurridos en Chile durante ese año.
- **Terremoto Concepción 2010:** El 27 de febrero del 2010 se registró un terremoto con una magnitud de 8,8 Mw, siendo este uno de los eventos sísmicos más grandes en la historia del país.
- **Terremoto Iquique 2014:** El 1 de abril de 2014 se produjo un terremoto de magnitud 8,2 Mw

frente a las costas de Iquique. Al día siguiente, el 2 de abril, ocurrió otro sismo de magnitud 7,7 Mw, el cual corresponde al evento incluido en el conjunto de testeo del modelo.

5.4.1. Modelo cGAN estándar (cGAN)

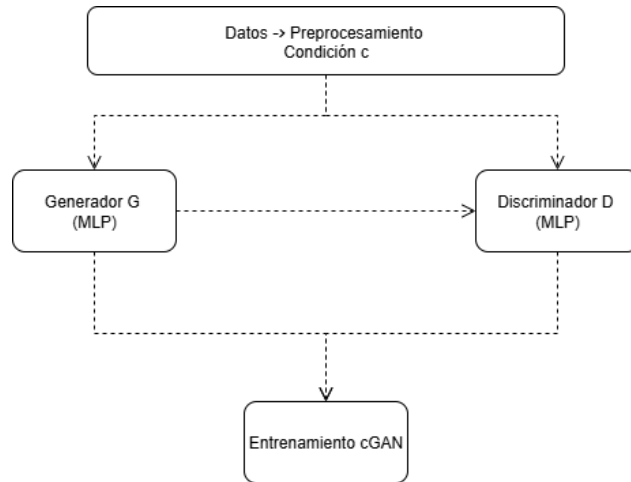


Figura 5.48: Esquema modelo cGAN

El modelo cGAN se implementa mediante una arquitectura relativamente simple pero funcional para la generación de secuencias sísmicas. Como se observa en la figura 5.48, el generador recibe como entradas un vector de ruido aleatorio $z \in \mathbb{R}^{100}$ y una condición $c \in \mathbb{R}^5$ compuesta por la información input del modelo (longitud, latitud, magnitud, profundidad y longitud en 360° para evitar problemas en las zonas cercanas a los 0°). Ambas entradas se concatenan y atraviesan varias capas densas MLP¹ de tamaño progresivo (256, 512 y 1024 neuronas), generando finalmente una secuencia sintética $\hat{y} \in \mathbb{R}^T$ escalada en el rango $[-1; 1]$ mediante una función de activación tangente hiperbólica. En paralelo, el discriminador recibe tanto secuencias reales y como generadas \hat{y} , concatenadas con la condición c , y las procesa a través de capas densas MLP decrecientes (1024, 512 y 256 neuronas). La salida del discriminador corresponde a una probabilidad de realismo en el intervalo $(0, 1)$, calculada mediante una función de activación sigmoide. El entrenamiento se realiza utilizando pérdidas de entropía binaria cruzada (Binary Cross-Entropy). La optimización se lleva a cabo con el algoritmo Adam ($lr = 2 \times 10^{-4}$; $\beta_1 = 0,5$; $\beta_2 = 0,999$).

¹Perceptrón Multicapa.

El modelo se entrena con 10.000 épocas y tuvo una duración de ejecución aproximada de 15 minutos utilizando GPU y utilizando el lenguaje de programación Python.

Tabla 5.3: Métricas modelo cGAN.

RMSE	MAE	MAPE	PI
1,2427	0,7696	28,23 %	-299,49 %

Las métricas obtenidas para el modelo cGAN indican un desempeño razonable en la generación de secuencias sísmicas, aunque con ciertas limitaciones en la precisión de los valores individuales. El RMSE y el MAE reflejan errores absolutos y cuadráticos moderados, lo que sugiere que el modelo logra capturar la estructura general de las secuencias, aunque con desviaciones puntuales. El MAPE indica que, en promedio, las predicciones presentan un error relativo cercano al 30 %, lo cual es aceptable en contextos de alta variabilidad como la sismicidad. Un PI de -299,49 % revela que el modelo tiene un rendimiento considerablemente inferior al de una estrategia de predicción persistente, lo que sugiere deficiencias en la representación de la dinámica temporal y la variabilidad de réplicas individuales. Este resultado podría estar asociado a errores de alineación secuencial, acumulación de ruido en la generación simultánea o limitaciones estructurales del modelo en la captura de dependencias locales.

A continuación se detallan los resultados obtenidos por el modelo cGAN.

Terremoto Tarapacá 2005

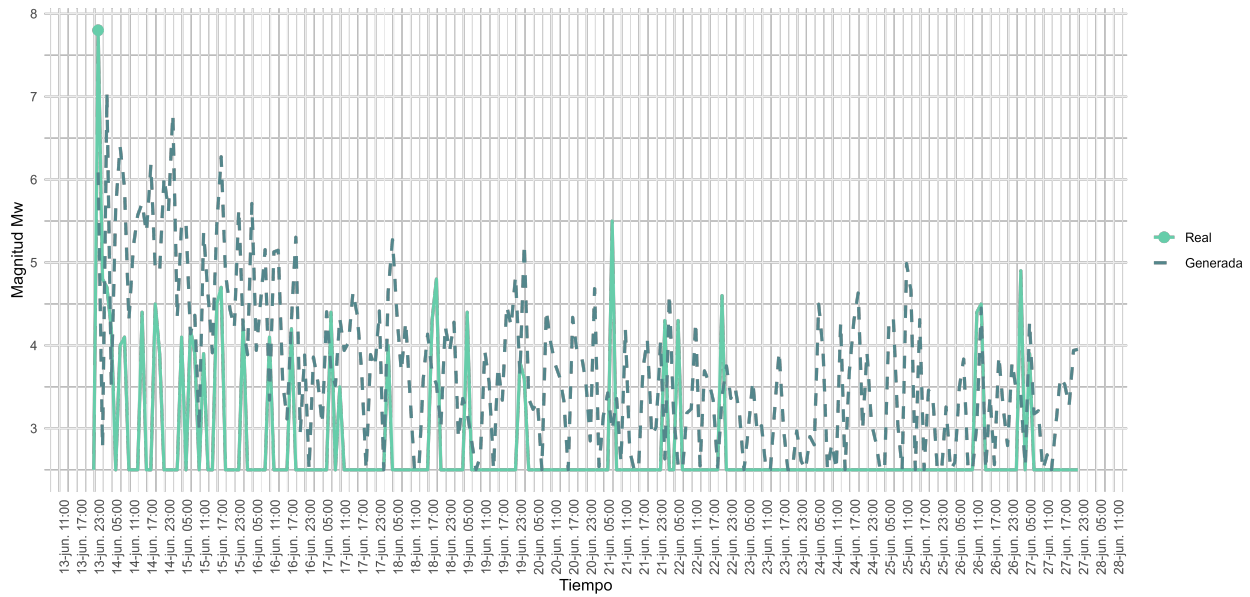


Figura 5.49: Resultados modelo cGAN Tarapacá 2005

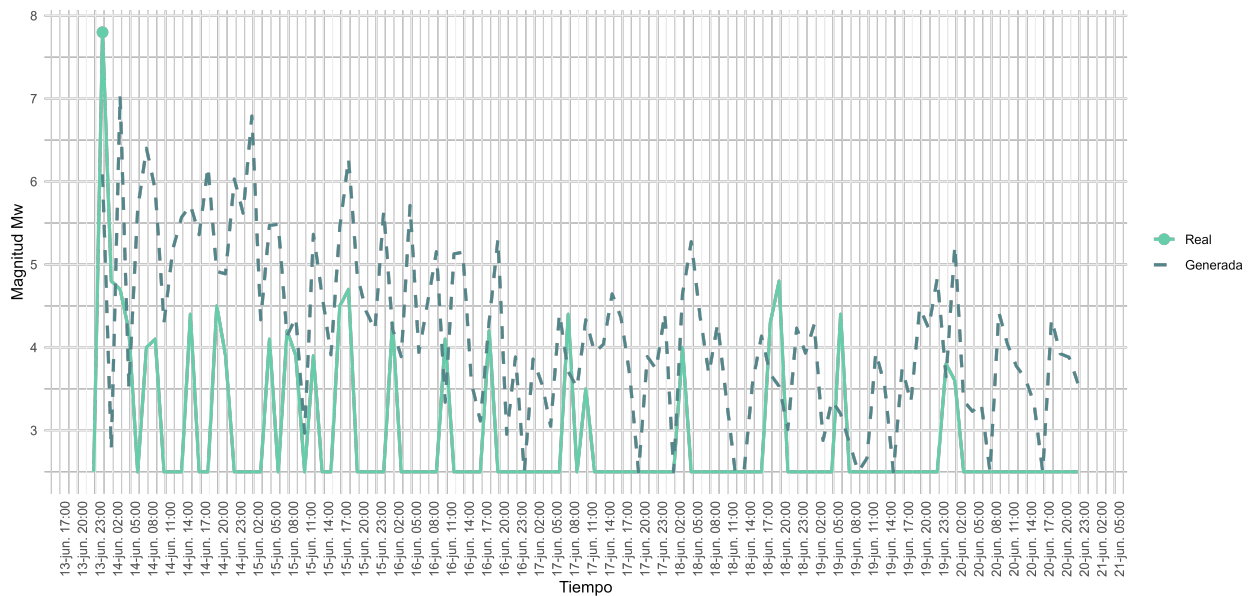


Figura 5.50: Resultados una semana modelo cGAN Tarapacá 2005

En la figura 5.49 se observa en color celeste el evento principal (representado por un punto) junto con toda la secuencia de réplicas reales asociadas al terremoto de Tarapacá de 2005. Por su parte, la secuencia generada por el modelo GAN se muestra en color gris punteado. Se aprecia

que, durante los primeros días posteriores al evento principal, el modelo tiende a sobreestimar la magnitud de las réplicas en comparación con los valores observados. A medida que transcurre el tiempo, las magnitudes generadas se ajustan de forma más cercana a las reales, aunque presentan un desfase temporal en su ocurrencia. A partir del 22 de junio, el modelo continúa generando réplicas incluso cuando en la secuencia real ya no se registran eventos, lo que sugiere una pérdida de precisión en la parte final de la secuencia.

Para facilitar el análisis, se recorta el gráfico y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.50. En esta visualización más acotada, se confirma que el modelo tiende a generar réplicas con magnitudes superiores a las reales y a generarlas cuando no se registran eventos. No obstante, algunas predicciones coinciden en magnitud con los eventos observados, como las réplicas del 15 y del 17 de junio a las 08:00 horas, aunque esta última presenta un leve desfase respecto al momento real de ocurrencia.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.4: Métricas modelo cGAN Terremoto Tarapacá 2005.

RMSE	MAE	MAPE	PI
1,3175	0,9120	21,10 %	-29,49 %

La tabla 5.4 evidencia un desempeño limitado del modelo cGAN en la generación de la secuencia de réplicas del terremoto de Tarapacá. El RMSE y el MAE indican errores absolutos elevados, lo que refleja una sobreestimación sistemática de las magnitudes, confirmando lo observado en los gráficos. El MAPE sugiere que, en promedio, las predicciones se desvían más de un 20 % respecto a los valores observados, lo cual es moderado pero relevante en contextos de alta sensibilidad como la sismicidad. El índice de persistencia revela que el modelo tiene un rendimiento inferior al de una estrategia de predicción persistente, lo que evidencia deficiencias en la representación temporal y en la calibración de la magnitud de las réplicas.

Terremoto Concepción 2010

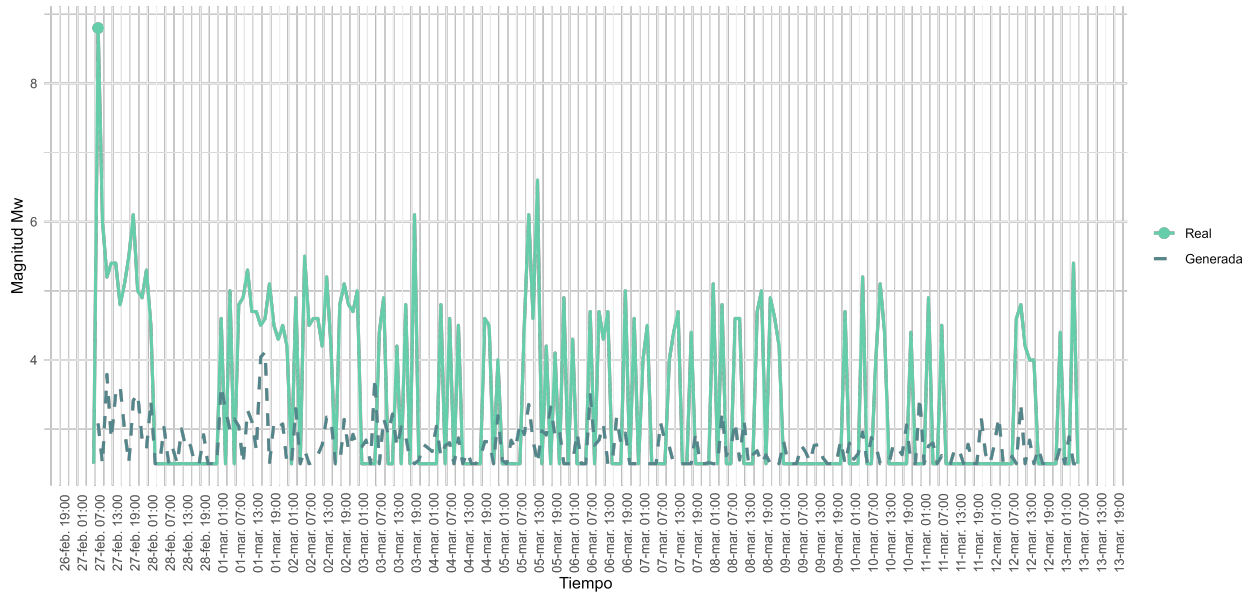


Figura 5.51: Resultados modelo cGAN Concepción 2010

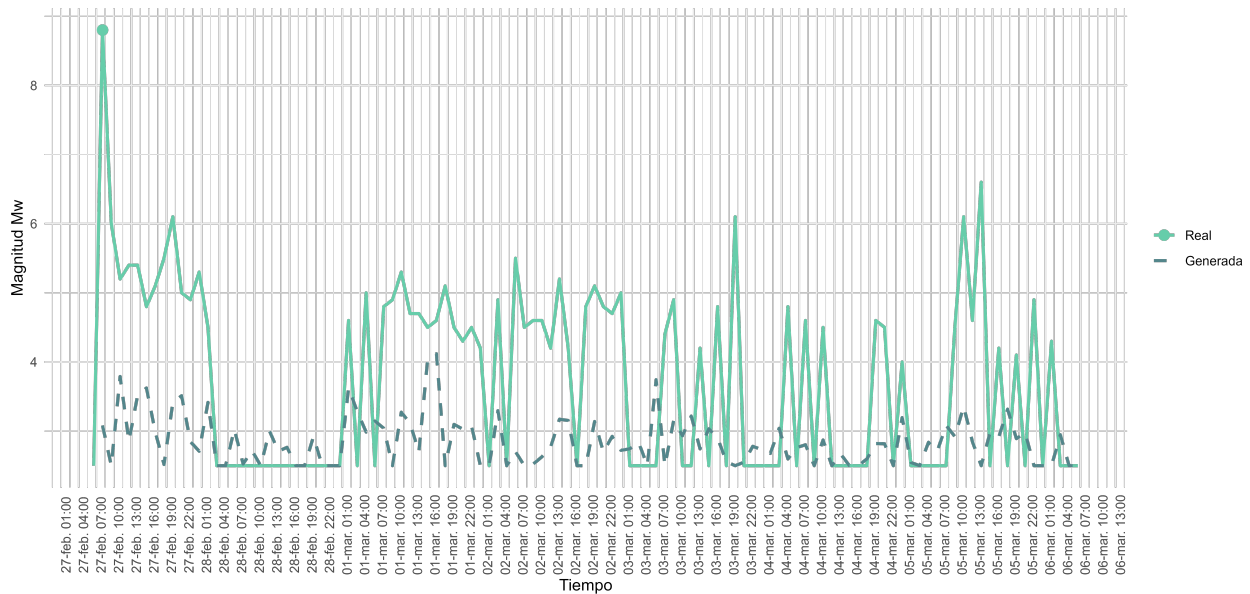


Figura 5.52: Resultados una semana modelo cGAN Concepción 2010

En la figura 5.51 se muestra en color celeste el evento principal (representado por un punto) junto con la secuencia completa de réplicas reales asociadas al terremoto de Concepción de 2010. La secuencia generada por el modelo cGAN se representa en color gris punteado. Se observa que

el modelo no logra reproducir adecuadamente la dinámica de las réplicas, ya que las magnitudes generadas son menores que las observadas en la secuencia real. Además, el modelo no consigue seguir las variaciones de magnitud cuando estas aumentan o disminuyen, lo que evidencia una limitada capacidad de adaptación a la evolución temporal de la secuencia.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.52. En esta representación más acotada, se confirma que el modelo genera réplicas con magnitudes inferiores a las reales. No obstante, se aprecia que en ciertos momentos logra aumentar ligeramente la magnitud cuando las réplicas reales presentan valores más altos, aunque sin alcanzar una correspondencia precisa ni sostenida.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.5: Métricas modelo cGAN Terremoto Concepción 2010.

RMSE	MAE	MAPE	PI
1,4251	1,1167	41,78 %	-374,98 %

La tabla 5.5 muestra que el modelo GAN presenta un desempeño limitado en la generación de la secuencia de réplicas del terremoto de Concepción. El RMSE y el MAE indican errores absolutos considerables, lo que indica una discrepancia sistemática entre las magnitudes reales y generadas. El MAPE sugiere que, en promedio, las predicciones se desvían un 40 % respecto a los valores reales, lo cual es considerable en contextos sísmicos. Finalmente, el índice de persistencia revela que el modelo tiene un rendimiento inferior al de una estrategia de predicción persistente, lo que refuerza la conclusión de que el modelo no logra capturar adecuadamente la estructura ni la variabilidad de la secuencia real.

Terremoto Iquique 2014

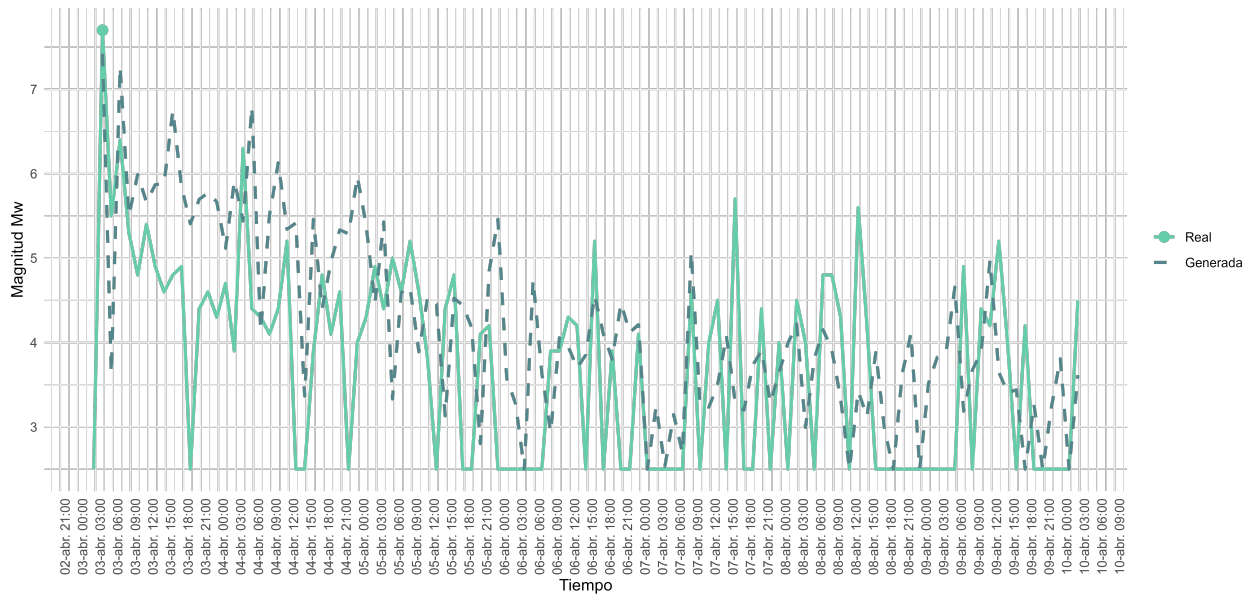


Figura 5.54: Resultados una semana modelo cGAN Iquique 2014

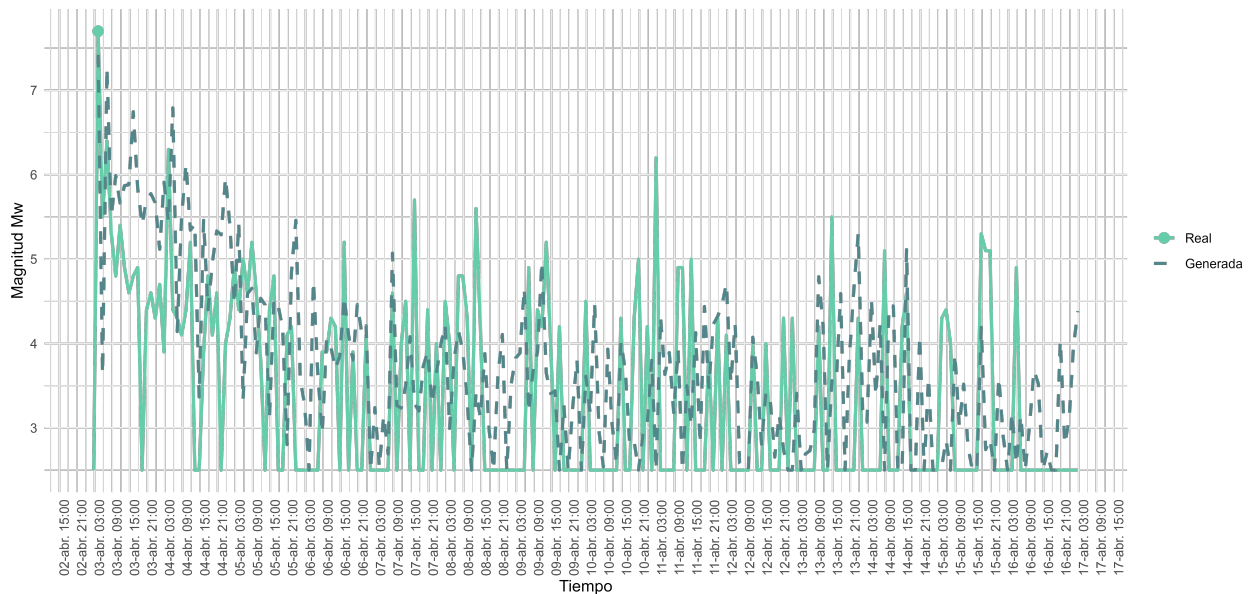


Figura 5.53: Resultados modelo cGAN Iquique 2014

En la figura 5.53 se muestra en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Iquique de 2014. La secuencia generada por el modelo cGAN se representa en color gris punteado. En términos

generales, el modelo logra ajustarse de manera razonable a las magnitudes observadas. A medida que transcurre el tiempo, las predicciones mantienen una proximidad relativa con las réplicas reales, aunque en algunos casos se observa una sobreestimación de las magnitudes. Ejemplos de coincidencia aceptable se registran el 6 de abril a las 21:00, el 10 de abril a las 15:00 y el 12 de abril a las 09:00 horas, entre otros.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.54. En esta representación más acotada, se confirma que el modelo tiende a generar réplicas con magnitudes superiores a las reales y, en pocos casos, predice eventos que no se registran en la secuencia observada. No obstante, se identifican instancias en las que las magnitudes generadas coinciden con las reales, como las réplicas del 5 de abril a las 15:00, del 6 de abril a las 21:00, del 8 de abril a las 03:00, y del 9 de abril a las 06:00 y 12:00 horas, aunque estas últimas presentan un leve desfase temporal respecto a su ocurrencia real.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.6: Métricas modelo cGAN Terremoto Iquique 2014.

RMSE	MAE	MAPE	PI
1,2948	1,0207	32,60 %	-42,26 %

La tabla 5.6 refleja un desempeño intermedio del modelo cGAN en la generación de la secuencia de réplicas del terremoto de Iquique. El RMSE y el MAE indican errores absolutos moderados, lo que sugiere que el modelo logra aproximarse a las magnitudes reales, aunque con desviaciones sistemáticas. El MAPE revela un error relativo significativo, propio de contextos de alta variabilidad como la sismicidad. El índice de persistencia indica que el modelo tiene un rendimiento inferior al de una estrategia de predicción persistente, aunque menos pronunciado que en otras secuencias analizadas. En conjunto, estos resultados sugieren que el modelo es capaz de capturar parcialmente la dinámica de las réplicas, pero aún presenta limitaciones en la calibración de magnitudes y en la sincronización temporal de los eventos generados.

5.4.2. Modelo GAN con capa Transformer (TGAN)

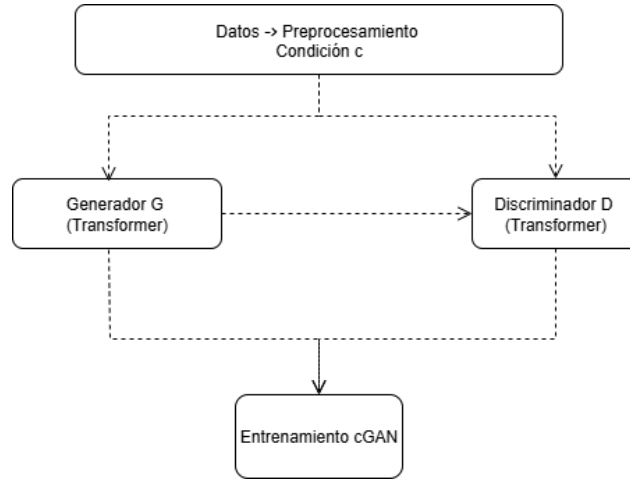


Figura 5.55: Esquema modelo TGAN

El modelo TGAN incorpora el mecanismo de self-attention de los Transformers [45] dentro del marco adversarial, con el objetivo de capturar dependencias temporales de largo alcance en las secuencias sísmicas. Como se aprecia en la figura 5.55, el generador recibe como entradas un vector de ruido global $z \in \mathbb{R}^{100}$, la condición $c \in \mathbb{R}^5$ compuesta por la información input del modelo (longitud, latitud, magnitud, profundidad y longitud en 360° para evitar problemas en las zonas cercanas a los 0°) y un ruido secuencial z_{seq} . Estas entradas se procesan mediante capas densas y un positional embedding, que permite codificar la posición temporal. A continuación, la información atraviesa L bloques transformer encoder, cada uno compuesto por un módulo de Multi-Head Self-Attention con máscara causal (cada paso solo atiende a su pasado), conexiones residuales normalizadas y capas feedforward con activación ReLU. La salida del generador corresponde a una secuencia $\hat{y} \in \mathbb{R}^{T \times 1}$ escalada en el rango $[-1, 1]$, obtenida a través de una capa TimeDistributed con función de activación tangente hiperbólica. El discriminador, por su parte, recibe secuencias reales o generadas junto con la condición c , replicada a lo largo de T . Estas entradas pasan por una capa densa, un positional embedding y bloques Transformer Encoder con Multi-Head Attention no causal, que permite atender a toda la secuencia. La salida se obtiene mediante Global Average Pooling seguido de una capa densa con activación sigmoide, que entrega la probabilidad de que la secuencia corresponda a datos reales en el intervalo $(0, 1)$. El entrenamiento sigue el esquema clásico de las GANs, con pérdidas de entropía binaria cruzada (BCE) y un término adicional de regularización $\lambda L_1 \cdot \|\hat{y} - y\|_1$ para mejorar la fidelidad de las secuencias generadas.

La optimización se realiza con el algoritmo Adam ($lr = [10^{-4}, 2 \times 10^{-4}]; \beta_1 = 0,5; \beta_2 = 0,999$), incorporando estrategias de estabilización como label smoothing (etiquetas reales = 0,9), dropout en el discriminador y normalizaciones intermedias. En conjunto, esta arquitectura aprovecha la capacidad de los Transformers para modelar patrones complejos en series temporales, mientras mantiene la estabilidad y el control propio del marco GAN.

El modelo se entrena con 10.000 épocas y tuvo una duración de ejecución aproximada de 30 minutos utilizando GPU y utilizando el lenguaje de programación Python.

Tabla 5.7: Métricas modelo TGAN.

RMSE	MAE	MAPE	PI
0,6829	0,2451	6,81 %	-20,63 %

La tabla 5.7 evidencia un desempeño significativamente superior del modelo TGAN en comparación con variantes generativas previas. El RMSE y el MAE reflejan errores absolutos bajos, lo que indica una alta precisión en la estimación de magnitudes dentro de todas las secuencias sísmicas generadas. El MAPE confirma que, en promedio, las predicciones se desvían menos de un 7% respecto a los valores reales, lo cual es notablemente bajo en contextos de alta variabilidad como la sismicidad. Por otro lado, el índice de persistencia sugiere que, aunque el modelo aún no supera la estrategia de predicción persistente, su rendimiento se acerca a ese umbral, lo que representa una mejora considerable respecto a modelos anteriores con PI mucho más negativos. En conjunto, estos resultados indican que el modelo TGAN logra capturar con mayor fidelidad tanto la magnitud como la estructura general de las secuencias sísmicas, posicionándose como una alternativa robusta para la generación de réplicas sintéticas en entornos tectónicos complejos.

A continuación se detallan los resultados obtenidos por el modelo TGAN.

Terremoto Tarapacá 2005

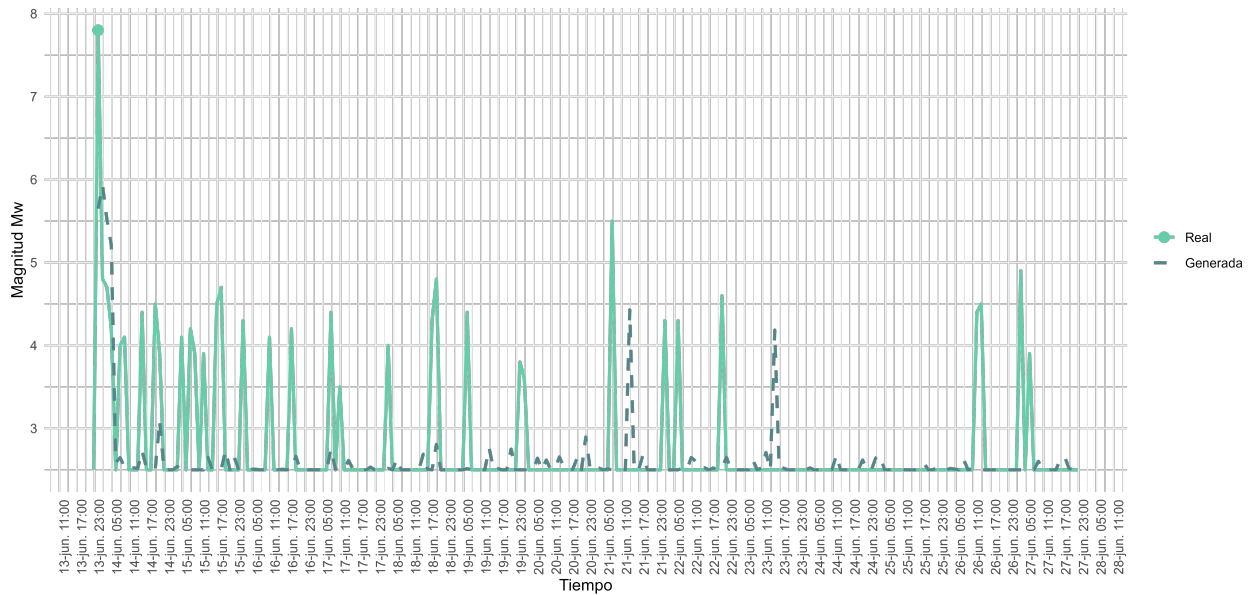


Figura 5.56: Resultados modelo TGAN Tarapacá 2005

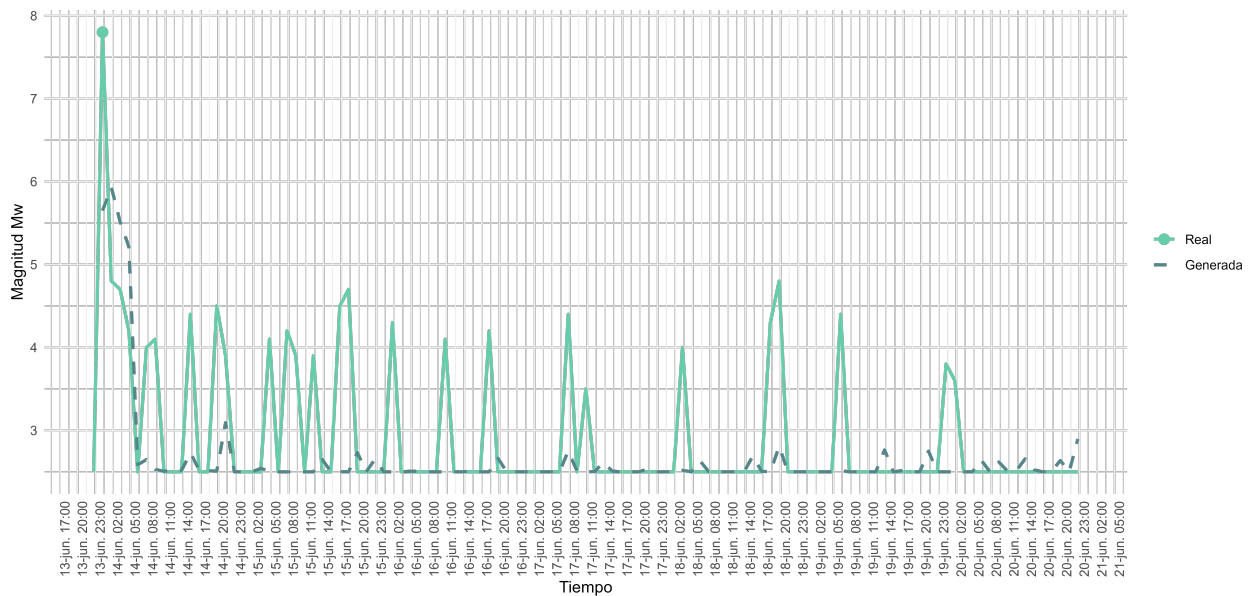


Figura 5.57: Resultados una semana modelo TGAN Tarapacá 2005

En la figura 5.56 se presenta en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Tarapacá de 2005. La secuencia generada por el modelo TGAN se muestra en color gris punteado. En

términos generales, el modelo no logra ajustarse adecuadamente a las magnitudes observadas, las réplicas generadas presentan valores significativamente inferiores a los reales lo que evidencia una subestimación sistemática. Si bien el modelo logra reproducir dos réplicas con magnitudes similares a las reales, estas se generan con un desfase temporal superior a seis horas, lo que limita su utilidad en aplicaciones sensibles al tiempo de ocurrencia.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.57. En esta representación más acotada, se confirma que el modelo tiende a generar réplicas con magnitudes considerablemente menores a las reales, sin lograr capturar la variabilidad ni la intensidad característica de la secuencia observada.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.8: Métricas modelo TGAN Terremoto Tarapacá 2005.

RMSE	MAE	MAPE	PI
1,4745	1,0034	28,24 %	-62,18 %

La tabla 5.8 refleja un desempeño limitado del modelo TGAN en la generación de la secuencia de réplicas del terremoto de Tarapacá. El RMSE y el MAE indican errores absolutos elevados, lo que confirma una discrepancia significativa entre las magnitudes reales y generadas. El MAPE sugiere que, en promedio, las predicciones se desvían más de un 28 % respecto a los valores observados, lo cual es considerable en contextos sísmicos. El índice de persistencia revela que el modelo tiene un rendimiento claramente inferior al de una estrategia de predicción persistente, lo que refuerza la conclusión de que el modelo no logra representar adecuadamente la dinámica ni la magnitud de las réplicas en esta secuencia específica.

Terremoto Concepción 2010

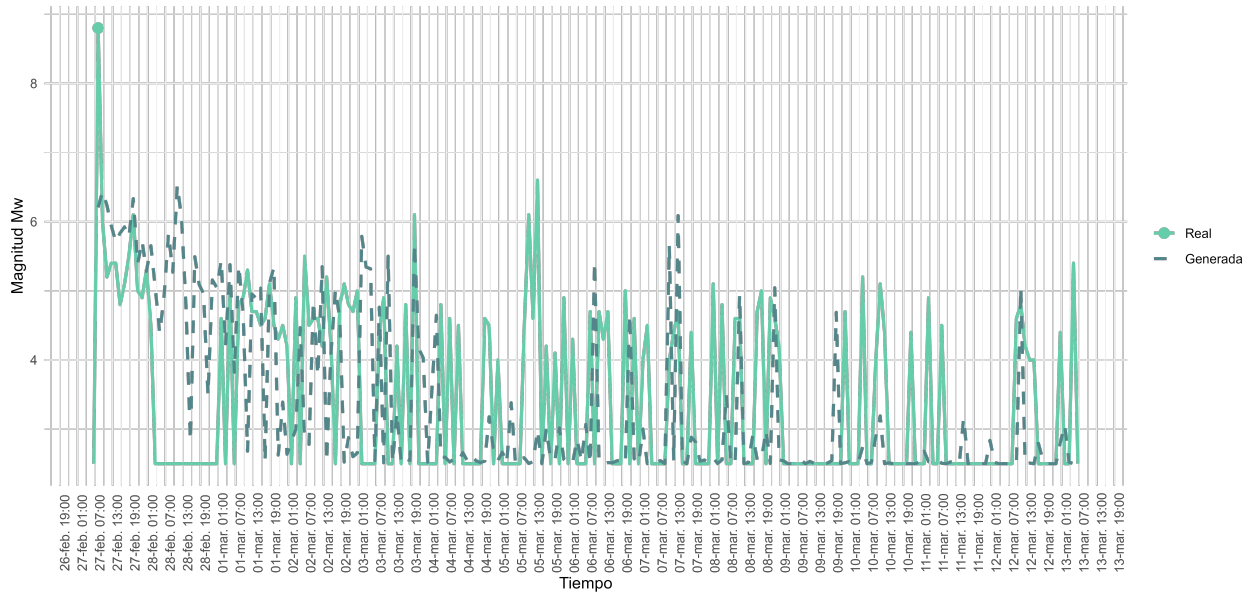


Figura 5.58: Resultados modelo TGAN Concepción 2010

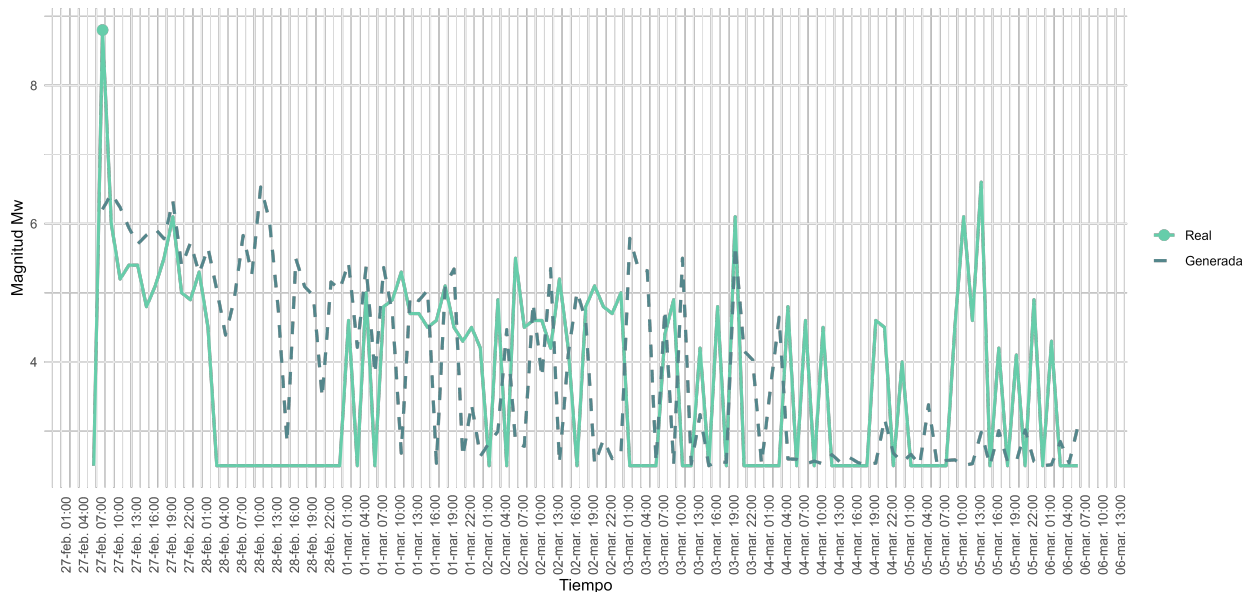


Figura 5.59: Resultados una semana modelo TGAN Concepción 2010

En la figura 5.58 se presenta en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Concepción de 2010. La secuencia generada por el modelo TGAN se muestra en color gris punteado. En términos

generales, el modelo logra ajustarse adecuadamente a las magnitudes observadas, generando réplicas con valores bastante similares a los reales. No obstante, se identifican ciertos casos en los que el modelo produce réplicas sin que se haya registrado actividad sísmica en la secuencia real pero con magnitudes de aproximadamente 3 Mw, así como instancias en las que las magnitudes generadas son inferiores a las observadas, lo que indica una subestimación en el final de la secuencia.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.59. En esta representación más acotada, se confirma que el modelo reproduce con mayor precisión las magnitudes de las réplicas reales. Sin embargo, persiste la generación de eventos y la dificultad para alcanzar las magnitudes reales en algunos casos, lo que sugiere una sensibilidad limitada ante variaciones abruptas en la secuencia.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.9: Métricas modelo TGAN Terremoto Concepción 2010.

RMSE	MAE	MAPE	PI
0,7212	0,3081	8,09 %	-21,65 %

La tabla 5.9 refleja un desempeño sólido del modelo TGAN en la generación de la secuencia de réplicas del terremoto de Concepción. El RMSE y el MAE indican errores absolutos bajos, lo que evidencia una buena aproximación a las magnitudes reales. El MAPE confirma que, en promedio, las predicciones se desvían menos de un 10 % respecto a los valores observados, lo cual es aceptable en contextos de alta variabilidad sísmica. El índice de persistencia sugiere que el modelo aún no supera la estrategia de predicción persistente, pero se acerca a ese umbral con un rendimiento notablemente mejor que el observado en otras secuencias. En conjunto, estos resultados posicionan al modelo TGAN como una herramienta prometedora para la generación de réplicas sintéticas en eventos de alta magnitud, aunque con oportunidades de mejora en la detección de eventos espurios y en la calibración de extremos.

Terremoto Iquique 2014

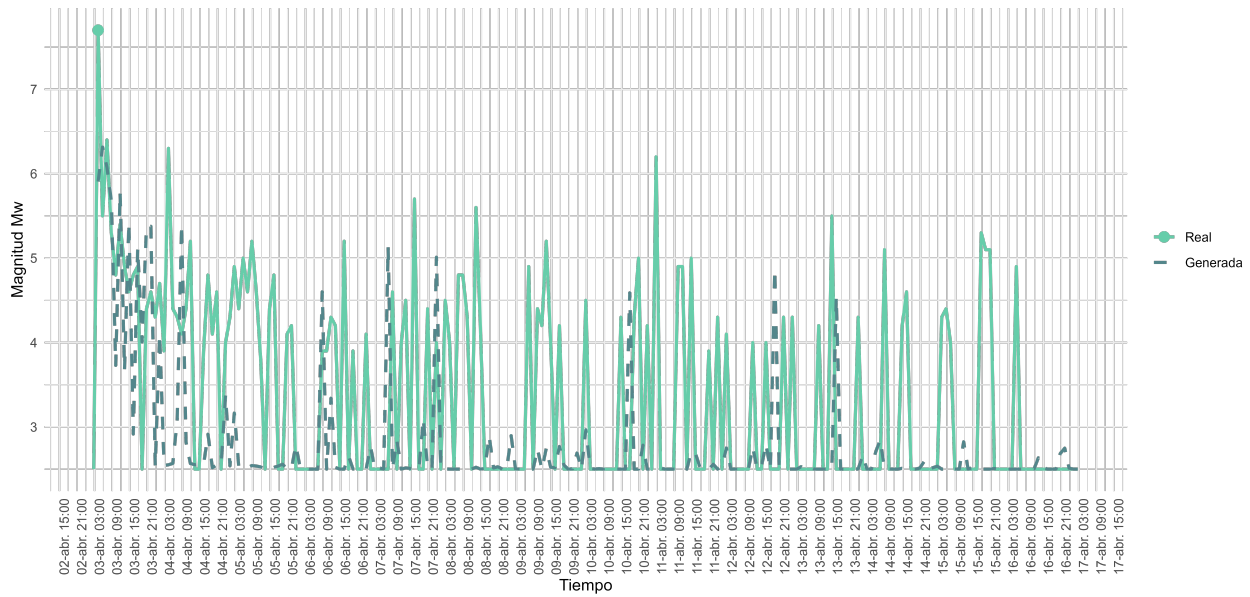


Figura 5.60: Resultados modelo TGAN Iquique 2014

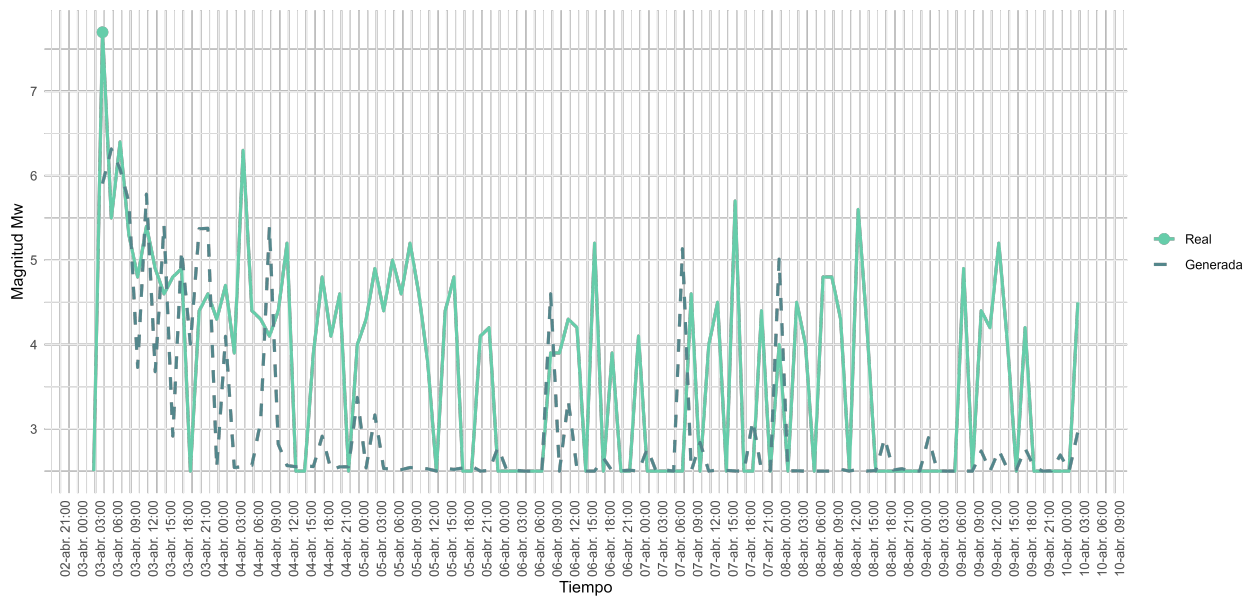


Figura 5.61: Resultados una semana modelo TGAN Iquique 2014

En la figura 5.60 se presenta en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Iquique de 2014. La secuencia generada por el modelo TGAN se muestra en color gris punteado. En términos generales,

el modelo logra ajustarse adecuadamente a las magnitudes observadas, generando algunas réplicas con valores similares a los reales. No obstante, se identifican algunos casos en los que el modelo produce réplicas sin que se haya registrado actividad sísmica en la secuencia real pero con magnitudes inferiores a 3 MW, así como algunas secuencias con pequeños desfases.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.59. En esta representación más acotada, se confirma que el modelo reproduce con precisión algunas de las magnitudes de las réplicas reales. Sin embargo, persiste la generación de eventos espurios y la dificultad para alcanzar las magnitudes reales en algunos casos.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.10: Métricas modelo TGAN Terremoto Iquique 2014.

RMSE	MAE	MAPE	PI
1,2526	0,7805	18,07 %	-33,14 %

La tabla 5.10 muestra un desempeño intermedio del modelo TGAN en la generación de la secuencia de réplicas asociadas al terremoto de Iquique. El RMSE y el MAE indican errores absolutos moderados, lo que sugiere que el modelo logra aproximarse parcialmente a las magnitudes reales, aunque con desviaciones sistemáticas. El MAPE refleja un error relativo aceptable en contextos sísmicos, especialmente considerando la variabilidad natural de las réplicas. Sin embargo, el índice de persistencia indica que el modelo aún presenta un rendimiento inferior al de una estrategia de predicción persistente. Esto sugiere que, aunque el TGAN logra capturar ciertos patrones de magnitud, su capacidad para representar la dinámica temporal y la ocurrencia precisa de las réplicas sigue siendo limitada. En conjunto, estos resultados posicionan al modelo como una herramienta con potencial, pero que requiere ajustes adicionales para mejorar su sensibilidad ante eventos complejos y su fidelidad en la generación de secuencias sísmicas realistas.

5.4.3. Modelo GAN con capa Wasserstein (WGAN)

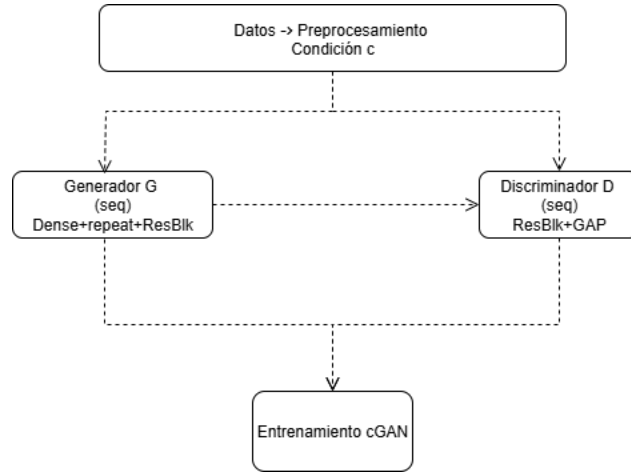


Figura 5.62: Esquema modelo WGAN

El modelo WGAN extiende el marco adversarial clásico al incorporar la distancia de Wasserstein como métrica de divergencia, junto con la penalización de gradiente para mejorar la suavidad de la función del crítico y mejorar la estabilidad del entrenamiento. Como se observa en la figura 5.62, el generador integra tres fuentes de variabilidad: un vector de ruido global $z \in \mathbb{R}^{100}$, la condición $c \in \mathbb{R}^5$ compuesta por la información input del modelo (longitud, latitud, magnitud, profundidad y longitud en 360° para evitar problemas en las zonas cercanas a los 0°) y un ruido secuencial $z_{seq} \in \mathbb{R}^{T \times Z_t}$. Estas entradas se procesan mediante capas densas, un RepeatVector que replica la condición a lo largo del horizonte temporal y una pila de bloques residuales (ResBlocks), produciendo una secuencia sintética $\hat{y} \in \mathbb{R}^{T \times 1}$ escalada en el rango $[-1, 1]$. El crítico (en lugar de un discriminador tradicional) D recibe secuencias reales o generadas junto con la condición replicada, las procesa mediante ResBlocks y aplica un Global Average Pooling que genera un escalar \mathcal{R} interpretado como el score Wasserstein. A diferencia de las GAN convencionales, no se emplea función de activación sigmoide, lo que permite gradientes más informativos para la retropropagación. El entrenamiento minimiza la diferencia de expectativas entre datos reales y generados, penalizando la norma del gradiente respecto de interpolaciones entre y y \hat{y} . Para estabilizar aún más el aprendizaje se utilizan varias actualizaciones del crítico por cada paso del generador ($n_{critico} > 1$), ruido de instancia (instance noise) que decae a lo largo de las épocas y un término de regularización L_1 que ayuda con la coherencia punto a punto entre las secuencias reales y generadas. La optimización se realiza con el algoritmo Adam ($lr = 1 \times 10^{-4}$; $\beta_1 = 0$; $\beta_2 = 0,99$),

incorporando técnicas de gradient clipping y métricas como la distancia de Wasserstein, el término de penalización y el error L_1 . Esta arquitectura resulta particularmente adecuada para secuencias largas o multimodales, donde el uso de pérdidas basadas en entropía cruzada binaria tiende a generar inestabilidad o colapso de modos.

El modelo se entrenó con 10.000 épocas y tuvo una duración de ejecución aproximada de 8 horas utilizando GPU y utilizando el lenguaje de programación Python.

Tabla 5.11: Métricas modelo WGAN.

RMSE	MAE	MAPE	PI
0,7430	0,4746	15,97 %	-42,82 %

La tabla 5.11 presenta el desempeño global del modelo WGAN en la generación de secuencias sísmicas. El RMSE y el MAE indican errores absolutos moderados, lo que sugiere que el modelo logra aproximarse razonablemente a las magnitudes reales en términos generales. El MAPE refleja un error relativo aceptable, especialmente considerando la variabilidad inherente a los procesos sísmicos. Sin embargo, el índice de persistencia revela que el modelo aún se desempeña por debajo de una estrategia de predicción persistente. Esto indica que, aunque el WGAN logra capturar ciertos patrones de magnitud, su capacidad para representar la dinámica temporal y la ocurrencia precisa de las réplicas sigue siendo limitada.

Terremoto Tarapacá 2005

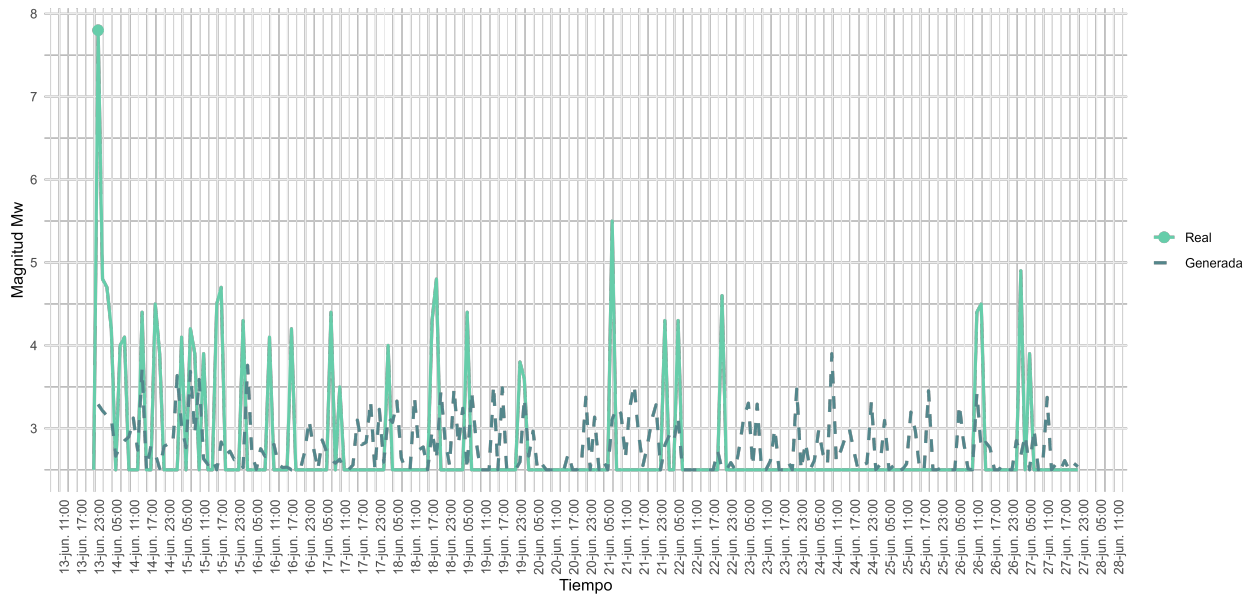


Figura 5.63: Resultados modelo WGAN Tarapacá 2005

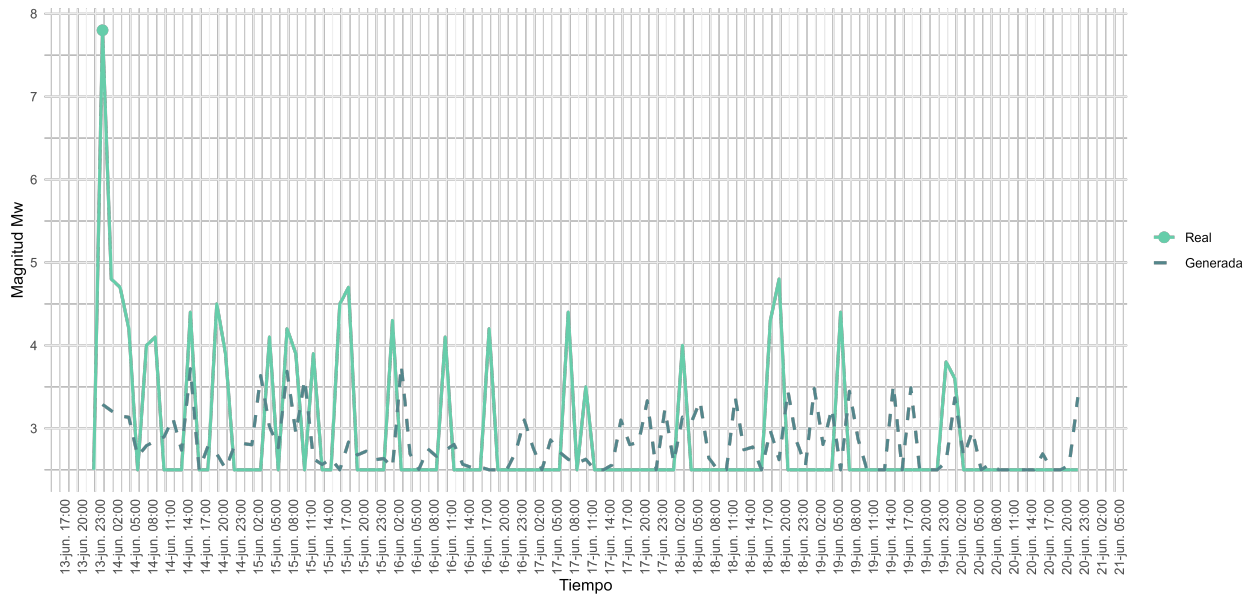


Figura 5.64: Resultados una semana modelo WGAN Tarapacá 2005

En la figura 5.63 se muestra en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Tarapacá de 2005. La secuencia generada por el modelo WGAN se representa en color gris punteado.

En términos generales, el modelo logra generar una secuencia estructuralmente coherente, pero con magnitudes considerablemente inferiores a las observadas en la secuencia real. Además, se identifican instancias en las que el modelo genera réplicas que no tienen correspondencia con eventos sísmicos reales, lo que evidencia la presencia de eventos espurios.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.64. En esta representación más acotada, se observa que el modelo reproduce de forma razonablemente cercana algunas de las magnitudes reales, aunque estas tienden a ser mayores que las generadas. Persiste, sin embargo, la dificultad del modelo para alcanzar los valores máximos de magnitud y para evitar la generación de réplicas no observadas en la secuencia original.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.12: Métricas modelo WGAN Terremoto Tarapacá 2005.

RMSE	MAE	MAPE	PI
0,7063	0,4361	13,80 %	-16,66 %

La tabla 5.12 evidencia un desempeño moderado del modelo WGAN en la generación de la secuencia de réplicas del terremoto de Tarapacá. El RMSE y el MAE indican errores absolutos relativamente bajos, lo que sugiere una aproximación aceptable a las magnitudes reales. El MAPE refleja un error relativo contenido, especialmente en comparación con otros modelos generativos evaluados. El índice de persistencia indica que el modelo aún no supera el rendimiento de una estrategia de predicción persistente, aunque se aproxima a ese umbral con mayor eficacia que otros enfoques.

Terremoto Concepción 2010

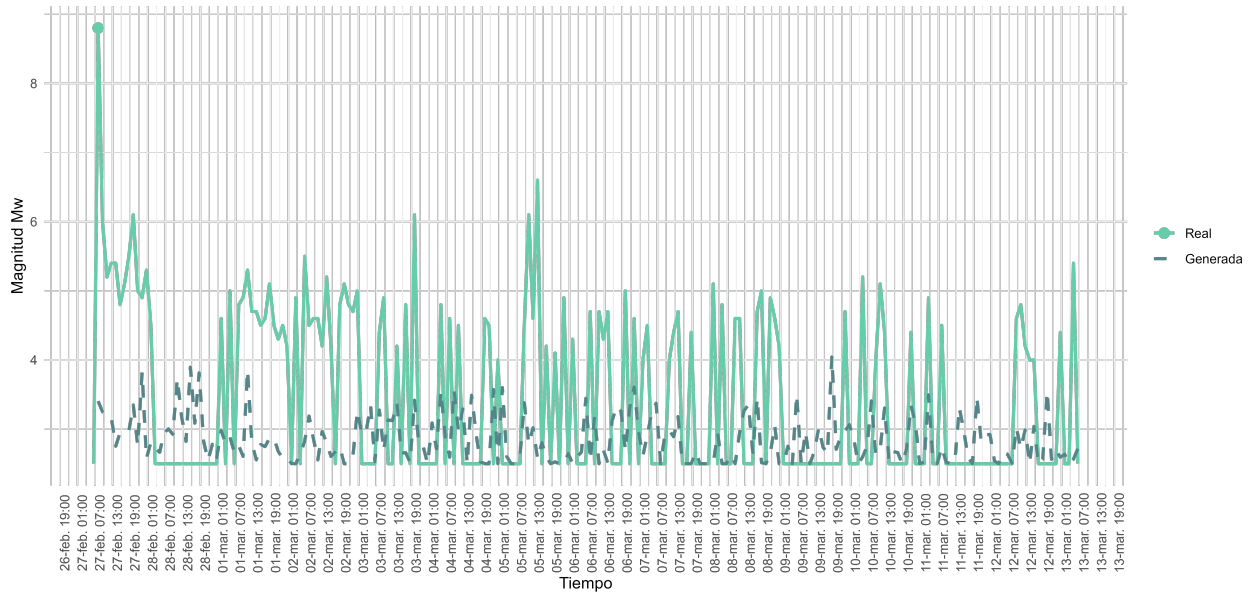


Figura 5.65: Resultados modelo WGAN Concepción 2010

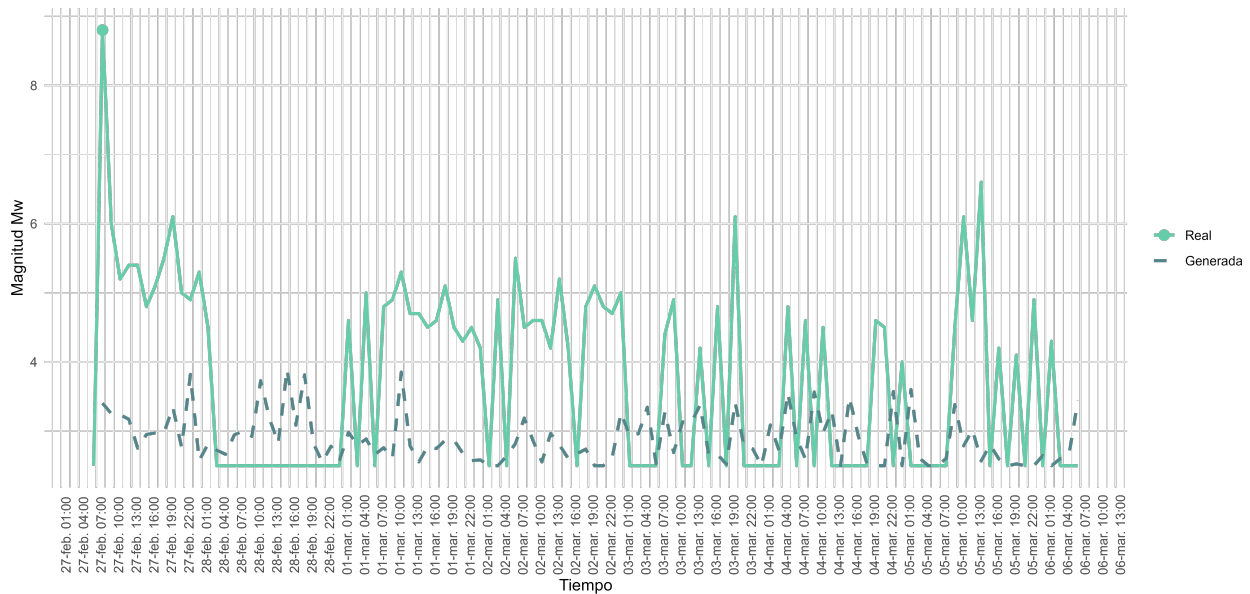


Figura 5.66: Resultados una semana modelo WGAN Concepción 2010

En la figura 5.65 se presenta en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Concepción de 2010. La secuencia generada por el modelo WGAN se muestra en color gris punteado. En términos

generales, el modelo no logra reproducir adecuadamente las magnitudes observadas, las réplicas generadas presentan valores significativamente inferiores a los reales. Además, se identifican periodos en los que el modelo genera eventos sísmicos sin que exista actividad registrada en la secuencia real, lo que evidencia la presencia de eventos espurios.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.66. En esta representación más acotada, se confirma que el modelo tiende a reproducir secuencias con magnitudes consistentemente menores a las reales, sin lograr capturar la intensidad ni la variabilidad característica de la secuencia observada.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.13: Métricas modelo WGAN Terremoto Concepción 2010.

RMSE	MAE	MAPE	PI
1,3471	1,0057	24,78 %	-35,37 %

La tabla 5.13 evidencia un desempeño limitado del modelo WGAN en la generación de la secuencia de réplicas del terremoto de Concepción. El RMSE y el MAE indican errores absolutos elevados, lo que refleja una discrepancia significativa entre las magnitudes reales y generadas. El MAPE sugiere que, en promedio, las predicciones se desvían cerca de un 25 % respecto a los valores observados, lo cual es considerable en contextos sísmicos. El índice de persistencia confirma que el modelo se desempeña por debajo de una estrategia de predicción persistente, lo que refuerza la conclusión de que el WGAN no logra representar adecuadamente la dinámica ni la magnitud de las réplicas en esta secuencia específica.

Terremoto Iquique 2014

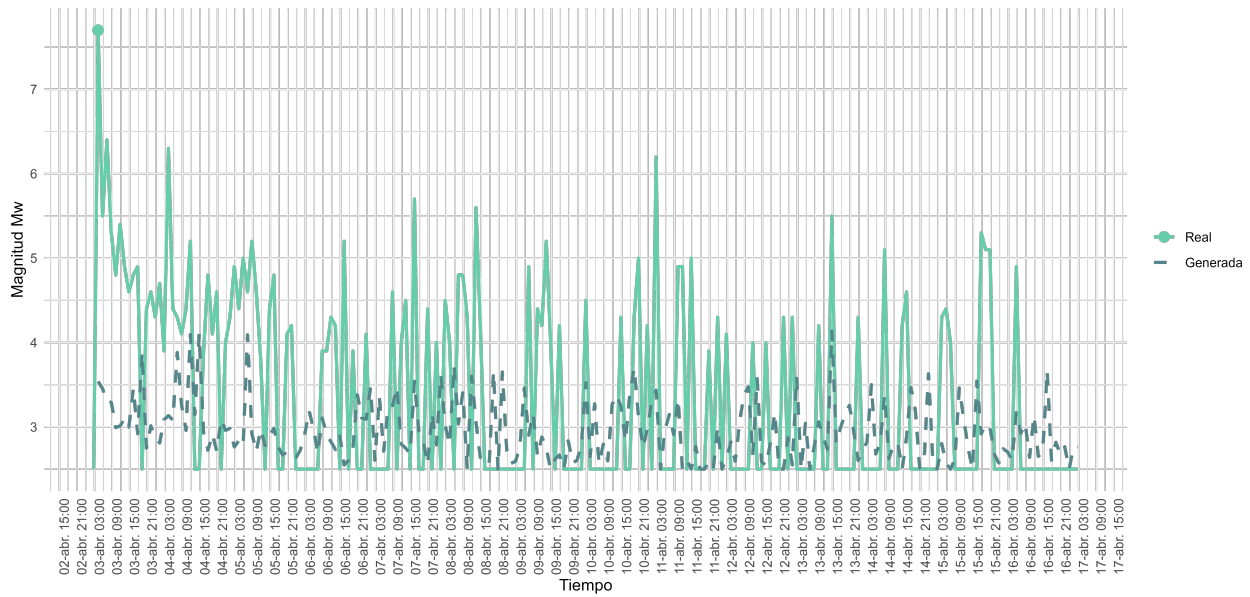


Figura 5.67: Resultados modelo WGAN Iquique 2014

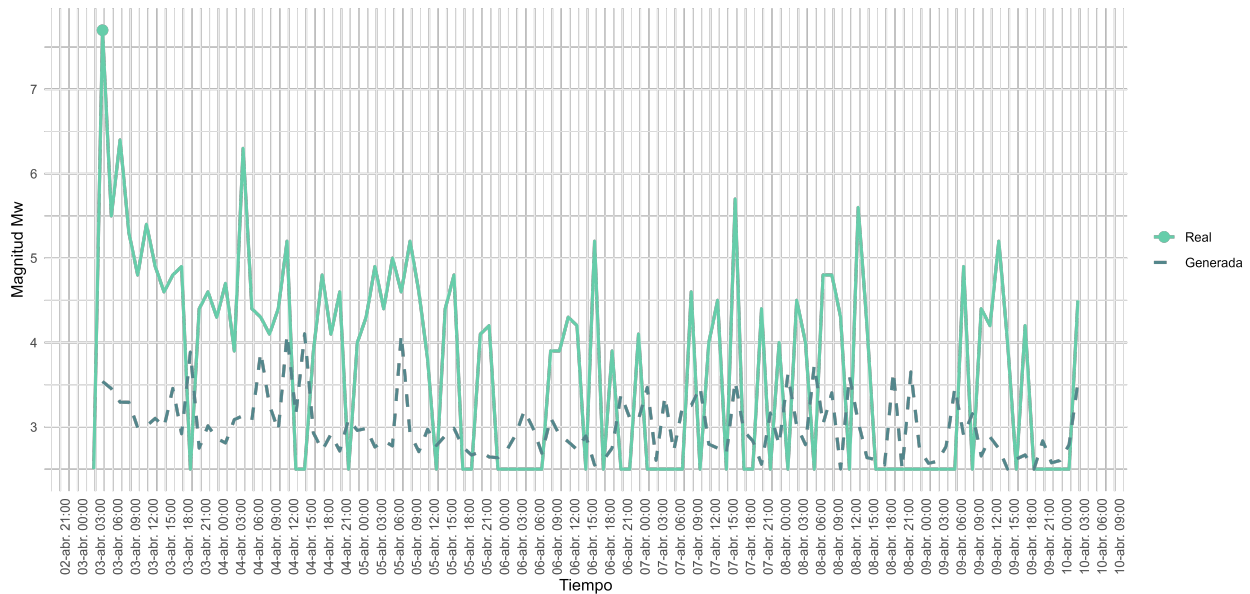


Figura 5.68: Resultados una semana modelo WGAN Iquique 2014

En la figura 5.67 se presenta en color celeste el evento principal (representado por un punto), junto con la secuencia completa de réplicas reales asociadas al terremoto de Iquique de 2014. La secuencia generada por el modelo WGAN se muestra en color gris punteado. En términos

generales, el modelo no logra reproducir adecuadamente las magnitudes observadas, las réplicas generadas presentan valores significativamente inferiores a los reales. Sin embargo, se observa que, a pesar de que son de menor magnitud, estas son exactas en el momento que se generan.

Para facilitar el análisis, se recorta la visualización y se considera únicamente la primera semana posterior al evento principal, como se muestra en la figura 5.68. En esta representación más acotada, se confirma que el modelo tiende a reproducir secuencias con magnitudes consistentemente menores a las reales, pero se ve la consistencia en los momentos que se generan.

A continuación, se presentan las métricas calculadas para esta secuencia específica.

Tabla 5.14: Métricas modelo WGAN Terremoto Iquique 2014.

RMSE	MAE	MAPE	PI
1,1472	0,8764	23,04 %	-11,69 %

La tabla 5.14 muestra un desempeño relativamente sólido del modelo WGAN en la generación de la secuencia de réplicas del terremoto de Iquique. El RMSE y el MAE reflejan errores absolutos moderados, lo que indica una aproximación razonable a las magnitudes reales, aunque con algunas desviaciones sistemáticas. El MAPE sugiere que, en promedio, las predicciones se desvían cerca de un 23 % respecto a los valores observados, lo cual es aceptable en contextos de alta variabilidad sísmica. El índice de persistencia revela que el modelo se acerca al rendimiento de una estrategia de predicción persistente, con una diferencia negativa menos pronunciada que en otras secuencias evaluadas. Este resultado sugiere que el WGAN logra capturar parcialmente la dinámica de las réplicas, con una mejor calibración de magnitudes y menor generación de eventos espurios en comparación con su desempeño en otros terremotos.

En conjunto, los resultados evidencian que las arquitecturas generativas implementadas presentan desempeños diferenciados al modelar secuencias de réplicas sísmicas. El cGAN estándar logró reproducir ciertos patrones generales, pero con limitaciones en la calibración de magnitudes y en la sincronización temporal, lo que derivó en errores elevados y en un rendimiento inferior al de una estrategia persistente. El GAN con capa Transformer (TGAN) mostró la mayor capacidad de aproximación a las secuencias reales, con métricas notablemente mejores y una mayor fidelidad en la representación de las réplicas, aunque aún con presencia de eventos espurios y dificultades para capturar extremos. Por su parte, el GAN con función de costo Wasserstein (WGAN) ofreció un desempeño intermedio: si bien mejoró la estabilidad y la coherencia estructural de las secuencias, tendió a subestimar magnitudes y a generar réplicas no observadas. En términos de eficiencia computacional, el modelo cGAN fue el más rápido en ejecutarse (aprox. 15 minutos con GPU),

seguido por el TGAN (20 minutos con GPU), mientras que el WGAN presentó un costo temporal significativamente mayor, alcanzando casi 8 horas de ejecución. En síntesis, los resultados sugieren que la incorporación de mecanismos de atención como los Transformers potencian la capacidad de los modelos generativos para capturar dependencias complejas y reproducir secuencias sísmicas más realistas, posicionándose como la alternativa más prometedora dentro de las variantes evaluadas tanto por los resultados obtenidos como por su eficiencia computacional.

Capítulo 6

Conclusiones

El presente trabajo abordó el modelamiento de secuencias de réplicas sísmicas en Chile mediante arquitecturas generativas adversariales (GAN), con el propósito de explorar nuevas metodologías que permitan comprender y reproducir los patrones espacio-temporales asociados a los terremotos.

En primer lugar, el análisis exploratorio permitió caracterizar la dinámica de los sismos en el territorio nacional, evidenciando la marcada dependencia temporal y espacial de las réplicas respecto a los eventos principales. Sin embargo, la limitada cantidad de secuencias en los registros nacionales llevó a ampliar la base de datos al Anillo de Fuego del Pacífico, lo que posibilitó un entrenamiento más robusto de los modelos.

Los experimentos realizados con las tres variantes de GAN entregaron hallazgos diferenciados. El cGAN estándar logró reproducir patrones generales de recurrencia, aunque con limitaciones en la calibración de magnitudes y sincronización temporal. El GAN con capa Transformer (TGAN) se consolidó como la variante con mejor capacidad de aproximación, mostrando métricas superiores y una mayor fidelidad en la representación de las secuencias, especialmente en la captura de dependencias complejas. Por su parte, el GAN con capa Wasserstein (WGAN) ofreció un desempeño intermedio, destacando por su mayor estabilidad y coherencia estructural, aunque con tendencia a subestimar magnitudes y generar eventos espurios.

En términos de eficiencia computacional, el modelo estándar fue el más rápido en ejecutarse (aproximadamente 15 minutos con GPU), seguido del TGAN (aproximadamente 20 minutos con GPU), mientras que el WGAN presentó un costo temporal significativamente mayor, alcanzando casi 8 horas de ejecución.

En síntesis, los resultados sugieren que la incorporación de mecanismos de atención como

los Transformers potencia la capacidad de los modelos generativos para capturar relaciones más difíciles de encontrar y reproducir secuencias sísmicas más realistas, posicionándose como la alternativa más prometedora entre las variantes evaluadas. No obstante, aún quedan desafíos en la calibración de magnitudes, la reducción de eventos espurios y la optimización de tiempos de cómputo.

Los resultados de esta investigación, además de aportar al ámbito académico, poseen aplicaciones prácticas relevantes en la gestión del riesgo sísmico. En particular, la generación de secuencias de réplicas a partir de modelos basados en GAN puede servir como insumo para el diseño de protocolos de respuesta temprana, planificación de evacuaciones y la definición de escenarios de simulación utilizados por organismos de protección civil como SENAPRED (Servicio Nacional de Prevención y Respuesta ante Desastres). Asimismo, este tipo de modelamiento podría integrarse como herramienta complementaria en sistemas de alerta sísmica, contribuyendo a anticipar la evolución de las réplicas posteriores a un evento principal y apoyando la toma de decisiones en políticas públicas orientadas a la seguridad de la población y la resiliencia urbana.

Finalmente, este trabajo demuestra el potencial de los enfoques basados en inteligencia artificial para complementar la sismología tradicional, abriendo nuevas líneas de investigación en la predicción de réplicas y en el diseño de herramientas de apoyo para la gestión del riesgo sísmico en Chile.

6.1. Trabajos futuros

A partir de los resultados obtenidos, se identifican diversas líneas de investigación que permitirán mejorar la precisión y aplicabilidad de los modelos propuestos.

1. **Integración del modelo ETAS:** Incorporar el modelo Epidemic-Type Aftershock Sequence (ETAS) como insumo adicional en el entrenamiento de los modelos generativos, con el fin de enriquecer la representación de las dinámicas de ocurrencia y dependencia temporal de las réplicas.
2. **Implementación de ventanas temporales adaptativas:** Desarrollar un esquema de discretización temporal dinámico, que utilice intervalos reducidos (por ejemplo, de 5 minutos) en los días posteriores al evento principal y que luego evolucione hacia intervalos de horas o días, según las características del sismo. Esto permitiría estructurar la base de datos de manera más flexible y acorde con la naturaleza cambiante de la actividad sísmica.

3. **Aplicación del algoritmo ST-DBSCAN-EV:** Utilizar este algoritmo de clustering espacio-temporal para ajustar los radios de agrupamiento en función de las características del sismo, con el objetivo de construir secuencias más precisas y representativas en la base de datos.
4. **Análisis de la influencia espacial:** Evaluar el impacto que tienen las distintas zonas geográficas de Chile (latitud y longitud) en el desempeño de los modelos generativos, explorando posibles diferencias regionales en la dinámica de réplicas y en la calidad de las predicciones obtenidas.
5. **Optimización de arquitecturas y parámetros:** Explorar ajustes más finos en los hiperparámetros utilizados (dimensión del ruido, tasas de aprendizaje, funciones de activación, entre otros) y mejorar la configuración de las capas Transformer y Wasserstein. Esto permitiría potenciar la capacidad de los modelos para capturar dependencias temporales y calibrar de manera más precisa la magnitud y ocurrencia de las réplicas.
6. **Extensión a otras regiones del Anillo de Fuego del Pacífico:** Replicar este análisis en distintos segmentos del Anillo de Fuego permitiría validar la generalización de los modelos propuestos y contrastar el comportamiento de las secuencias generadas en contextos tectónicos diversos.

Este trabajo se realizó en el marco del Proyecto FONDECYT de Iniciación N.º 11251120 (ANID, Chile). Sus herramientas y/o resultados forman parte de la investigación del proyecto y podrán ser aplicados o extendidos en futuras líneas de estudio vinculadas al mismo.

ANEXO A

Anexo

A.1. Dependencia temporal en terremotos de Chile

En esta sección se examina con mayor detalle la dinámica sísmica en Chile, considerando la frecuencia de eventos en distintas zonas del país.

Inicialmente, se evaluó la posibilidad de considerar únicamente los sismos perceptibles por la población chilena. Para ello, se propuso utilizar como umbral mínimo aquellos eventos con una magnitud mayor o igual a 5 Mw. Sin embargo, al analizar las diferencias temporales entre eventos, se observó que algunos terremotos registran muy pocos sismos que cumplen con este criterio, lo que limita la representatividad del análisis.

Por esta razón, se decidió estudiar la cantidad de sismos registrados bajo distintos umbrales de magnitud, con el objetivo de identificar un valor que permita un análisis más robusto y consistente entre los distintos eventos.

Tabla A.1: Cantidad de sismos por magnitud y terremoto.

Terremoto	$Mw > 4$	$Mw \geq 4$	$Mw > 4,5$	$Mw \geq 4,5$	$Mw > 5$	≥ 5
Terremoto 2005	39	45	15	18	2	3
Terremoto 2007	57	64	34	38	18	21
Terremoto 2010	299	336	183	207	75	94
Terremoto 2014	228	265	107	132	39	52
Terremoto 2015	329	394	139	162	50	67
Terremoto 2016	21	25	8	8	4	5
Terremoto 2024	15	19	4	6	1	1

A partir de los datos presentados en la tabla A.1, se concluye que utilizar un umbral de magnitud mayor o igual a 4 Mw permite incluir una mayor cantidad de eventos en el análisis. Aunque estos sismos no siempre son perceptibles por la población, su acumulación puede representar un riesgo significativo, por lo que se justifica su inclusión en el estudio.

Por otro lado, se calcularon las diferencias temporales entre sismos consecutivos, obtenidas al restar los minutos de ocurrencia de cada evento respecto al inmediatamente anterior. Este procedimiento permitió visualizar con mayor claridad los intervalos entre eventos, lo cual resulta fundamental para el análisis de la distribución temporal de los sismos.

Para la elaboración de los gráficos, se consideraron únicamente aquellas diferencias de tiempo que no superaban los 90 minutos, con el objetivo de enfocar el análisis en los eventos más cercanos entre sí.

Terremoto: Tarapacá 2005

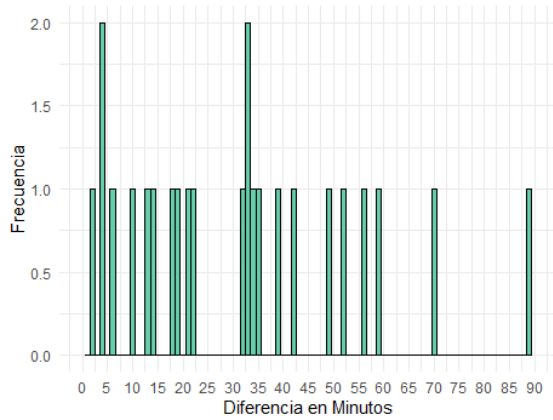


Figura A.1: Diferencia en minutos para magnitudes $M_w \geq 4$.

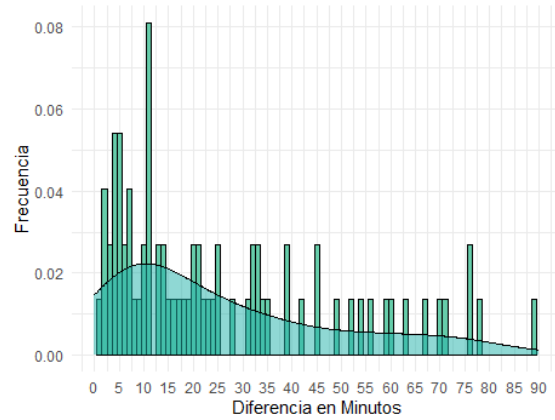


Figura A.2: Densidad de la diferencia en minutos.

En la figura A.1 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen dos eventos con diferencias temporales de aproximadamente 5 y 35 minutos respecto al sismo anterior. Por su parte, la figura A.2 muestra que la mayor densidad de ocurrencia se concentra en torno a los 10 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

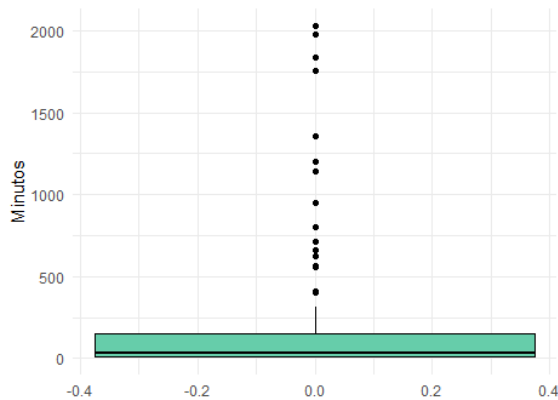


Figura A.3: Boxplot diferencia de minutos.

Tabla A.2: Summary terremoto Tarapacá.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,55	11,25	34,39	203,33	151,09	2033,02

Los resultados de la figura A.3 y de la tabla A.2 muestran una alta dispersión en los intervalos de tiempo entre sismos. Aunque la mediana es de aproximadamente 34,39 minutos, lo que indica que la mitad de los eventos ocurren con menos de esa diferencia temporal, el valor máximo alcanza los 2033,02 minutos, lo que evidencia la presencia de intervalos significativamente largos.

Terremoto: Tocopilla 2007

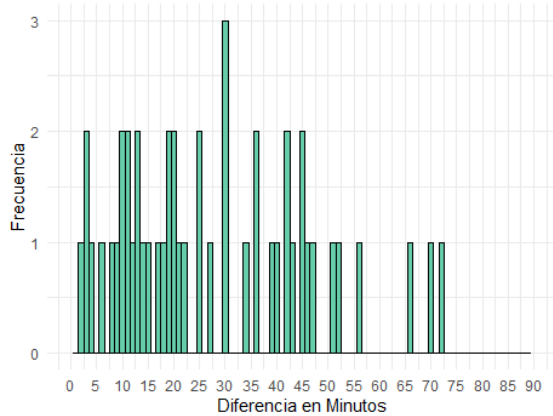


Figura A.4: Diferencia en minutos para magnitudes $M_w \geq 4$.

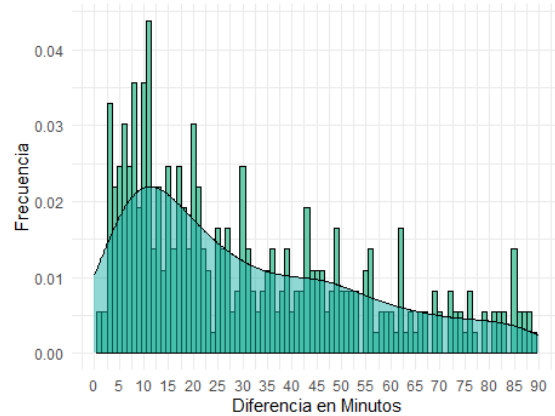


Figura A.5: Densidad de la diferencia en minutos.

En la figura A.4 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existe un eventos con diferencias temporales de aproximadamente 30 minutos respecto al sismo anterior. Por su parte, la figura A.5 muestra que la mayor densidad de ocurrencia se concentran en torno a los 10 y 15 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

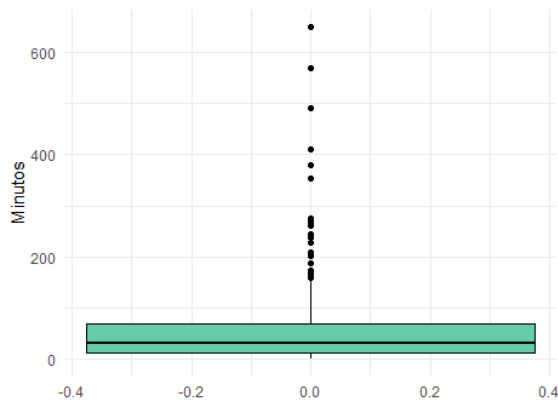


Figura A.6: Boxplot diferencia de minutos.

Tabla A.3: Summary terremoto Tocopilla.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,01	12,70	31,53	54,93	69,02	650,80

Los resultados de la figura A.6 y de la tabla A.3 muestran que los intervalos de tiempo entre sismos presentan una distribución más concentrada en comparación con el caso anterior. Tanto

el primer como el tercer cuartil (12,70 y 69,02 minutos, respectivamente) indican que la mayoría de los eventos ocurren en un rango relativamente estrecho. El valor máximo alcanza los 650,80 minutos, lo que sugiere la presencia de algunos intervalos largos.

Terremoto: Concepción 2010

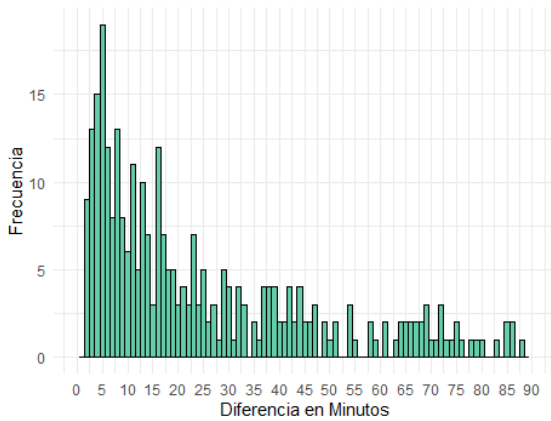


Figura A.7: Diferencia en minutos para magnitudes $M_w \geq 4$.

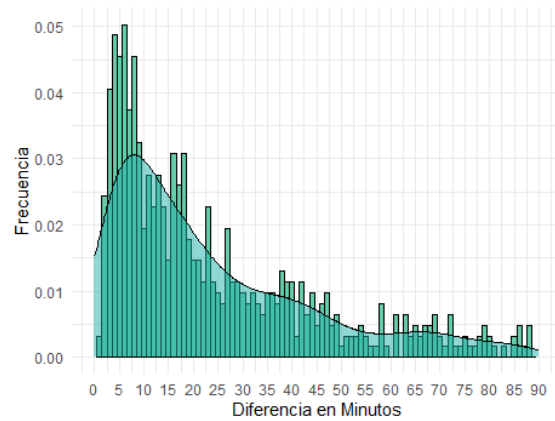


Figura A.8: Densidad de la diferencia en minutos.

En la figura A.7 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen más de 15 eventos con diferencias temporales de aproximadamente 5 minutos respecto al sismo anterior. Por su parte, la figura A.8 muestra que la mayor densidad de ocurrencia se concentra en torno a los 10 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

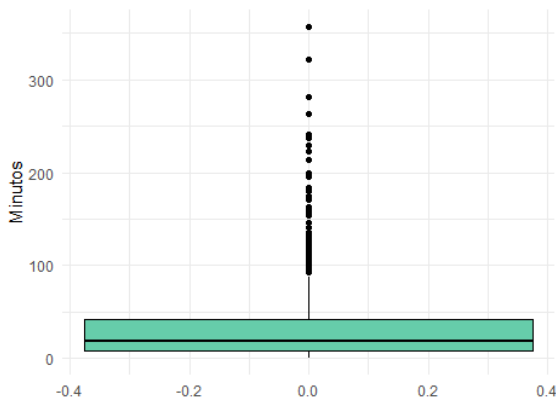


Figura A.9: Boxplot diferencia de minutos.

Tabla A.4: Summary terremoto Concepción.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,016	7,97	18,25	33,93	41,48	357,75

Los resultados de la figura A.9 y de la tabla A.4 muestran una distribución de intervalos de tiempo entre sismos aún más concentrada. La mediana es de 18,25 minutos, y el 50 % central de los datos se encuentra entre 7,97 y 41,48 minutos, lo que indica una alta frecuencia de eventos en intervalos cortos. El valor máximo alcanza los 357,75 minutos, la media de 33,93 minutos sugiere que los valores extremos tienen una influencia moderada.

Terremoto: Iquique 2014

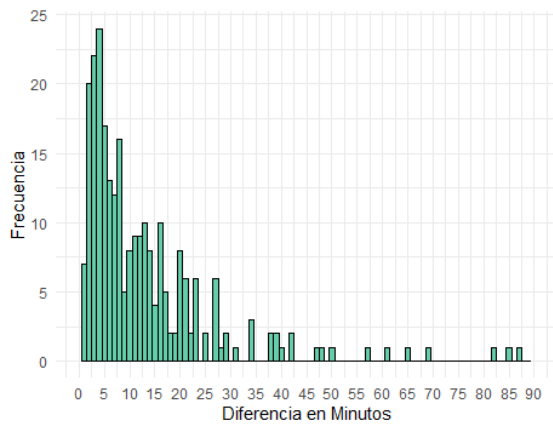


Figura A.10: Diferencia en minutos para magnitudes $M_w \geq 4$.

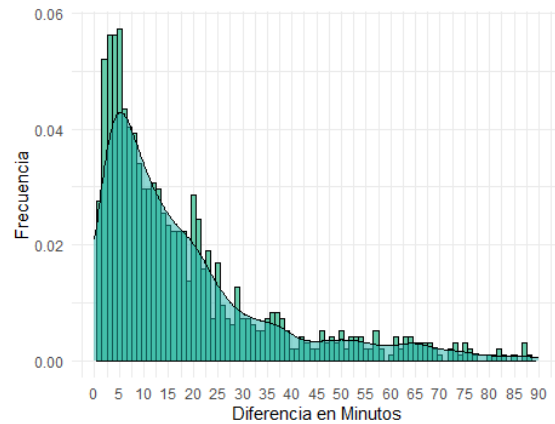


Figura A.11: Densidad de la diferencia en minutos.

En la figura A.10 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen más de 20 eventos con diferencias temporales de aproximadamente 5 minutos respecto al sismo anterior. Por su parte, la figura A.11 muestra que la mayor densidad de ocurrencia se concentran en torno a los 5 y 10 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

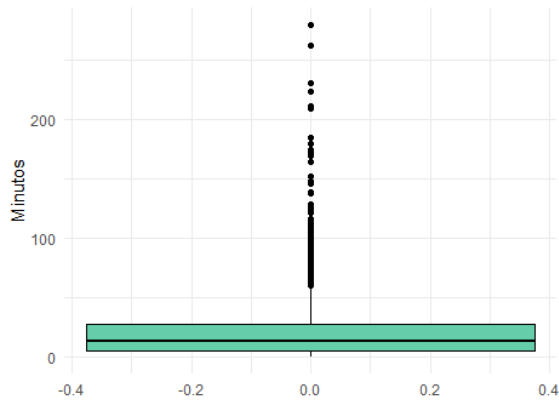


Tabla A.5: Summary terremoto Iquique.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,01	5,67	13,27	24,53	27,39	280,41

Figura A.12: Boxplot diferencia de minutos.

Los resultados de la figura A.12 y de la tabla A.5 indican una distribución de intervalos de tiempo entre sismos altamente concentrada en valores bajos. La mediana es de 13,27 minutos, y el 50 % de los datos se encuentra entre 5,67 y 27,39 minutos, lo que sugiere una ocurrencia frecuente de eventos en lapsos cortos. Aunque el valor máximo alcanza los 280,41 minutos, la media de 24,53 minutos muestra que los valores extremos tienen una influencia limitada.

Terremoto: Coquimbo 2015

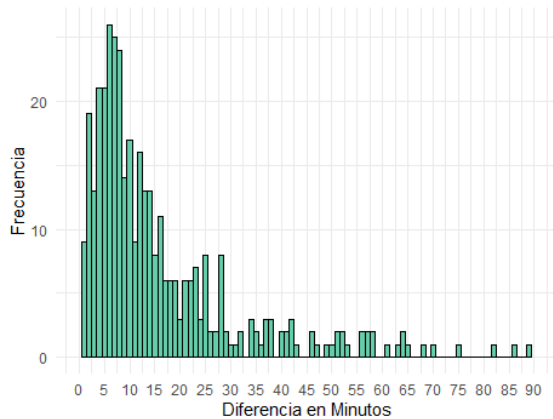


Figura A.13: Diferencia en minutos para magnitudes $M_w \geq 4$.

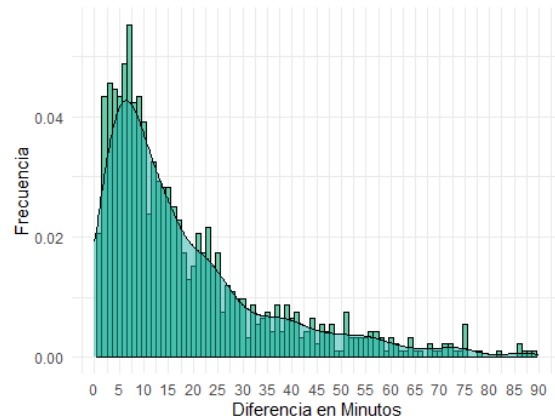


Figura A.14: Densidad de la diferencia en minutos.

En la figura A.13 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen más de 20 eventos con diferencias temporales de aproximadamente 5 y 10 minutos respecto

al sismo anterior. Por su parte, la figura A.14 muestra que la mayor densidad de ocurrencia se concentran en torno a los 5 y 10 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

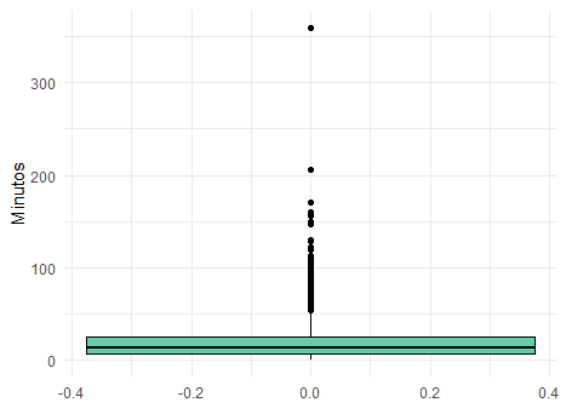


Tabla A.6: Summary terremoto Coquimbo.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,00	6,32	12,85	22,64	25,50	1320,76

Figura A.15: Boxplot diferencia de minutos.

Los resultados de la figura A.15 y de la tabla A.6 muestran una fuerte concentración de sismos en intervalos de tiempo cortos. La mediana es de 12,85 minutos, y el 50 % central de los eventos ocurre entre 6,32 y 25,50 minutos, lo que indica una alta frecuencia de ocurrencia en lapsos breves. Aunque el valor máximo alcanza los 1320,76 minutos, lo que representa un caso atípico considerable, la media de 22,64 minutos sugiere que estos valores extremos tienen una influencia moderada. En general, la distribución refleja una clara tendencia a la agrupación temporal de los sismos, con algunos eventos aislados que se alejan significativamente del patrón general.

Terremoto: Chiloé 2016

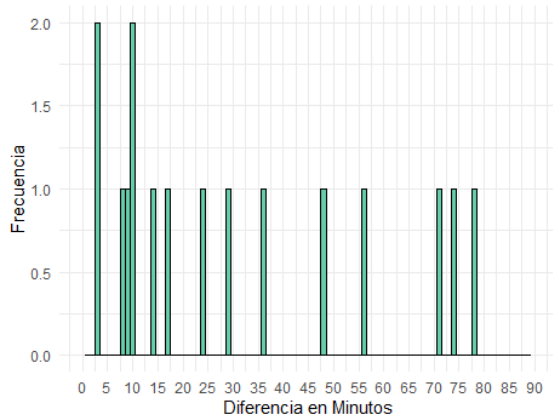


Figura A.16: Diferencia en minutos para magnitudes $M_w \geq 4$.

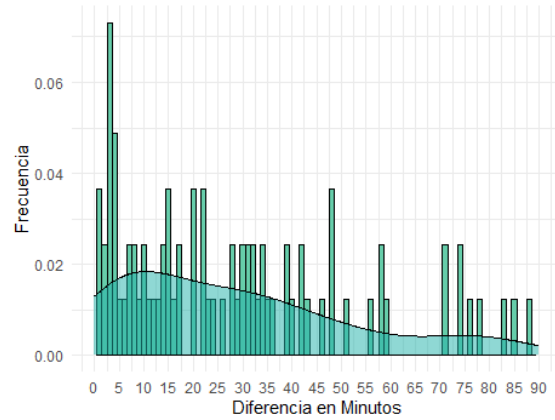


Figura A.17: Densidad de la diferencia en minutos.

En la figura A.16 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen dos eventos con diferencias temporales de aproximadamente 3 y 10 minutos respecto al sismo anterior. Por su parte, la figura A.17 muestra que la mayor densidad de ocurrencia se concentra en torno a los 10 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

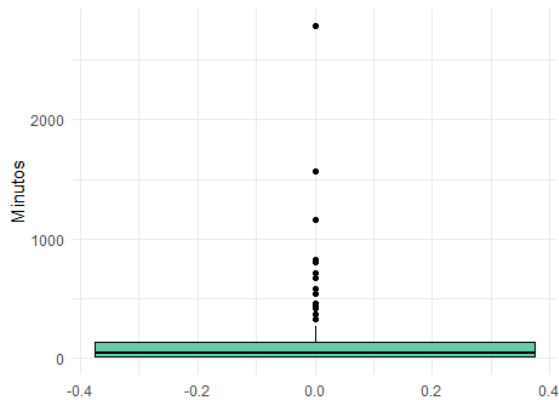


Figura A.18: Boxplot diferencia de minutos.

Tabla A.7: Summary terremoto Chiloé.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,93	15,33	42,36	151,75	138,55	2791,96

Los resultados de la figura A.18 y de la tabla A.7 reflejan una mayor dispersión en los intervalos de tiempo entre sismos. La mediana es de 42,36 minutos, lo que indica que la mitad de los eventos ocurren con una diferencia menor a ese valor. Sin embargo, la media asciende a 151,75 minutos, lo que sugiere la presencia de valores atípicos que elevan el promedio. Esto se confirma con un valor máximo de 2791,96 minutos, considerablemente más alto que el tercer cuartil (138,55

minutos). En conjunto, estos resultados evidencian una distribución con alta variabilidad y algunos intervalos excepcionalmente largos entre eventos.

Terremoto: Atacama 2024

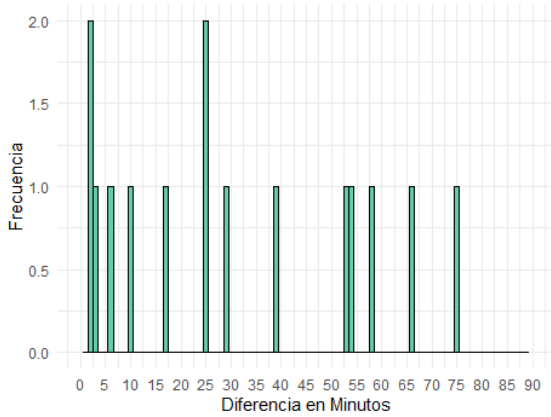


Figura A.19: Diferencia en minutos para magnitudes $M_w \geq 4$.

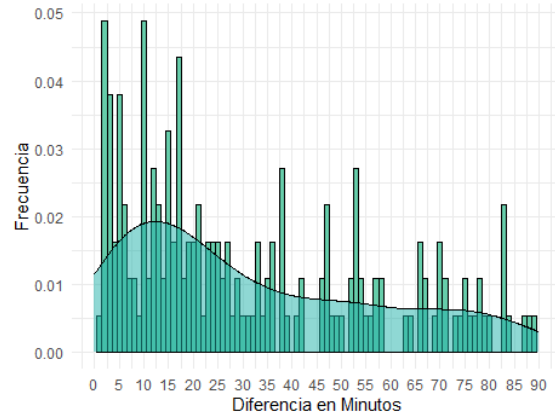


Figura A.20: Densidad de la diferencia en minutos.

En la figura A.19 se observa que, para sismos con magnitudes menores o iguales a 4 Mw, existen dos eventos con diferencias temporales de aproximadamente 3 y 25 minutos respecto al sismo anterior. Por su parte, la figura A.20 muestra que la mayor densidad de ocurrencia se concentran en torno a los 10 y 15 minutos, lo que sugiere que los sismos tienden a agruparse en intervalos cortos de tiempo.

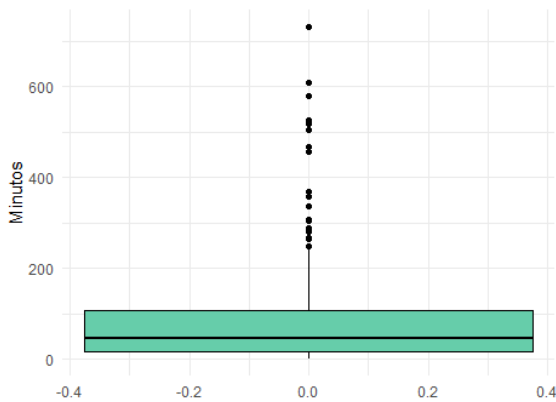


Figura A.21: Boxplot diferencia de minutos.

Tabla A.8: Summary terremoto Atacama.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,05	15,58	47,18	86,47	108,45	731,93

Los resultados de la figura A.21 y de la tabla A.8 muestran una distribución moderadamente dispersa en los intervalos de tiempo entre sismos. La mediana es de 47,18 minutos, lo que indica que la mitad de los eventos ocurren con menos de una hora de diferencia. El primer y tercer cuartil (15,58 y 108,45 minutos, respectivamente) reflejan una concentración significativa de eventos en intervalos relativamente cortos. Sin embargo, la media de 86,47 minutos y el valor máximo de 731,93 minutos sugieren la presencia de algunos intervalos considerablemente más largos, que elevan el promedio general. En conjunto, estos resultados indican una tendencia a la agrupación temporal, aunque con algunos eventos aislados que se distancian del patrón predominante.

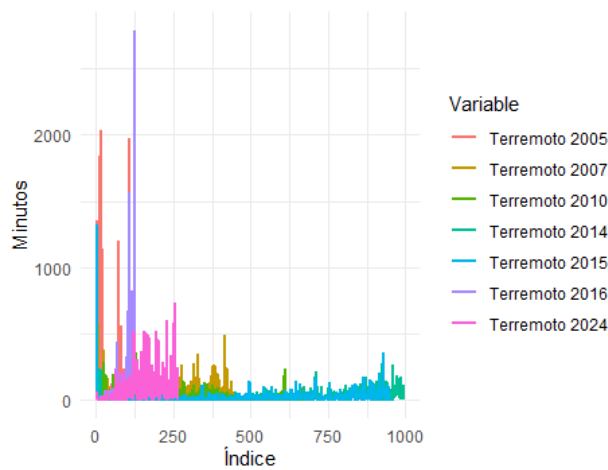


Figura A.22: Minutos de cada Terremoto.

En la figura A.22 se observa que el terremoto de Chiloé presenta la mayor diferencia temporal entre sismos consecutivos, superando los 2.000 minutos, lo que lo convierte en un caso atípico dentro del conjunto analizado.

A.2. Código

A continuación se presenta el código utilizado para entrenar los modelos propuestos.

A.2.1. cGAN

```

1 # =====
2 # 3) Hiperpar metros
3 # =====

```

ANEXO A. ANEXO

```
4 SEQ_LENGTH = df_aftershock.shape[1]
5 NOISE_DIM = 100
6 COND_DIM = df_main.shape[1]
7 LR = 0.0002
8 BETA_1 = 0.5
9 EPOCHS = 10000 #10.000
10 BATCH_SIZE = 64
11 TEST_SIZE = 0.2
12 VALIDATION_SPLIT = 0.1 # Porcentaje del training set para validaci n interna
13 SEED = 42 # Semilla para la divisi n de datos
14
15 # =====
16 # 4) Construcci n de Modelos (cGAN)
17 # =====
18 def build_generator():
19     noise_input = Input(shape=(NOISE_DIM,), name="noise_input")
20     cond_input = Input(shape=(COND_DIM,), name="cond_input")
21     x = Concatenate()([noise_input, cond_input])
22     for units in [256, 512, 1024]:
23         x = Dense(units)(x)
24         x = LeakyReLU(alpha=0.2)(x)
25         x = BatchNormalization()(x)
26     seq_output = Dense(SEQ_LENGTH, activation='tanh', name="gen_output")(x)
27     return Model([noise_input, cond_input], seq_output, name="Generator")
28
29 def build_discriminator():
30     seq_input = Input(shape=(SEQ_LENGTH,), name="seq_input")
31     cond_input = Input(shape=(COND_DIM,), name="cond_input")
32     x = Concatenate()([seq_input, cond_input])
33     for units in [1024, 512, 256]:
34         x = Dense(units)(x)
35         x = LeakyReLU(alpha=0.2)(x)
36     validity = Dense(1, activation='sigmoid', name="disc_output")(x)
37     return Model([seq_input, cond_input], validity, name="Discriminator")
38
39 # Instanciar y compilar los modelos
40 generator = build_generator()
41 discriminator = build_discriminator()
42
43 discriminator.compile(
44     optimizer=Adam(LR, beta_1=BETA_1),
45     loss='binary_crossentropy',
46     metrics=['accuracy']
47 )
48
49 discriminator.trainable = False
50
51 noise_in = Input(shape=(NOISE_DIM,))
52 cond_in = Input(shape=(COND_DIM,))
53 gan_seq = generator([noise_in, cond_in])
54 valid = discriminator([gan_seq, cond_in])
55 gan = Model([noise_in, cond_in], valid, name="cGAN")
```

```

56 gan.compile(
57     optimizer=Adam(LR, beta_1=BETA_1),
58     loss='binary_crossentropy'
59 )
60
61 # =====
62 # 5) Preprocesamiento de Datos
63 # =====
64 X = df_main[['longitud', 'latitud', 'magnitud', 'profundidad', 'long360']].values
65 Y = df_aftershock.values
66
67 # Normalizar datos en el rango [-1, 1] que es compatible con la activaci n 'tanh' del generador
68 scaler_X = MinMaxScaler(feature_range=(-1, 1))
69 X_norm = scaler_X.fit_transform(X)
70
71 scaler_Y = MinMaxScaler(feature_range=(-1, 1))
72 Y_norm = scaler_Y.fit_transform(Y)
73
74 # Divisi n en conjuntos de entrenamiento y prueba con semilla
75 X_train, X_test, Y_train, Y_test = train_test_split(
76     X_norm, Y_norm, test_size=TEST_SIZE, random_state=SEED
77 )
78
79 # =====
80 # 6) Entrenamiento del Modelo con Model Checkpoint
81 # =====
82 # El ModelCheckpoint guardar el generador con la mejor p rdida en un conjunto de validaci n
83 # Nota: La "p rdida de validaci n" para una GAN es un concepto complejo.
84 # Aqu , usaremos la p rdida del generador en un set de validaci n como proxy.
85
86 checkpoint_callback = ModelCheckpoint(
87     filepath=WEIGHTS_BEST, # sin replace
88     save_weights_only=True,
89     monitor='val_loss',
90     mode='min',
91     save_best_only=True,
92     verbose=1
93 )
94
95 def train(epochs, batch_size):
96     # Crear un peque o set de validaci n desde el training set
97     val_size = int(X_train.shape[0] * VALIDATION_SPLIT)
98     X_val, Y_val = X_train[:val_size], Y_train[:val_size]
99     X_train_sub, Y_train_sub = X_train[val_size:], Y_train[val_size:]
100
101     half_batch = batch_size // 2
102     history = {'d_loss': [], 'g_loss': [], 'val_loss': []}
103
104     for epoch in range(1, epochs + 1):
105         # -----
106         # Entrenar Discriminador
107         # -----

```

```

108     idx = np.random.randint(0, X_train_sub.shape[0], half_batch)
109     real_seqs, real_conds = Y_train_sub[idx], X_train_sub[idx]
110
111     noise = np.random.normal(0, 1, (half_batch, NOISE_DIM))
112     fake_seqs = generator.predict([noise, real_conds], verbose=0)
113
114     d_loss_real = discriminator.train_on_batch([real_seqs, real_conds], np.ones((half_batch, 1)))
115     d_loss_fake = discriminator.train_on_batch([fake_seqs, real_conds], np.zeros((half_batch, 1)))
116     d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)
117
118     # -----
119     # Entrenar Generador
120     # -----
121     idx = np.random.randint(0, X_train_sub.shape[0], batch_size)
122     gen_conds = X_train_sub[idx]
123     noise = np.random.normal(0, 1, (batch_size, NOISE_DIM))
124     valid_y = np.ones((batch_size, 1))
125     g_loss = gan.train_on_batch([noise, gen_conds], valid_y)
126
127     # -----
128     # Validaci n y Logging
129     # -----
130     if epoch % 100 == 0:
131         # Calcular p rdida de validaci n para el ModelCheckpoint
132         val_noise = np.random.normal(0, 1, (val_size, NOISE_DIM))
133         val_loss = gan.test_on_batch([val_noise, X_val], np.ones((val_size, 1)))
134
135         history['d_loss'].append(d_loss[0])
136         history['g_loss'].append(g_loss)
137         history['val_loss'].append(val_loss)
138
139         print(f"Epoch_{epoch}/{epochs}|_D_loss:_{d_loss[0]:.4f}|_acc:_{100*d_loss[1]:.1f}%|_G_loss:_{g_loss:.4f}|_Val_loss:_{val_loss:.4f}")
140
141         # L gica del Model Checkpoint manual
142         if val_loss <= min(history['val_loss']):
143             print(f"Validation_loss_mejor _a_{val_loss:.4f}|_guardando_modelo_en_{WEIGHTS_BEST}")
144             generator.save_weights(WEIGHTS_BEST)
145
146
147     return history
148
149 # Iniciar el entrenamiento
150 train(EPOCHS, BATCH_SIZE)

```

A.2.2. TGAN

```

1
2 # =====
3 # 3) Hiperpar metros
4 # =====

```

ANEXO A. ANEXO

```
5 SEQ_LENGTH = Y.shape[1]
6 FEATURES_Y = 1
7
8 NOISE_DIM = 100
9 ZT_DIM = 16
10
11 LR_G = 2e-4
12 LR_D = 8e-5
13 BETA_1 = 0.5
14 EPOCHS = 10000
15 BATCH_SIZE = 64
16 TEST_SIZE = 0.2
17 VALIDATION_SPLIT = 0.10 # 10% del training para validaci n interna
18
19 D_MODEL = 128
20 NUM_HEADS = 4
21 FF_DIM = 256
22 NUM_LAYERS = 2
23 DROPOUT_RATE = 0.10
24
25 LAMBDA_L1_BASE = 2.0
26 LAMBDA_L1_FLOOR = 0.2
27
28 COND_DIM = X.shape[1]
29
30 # =====
31 # 4) Bloques Transformer
32 # =====
33 class PositionalEmbedding(layers.Layer):
34     def __init__(self, seq_len, d_model, name=None, **kwargs):
35         super().__init__(name=name, **kwargs)
36         self.pos_emb = self.add_weight(
37             name="pos_emb",
38             shape=(1, seq_len, d_model),
39             initializer="glorot_uniform",
40             trainable=True
41         )
42
43     def call(self, x):
44         return x + self.pos_emb
45
46 def transformer_encoder(x, num_heads=NUM_HEADS, ff_dim=FF_DIM, dropout=DROPOUT_RATE, causal=False):
47     d_model = int(x.shape[-1])
48     d_k = max(d_model // num_heads, 1)
49     attn = MultiHeadAttention(num_heads=num_heads, key_dim=d_k)
50     attn_out = attn(x, x, use_causal_mask=causal)
51     x = LayerNormalization(epsilon=1e-6)(x + Dropout(dropout)(attn_out))
52     ff = Dense(ff_dim, activation="relu")(x)
53     ff = Dense(d_model)(ff)
54     x = LayerNormalization(epsilon=1e-6)(x + Dropout(dropout)(ff))
55     return x
56
```

```

57 # =====
58 # 5) Modelos: Generador y Discriminador
59 # =====
60 def build_generator_transformer():
61     z = Input(shape=(NOISE_DIM,), name="noise_input")
62     c = Input(shape=(COND_DIM,), name="cond_input")
63     zseq = Input(shape=(SEQ_LENGTH, ZT_DIM), name="noise_seq_input")
64
65     h = Dense(D_MODEL, activation="relu")(Concatenate()([z, c]))
66     H = RepeatVector(SEQ_LENGTH)(h)
67     H = Concatenate(axis=-1)([H, zseq])
68     H = Dense(D_MODEL, activation="relu")(H)
69     H = PositionalEmbedding(SEQ_LENGTH, D_MODEL, name="pos_g")(H)
70
71     for _ in range(NUM_LAYERS):
72         H = transformer_encoder(H, causal=True)
73
74     y_lin = TimeDistributed(Dense(FEATURES_Y), name="gen_lin")(H)
75     yhat = Lambda(lambda t: tf.clip_by_value(t, -1.0, 1.0), name="gen_clip")(y_lin)
76     return Model([z, c, zseq], yhat, name="Generator_Transformer")
77
78 def build_discriminator_transformer():
79     y = Input(shape=(SEQ_LENGTH, FEATURES_Y), name="seq_input")
80     c = Input(shape=(COND_DIM,), name="cond_input")
81     cT = RepeatVector(SEQ_LENGTH)(c)
82     Xd = Concatenate(axis=-1)([y, cT])
83     Xd = Dense(D_MODEL, activation="relu")(Xd)
84     Xd = PositionalEmbedding(SEQ_LENGTH, D_MODEL, name="pos_d")(Xd)
85     for _ in range(NUM_LAYERS):
86         Xd = transformer_encoder(Xd, causal=False)
87     Xd = GlobalAveragePooling1D()(Xd)
88     out = Dense(1, activation="sigmoid", name="disc_output")(Xd)
89     return Model([y, c], out, name="Discriminator_Transformer")
90
91 generator = build_generator_transformer()
92 discriminator = build_discriminator_transformer()
93
94 discriminator.compile(optimizer=Adam(LR_D, beta_1=BETA_1),
95                       loss='binary_crossentropy', metrics=['accuracy'])
96
97 # cGAN compuesto para entrenar G (D congelado dentro de g_train)
98 discriminator.trainable = False
99 z_in = Input(shape=(NOISE_DIM,), name="z_in")
100 c_in = Input(shape=(COND_DIM,), name="c_in")
101 zseq_in = Input(shape=(SEQ_LENGTH, ZT_DIM), name="zseq_in")
102 fake_y = generator([z_in, c_in, zseq_in])
103 valid = discriminator([fake_y, c_in])
104
105 g_train = Model([z_in, c_in, zseq_in], [valid, fake_y], name="g_train")
106 g_train.compile(optimizer=Adam(LR_G, beta_1=BETA_1),
107                loss=['binary_crossentropy', 'mae'],
108                loss_weights=[1.0, LAMBDA_LL_BASE])

```

ANEXO A. ANEXO

```
109 discriminator.trainable = True
110
111 # =====
112 # 6) Preprocesamiento (sin fuga de datos)
113 # =====
114 X_train_raw, X_test_raw, Y_train_2d_raw, Y_test_2d_raw = train_test_split(
115     X, Y, test_size=TEST_SIZE, random_state=SEED
116 )
117
118 scaler_X = MinMaxScaler(feature_range=(-1, 1)).fit(X_train_raw)
119 scaler_Y = MinMaxScaler(feature_range=(-1, 1)).fit(Y_train_2d_raw)
120
121 X_train = scaler_X.transform(X_train_raw)
122 X_test = scaler_X.transform(X_test_raw)
123
124 Y_train_2d = scaler_Y.transform(Y_train_2d_raw)
125 Y_test_2d = scaler_Y.transform(Y_test_2d_raw)
126
127 Y_train = Y_train_2d.reshape((-1, SEQ_LENGTH, FEATURES_Y))
128 Y_test = Y_test_2d.reshape((-1, SEQ_LENGTH, FEATURES_Y))
129
130 # =====
131 # 7) Entrenamiento + Checkpoint Manual (mejor val_loss)
132 # =====
133 def linear_decay(val0, valf, step, total):
134     alpha = min(max(step / float(total), 0.0), 1.0)
135     return val0 + (valf - val0) * alpha
136
137 def train(epochs, batch_size):
138     # peque o set de validaci n interno
139     val_size = int(X_train.shape[0] * VALIDATION_SPLIT)
140     X_val, Y_val = X_train[:val_size], Y_train[:val_size]
141     X_train_sub, Y_train_sub = X_train[val_size:], Y_train[val_size:]
142
143     n_train = X_train_sub.shape[0]
144     half_batch = batch_size // 2
145     min_val_loss = float('inf')
146
147     for epoch in range(1, epochs + 1):
148         # decaimiento de L1 para el generador
149         lam_l1 = linear_decay(LAMBDA_L1_BASE, LAMBDA_L1_FLOOR, epoch, epochs)
150         g_train.loss_weights = [1.0, lam_l1]
151
152         # --- Discriminador ---
153         idx_real = np.random.randint(0, n_train, half_batch)
154         Xr, Yr = X_train_sub[idx_real], Y_train_sub[idx_real]
155         noise_vec = np.random.normal(0, 1, (half_batch, NOISE_DIM))
156         noise_seq = np.random.normal(0, 1, (half_batch, SEQ_LENGTH, ZT_DIM))
157         Yf = generator.predict([noise_vec, Xr, noise_seq], verbose=0)
158
159         d_loss_real = discriminator.train_on_batch([Yr, Xr], np.ones((half_batch, 1)) * 0.9)
160         d_loss_fake = discriminator.train_on_batch([Yf, Xr], np.zeros((half_batch, 1)))
```

```

161     d_loss = 0.5 * np.add(d_loss_real , d_loss_fake)
162
163     # --- Generador (2 pasos) ---
164     for _ in range(2):
165         noise_vec = np.random.normal(0, 1, (batch_size , NOISE_DIM))
166         noise_seq = np.random.normal(0, 1, (batch_size , SEQ_LENGTH, ZT_DIM))
167         idx_g     = np.random.randint(0, n_train , batch_size)
168         Xb, Yb    = X_train_sub[idx_g], Y_train_sub[idx_g]
169         g_loss    = g_train.train_on_batch([noise_vec , Xb, noise_seq],
170                                           [np.ones((batch_size , 1)), Yb])
171
172     # --- Validaci n + checkpoint cada 100 pocas ---
173     if epoch % 100 == 0:
174         val_noise_vec = np.random.normal(0, 1, (val_size , NOISE_DIM))
175         val_noise_seq = np.random.normal(0, 1, (val_size , SEQ_LENGTH, ZT_DIM))
176         val_loss      = g_train.test_on_batch([val_noise_vec , X_val, val_noise_seq],
177                                             [np.ones((val_size , 1)), Y_val])
178
179         g_total , g_bce , g_l1 = g_loss [0], g_loss [1], g_loss [2]
180         print (f"Epoch_{epoch}/{epochs}_|_D_loss :_{d_loss [0]:.4 f} ,_acc :_{100*d_loss [1]:.1 f}%_
181               f"|_G_loss :_{g_total :.4 f}|_Val_G_loss :_{val_loss [0]:.4 f}")
182
183         # === Model Checkpoint manual (mejor val_loss) ===
184         if val_loss [0] < min_val_loss :
185             min_val_loss = val_loss [0]
186             print (f"Validation_loss_mejor _a_{min_val_loss :.4 f} ,_guardando_modelo_en_{MODEL_SAVE_PATH}\n")
187             generator.save_weights(MODEL_SAVE_PATH)
188
189 # Ejecutar entrenamiento (descomenta para entrenar)
190 train(EPOCHS, BATCH_SIZE)

```

A.2.3. WGAN

```

1
2 # =====
3 # 3) Hiperpar metros (ajustes WGAN-GP estables)
4 # =====
5 SEQ_LENGTH   = Y.shape[1]
6 FEATURES_Y   = 1
7
8 NOISE_DIM    = 100
9 ZT_DIM       = 16
10
11 # WGAN-GP estable
12 LR_G         = 1e-4
13 LR_D         = 1e-4
14 BETA_1       = 0.0
15 BETA_2       = 0.99
16
17 EPOCHS       = 10000

```

```

18 BATCH_SIZE = 64
19 TEST_SIZE = 0.2
20 VALIDATION_SPLIT = 0.10
21
22 D_MODEL = 128
23 NUM_LAYERS = 4
24
25 # Límites suave (coherente con cGAN/GANT previos)
26 LAMBDA_LI_BASE = 0.5
27 LAMBDA_LI_FLOOR = 0.05
28
29 LAMBDA_GP = 10.0
30 COND_DIM = X.shape[1]
31 N_CRITIC = 5
32
33 # Instance noise (decae a 0)
34 SIGMA_START = 0.05
35 SIGMA_END = 0.0
36
37 # =====
38 # 4) Bloques y Construcción de Modelos
39 # =====
40 def temporal_res_block(x, units=D_MODEL):
41     h = Dense(units, activation="relu")(x)
42     h = Dense(units)(h)
43     return LayerNormalization(epsilon=1e-6)(x + h)
44
45 def build_generator():
46     z = Input(shape=(NOISE_DIM,))
47     c = Input(shape=(COND_DIM,))
48     zseq = Input(shape=(SEQ_LENGTH, ZT_DIM))
49
50     h0 = Dense(D_MODEL, activation="relu")(Concatenate()([z, c]))
51     H = RepeatVector(SEQ_LENGTH)(h0)
52     H = Concatenate(axis=-1)([H, zseq])
53     H = Dense(D_MODEL, activation="relu")(H)
54
55     for _ in range(NUM_LAYERS):
56         H = temporal_res_block(H, D_MODEL)
57
58     y_lin = TimeDistributed(Dense(FEATURES_Y))(H)
59     yhat = Lambda(lambda x: tf.clip_by_value(x, -1.0, 1.0))(y_lin)
60     return Model([z, c, zseq], yhat, name="Generator_WGAN")
61
62 def build_critic():
63     y = Input(shape=(SEQ_LENGTH, FEATURES_Y))
64     c = Input(shape=(COND_DIM,))
65     cT = RepeatVector(SEQ_LENGTH)(c)
66     Xd = Concatenate(axis=-1)([y, cT])
67     Xd = Dense(D_MODEL, activation="relu")(Xd)
68     for _ in range(NUM_LAYERS):
69         Xd = temporal_res_block(Xd, D_MODEL)

```

```

70     Xd = GlobalAveragePooling1D()(Xd)
71     out = Dense(1)(Xd)
72     return Model([y, c], out, name="Critic_WGAN")
73
74 generator = build_generator()
75 critic    = build_critic()
76
77 # Optimizers con clip de gradiente
78 g_opt = Adam(learning_rate=LR_G, beta_1=BETA_1, beta_2=BETA_2, clipnorm=1.0)
79 d_opt = Adam(learning_rate=LR_D, beta_1=BETA_1, beta_2=BETA_2, clipnorm=1.0)
80
81 # =====
82 # 5) Preprocesamiento
83 # =====
84 assert Y.shape[1] == SEQ_LENGTH, f"Y_debe_tener_{SEQ_LENGTH}_columnas."
85
86 X_train_raw, X_test_raw, Y_train_2d_raw, Y_test_2d_raw = train_test_split(
87     X, Y, test_size=TEST_SIZE, random_state=SEED
88 )
89
90 scaler_X = MinMaxScaler(feature_range=(-1, 1)).fit(X_train_raw)
91 scaler_Y = MinMaxScaler(feature_range=(-1, 1)).fit(Y_train_2d_raw)
92
93 X_train = scaler_X.transform(X_train_raw)
94 X_test  = scaler_X.transform(X_test_raw)
95
96 Y_train_2d = scaler_Y.transform(Y_train_2d_raw)
97 Y_test_2d  = scaler_Y.transform(Y_test_2d_raw)
98
99 Y_train = Y_train_2d.reshape((-1, SEQ_LENGTH, FEATURES_Y))
100 Y_test  = Y_test_2d.reshape((-1, SEQ_LENGTH, FEATURES_Y))
101
102 # =====
103 # 6) Helpers: decaimientos e instance noise
104 # =====
105 def linear_decay(val0, valf, step, total):
106     alpha = min(max(step / float(total), 0.0), 1.0)
107     return val0 + (valf - val0) * alpha
108
109 def sigma_decay(step, total):
110     return linear_decay(SIGMA_START, SIGMA_END, step, total)
111
112 @tf.function
113 def add_instance_noise(y, sigma):
114     # si sigma=0 retorna y; si no, y + N(0, sigma)
115     return y + tf.random.normal(tf.shape(y), stddev=sigma) if sigma > 0.0 else y
116
117 # =====
118 # 7) Lógica de Entrenamiento (WGAN-GP)
119 # =====
120 @tf.function
121 def critic_step(real_y, cond_x, sigma):

```

```

122     batch_size = tf.shape(real_y)[0]
123     z = tf.random.normal((batch_size, NOISE_DIM))
124     zseq = tf.random.normal((batch_size, SEQ_LENGTH, ZT_DIM))
125
126     with tf.GradientTape() as tape:
127         fake_y = generator([z, cond_x, zseq], training=True)
128
129         # instance noise en entradas al cr tico
130         real_y_noisy = add_instance_noise(real_y, sigma)
131         fake_y_noisy = add_instance_noise(fake_y, sigma)
132
133         real_score = critic([real_y_noisy, cond_x], training=True)
134         fake_score = critic([fake_y_noisy, cond_x], training=True)
135
136         # Wasserstein distance term
137         w_dist = tf.reduce_mean(fake_score) - tf.reduce_mean(real_score)
138
139         # Gradient penalty en trayectorias entre real y fake (sin ruido)
140         epsilon = tf.random.uniform((batch_size, 1, 1), 0.0, 1.0)
141         interp = epsilon * real_y + (1.0 - epsilon) * fake_y
142         with tf.GradientTape() as gp_tape:
143             gp_tape.watch(interp)
144             interp_score = critic([interp, cond_x], training=True)
145             grads = gp_tape.gradient(interp_score, interp)
146             grad_norm = tf.sqrt(tf.reduce_sum(tf.square(grads), axis=[1, 2]))
147             gp = tf.reduce_mean(tf.square(grad_norm - 1.0))
148
149         d_loss = w_dist + LAMBDA_GP * gp
150
151     d_grads = tape.gradient(d_loss, critic.trainable_variables)
152     d_opt.apply_gradients(zip(d_grads, critic.trainable_variables))
153     return d_loss, w_dist, gp
154
155 @tf.function
156 def generator_step(real_y, cond_x, lambda_l1):
157     batch_size = tf.shape(real_y)[0]
158     z = tf.random.normal((batch_size, NOISE_DIM))
159     zseq = tf.random.normal((batch_size, SEQ_LENGTH, ZT_DIM))
160     with tf.GradientTape() as tape:
161         fake_y = generator([z, cond_x, zseq], training=True)
162         fake_score = critic([fake_y, cond_x], training=True)
163         g_adv = -tf.reduce_mean(fake_score) # objetivo WGAN
164         g_l1 = tf.reduce_mean(tf.abs(fake_y - real_y))
165         g_loss = g_adv + lambda_l1 * g_l1
166     g_grads = tape.gradient(g_loss, generator.trainable_variables)
167     g_opt.apply_gradients(zip(g_grads, generator.trainable_variables))
168     return g_loss, g_adv, g_l1
169
170 def train(epochs, batch_size):
171     # set de validaci n para checkpoint (m trica: L1 en validaci n)
172     val_size = int(X_train.shape[0] * VALIDATION_SPLIT)
173     X_val = tf.convert_to_tensor(X_train[:val_size], dtype=tf.float32)

```

```

174 Y_val = tf.convert_to_tensor(Y_train[:val_size], dtype=tf.float32)
175
176 X_train_sub, Y_train_sub = X_train[val_size:], Y_train[val_size:]
177 n_train = X_train_sub.shape[0]
178 steps_per_epoch = max(1, n_train // batch_size)
179
180 best_val_l1 = float('inf')
181
182 for epoch in range(1, epochs + 1):
183     lam_l1 = linear_decay(LAMBDA_L1_BASE, LAMBDA_L1_FLOOR, epoch, epochs)
184     sigma = sigma_decay(epoch, epochs)
185
186     idx_perm = np.random.permutation(n_train)
187     for step in range(steps_per_epoch):
188         start = step * batch_size
189         end = start + batch_size
190         idx = idx_perm[start:end]
191         Xb = tf.convert_to_tensor(X_train_sub[idx], dtype=tf.float32)
192         Yb = tf.convert_to_tensor(Y_train_sub[idx], dtype=tf.float32)
193
194         for _ in range(N_CRITIC):
195             d_loss, w_dist, gp = critic_step(Yb, Xb, tf.constant(sigma, dtype=tf.float32))
196             g_loss, g_adv, g_l1 = generator_step(Yb, Xb, tf.constant(lam_l1, dtype=tf.float32))
197
198         # Validaci n + checkpoint cada 100 pocas
199         if epoch % 100 == 0:
200             z_val = tf.random.normal((tf.shape(X_val)[0], NOISE_DIM))
201             zseq_val = tf.random.normal((tf.shape(X_val)[0], SEQ_LENGTH, ZT_DIM))
202             y_val_hat = generator([z_val, X_val, zseq_val], training=False)
203             val_l1 = tf.reduce_mean(tf.abs(y_val_hat - Y_val)).numpy()
204
205             print(f"Epoch_{epoch}/{epochs}_d_loss:_{d_loss.numpy():.4f}_
206                   f"_w_dist:_{w_dist.numpy():.4f}_gp:_{gp.numpy():.4f}_
207                   f"_g_loss:_{g_loss.numpy():.4f}_adv_{g_adv.numpy():.4f}_+_lam_l1:_{lam_l1:.3f}*L1_{g_l1.numpy():.4f}_
208                   f"_val_l1:_{val_l1:.4f}")
209
210             if val_l1 < best_val_l1:
211                 best_val_l1 = val_l1
212                 print(f"Val_L1_mejor _a_{best_val_l1:.4f}_guardando_modelo_en_{MODEL_SAVE_PATH}\n")
213                 generator.save_weights(MODEL_SAVE_PATH)
214
215 # === Entrenamiento (descomenta si quieres entrenar de inmediato) ===
216 train(EPOCHS, BATCH_SIZE)

```

Referencias bibliográficas

- [1] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.
- [2] Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical report, Stanford.
- [3] Bao, Z., Zhao, J., Huang, P., Yong, S., and Wang, X. (2021). A deep learning-based electromagnetic signal for earthquake magnitude prediction. *Sensors*, 21(13):4434.
- [4] Bilek, S. L. and Lay, T. (2018). Subduction zone megathrust earthquakes. *Geosphere*, 14(4):1468–1500.
- [5] Centro Sismológico Nacional de Chile (2024). <https://www.sismologia.cl/>. Accedido el 25 de noviembre de 2024.
- [6] Congreso Nacional de Chile (2023). Ley n° 21.302: Modifica la ley n° 16.744 sobre accidentes del trabajo y enfermedades profesionales, para mejorar el régimen de prestaciones y la eficiencia de la administración de las instituciones del sistema de seguridad social. <https://www.bcn.cl/leychile/navegar?idNorma=1188583&idParte=10404398&idVersion=2023-02-03&form=MG0AV3>. Accedido el 10 de diciembre de 2024.
- [7] Cox, D. R. and Isham, V. (1980). *Point processes*, volume 12. CRC Press.
- [8] Cressie, N. and Wikle, C. K. (2011). *Statistics for spatio-temporal data*. John Wiley & Sons.
- [9] Dzierma, Y., Thorwart, M., Rabbell, W., Siegmund, C., Comte, D., Bataille, K., Iglesia, P., and Prezzi, C. (2012). Seismicity near the slip maximum of the 1960 mw 9.5 valdivia earthquake (chile): Plate interface lock and reactivation of the subducted valdivia fracture zone. *Journal of Geophysical Research: Solid Earth*, 117(B6).
- [10] Fan, Z. and Xu, X. (2019). Application and visualization of typical clustering algorithms in seismic data analysis. *Procedia Computer Science*, 151:171–178. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd

International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

- [11] Ghofrani, H. and Atkinson, G. M. (2015). Duration of the 2011 tohoku earthquake ground motions. *Journal of Seismology*, 19:9–25.
- [12] Ghosh, S. K. and Cleland, N. (2012). Observations from the february 27, 2010, earthquake in chile. *PCI journal*, 57(1).
- [13] Gobierno Regional de Santiago (2024). Gobernador orrego por solicitud del gobierno regional: Se inició modificación al prms por falla de san ramón. <https://www.gobiernosantiago.cl/gobernador-orrego-por-solicitud-del-gobierno-regional-se-inicio-modificacion-al-prms-por-falla-de-san-ramon/>. Accedido el 10 de diciembre de 2024.
- [14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014a). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- [15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014b). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [16] Hall-Wallace, M. K. (1998). Can earthquakes be predicted? *Journal of Geoscience Education*, 46(5):439–449.
- [17] Hayes, G. P., Smoczyk, G. M., Villaseñor, A. H., Furlong, K. P., and Benz, H. M. (2020). Seismicity of the earth 1900–2018. Technical report, US Geological Survey.
- [18] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer US.
- [19] Kagan, Y. Y. (1997). Are earthquakes predictable? *Geophysical Journal International*, 131(3):505–525.
- [20] Kodinariya, T. M., Makwana, P. R., et al. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.
- [21] Madariaga, R. (1998). Sismicidad de chile. *Física de la Tierra*, 10(1):221–258.
- [22] McGuire, J. J. and Kaneko, Y. (2018). Directly estimating earthquake rupture area using second moments to reduce the uncertainty in stress drop. *Geophysical Journal International*, 214(3):2224–2235.

- [23] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [24] Molina, D., Tassara, A., Abarca, R., Melnick, D., and Madella, A. (2021). Frictional segmentation of the Chilean megathrust from a multivariate analysis of geophysical, geological, and geodetic data. *Journal of Geophysical Research: Solid Earth*, 126(6):e2020JB020647.
- [25] Montero, J.-M., Fernández-Avilés, G., and Mateu, J. (2015). *Spatial and spatio-temporal geostatistical modeling and kriging*. John Wiley & Sons.
- [26] Mousavi, S. M. and Beroza, G. C. (2020). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1):e2019GL085976.
- [27] Narteau, C. and Shebalin, P. (2017). Depth dependent stress revealed by aftershocks. In *AGU Fall Meeting Abstracts*, volume 2017, pages S23F–03.
- [28] Nicolis, O., Delgado, L., Peralta, B., Díaz, M., and Chiodi, M. (2024). Space-time clustering of seismic events in Chile using st-dbscan-ev algorithm. *Environmental and Ecological Statistics*, pages 1–28.
- [29] Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402.
- [30] Palma, W. (2016). *Time series analysis*. John Wiley & Sons.
- [31] Parihar, D., Ghosh, R., Akula, A., Kumar, S., and Sardana, H. (2018). Machine learning based comparative analysis for the classification of earthquake signals. In *Proceedings of the International Conference on Computing and Communication Systems: I3CS 2016, NEHU, Shillong, India*, pages 369–376. Springer.
- [32] Pulido-Calvo, I., Gutiérrez-Estrada, J. C., and Sanz-Fernández, V. (2024). Persistent neural calibration for discharges modelling in drought-stressed catchments. *Expert Systems with Applications*, 249:123785.
- [33] Python Software Foundation (2023). Python programming language. <https://www.python.org>.
- [34] R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- [35] Raschka, S. and Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd.

- [36] Rietbrock, A., Ryder, I., Hayes, G., Haberland, C., Comte, D., Roecker, S., and Lyon-Caen, H. (2012). Aftershock seismicity of the 2010 maule mw= 8.8, chile, earthquake: Correlation between co-seismic slip models and aftershock distribution? *Geophysical Research Letters*, 39(8).
- [37] Ruiz, S., Madariaga, R., Astroza, M., Saragoni, G. R., Lancieri, M., Vigny, C., and Campos, J. (2012). Short-period rupture process of the 2010 mw 8.8 maule earthquake in chile. *Earthquake Spectra*, 28(1_suppl1):1–18.
- [38] Senapred (2024). Autoridades monitorearon respuesta de planes de seguridad y de emergencia durante simulacro de terremoto para el sector educación de la región metropolitana. <https://web.senapred.cl/noticias/autoridades-monitorearon-respuesta-de-planes-de-seguridad-y-de-emergencia-durante-simulacro-de-terremoto-para-el-sector-educacion-de-la-region-metropolitana/?form=MGOAV3>. Accedido el 10 de diciembre de 2024.
- [39] Shearer, P. M. (2019). *Introduction to seismology*. Cambridge University Press.
- [40] Stein, S. and Okal, E. A. (2005). Speed and size of the sumatra earthquake. *Nature*, 434(7033):581–582.
- [41] Stern, R. J. (2002). Subduction zones. *Reviews of geophysics*, 40(4):3–1.
- [42] United States Geological Survey (2024). Usgs earthquake hazards program. <https://earthquake.usgs.gov>. Base de datos y monitoreo global de terremotos proporcionado por el Servicio Geológico de los Estados Unidos.
- [43] Utsu, T., Ogata, Y., et al. (1995a). The centenary of the omori formula for a decay law of aftershock activity. *Journal of Physics of the Earth*, 43(1):1–33.
- [44] Utsu, T., Ogata, Y., S, R., and Matsu'ura (1995b). The centenary of the omori formula for a decay law of aftershock activity. *Journal of Physics of the Earth*, 43(1):1–33.
- [45] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [46] Zaliapin, I. and Ben-Zion, Y. (2013). Earthquake clusters in southern california i: Identification and stability. *Journal of Geophysical Research: Solid Earth*, 118(6):2847–2864.
- [47] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.