



ESPE
ESCUELA POLITÉCNICA DEL EJÉRCITO
CAMINO A LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES CARRERA DE TELECOMUNICACIONES

**“CREACIÓN DE BASES DE DATOS SINTÉTICAS DE EVENTOS
SÍSMICOS PRODUCIDOS EN EL VOLCÁN LLAIMA CON DEEP
LEARNING.”**

Trabajo de integración curricular previo a la obtención del título
de:
Ingeniera en Telecomunicaciones

Autora: Myriam Cristina Estrella Caicedo

Director del Proyecto: Ing. Román Alcides Lara Cueva, PhD.



Agenda

- 1.- **Introducción**
- 2.- **Desarrollo**
- 3.- **Pruebas y Resultados**
- 4.- **Conclusiones y Trabajos futuros**



1. Introducción

Motivación e Importancia



- Generación.
- Clasificación.
- Detección.
- Caracterización.
- Monitorización.

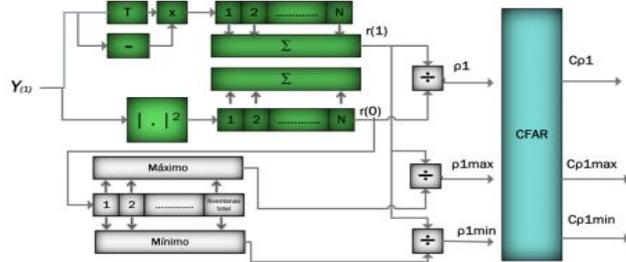


Sistemas de detección automático.

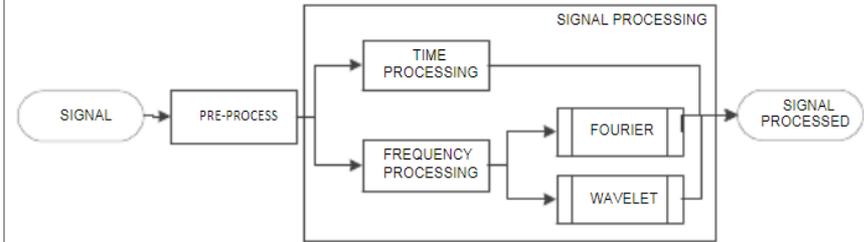
1. Introducción

Antecedentes

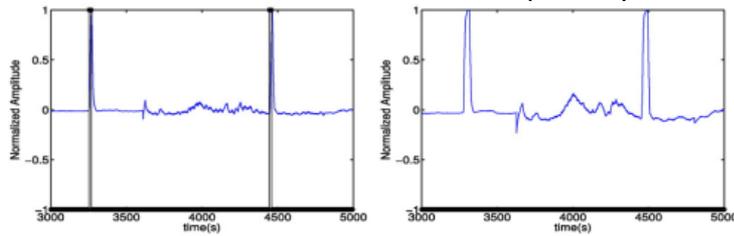
Estimadores de entropía



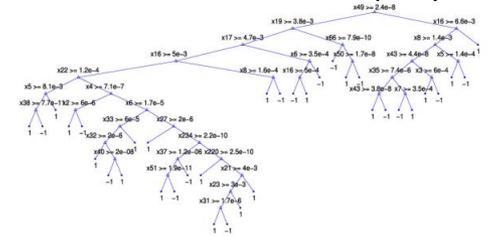
Análisis con transformadas de Fourier y Wavelet



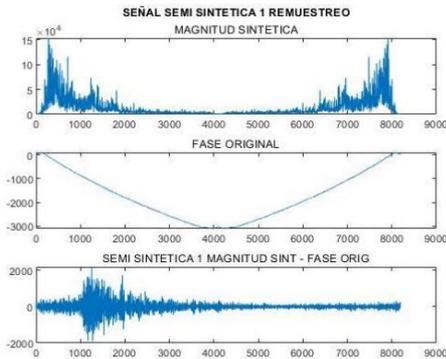
k vecinos más cercanos (k-NN)



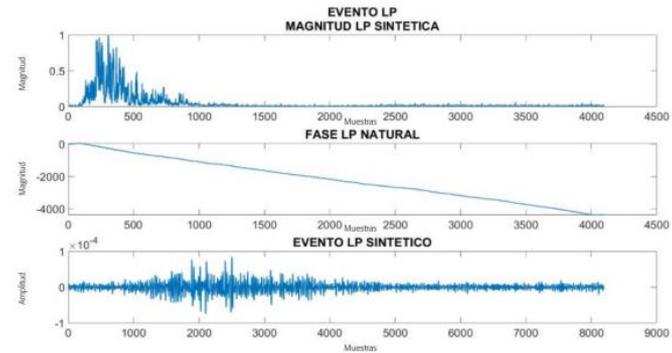
Árboles de decisión (DT)



Bootstrap



CGAN



1. Introducción

Objetivo

- Crear bases de datos sintéticas de eventos sísmicos producidos en el volcán Llaima con Deep Learning.

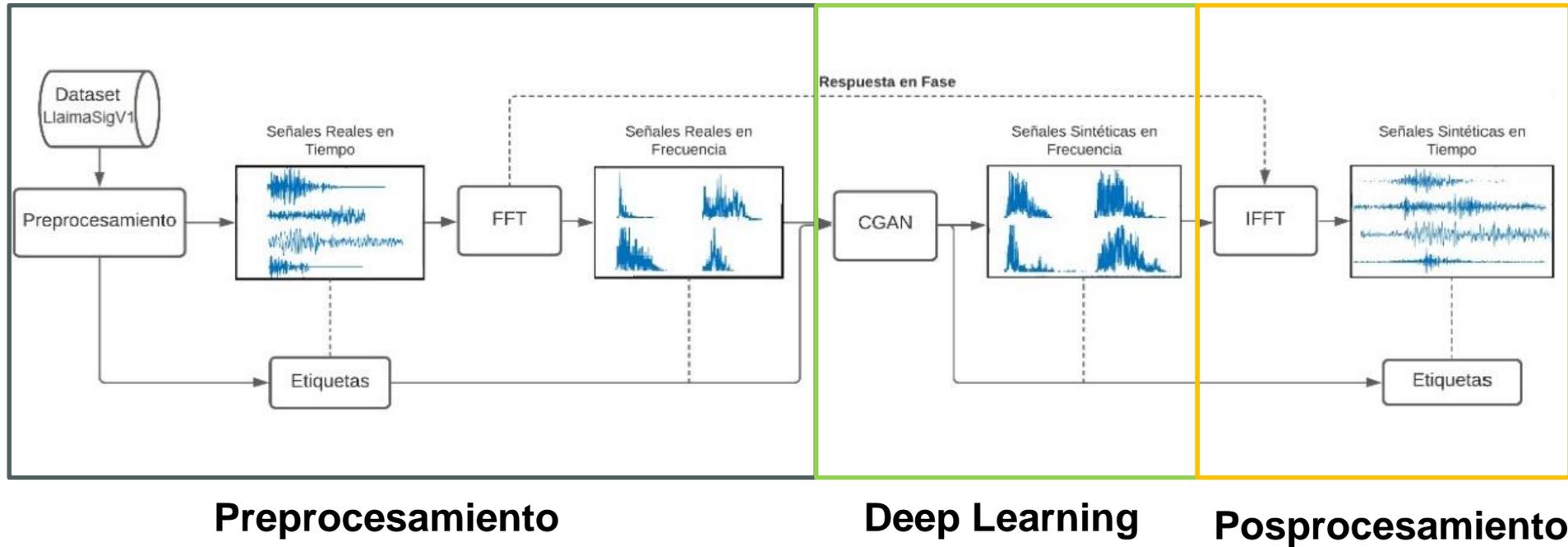
Actividades

- Identificación de las propiedades y variables del problema a tener en cuenta.
- **Reunión de la base de datos** representativa, **con datos provistos por el Observatorio Vulcanológico de los Andes del Sur.**
- Creación de una estructura de datos (temporales, espaciales, otros.) con un soporte común.
- **Caracterización de microsismos.**
- **Generación de bases sintéticas con Deep Learning.**
- **Pruebas y evaluación de desempeño.**



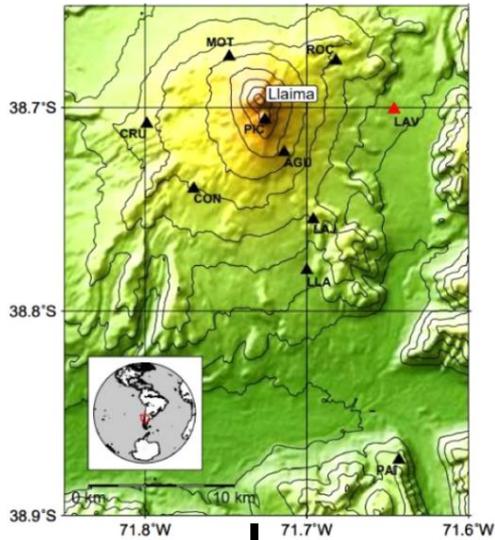
2. Desarrollo

Diagrama General



2. Desarrollo

Base de Datos



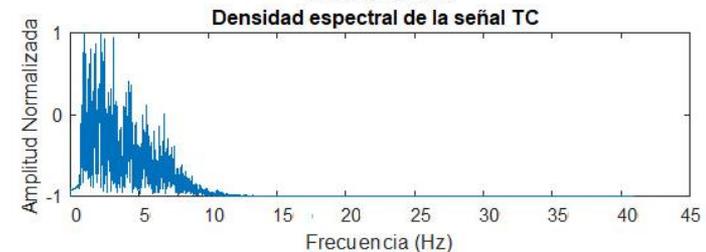
- LAV: una de las siete estaciones.
- 3592 señales de tiempo-frecuencia.
- 2010 hasta 2016

Largo Periodo (LP)
1310

Volcano Tectónico (VT)
304

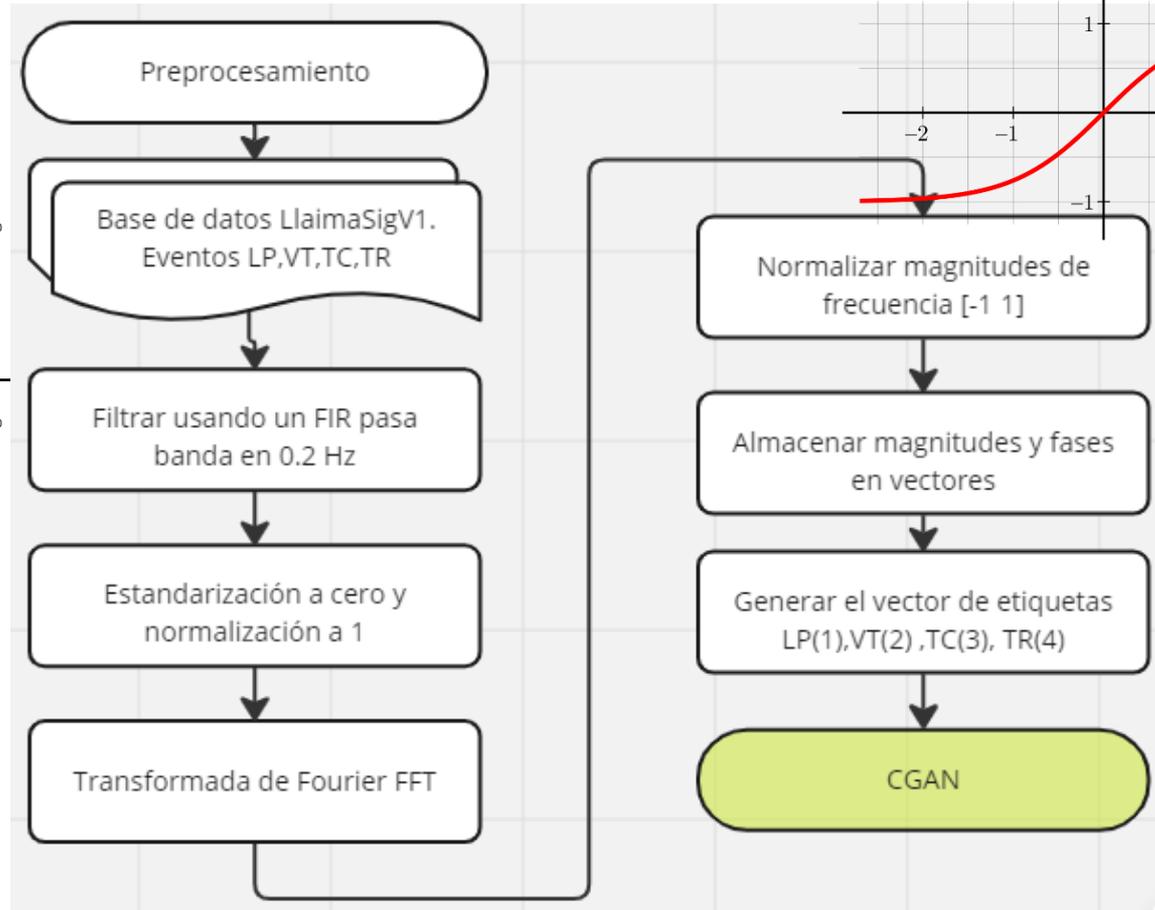
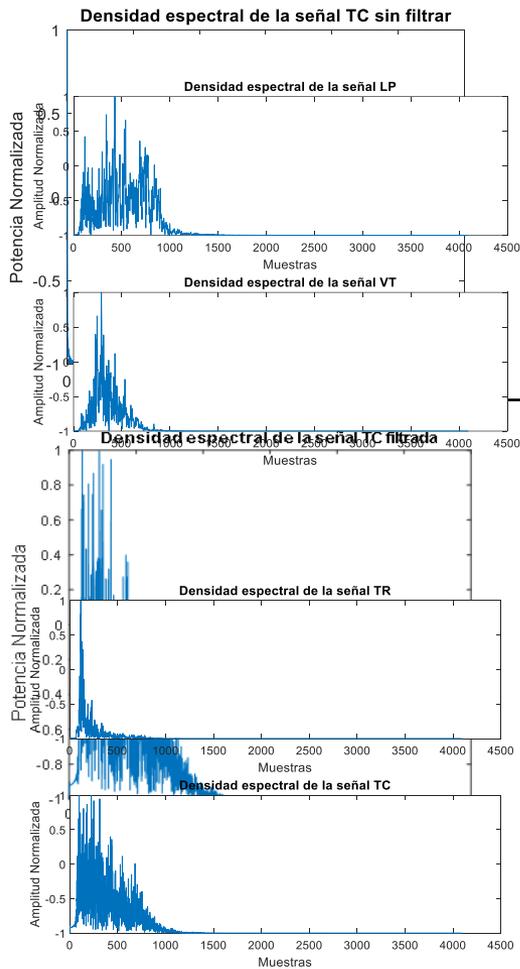
Tremor (TR)
490

Tectónico (TC)
1488



2. Desarrollo

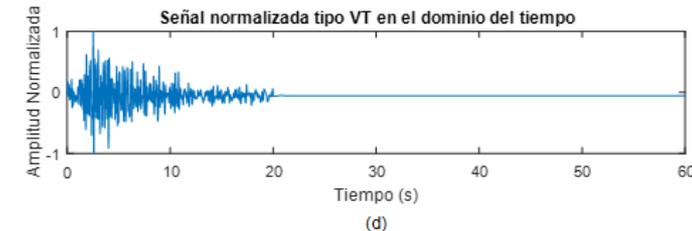
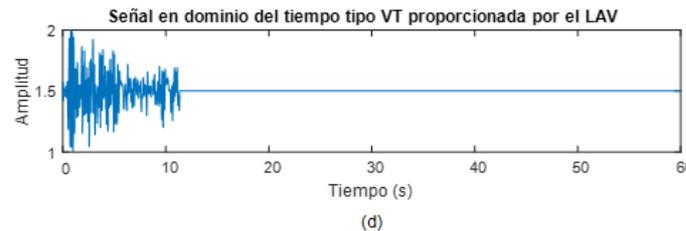
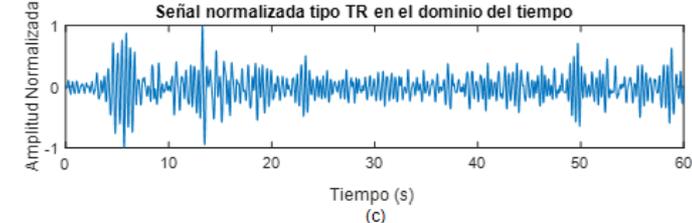
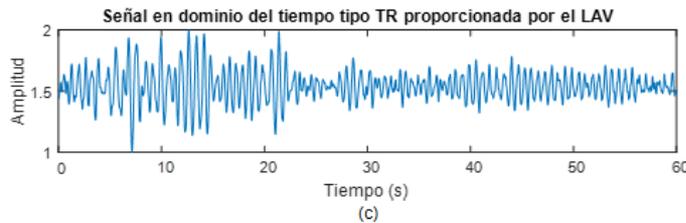
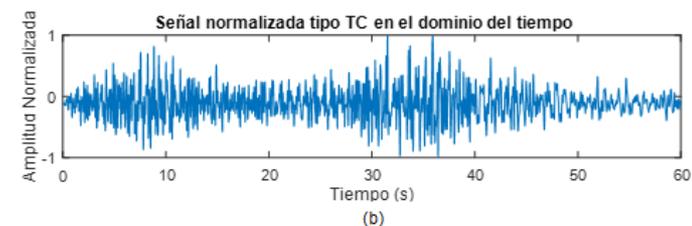
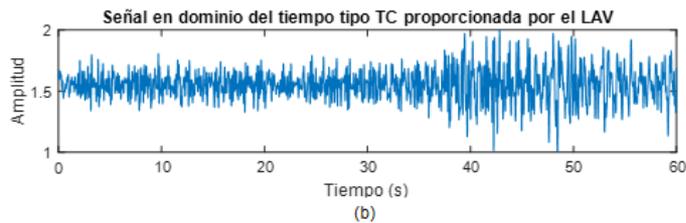
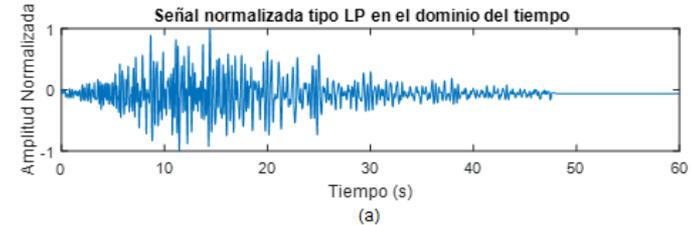
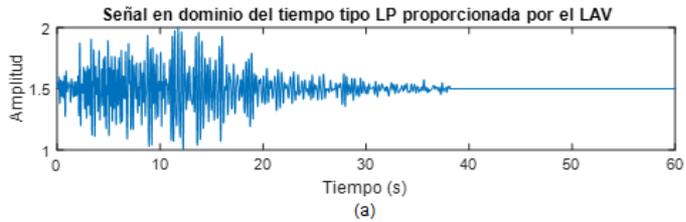
Preprocesamiento



2. Desarrollo

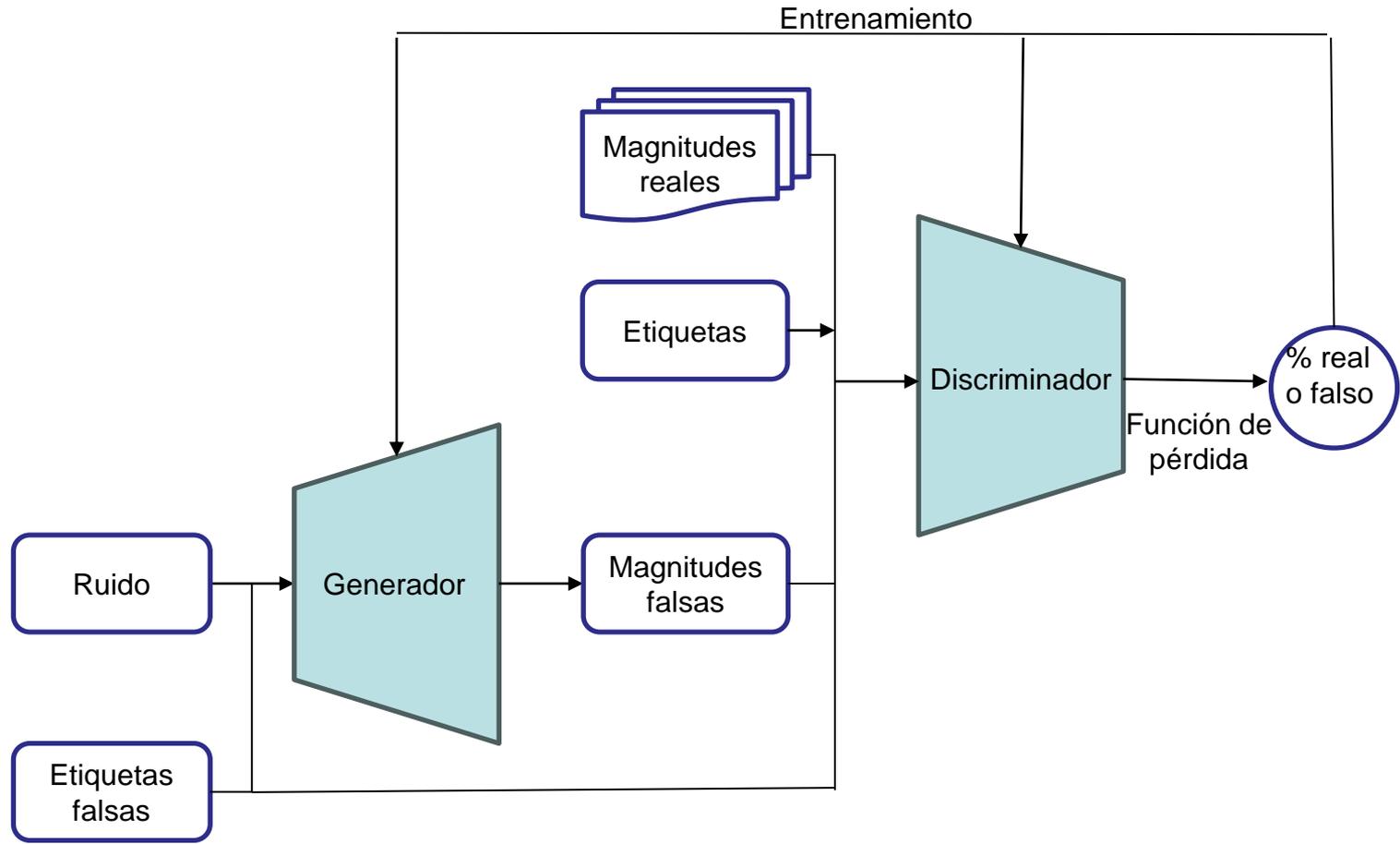
Preprocesamiento

Señales originales en el dominio del tiempo Señales normalizadas en el dominio del tiempo



2. Desarrollo

Red Generativa Antagónica Condicional (CGAN)



2. Desarrollo

Red Generativa Antagónica Condicional (CGAN)

Generador

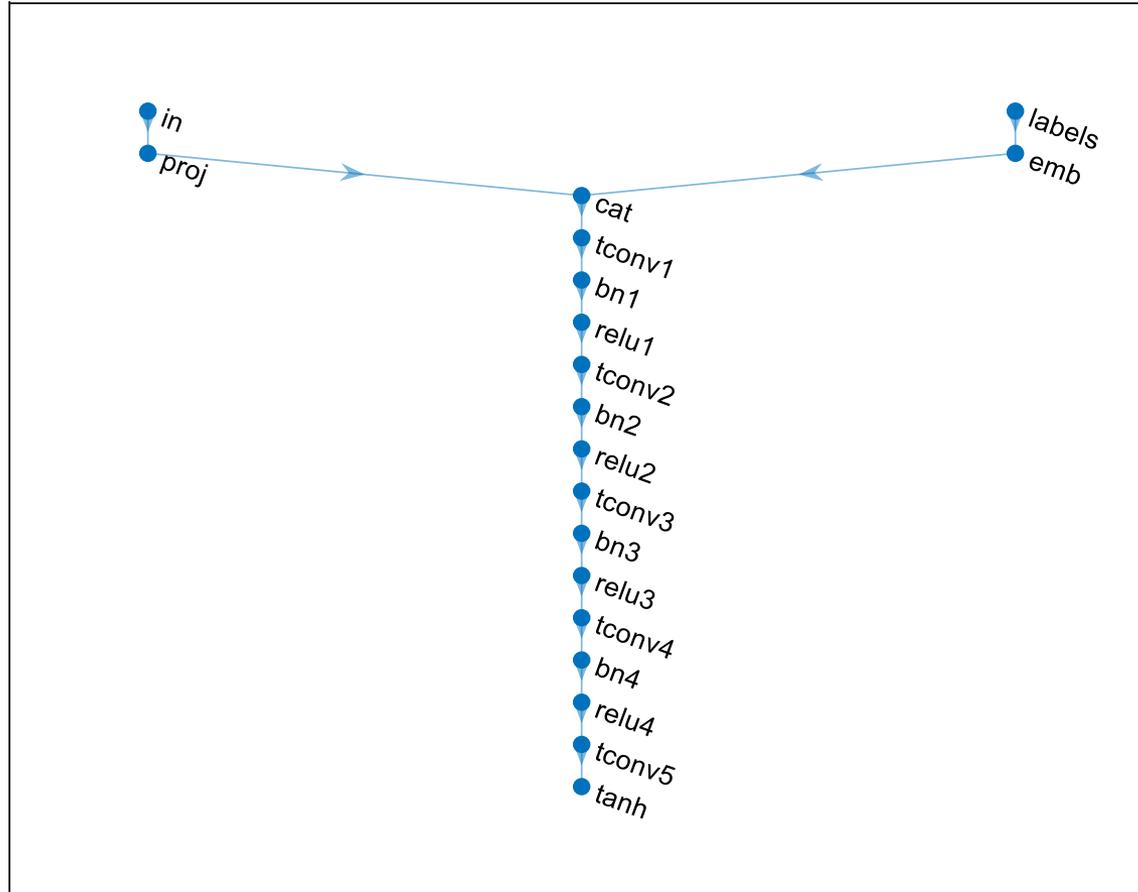
Operación	Kernel Size	Salida
Entrada z	100	
projectandReshape	100	(4x1x1024)
Etiquetas	(1x1x1)	
Concat (proj + etiquetas)	(4x1x1024) + (1x1x1)	(4x1x1025)
Trans Conv2D (Stride=1) batchNormalization ReLU	(8xd, kernel_size=13)	(16x1x512)
Trans Conv2D (Stride=4) batchNormalization ReLU	(4xd, kernel_size=6)	(64x1x256)
Trans Conv2D (Stride=4) batchNormalization ReLU	(2xd, kernel_size=6)	(256x1x128)
Trans Conv2D (Stride=4) batchNormalization ReLU	(1xd, kernel_size=6)	(1024x1x64)
Trans Conv2D (Stride=4) batchNormalization Tanh	(1, kernel_size=6)	(4096x1x1)



2. Desarrollo

Red Generativa Antagónica Condicional (CGAN)

GENERADOR



2. Desarrollo

Red Generativa Antagónica Condicional (CGAN)

Discriminador

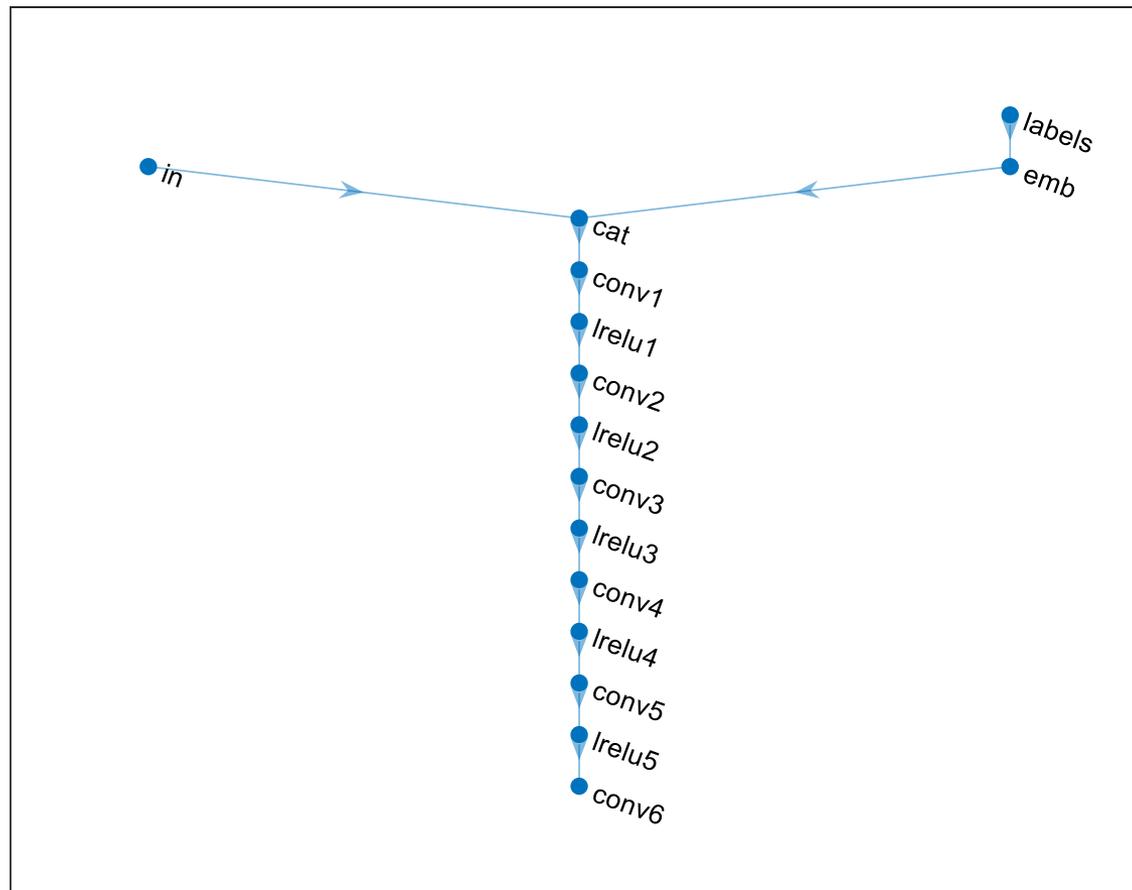
Operación	Kernel Size	Salida
Entrada x G(z) Etiquetas	(4096x1x1) (1x1x1)	(4096x1x2)
Conv1D (Stride=4) LReLU ($\alpha=0.2$)	(d,kernel_size=6)	(1024x1x64)
Conv1D (Stride=4) LReLU ($\alpha=0.2$)	(2xd,kernel_size=6)	(256x1x128)
Conv1D (Stride=4) LReLU ($\alpha=0.2$)	(4xd,kernel_size=6)	(64x1x256)
Conv1D (Stride=4) LReLU ($\alpha=0.2$)	(8xd,kernel_size=6)	(16x1x512)
Conv1D (Stride=4) LReLU ($\alpha=0.2$)	(16xd,kernel_size=6)	(4x1x1024)
Conv1D (Stride=1)	(1,kernel_size=4)	(1x1x1)



2. Desarrollo

Red Generativa Antagónica Condicional (CGAN)

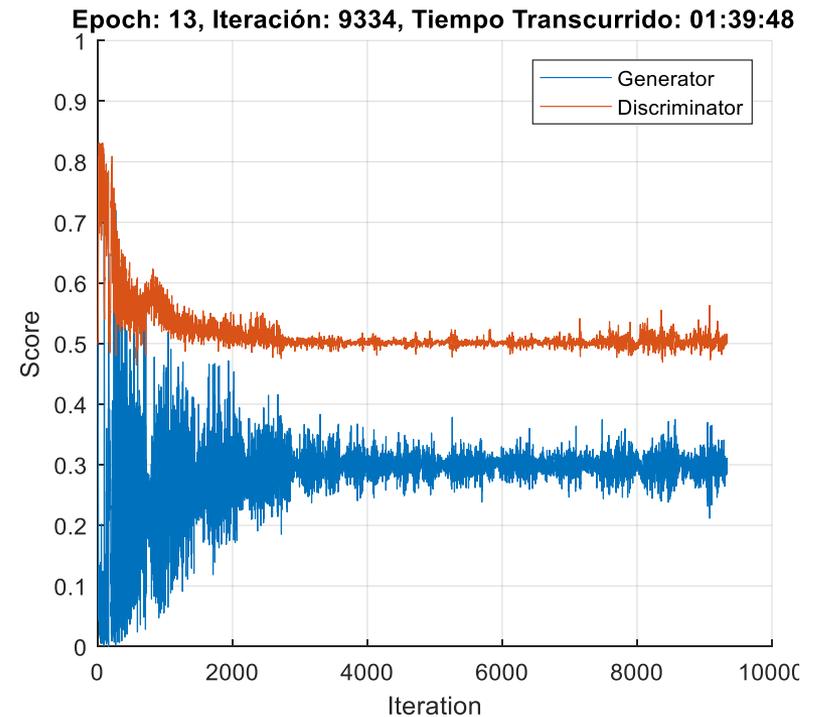
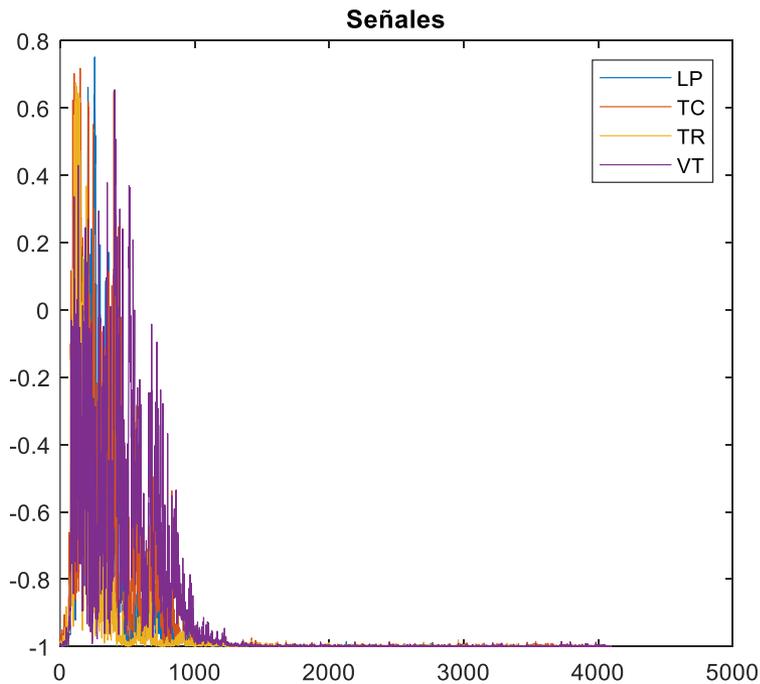
DISCRIMINADOR



2. Desarrollo

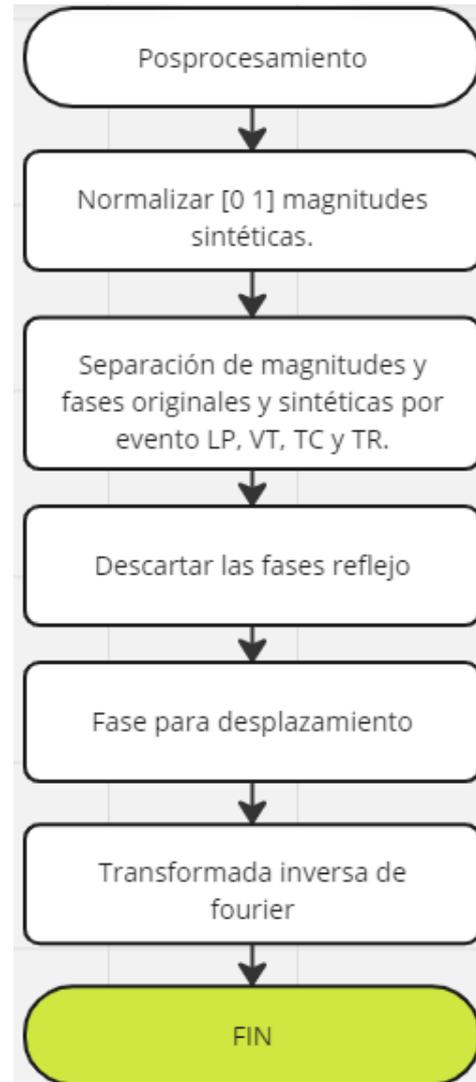
Entrenamiento

Monitorización de entrenamiento usando 13 épocas y 3 de minibatch size



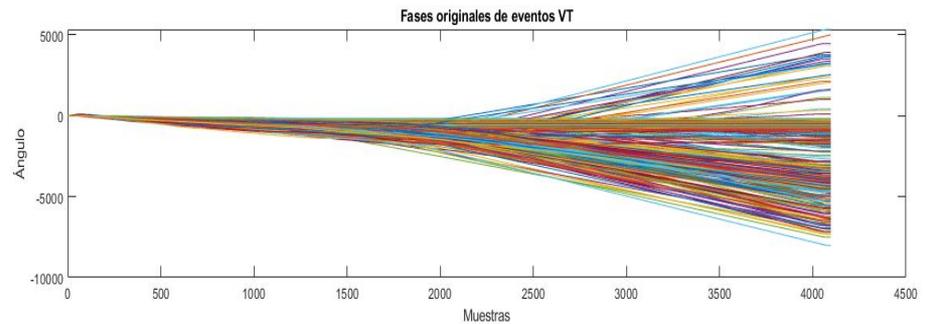
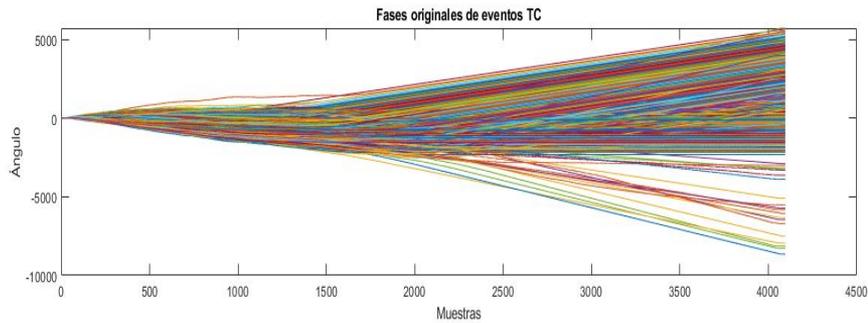
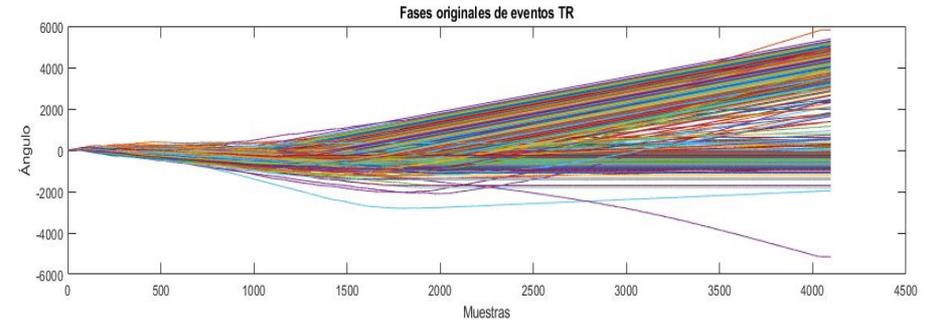
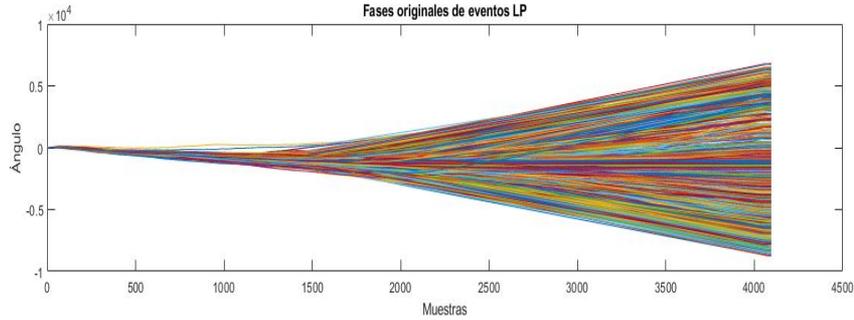
2. Desarrollo

Posprocesamiento



2. Desarrollo

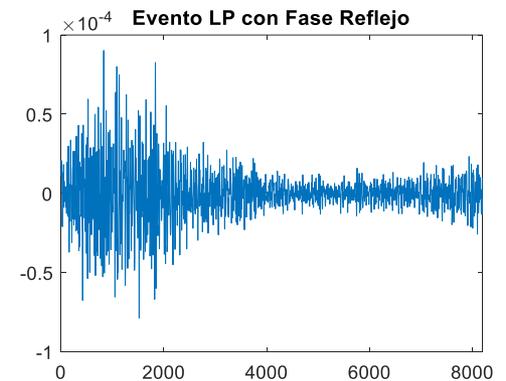
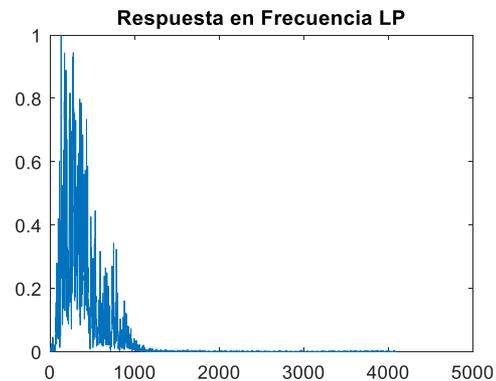
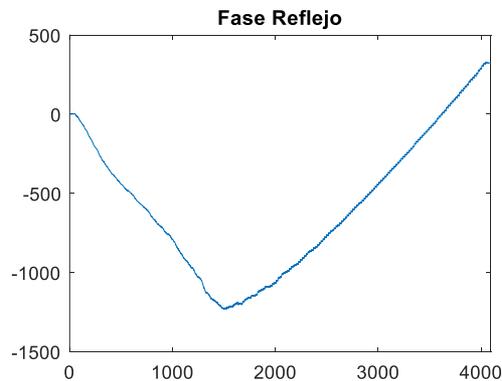
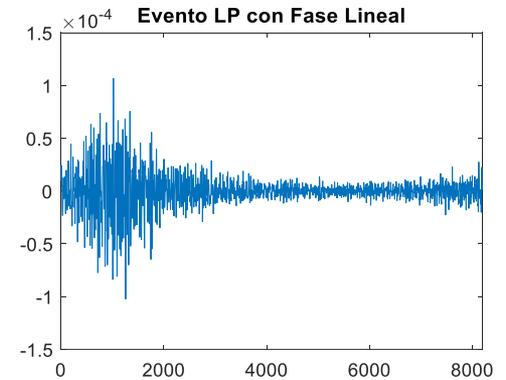
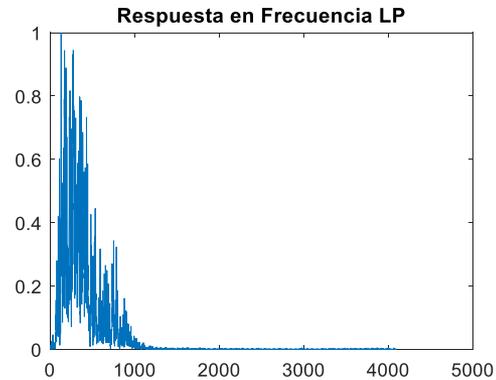
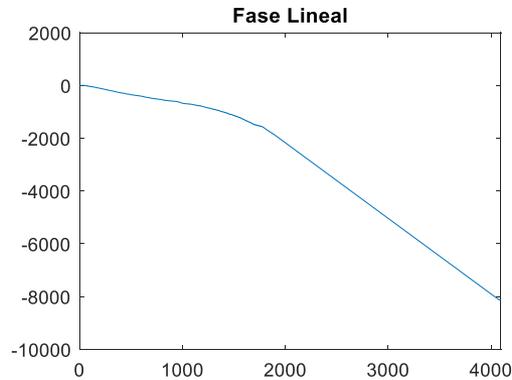
Posprocesamiento



2. Desarrollo

Posprocesamiento

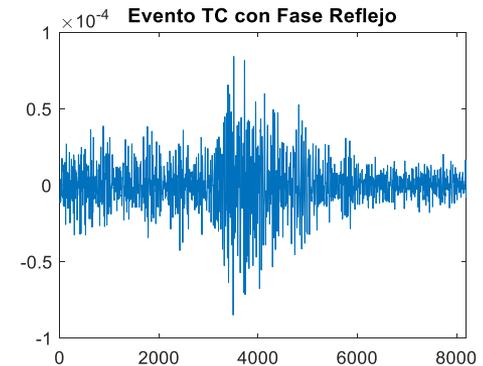
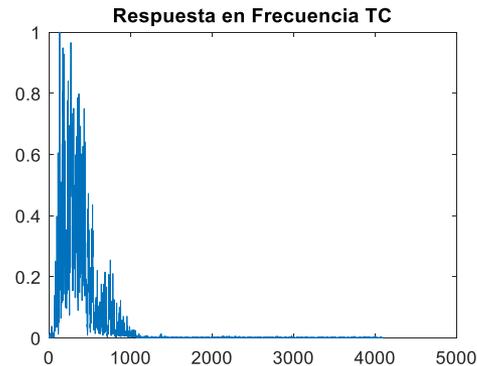
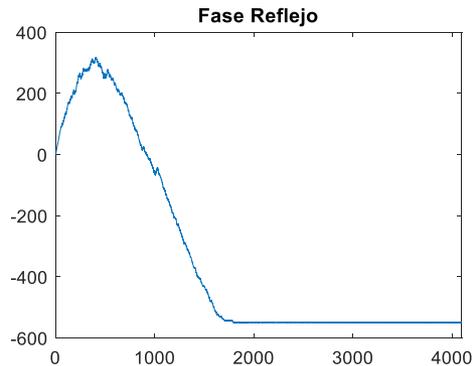
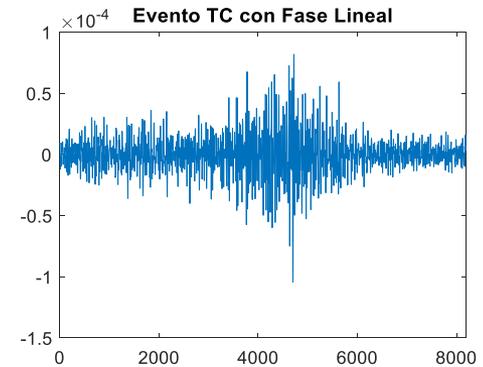
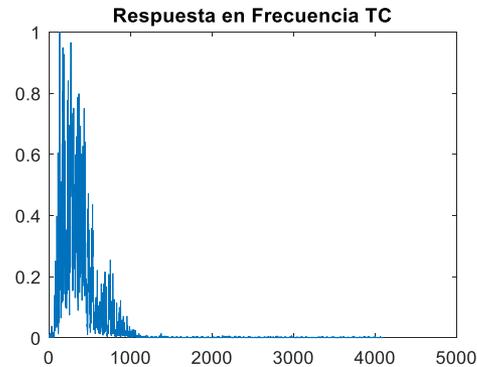
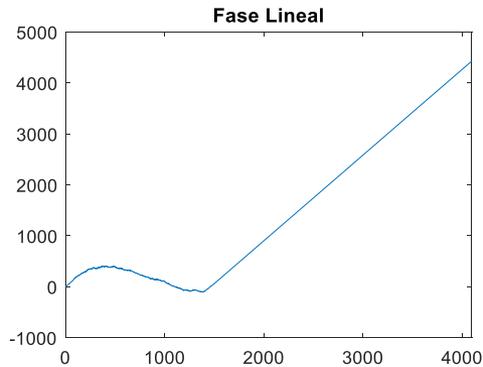
Ejemplo de criterio en señales LP para selección de fases



2. Desarrollo

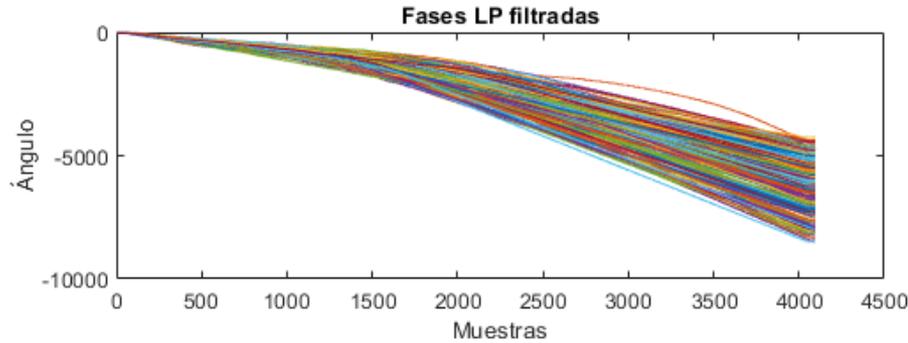
Posprocesamiento

Ejemplo de criterio en señales TC para selección de fases

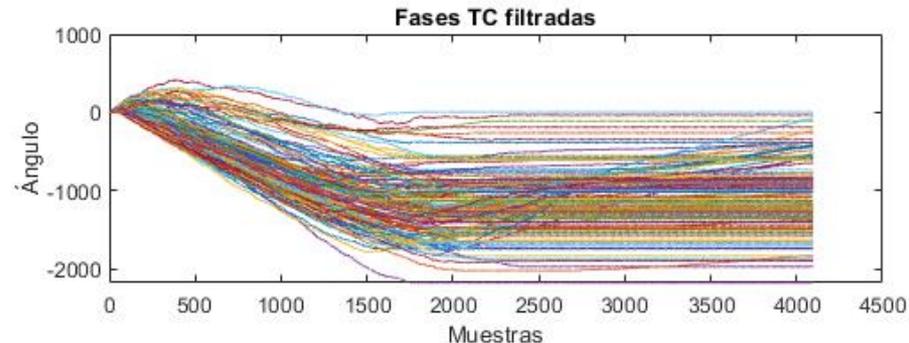


2. Desarrollo

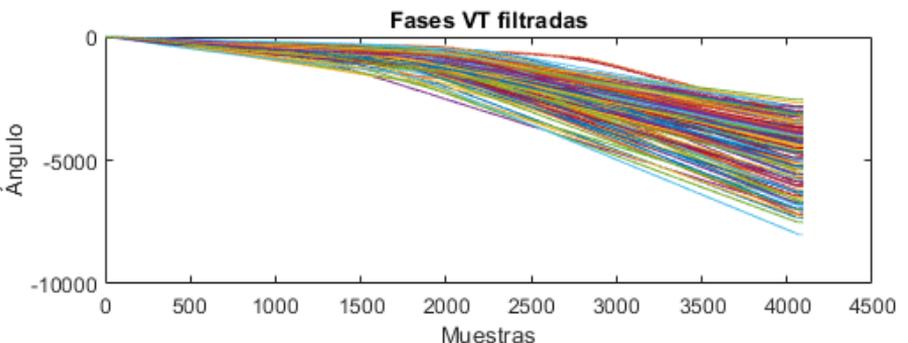
Posprocesamiento



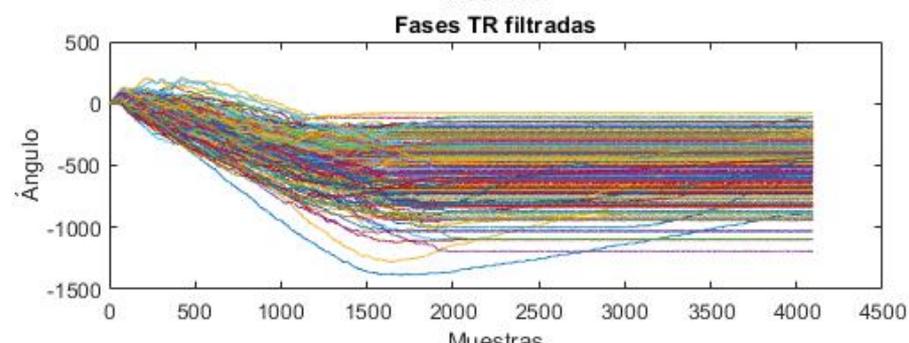
$$\overline{\overline{x}}_{LP} = \frac{\sum_{i=1}^{N_{LP}} X_{i_{LP}}}{N_{LP}} = \frac{\sum_{i=1}^{N_{LP}} X_{i_{LP}}}{1310} = -2723$$



$$\overline{\overline{x}}_{TC} = \frac{\sum_{i=1}^{N_{TC}} X_{i_{TC}}}{N_{TC}} = \frac{\sum_{i=1}^{N_{TC}} X_{i_{TC}}}{1488} = -1962$$



$$\overline{\overline{x}}_{VT} = \frac{\sum_{i=1}^{N_{VT}} X_{i_{VT}}}{N_{VT}} = \frac{\sum_{i=1}^{N_{VT}} X_{i_{VT}}}{304} = VT = -2455$$



$$\overline{\overline{x}}_{TR} = \frac{\sum_{i=1}^{N_{TR}} X_{i_{TR}}}{N_{TR}} = \frac{\sum_{i=1}^{N_{TR}} X_{i_{TR}}}{490} = -1787$$



2. Desarrollo

Métricas de Rendimiento

Matriz de confusión para cuatro clases

		Clase Predicha			
		1	2	3	4
Clase Real	1	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)
	2	Falso Negativo (FN)	Verdadero Positivo (VP)	Falso Negativo (FN)	Falso Negativo (FN)
	3	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)
	4	Verdadero Negativo (VN)	Falso Positivo (FP)	Verdadero Negativo (VN)	Verdadero Negativo (VN)

$$A(\%) = \frac{VP + VN}{VN + VP + FN + FP} \times 100$$

$$P(\%) = \frac{VP}{VP + FP} \times 100$$

$$R(\%) = \frac{VP}{VP + FN} \times 100$$

$$S(\%) = \frac{VN}{VN + FP} \times 100$$

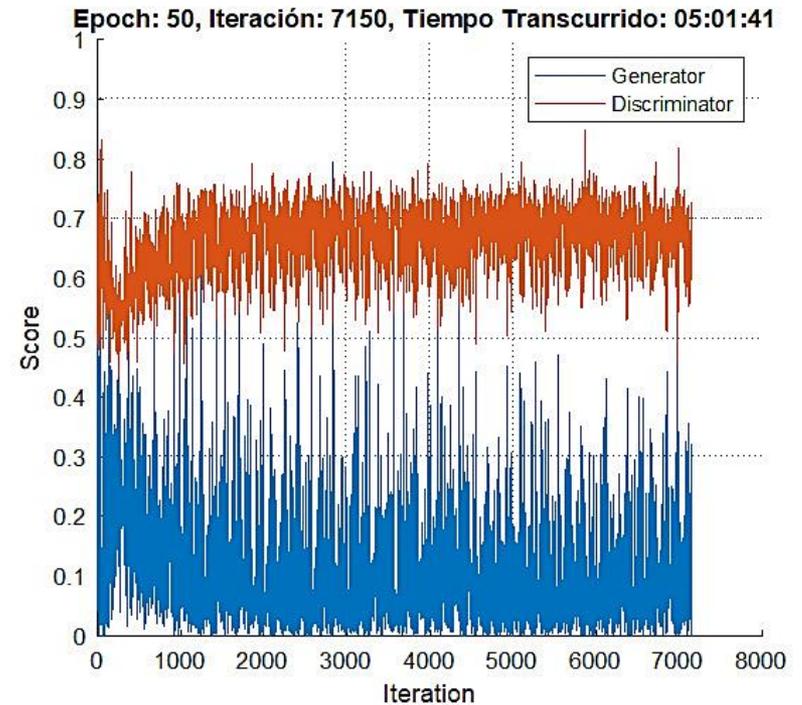
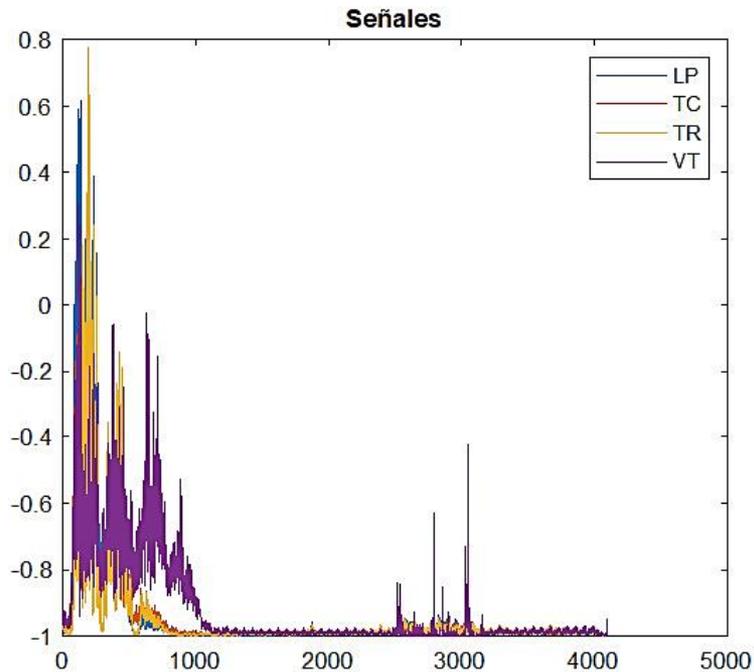
$$BER = 1 - \frac{R + S}{200}$$

3. Pruebas y Resultados

Pruebas

Modo colapso con exceso de filtros

- 128 filtros para el generador y para el discriminador.
- 50 epoch y 25 *minibatch*.

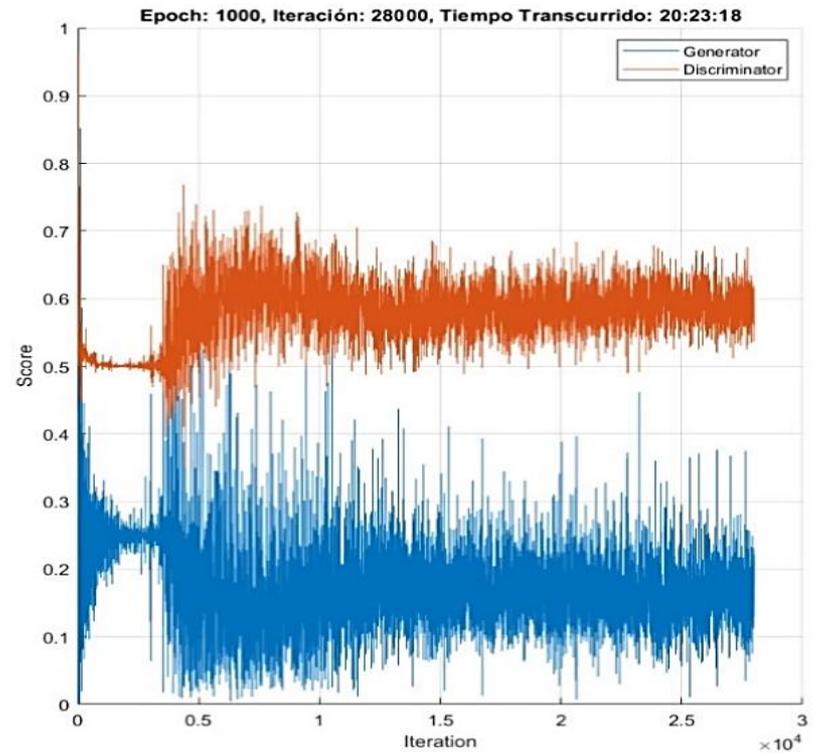
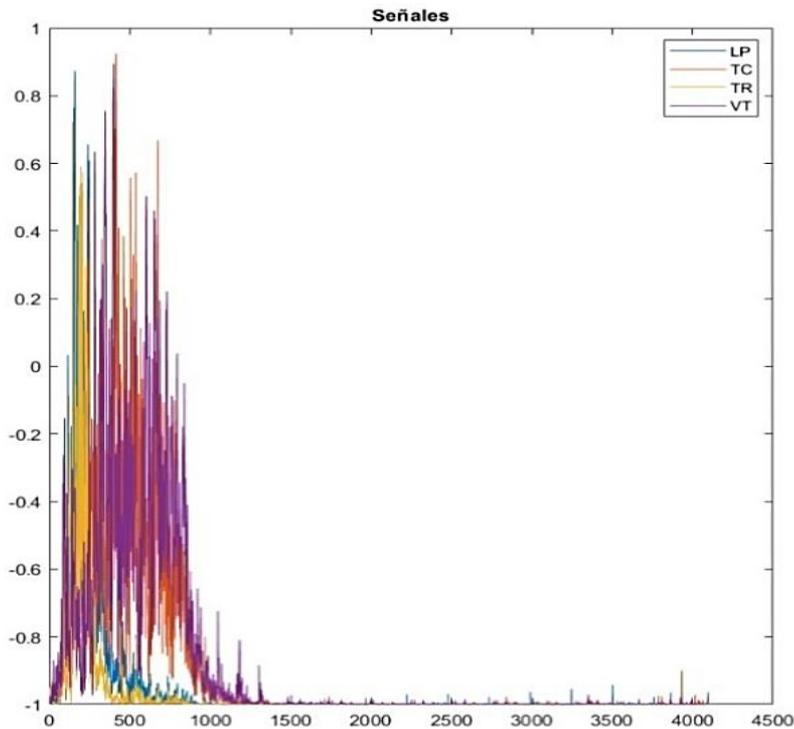


3. Pruebas y Resultados

Pruebas

Falla de convergencia con un epoch grande

- 1000 epoch y 128 *minibatch*.



3. Pruebas y Resultados

Pruebas

Parámetros variantes para pruebas en el modelo CGAN

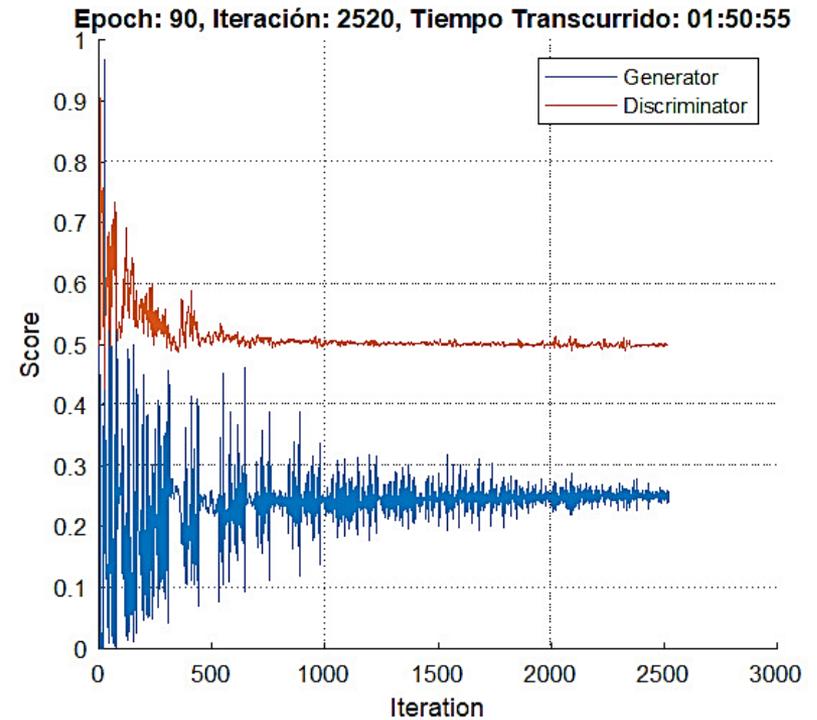
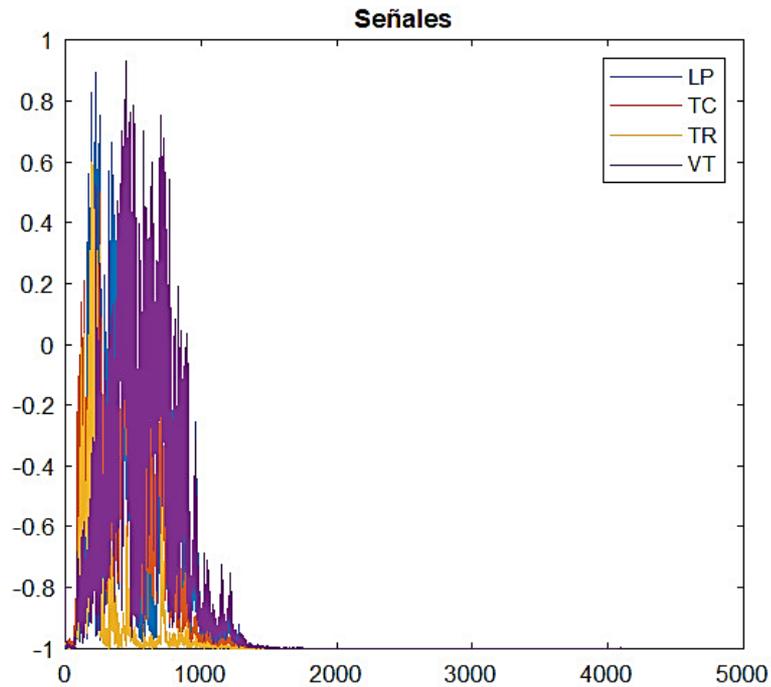
Epoch	Iteraciones	Minibatch	Puntaje Generador	Puntaje de discriminador
1000	28000	128	0.16	0.6
90	2520	128	0.25	0.5
90	21510	15	0.15	0.65

3. Pruebas y Resultados

Pruebas

Caso 1

- 90 epoch y 128 minibatch

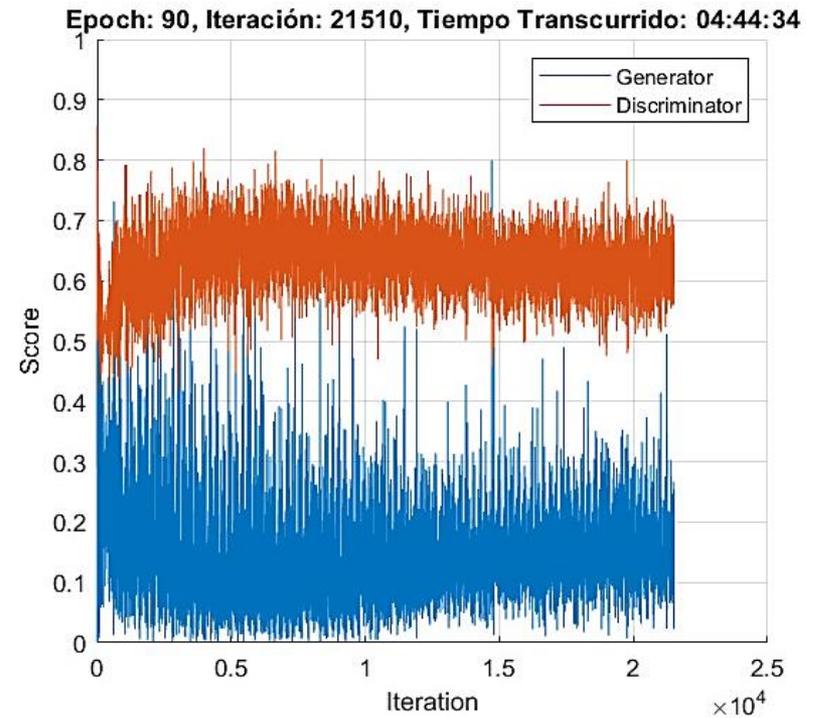
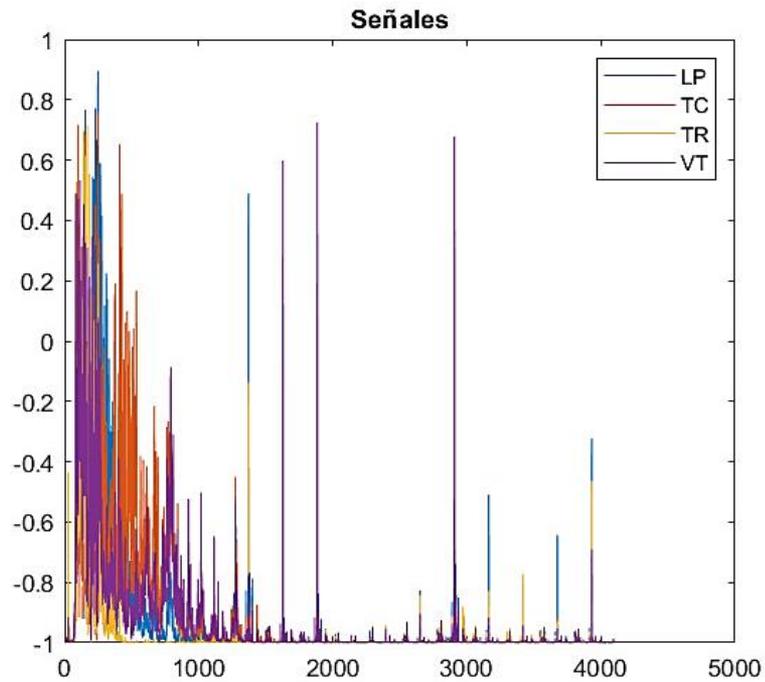


3. Pruebas y Resultados

Pruebas

Caso 2

- 90 Epoch, 15 minibatch

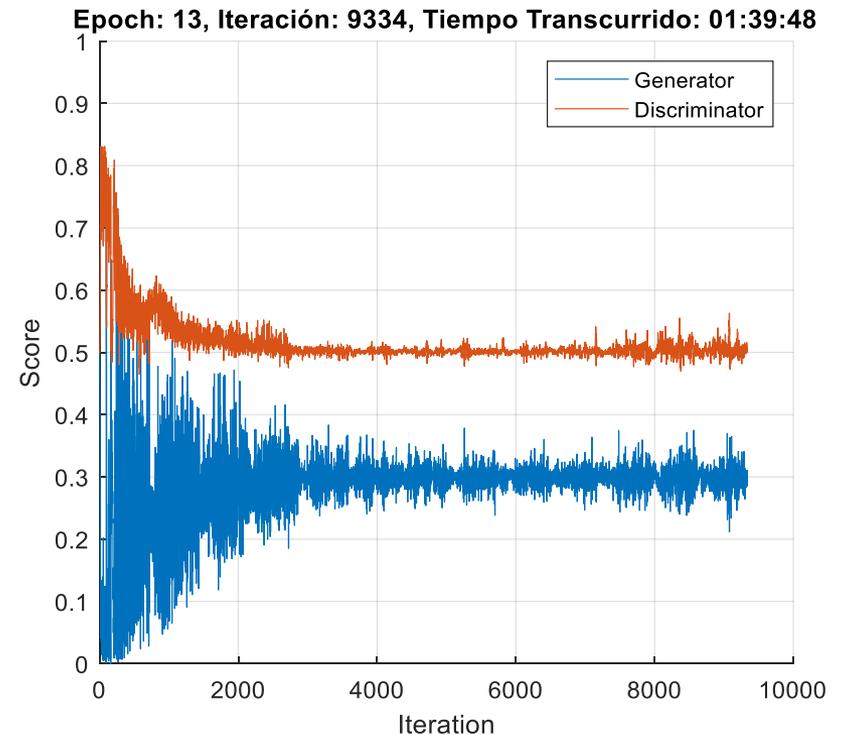
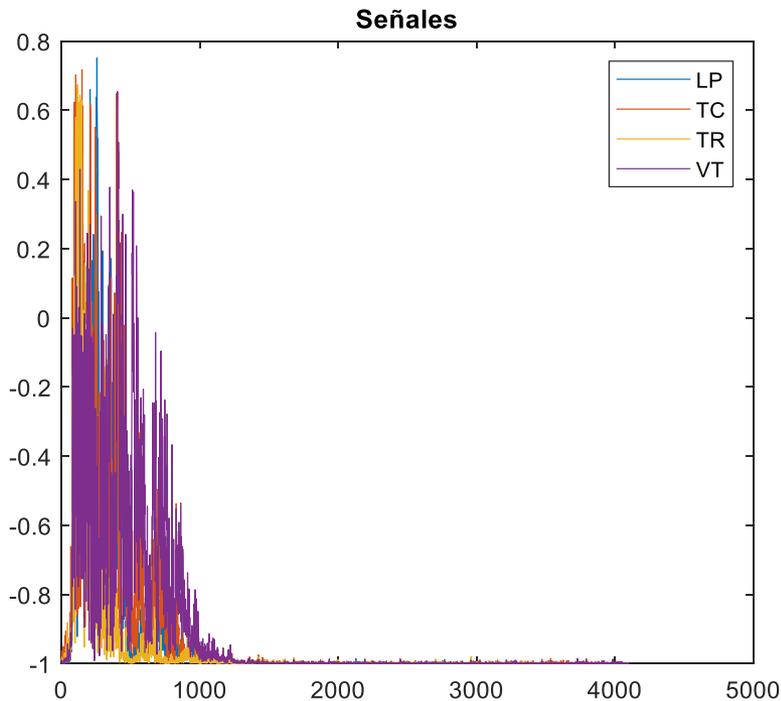


3. Pruebas y Resultados

Pruebas

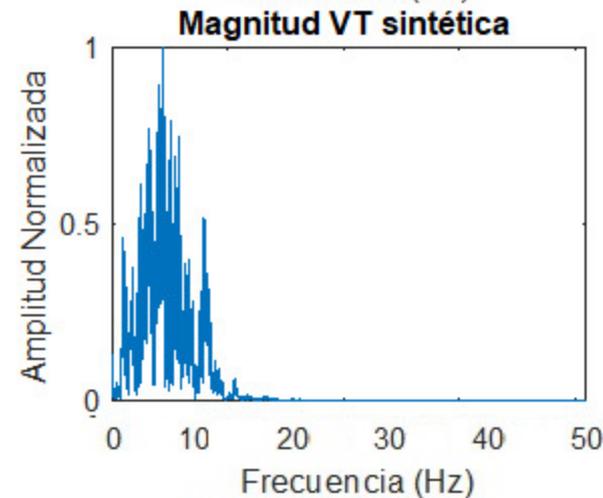
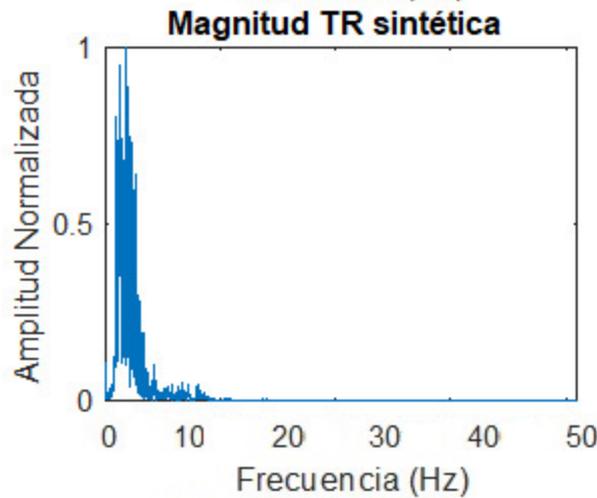
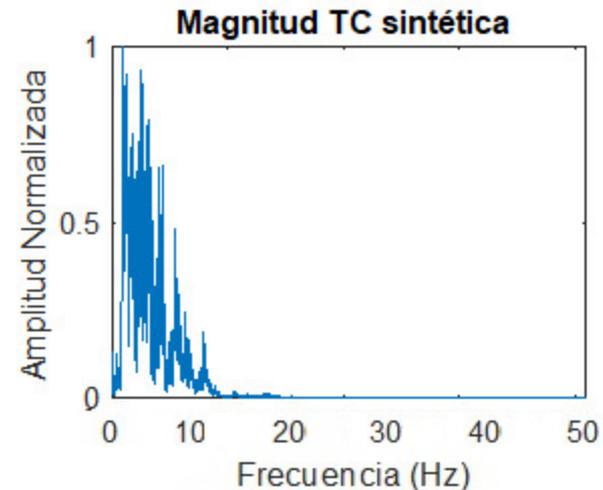
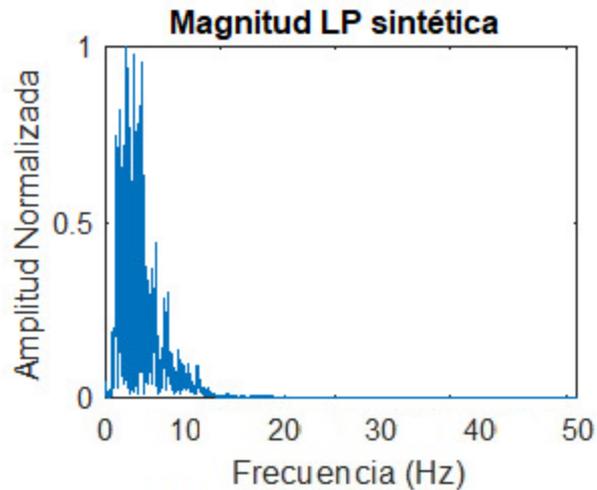
Entrenamiento adicional

- $\alpha = 0.001 \rightarrow$ tasa de aprendizaje, el término $\beta_1 = 0.5$ y $\beta_2 = 0.999$
- $\alpha = 0.001$ es remplazado por el valor de $\alpha = 0.0002$.
- 13 epoch y 5 de minibatch size



3. Pruebas y Resultados

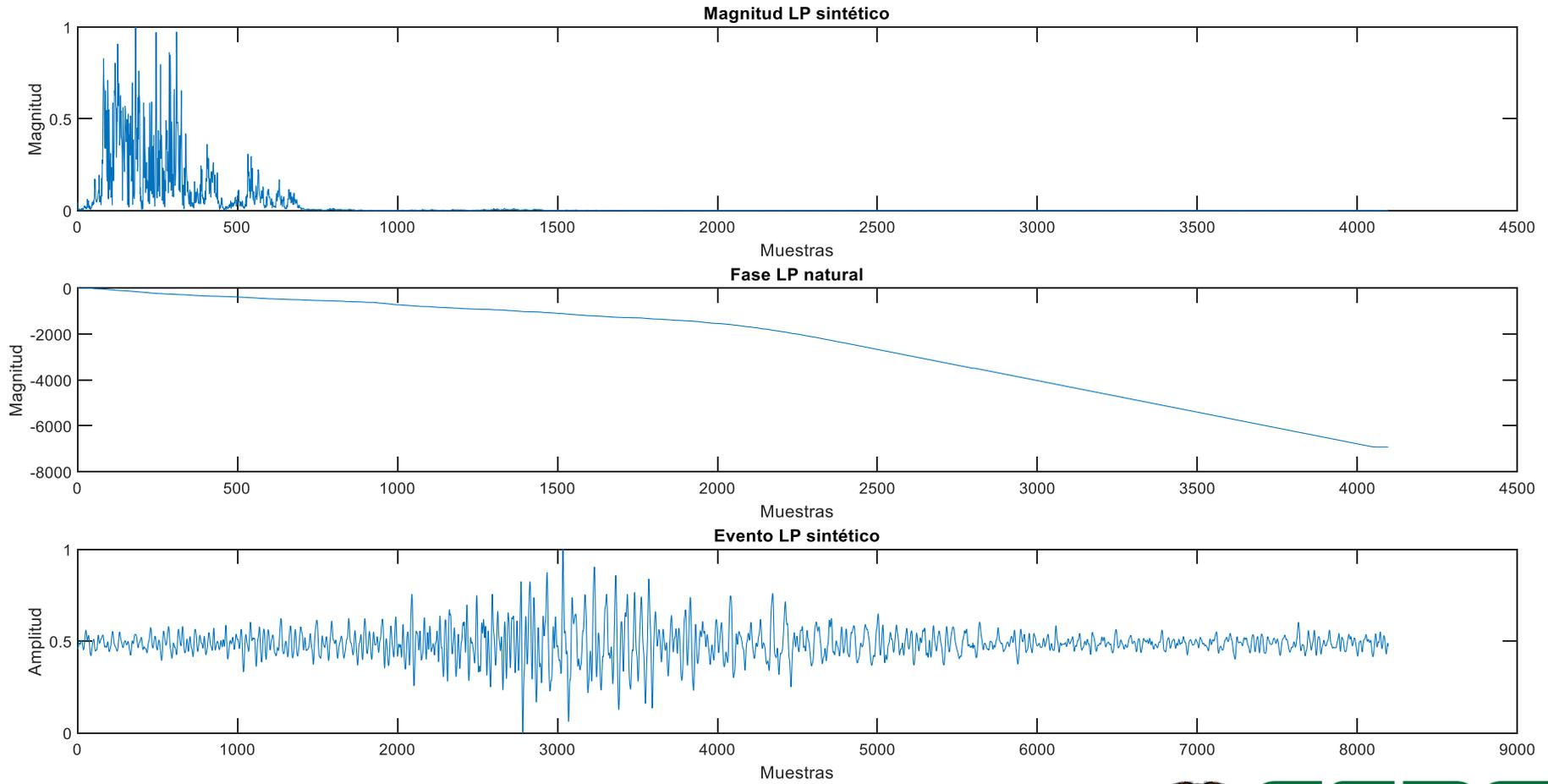
Resultados



3. Pruebas y Resultados

Resultados

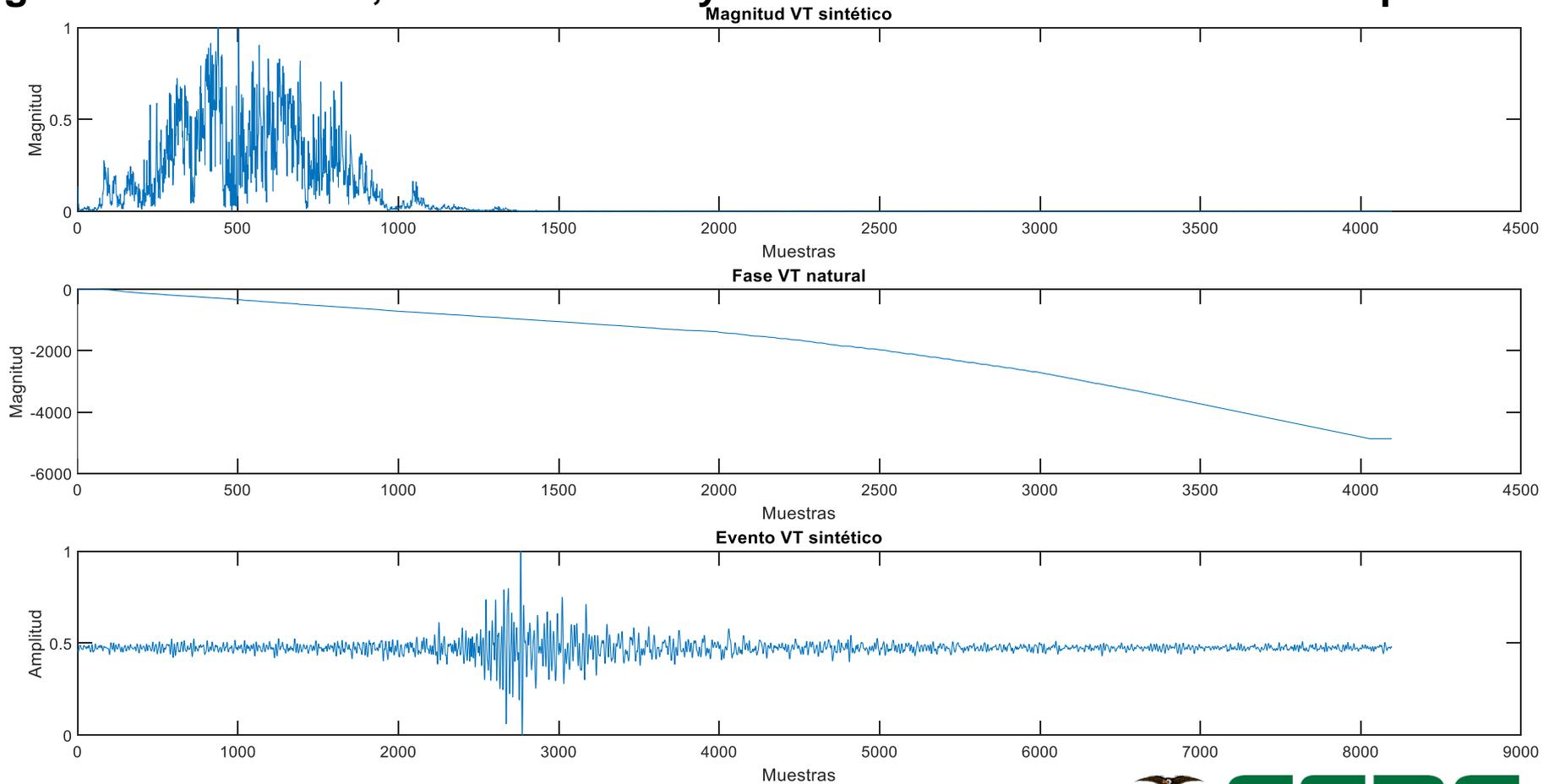
Magnitud LP sintético, fase LP natural y señal sintética microsísmica LP



3. Pruebas y Resultados

Resultados

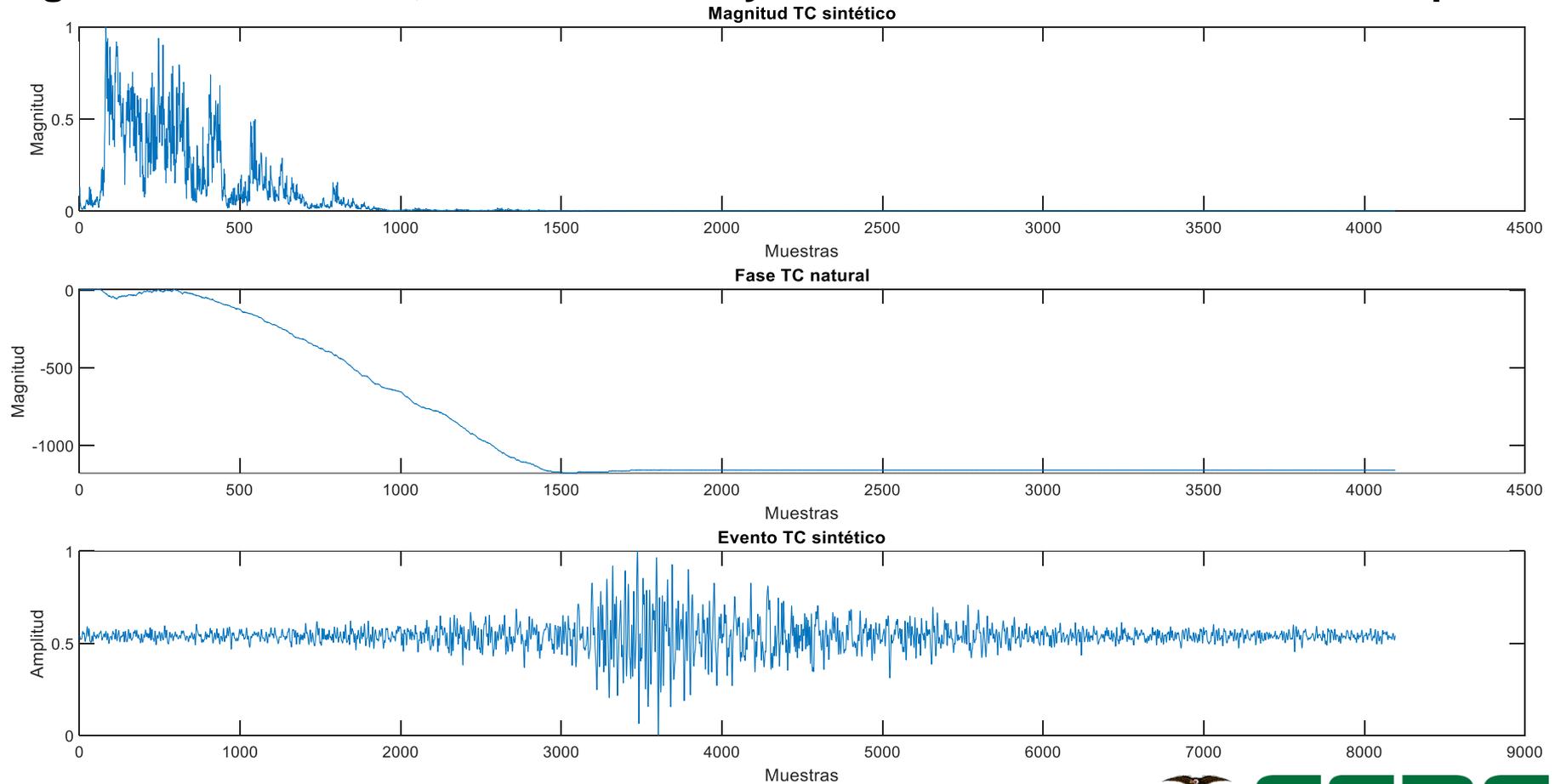
Magnitud VT sintético, fase VT natural y señal sintética microsísmica de tipo VT



3. Pruebas y Resultados

Resultados

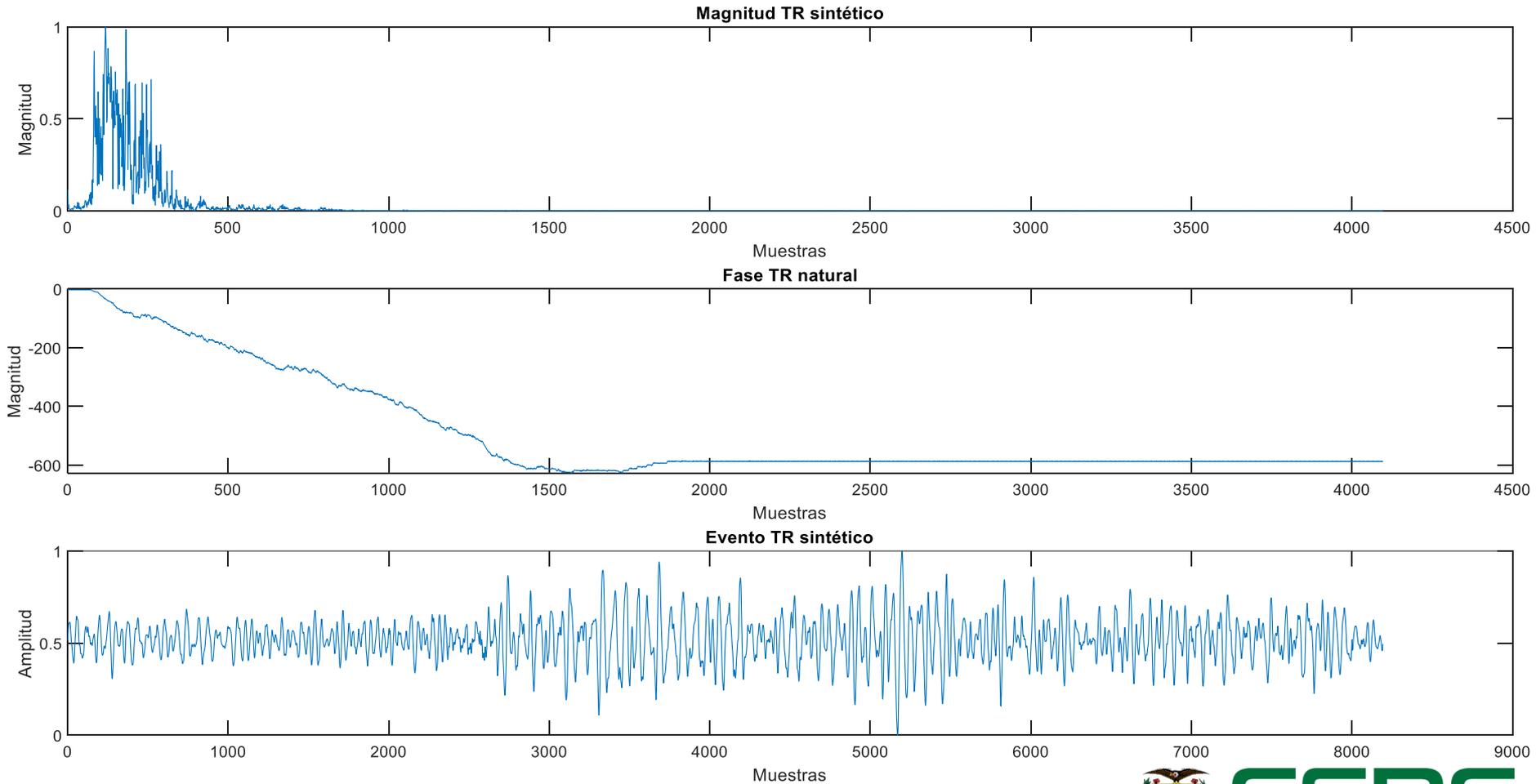
Magnitud TC sintético, fase TC natural y señal sintética microsísmica de tipo TC



3. Pruebas y Resultados

Resultados

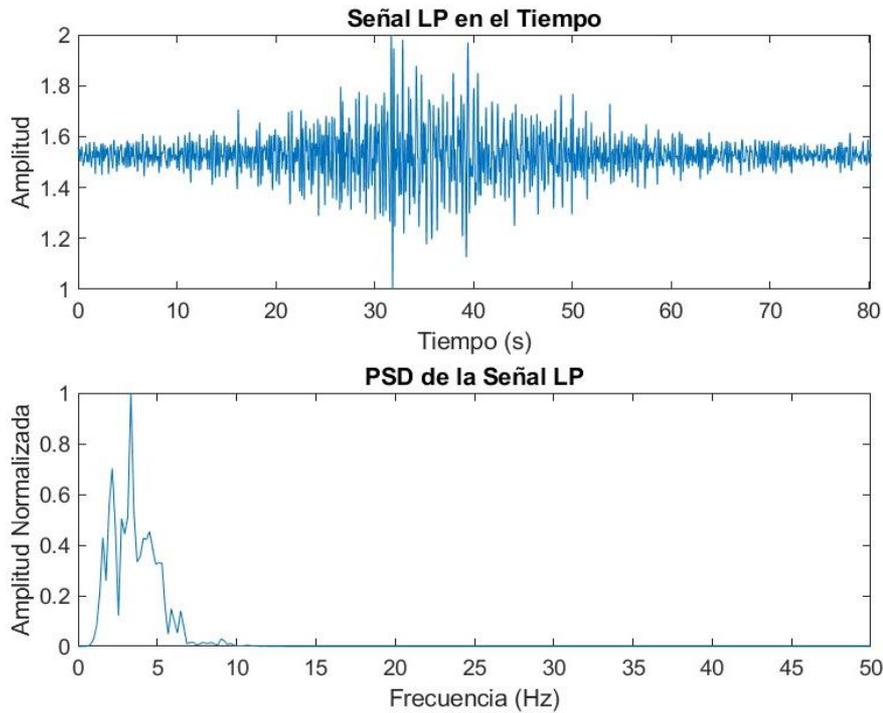
Magnitud TR sintético, fase TR natural y señal sintética microsísmica de tipo TR



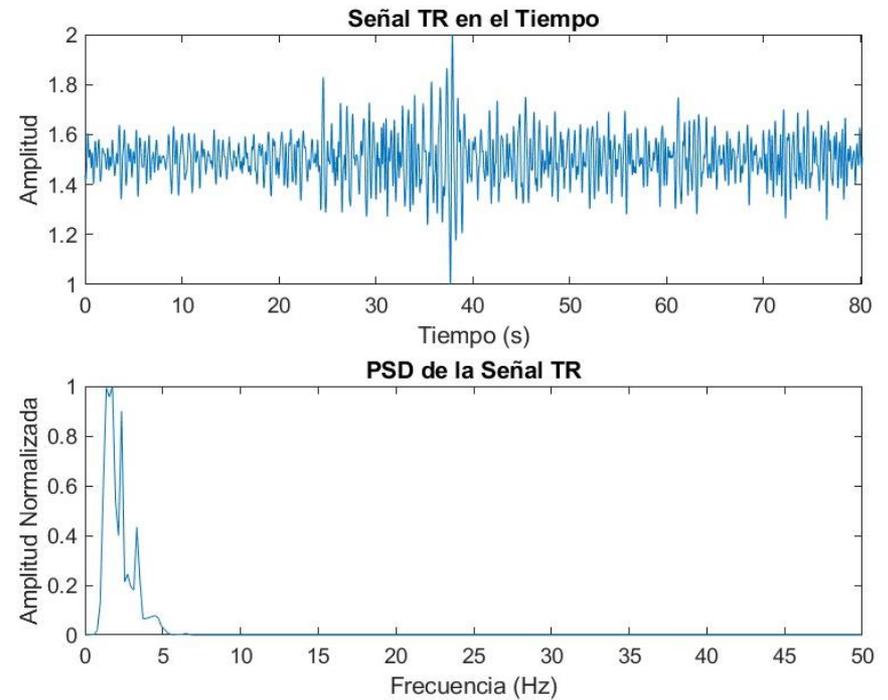
3. Pruebas y Resultados

Resultados

Señal sintética LP y su PSD



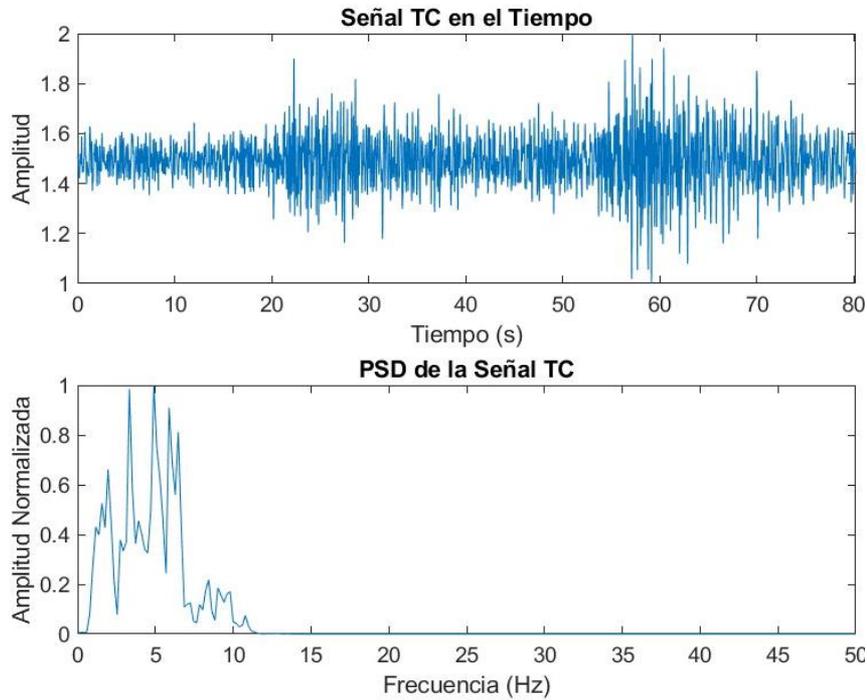
Señal sintética TR y su PSD



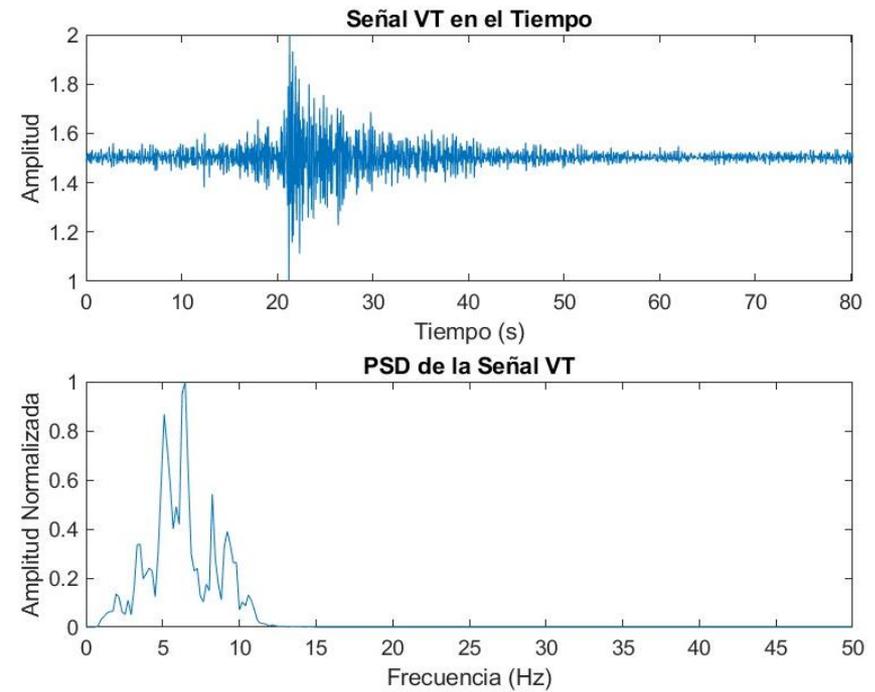
3. Pruebas y Resultados

Resultados

Señal sintética TC y su PSD



Señal sintética VT y su PSD



3. Pruebas y Resultados Evaluación

Primer autoencoder

Confusion Matrix

Output Class	1	2	3	4	
1	2067 20.7%	207 2.1%	115 1.1%	37 0.4%	85.2% 14.8%
2	403 4.0%	2234 22.3%	5 0.1%	1043 10.4%	60.6% 39.4%
3	8 0.01%	3 0.0%	2380 23.08%	0 0.0%	99.5% 0.5%
4	22 0.2%	56 0.6%	0 0.0%	1420 14.2%	94.8% 5.2%
	82.7% 17.3%	89.4% 10.6%	95.2% 4.8%	56.8% 43.2%	81.0% 19.0%
	1	2	3	4	
	Target Class				

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	92.08	85.20	82.68	94.28	0.10
TC	82.83	60.62	89.36	95.79	0.22
TR	98.69	99.54	95.20	98.42	0.01
VT	88.42	94.79	56.80	87.30	0.09
General	81.00	85.00	81.00	94.00	0.11

- Las señales LP exhiben una tendencia a ser confundidas con las señales TC.
- Las señales TR tienden a confundirse con las señales LP.
- Las señales VT muestran una confusión predominante con las señales TC.



3. Pruebas y Resultados Evaluación

Segundo autoencoder

Confusion Matrix

Output Class	1	2	3	4	
1	2499 25.0%	1781 17.8%	24 0.2%	0 0.0%	58.1% 41.9%
2	0 0.0%	621 6.2%	54 0.5%	29 0.3%	88.2% 11.8%
3	1 0.0%	0 0.0%	2422 24.2%	0 0.0%	100.0% 0.0%
4	0 0.0%	98 1.0%	0 0.0%	2471 24.7%	96.2% 3.8%
	100.0% 0.0%	24.8% 75.2%	96.9% 3.1%	98.8% 1.2%	80.1% 19.9%
	1	2	3	4	
	Target Class				

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	81.94	58.06	99.96	99.98	0.21
TC	80.38	88.21	94.84	79.79	0.16
TR	99.21	99.96	96.88	98.97	0.01
VT	98.73	96.18	98.84	99.61	0.02
General	80.10	85.60	80.13	94.59	0.10

- La confusión principal se produce al clasificar TC que en su mayoría se clasifican como LP.



3. Pruebas y Resultados Evaluación

DT

Confusion Matrix

LP	2483 24.8%	67 0.7%	81 0.8%	2 0.0%	94.3% 5.7%
TC	7 0.1%	221 2.2%	2362 23.6%	14 0.1%	8.5% 91.5%
TR	1 0.0%	0 0.0%	0 0.0%	2484 24.8%	0.0% 100%
VT	9 0.1%	2212 22.1%	57 0.6%	0 0.0%	0.0% 100%
	99.3% 0.7%	8.8% 91.2%	0.0% 100%	0.0% 100%	27.0% 73.0%
	LP	TC	TR	VT	

Target Class

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.33	94.30	99.32	99.77	0.03
TC	53.38	8.49	8.84	69.19	0.61
TR	50.15	0.00	0.00	66.73	0.67
VT	52.22	0.00	0.00	67.63	0.66
General	27.00	25.69	27.04	75.83	0.49



3. Pruebas y Resultados Evaluación

k-NN

Confusion Matrix

LP	2498 25.0%	0 0.0%	158 1.6%	12 0.1%	93.6% 6.4%
TC	0 0.0%	15 0.1%	2315 23.2%	3 0.0%	0.6% 99.4%
TR	0 0.0%	0 0.0%	0 0.0%	2485 24.9%	0.0% 100%
VT	2 0.0%	2485 24.9%	27 0.3%	0 0.0%	0.0% 100%
	99.9% 0.1%	0.6% 99.4%	0.0% 100%	0.0% 100%	25.1% 74.9%
	LP	TC	TR	VT	

Target Class

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.28	93.63	99.92	99.97	0.03
TC	51.97	0.64	0.60	67.59	0.65
TR	50.15	0.00	0.00	0.67	0.67
VT	49.86	0.00	0.00	0.67	0.67
General	25.10	23.57	25.13	42.22	0.51

3. Pruebas y Resultados Evaluación

SVM

Restricción de caja: 6

Confusion Matrix

	LP	TC	TR	VT	
LP	2498 25.0%	0 0.0%	130 1.3%	3 0.0%	94.9% 5.1%
TC	0 0.0%	0 0.0%	2336 23.4%	2 0.0%	0.0% 100%
TR	0 0.0%	0 0.0%	0 0.0%	2495 24.9%	0.0% 100%
VT	2 0.0%	2500 25.0%	34 0.3%	0 0.0%	0.0% 100%
	99.9% 0.1%	0.0% 100%	0.0% 100%	0.0% 100%	25.0% 75.0%
	LP	TC	TR	VT	

Target Class

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.65	94.94	99.92	99.97	0.03
TC	51.62	0.00	0.00	67.37	0.67
TR	50.05	0.00	0.00	66.68	0.67
VT	49.64	0.00	0.00	66.51	0.67
General	25.00	23.74	24.98	75.13	0.51



3. Pruebas y Resultados Evaluación

Votación

Confusion Matrix

	LP	TC	TR	VT	
LP	2498 25.0%	0 0.0%	122 1.2%	5 0.1%	95.2% 4.8%
TC	0 0.0%	221 2.2%	2334 23.3%	15 0.1%	8.6% 91.4%
TR	0 0.0%	0 0.0%	0 0.0%	2480 24.8%	0.0% 100%
VT	2 0.0%	2279 22.8%	44 0.4%	0 0.0%	0.0% 100%
	99.9% 0.1%	8.8% 91.2%	0.0% 100%	0.0% 100%	27.2% 72.8%
	LP	TC	TR	VT	

Target Class

Evento	A (%)	R (%)	P (%)	S (%)	BER
LP	98.71	95.16	99.92	99.97	0.02
TC	53.72	8.60	8.84	69.33	0.61
TR	50.20	0.00	0.00	66.76	0.67
VT	51.75	0.00	0.00	67.43	0.67
General	27.20	25.94	27.19	75.87	0.49

- Las señales TC, se clasifican erróneamente como VT.
- Para las señales TR, en su totalidad se clasifican erróneamente como señales TC.
- Para señales VT, en su totalidad se clasifican como TR.



3. Pruebas y Resultados Evaluación

Resultados de las herramientas empleadas

Herramienta	A (%)	R (%)	P (%)	S (%)	BER
Primer Autoencoder	81.00	85.00	81.00	94.00	0.11
Segundo Autoencoder	80.10	85.60	80.13	94.59	0.10
Sistema de clasificación DT	27.00	25.69	27.04	75.83	0.49
Sistema de clasificación k-NN	25.10	23.57	25.13	42.22	0.51
Sistema de clasificación SVM	25.00	23.74	24.98	75.13	0.51
Sistema de clasificación Votación	27.20	25.94	27.19	75.87	0.49



4. Conclusiones

- **Para este trabajo** de integración curricular, se utilizó la **base de datos** que fue **adaptada** para que pueda trabajarse en el software de **MATLAB®**. La base de datos usada se denomina **LlaimaSigV1** que fue recolectado de la estación LAV que monitorea el **volcán Llaima**, y los eventos etiquetados por especialistas del OVDAS. Contiene un total de **3592 eventos de cuatro clases** diferentes de las cuales se tiene 304 VT, 1310 LP, 490 TR y 1488 TC. Las señales con las que se trabajó son todas las disponibles en la base de datos. Lo que se hizo de primera instancia es **filtrar** todas las señales de los eventos. Esto se hace debido al **choque de las olas del mar contra la superficie terrestre**, esto da como consecuencia que las señales presentan un componente frecuencial en **0.2 Hz**. Esta característica conlleva al enmascaramiento de la señal real. Por lo tanto, se implementó un filtro FIR pasa banda. Seguido de esto, cada señal se **estandarizó** y se aplicó **normalización de tipo máximo**, se obtuvo la respuesta en frecuencia por medio del uso de la **FFT** y finalmente se **normalizó entre -1 y 1**.



4. Conclusiones

- En la etapa en la que se diseñó la **CGAN multiclase**, se encontró que las redes componentes, es decir, tanto el **generador como el discriminador**, requerían de parámetros específicos. Estos parámetros incluyeron determinar el **número de filtros**, la profundidad de las redes en función del **número de capas** utilizadas, el tamaño del **kernel**, el **stride** y la elección de la **función de activación**, la cual está intrínsecamente relacionada con la naturaleza de cada capa. Al afinar estos parámetros se logró obtener un **modelo final**, el cual tuvo como objetivo la **generación de señales sintéticas de magnitudes frecuenciales** de los cuatro diferentes tipos de eventos. La variación de distintos parámetros y sus respectivos resultados obtenidos fueron **monitoreados** gracias a la ayuda de herramientas diseñadas para este fin.



4. Conclusiones

- En la **etapa de entrenamiento**, con la ayuda de la monitorización, se comprobó que, al **utilizar excesiva cantidad de filtros**, el **trabajo computacional** es demasiado e incluso inmanejable para las limitaciones de las máquinas. Además, provoca un **error de convergencia**, es decir, la red no converge, por tanto, **se fijó el número de filtros** en un valor intermedio. También se probó el entrenamiento con un **número de épocas significativo**, en el cual se da otro error conocido como **sobre entrenamiento**. Hasta cierto punto el aprendizaje es correcto hasta que la convergencia se dispara hasta otros valores. Con **este precedente** se determinó que las **épocas** deberían ser menor a 100, por tanto, se toma **90 épocas** y con **esto se varía el valor de *minibatch***, se determinó que el mejor es **128**. Finalmente, se llevó a cabo un **entrenamiento adicional** que incorporaba una variación en uno de los parámetros del **optimizador de Adam**.

4. Conclusiones

Como resultado de este **ajuste**, la red no requirió de valores elevados de **épocas** ni de **minibatch size**; en este caso, bastaron **13 y 5** respectivamente. El resultado final fue que el uso de **recursos** computacionales **disminuyó** y, además, se logró una **convergencia más eficiente**.

4. Conclusiones

- Con la red CGAN entrenada, se creó una **aplicación** por medio del uso de **App Designer** de MATLAB® 2022. La aplicación permite la **generación de señales sintéticas etiquetadas de los cuatro tipos de microsismos**. En la aplicación se puede ingresar la **cantidad de señales deseadas**, las cuales para cada evento se generan **1/4** del total. **La tabla** obtenida está en .mat, en donde se mantuvo el formato de la base de datos original que **contiene: procedencia de la señal, la estación base que la monitorea, la frecuencia de muestreo, el componente, el año y el mes, la etiqueta del evento, la duración del evento, el punto de inicio y fin del evento y finalmente la señal sintética temporal**. La aplicación también proporciona un **gráfico** de cada uno de los tipos de señales en el **tiempo** con su **PSD** respectiva.



4. Conclusiones

- De las herramientas para analizar las señales se obtuvo las exactitudes de la clasificación para los eventos LP, TC, TR y VT, con la utilización del **primer Autoencoder** el valor de **81%** en exactitud, **segundo Autoencoder** con **80.10%** en exactitud. Para el **sistema de clasificación** por medio del uso de **DT 27%**, cuando este sistema usó **k-NN 25.10%**, al haber usado **SVM** se obtuvo **25%** y con el sistema de **votación 27.20%**. Los **mejores resultados** se dieron con el uso del primer Autoencoder, debido a que no solo se toman las características de **tiempo** si no también se lo hace con el uso de características de **frecuencia**.



4. Trabajos Futuros

- Los **resultados** obtenidos para **exactitud mayor a 80%** y con **BER menor a 0.11** no aseguran que sean señales **malas**, pero **tampoco que se deban usar para aumentar las bases de datos** en estudios futuros que utilicen Deep Learning. **Para solventar este problema** y poder usar las señales generadas por el modelo planteado en este trabajo, se sugiere realizarlo bajo la **utilización de un clasificador previo** para garantizar y **conservar solo las señales confiables**.
- **Para mejorar** estos resultados, en el entrenamiento de la CGAN se debe trabajar con **bases de datos que proporcionen mayor variabilidad**, ya que es importante, para técnicas de *Deep Learning*, trabajar con gran cantidad de señales, pero se debe considerar que tendrá mayor costo computacional.
- En el presente trabajo **se utilizaron las fases reales** de cada uno de los microsismos, pero **se podrían generar estas fases de manera artificial** para ver qué tipo de resultados se tendrían.



4. Trabajos Futuros

- Con el fin de reducir **el costo computacional** y lograr mejoras en los resultados, se espera implementar técnicas como ***Transfer Learning***. Esto se basa en la premisa de utilizar redes previamente entrenadas en tareas de **clasificación binaria**. Se contempla la posibilidad de **ajustar los parámetros internos de las redes generadora y discriminadora** para **adaptar el modelo CGAN a un enfoque multiclase**, incluso se considera la duración de las nuevas señales. Otras técnicas de ***Transfer Learning*** que pueden ser tomadas en cuenta es hacerlo con redes entrenadas como ***AlexNet*** o ***GoogLeNet***, debido a que estas han sido entrenadas con una gran cantidad de imágenes y los resultados que se esperan serán mejores. Como en el anterior caso, para utilizar estas redes ya entrenadas, se debe considerar las modificaciones en las capas iniciales o finales e incluso el tamaño de entrada de las imágenes.





ESPE
ESCUELA POLITÉCNICA DEL EJÉRCITO
CAMINO A LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

**“CREACIÓN DE BASES DE DATOS SINTÉTICAS DE EVENTOS
SÍSMICOS PRODUCIDOS EN EL VOLCÁN LLAIMA CON DEEP
LEARNING.”**

**Trabajo de integración curricular previo a la obtención del título
de:
Ingeniera en Telecomunicaciones**

Autora: Myriam Cristina Estrella Caicedo

Director del Proyecto: Ing. Román Alcides Lara Cueva, PhD.

