



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Detección y clasificación  
automática de señales  
volcánicas en el  
Popocatépetl**

**TESIS**

Que para obtener el título de  
**Ingeniero Geofísico**

**P R E S E N T A**

Alejandro Reyes Romero

**DIRECTOR DE TESIS**

Dr. Denis Xavier Francoise Legrand



Ciudad Universitaria, Cd. Mx., 2022

## ***DEDICATORIA***

*A mis padres, por enseñarme los valores de disciplina, esfuerzo y dedicación.*

*A mi awe, a quien admiro demasiado.*

*A mi familia y amigos por creer en mí.*

*A mi asesor de tesis, por tenerme paciencia y guiarme en este camino.*

*A los que ya no están, por formar parte de mi vida en algún momento.*

*A ti que te das el tiempo de leer la culminación de 5 años de trabajo.*

*A Will Smith.*

*Este es mi reino en el Sol.*

## **AGRADECIMIENTOS**

Antes de empezar, quiero escribir unas palabras en agradecimiento a las personas que me ayudaron en el proceso de realización de este trabajo.

Quiero empezar por agradecer a mis padres, por enseñarme, apoyarme y estar siempre para mí cuando más los he necesitado. Siempre me han dicho que si las cosas que valieran la pena fueran fáciles cualquiera las haría y espero que este trabajo los haga sentir orgullosos de mí.

De igual manera agradezco a mi hermana, a Will Smith, a una de las personas que he conocido por más tiempo, por todo lo que ha hecho por mí, por estar a mi lado estos 5 años, por apoyarme y por tenerme paciencia cuando hacía mis dramas en un vaso de agua. Eres una inspiración para mí y un gran ejemplo a seguir.

Quiero agradecerle también a mi awe, por estar para mí, por apoyarnos a mí y a mi hermana en estos 5 años que estuvimos estudiando la universidad, te admiro mucho abuelita.

Quiero darle las gracias al resto de mi familia, por extenderme su apoyo, siempre vernos con una sonrisa y por siempre estar presentes.

Le doy las gracias también al Dr. Denis Legrand, por aceptarme como tutorado y por ver mucho potencial en mí, igual por tenerme paciencia y por darme los ánimos que necesitaba en ciertas etapas para poder terminar este trabajo.

Igual le doy las gracias al Dr. Luis Antonio, ya que, sin su ayuda, me hubiera costado mucho trabajo entenderle a la técnica de FAST, le doy gracias por siempre contestar mis dudas respecto al programa.

Quiero darles igualmente las gracias a los miembros del jurado, y a las aportaciones que hicieron para que este trabajo fuera de la mejor calidad posible.

Quiero darle las gracias a los amigos que hice durante mi formación profesional, ya que sin ellos hubiera sido imposible llegar a este punto, me es imposible mencionarlos a todos, sin embargo, quiero agradecer especialmente por su amistad a: Cristian, Gaby, Rulo, Joel, Luis, Xavier, Estefania, Jasso, Carlos Eduardo, Carlos, Kuswara, Alejandro, Ludwig, Meli, Emma y a Dany.

De igual manera, quiero darles las gracias a mis amigos fuera de la facultad, a Fany, a Lejlle y a Yessica. Por escuchar cuando lo necesitaba y por mostrarme su apoyo.

Quiero darle las gracias igualmente a Eugenia, ya que gracias a su apoyo hacer este proyecto me fue más ligero. Gracias por escucharme y por estar siempre. Igual por mostrarme que un acto de bondad verdadera siempre inspira otro.

Quiero darles las gracias en especial a todas las personas a las que les pregunté o mostré mi tesis antes de terminarla. Sus comentarios me ayudaron a que este trabajo sea el mejor posible. Espero les guste el resultado.

Quiero agradecer al Dr. Allen Husker, que, aunque no participo directamente en este trabajo, sin su guía durante mi servicio social no hubiera encontrado el tema sobre el cual quería trabajar mi tesis.

A la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México, por ser donde tuve muchas experiencias en estos 5 años, estas experiencias hicieron posible que terminara el camino de la licenciatura y que quisiera hacer este trabajo, espero poder regresar pronto.

Finalmente te agradezco a ti estimado lector, que quizás nunca tengamos la oportunidad de conocernos, pero te diste el tiempo de escoger esta tesis y leerla, espero encuentres la información que buscas.

Ahora sí, sin nada más que agregar, espero que disfruten este trabajo.

# ÍNDICE

<i>Dedicatoria</i> .....	2
Agradecimientos.....	3
Resumen.....	6
Abstract.....	7
Capítulo 1: Introducción.....	8
Capítulo 2: Antecedentes .....	11
STA/LTA.....	11
Correlación cruzada de formas de onda .....	21
Sismología en volcanes .....	22
Clasificación de señales en volcanes.....	23
Capítulo 3: Acerca de los datos y las estaciones.....	25
Estaciones Alrededor del Popocatépetl .....	25
Disponibilidad de datos.....	26
Instrumentación en las estaciones.....	28
Capítulo 4: Metodología .....	33
Comparación entre las metodologías anteriores con FAST .....	34
Requisitos para utilizar FAST.....	35
Preprocesado .....	35
Extracción de características .....	37
Búsqueda de similitudes .....	41
Postprocesado.....	48
Capítulo 5: Resultados .....	49
Capítulo 6: Conclusiones y Recomendaciones.....	55
Bibliografía .....	57

## **RESUMEN**

La sismología moderna es capaz de obtener bases de datos sismológicos continuos en tiempo real en redes sismológicas que cubren grandes áreas. Sin embargo, esto ocasiona un incremento en la cantidad de datos que los sismólogos deben analizar manualmente, haciendo esta tarea difícil. En la actualidad, se han explorado alternativas de detección y clasificación automática de sismos utilizando algoritmos escalables que permiten sacar información de estos grandes volúmenes de datos.

En este trabajo se utiliza la técnica FAST, por sus siglas en inglés “*Fingerprint and Similarity Thresholding*”, la que tiene un enfoque de minería de datos, en los datos sísmicos del volcán Popocatepetl. Aquí se estudiará la eficacia de ésta para hacer detecciones de eventos sísmicos volcánicos, y poder usar esta herramienta de manera automática para el monitoreo volcánico. FAST fue desarrollado por la Dra. Clara E. Yoon de la Universidad de Stanford (trabaja actualmente en el *United States Geological Survey*), FAST consiste en extraer de manera automática eventos sísmicos similares caracterizados por un “*fingerprint*” de la forma de onda y mediante un proceso de clasificación agrupa los *fingerprints* similares en una base de datos, para búsquedas de pares similares para obtener una lista de detecciones de terremotos.

Se utiliza FAST sobre una base de datos de una de las estaciones de la red de monitoreo volcánico del volcán Popocatepetl. Para ello se toman 45 días de datos, desde mediados de noviembre hasta finales de diciembre del año 2018. Logrando realizar detecciones de eventos sísmicos volcánicos. Se realizaron cerca de 2500 detecciones, donde la mayoría corresponden a eventos sísmicos volcánicos. No obstante, dado el funcionamiento de FAST, no es posible obtener una clasificación directa de los eventos, por lo que se hacen sugerencias para lograrlo.

## **ABSTRACT**

Modern seismology records continuous real-time seismic data through regional networks. Until few years ago, a large portion of the processing done to the acquired data was performed manually. Nowadays, as data sizes continue to grow, alternatives for automatic detection have been tested using scalable algorithms that allow extracting information from huge volumes of data. One of these techniques can identify similar seismic signals from continuous seismic records.

This technique is applied on seismic data at the Popocatepetl volcano. We study the efficiency of this approach to detect volcanic seismic events and the possibility of its usage for volcanic monitoring. FAST is a code developed by Dr. Clara E. Yoon from the University of Stanford (she now works at USGS). FAST works by extracting information from waveforms identified by key features or “fingerprints” from waveforms and through a classification process groups similar fingerprints within a database, for similar pair searches. The output of the program is a list of earthquake detections.

Using the FAST code, 45 days of continuous data recorded on a broad-band seismometer located at the Popocatepetl volcano were processed. FAST could detect about 2500 seismic events from mid-November 2018 to the end of December 2018, most of them being volcano-tectonic earthquakes. This result is an expected one because FAST cannot make a classification of seismic events, but it is instead focused on detecting tectonic events that possess similar characteristics. The code will be improved in the future to detect other kinds of volcanic events and their classification.

## **CAPÍTULO 1: INTRODUCCIÓN.**

Uno de los aspectos más importantes de la sismología moderna es la capacidad de obtener bases de datos de manera continua, lo que muestra los avances tecnológicos en adquisición de datos. Estos avances permiten la creación de más redes sismológicas con el fin de monitorear áreas de interés. Esto ocasiona un incremento en el tamaño de la cantidad de datos que los sismólogos que deben de ser analizados manualmente, haciendo que esta tarea se vuelva cada vez más lenta y se preste a llevar consigo errores humanos. Esta gran cantidad de datos genera uno de los problemas de la sismología moderna, y por lo que se generan datos mucho más rápido de lo que es posible procesar por técnicas tradicionales.

A lo largo de los años se han desarrollado diferentes técnicas de detección automática para facilitar la tarea de procesamiento de datos. Algunas de estas técnicas son: 1) el método llamado STA/LTA (Trnkoczy, 1999) y 2) el uso de la correlación cruzada de formas de onda (Gibbons, 2006). Es importante mencionar que todas estas técnicas han hecho grandes aportes a la sismología, sin embargo, cada una tiene sus propias limitaciones. Hoy en día, la tecnología se encuentra en una época donde no es difícil conseguir capacidad computacional de alto nivel, por lo que se han explorado otras alternativas de detección automática más eficientes. Muchos programas fueron desarrollados estos últimos cinco años. En este trabajo se usará uno de ellos llamado FAST (Yoon, 2015).

Teniendo en cuenta la importancia que tienen los volcanes para las comunidades que se establecieron cerca de ellos, así como los avances tecnológicos en adquisición de datos sísmicos y en el desarrollo de técnicas de detección automática. Hay un gran interés en realizar monitoreo volcánico, para poder informar a la población de posibles erupciones futuras. El monitoreo volcánico se hace gracias a la aportación de diferentes disciplinas, entre las cuales resaltan: geodesia, geoquímica, gravimetría, acústica, vulcanología física y sismología. Esta última es una de las más desarrolladas, y es la más eficiente al momento de pronosticar una erupción. Ya que esta disciplina genera una gran cantidad de datos



de alta resolución obtenida por redes de sensores en tiempo real, permitiendo un monitoreo continuo y eficaz de un volcán.

Debido a la problemática reciente con respecto a la gran cantidad de datos (*Big Data*), se han realizado algoritmos escalables con el fin de realizar búsquedas efectivas en bases de datos enormes. Algunos algoritmos de búsqueda son conocidos en la vida cotidiana. Por ejemplo, en aplicaciones que permiten encontrar canciones a partir de la identificación de música, encontrar sitios Web casi idénticos, detección de plagios, identificación de una persona por medio de sus huellas dactilares, entre otros usos.

Considerando lo mencionado anteriormente, Clara E. Yoon, desarrolló en la Universidad de Stanford la técnica FAST. La cual será utilizada en este trabajo y de aquí en adelante será referida como FAST. FAST permite analizar grandes bases de datos sísmicos continuos, con duración de meses, años o décadas de una manera rápida, efectiva y fácil de implementar. Es posible utilizar FAST exitosamente en bases de datos sísmicos continuos para detectar terremotos generados en diferentes ambientes tectónicos, uno de los ejemplos de la aplicación de FAST es el caso de la detección de réplicas del sismo con valor de magnitud (Mw) 7.4 ocurrido en Ometepe del año 2012 (Legrand, 2021). Hasta ahora, no se ha utilizado a FAST para realizar detecciones de sismos volcánicos, por lo que aplicar FAST para detectarlos, es el objetivo de esta tesis. FAST, puede ser muy útil en situaciones de emergencia, por ejemplo, en períodos de crisis volcánica o de réplicas de un terremoto muy grande, porque es muy difícil para los analistas realizar la detección y clasificación de eventos de manera continua, en tiempo real, rápida, completa y efectiva.

Por esta razón, el presente trabajo pretende aplicar la técnica de detección automática de terremotos, llamada FAST, en un contexto volcánico con el fin de determinar si está es efectiva haciendo detecciones en este tipo de ambiente. Además, de decidir si es viable utilizar esta técnica en estaciones de monitoreo volcánico, al observar la manera en la que se puede utilizar FAST para realizar

clasificaciones de familias de eventos, en específico de familias de eventos volcanotectónicos lo que no se ha hecho hasta ahora con FAST.

Así mismo, en el capítulo 2, se presenta un resumen de las diferentes técnicas de detección automáticas que se han propuesto en el pasado, describiendo sus ventajas y desventajas. Las cuales se van a comparar con la técnica FAST, la cual es utilizada en este trabajo. Haciendo énfasis en que una técnica dada de detección se debe utilizar en función del objetivo, porque cada técnica tiene sus propias ventajas y carencias. En el capítulo 3, se da una explicación acerca de los datos que se utilizaron para este trabajo. Los cuales fueron obtenidos de una base de datos del monitoreo volcánico del volcán Popocatepetl durante la segunda mitad del año 2018. Se mostrará el equipo de adquisición, así como sus características y algunas formas de onda de ejemplo, para mostrar las condiciones de la base de datos.

Posteriormente, en el capítulo 4, se explicará el funcionamiento de FAST, describiendo de manera breve, pero clara la metodología con la que trabaja para realizar detecciones de una sola estación, al igual que los parámetros utilizados para hacer detecciones y los criterios tomados en cuenta para seleccionar los valores de éstos.

Finalmente, se aplicará FAST a datos volcánicos reales, por lo que es importante entender el funcionamiento del programa para poder interpretar correctamente los resultados, los cuales se mostrarán en el capítulo 5. Se observará el tipo de señales que FAST detecta en la base de datos. Para medir la eficacia del programa y se comparan las detecciones obtenidas con el programa con unas detecciones hechas a mano. Finalmente, en el capítulo 6, se muestran diferentes maneras en las que se pueden clasificar las detecciones con base en los resultados obtenidos de la detección y como esto podría ayudar a realizar clasificación de señales volcánicas.

## **CAPÍTULO 2: ANTECEDENTES**

Con la llegada de la instrumentación digital a la sismología y el incremento en la cobertura de redes sismológicas a nivel global. El registro de señales sísmicas continuo a una alta frecuencia de muestreo se ha convertido en una realidad. Como consecuencia, se han generado enormes volúmenes de datos provenientes de redes que abarcan miles de kilómetros cuadrados por un largo periodo de tiempo. A medida que la problemática fue creciendo, el trabajo de un analista se ha convertido en uno casi imposible de llevar a cabo manualmente, ya que se generan datos a una velocidad mayor de lo que es posible analizarlos.

Por este motivo en los últimos años se han desarrollado técnicas automáticas para la detección de eventos sísmicos en bases de datos continuas. Algunas de ellas son: STA/LTA, y la correlación cruzada de formas de onda; teniendo cada una de ellas sus ventajas y desventajas. Es importante mencionar que los métodos automáticos no tienen como objetivo reemplazar al analista humano, pero sí el ayudarlo con su tarea.

### **STA/LTA**

El método STA/LTA, o "*Short-Term Average/Long-Term Average*", fue uno de los primeros métodos usados para realizar detecciones. Por ejemplo, en el pasado, la memoria de almacenamiento de datos no era suficiente, por lo que no había otra alternativa más que priorizar el registro de información útil (tal como, terremotos grandes), lo que resultaba en la ausencia de grabación de otros tipos de información valiosa (como, terremotos pequeños o ruido sísmico).

Este método es un algoritmo de detección automático fue descrito extensamente por Trnkoczy (1999), pero desarrollada anteriormente por R. Allen (1978), P. Shearer (1994) y M. Withers (1998). El método consiste en el cálculo de manera continua de los valores promedio de amplitud absoluta de una señal sísmica en dos ventanas: una ventana corta (STA) y una ventana larga (LTA). A cada instante del registro se calculan los valores promedios de la amplitud de la señal sísmica en ambas ventanas, STA y LTA. El promedio calculado sobre la ventana

STA es una medición de la amplitud instantánea, mientras que el valor promedio de la amplitud calculado sobre la ventana LTA da un valor promedio de amplitudes del ruido sísmico.

Una vez obtenidas estas ventanas, es calculada la razón STA/LTA a cada instante de la señal sísmica. Cuando el valor calculado sobre una ventana STA es mucho más grande que el valor calculado sobre una ventana LTA, hay una detección de señal de gran amplitud, como en el caso de un sismo.

El valor obtenido por la razón STA/LTA se debe comparar de manera continua con valores de activación y desactivación ingresados por el usuario. Cuando se supere el valor de activación se declara el inicio de un evento, de igual manera cuando la razón STA/LTA sea menor al valor de desactivación el evento habrá terminado.

La técnica depende de los siguientes parámetros definidos por el usuario:

- Duración de la ventana STA.
- Duración de la ventana LTA.
- Valor de activación STA/LTA.
- Valor de desactivación STA/LTA.

Los factores se asignan en función del tipo de eventos que quiera detectar el operador, por ejemplo: sismos grandes, sismos pequeños cercanos, etc.

El método STA/LTA consiste en un algoritmo de detección que se caracteriza por hacer detecciones en todo tipo de señales sísmicas.

Algunas de sus ventajas incluyen:

- 1) Costo computacional bajo;
- 2) Facilidad de utilizar; y
- 3) Adaptable a diferentes tipos de señales sísmicas tanto sismos cercanos como lejanos.
- 4) Permite grabar solamente la información “útil”, por lo que necesita de poca memoria de almacenamiento.

Esto hace que STA/LTA fuera una técnica usada frecuentemente cuando no había memorias de gran capacidad. Hoy en día, STA/LTA ha caído en desuso como

métrica para la activación del registro en memoria, debido a que la capacidad de almacenamiento ha crecido considerablemente. En cambio, se sigue usando en procesos de detección post grabación. Sin embargo, STA/LTA tiene algunas limitaciones. Por ejemplo: es difícil detectar sismos cercanos y lejanos de manera conjunta porque los parámetros anteriores son específicos para cada tipo de detecciones. Si se quiere hacer sobre datos, hay que hacer 2 detecciones con valores diferentes en los parámetros. Otras limitaciones se mostrarán con varios ejemplos del funcionamiento de la técnica.

Para ilustrar cómo funciona STA/LTA, se realizaron los siguientes ejemplos que utilizaron el comando STA/LTA integrado en la librería *ObsPy*, sobre un segmento de la componente vertical de un registro sísmico de 40 segundos de duración registrado por una estación en el volcán Chichonal el 5 de enero del 2017, con el fin de detectar sismicidad cercana al volcán. Los datos se grabaron con una tasa de muestreo de 100 Hz, y se aplicó un filtro pasa banda, entre 2 y 10 Hz.

En la parte superior de la *Figura 1* se muestra el registro sísmico y en la parte inferior se muestra el resultado de STA/LTA. Los parámetros utilizados para este caso fueron:

- STA: 0.5 segundos.
- LTA: 2 segundos.
- Valor de activación: 2.1
- Valor de desactivación: 0.1

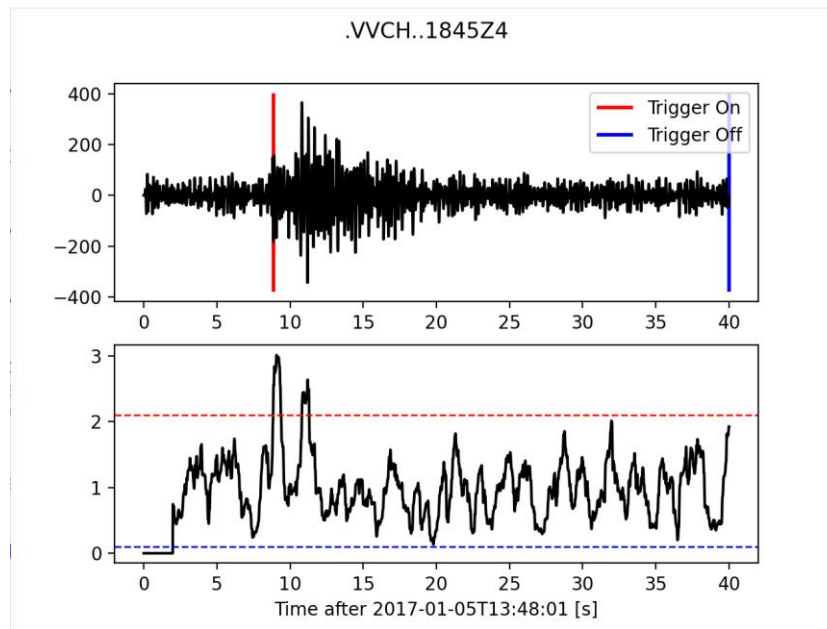


Figura 1. Ejemplo 1 de aplicación de STA/LTA al registro sísmico en el volcán Chichonal del 5 de enero del 2017.

Al observar la *Figura 1*, parece que el método STA/LTA está funcionando correctamente, debido a que está que detecta el evento adecuadamente. Es decir, detecta bien la onda P, onda primaria, cerca de los 8 segundos, y la onda S, onda secundaria, cerca de los 12 segundos. Pero al observar el evento, se aprecia que STA/LTA se desactiva al final del registro, lo que quiere decir que la desactivación no funcionó. De este ejemplo se pueden sacar dos conclusiones: o el evento duró más de medio minuto o que STA/LTA no alcanzó el valor de desactivación y el evento posiblemente terminó antes.

En el ejemplo, lo más probable que suceda es lo segundo, debido al alto nivel de ruido sísmico de la señal. Por lo que una de las desventajas de STA/LTA es que si el valor de desactivación es mal seleccionado (demasiado bajo, como en el ejemplo), se tienen detecciones muy largas sin sismos. Debido a que una de las razones por la que se desarrolló STA/LTA fue para salvar memoria, generalmente se escogían niveles de desactivación bastante altos para evitar este problema y esto ocasionaba que algunas señales sísmicas no fueran completas (cortadas antes del final de la coda S).

Como el valor de desactivación anterior fue mal seleccionado (demasiado pequeño), se aumentará para limitar la duración de la detección. La *Figura 2*

muestra la aplicación del comando STA/LTA de la librería *ObsPy*. Los parámetros para este caso fueron (únicamente se cambió el valor de desactivación):

- STA: 0.5 segundos.
- LTA: 2 segundos.
- Valor de activación: 2.1
- Valor de desactivación: 0.2

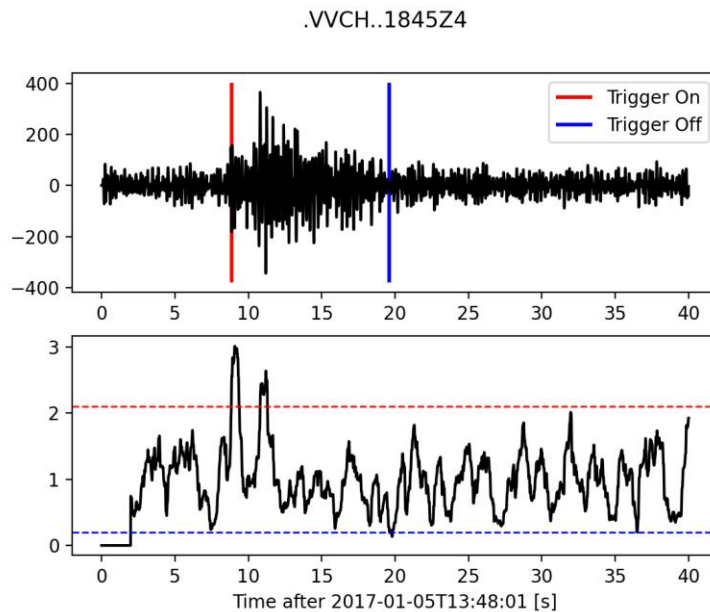


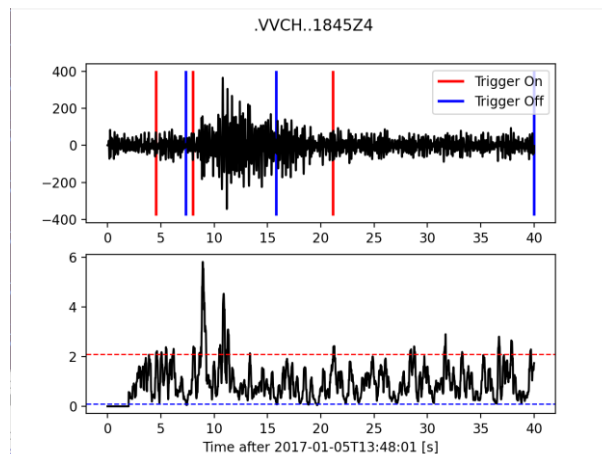
Figura 2. Ejemplo 2 de aplicación de STA/LTA al registro sísmico en el volcán Chichonal del 5 de enero del 2017

En este caso, se incrementó el valor de desactivación de STA/LTA en un valor de 0.2, cuando antes era de 0.1. Se puede observar que ahora el evento ya no tiene la duración de 32 segundos como en la *Figura 1*. La duración del evento es de poco más de 10 segundos, dado que STA/LTA alcanza el valor de desactivación alrededor del segundo 20. Asimismo, se puede apreciar otra desventaja del método, la cual es que dos valores de desactivación parecidos, 0.1 y 0.2, dan resultados completamente diferentes. En la práctica, es difícil escoger bien todos estos parámetros, por lo que STA/LTA, resulta efectivo una vez que se tiene experiencia al seleccionar los parámetros, dada su sensibilidad al hacer detecciones.

Ahora, se revisará el efecto de la ventana STA sobre la detección. La *Figura 3* muestra la aplicación del comando STA/LTA de la librería *ObsPy*, los parámetros

(en este ejemplo únicamente se cambiará a STA y se dejarán los otros parámetros iguales al ejemplo de la *Figura 1*), para este caso fueron:

- STA: 0.2 segundos.
- LTA: 2 segundos.
- Valor de activación: 2.1
- Valor de desactivación: 0.1



*Figura 3. Ejemplo 3 de aplicación de STA/LTA al registro sísmico en el volcán Chichonal del 5 de enero del 2017.*

En el caso de la *Figura 3*, se puede observar que con una ventana STA más corta, en este caso de 0.2 segundos, se detectan tres eventos. Dos falsas detecciones y una positiva. En este segmento de 40 segundos, resultando ser ruido para la primera y última detección. Este es otro ejemplo que ilustra la dificultad en escoger los parámetros; si STA es demasiado pequeño va a detectar muchos eventos falsos y cortar un sismo, lo que no es deseable.

En el caso del efecto de la ventana LTA. La *Figura 4* muestra la aplicación del comando STA/LTA de la librería *ObsPy*. Los parámetros (únicamente se cambió el valor de LTA y los demás parámetros permanecen igual al ejemplo de la *Figura 1*), para este caso fueron:

- STA: 0.5 segundos.
- LTA: 10 segundos.
- Valor de activación: 2.1.
- Valor de desactivación: 0.1.



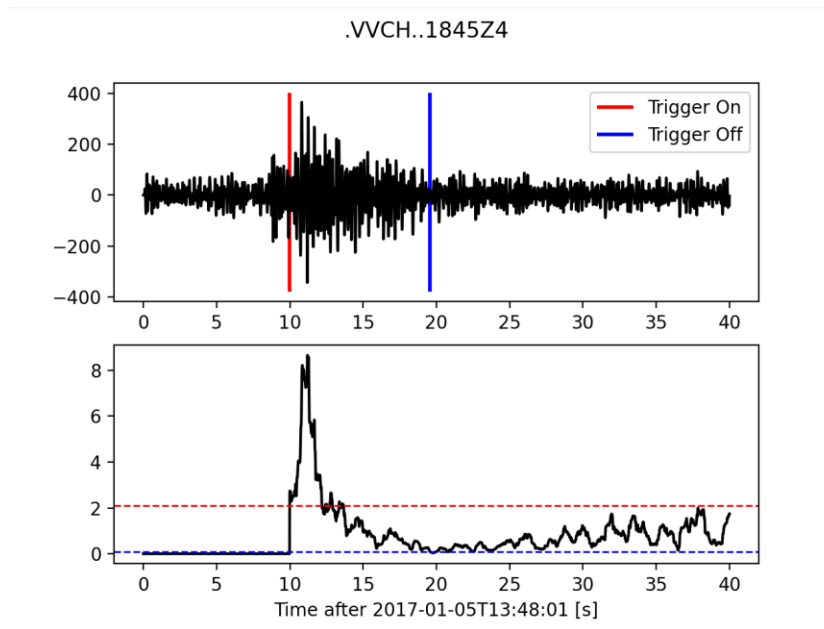


Figura 4. Ejemplo 4 de aplicación de STA/LTA al registro sísmico en el volcán Chichonal del 5 de enero del 2017. Elaboración propia.

En este ejemplo se utilizó una ventana LTA más larga de 10 segundos de duración, por lo que durante los primeros 10 segundos, no hay un cálculo de STA/LTA. En este caso, se cortó la onda P y la detección se hizo sobre la onda S, lo que no es deseable. Este es otro ejemplo que muestra la dificultad al escoger los parámetros: si LTA es demasiado grande va a detectar ondas de altas amplitudes como la onda S y puede no detectar a la onda P, que faltara en la señal final cortada.

Finalmente, se observa en la *Figura 5* el efecto del parámetro de activación de la detección, mediante la aplicación del comando STA/LTA de la librería *ObsPy*. Los parámetros (únicamente se cambió el valor de activación y los otros parámetros iguales al ejemplo de la *Figura 1*), para este caso fueron:

- STA: 0.5 segundos.
- LTA: 2 segundos.
- Valor de activación: 1.1.
- Valor de desactivación: 0.1.

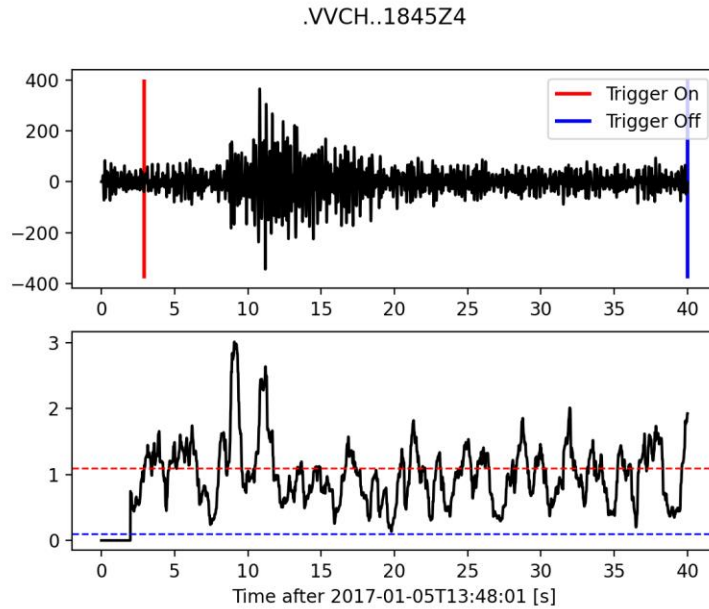


Figura 5. Ejemplo 5 de aplicación de STA/LTA al registro sísmico en el volcán Chichonal del 5 de enero del 2017.

En este ejemplo se utilizó un valor de activación más pequeño, y se observa que la ventana de detección es más grande que en la *Figura 1*. La detección inicia mucho antes de la onda P, esto debido a que alcanza el valor de activación más rápido al ser este más pequeño. En algunos casos también puede ser un inconveniente porque puede producir registros muy largos y llenar la memoria.

En conclusión, se observa una de las grandes limitaciones del método, que a partir de la elección de los cuatro parámetros definidos anteriormente se van a obtener diferentes detecciones, algunas muy deficientes. Este método es efectivo para realizar detecciones, pero solamente si los parámetros son seleccionados para tener ventanas de tamaño adecuado. Ciertos parámetros de detección pueden detectar eventos de manera correcta, sin embargo, también es posible que se realicen falsas detecciones tal como se vio en la *Figura 3*, o señales muy largas con mucho ruido. O por el contrario, señales cortadas sin la onda P o con poca onda S.

A continuación, se mostrarán las detecciones realizadas con STA/LTA usando los parámetros utilizados en la *Figura 1*, para el día entero de registro, acercándose a la hora del evento de las figuras anteriores.

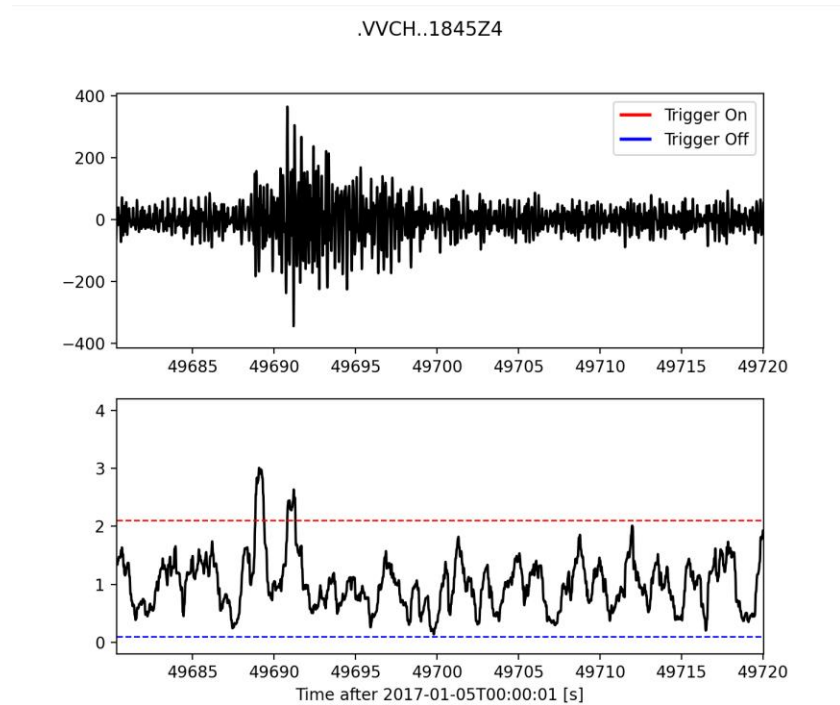
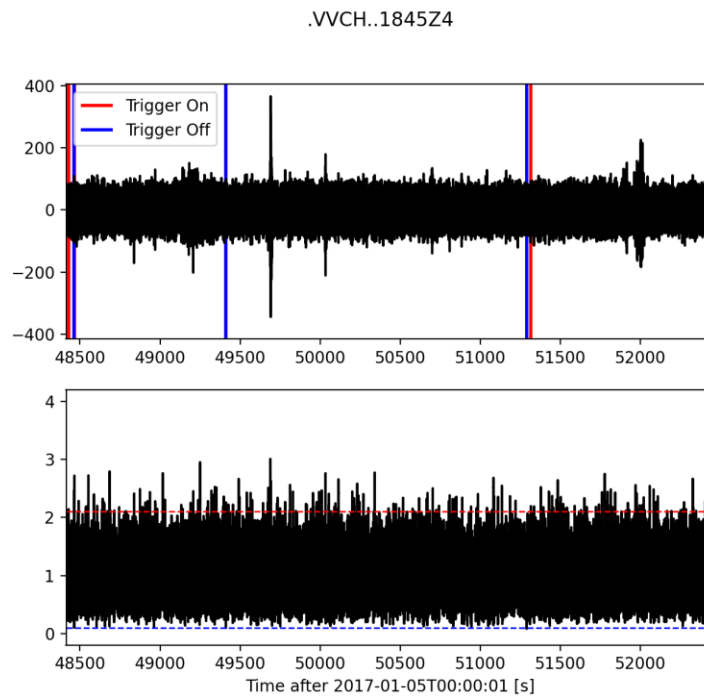


Figura 6. Aplicación de STA/LTA al día entero de registro en el volcán Chichonal del 5 de enero del 2017, haciendo acercamiento a la hora estudiada en los ejemplos anteriores con los parámetros STA/LTA utilizados en el Ejemplo 1.

Se puede observar en la *Figura 6*, que al aplicar STA/LTA en todo el día de registro, el evento que se ha analizado en las figuras anteriores no es identificado por el método, aunque alcanza el valor de activación, lo que quiere decir que se encuentra contenido en otra detección, tal como es posible observar en la *Figura 7*.



*Figura 7. Aplicación de STA/LTA al día entero de registro en el volcán Chichonal del 5 de enero del 2017, haciendo acercamiento a la hora estudiada en los ejemplos anteriores con los parámetros STA/LTA utilizados en el Ejemplo 1.*

El evento que se ha analizado de manera aislada, la ventana que contiene este evento inicia alrededor de la marca de 49,680 y termina aproximadamente en la marca 49,720. En los ejemplos anteriores no fue identificado al utilizar STA/LTA en todo el día de registro. Este último ejemplo muestra otra de las desventajas del método, la cual es que de acuerdo con los parámetros se pueden omitir eventos, a pesar de que estos parámetros funcionen bien sobre una ventana corta.

## **Correlación cruzada de formas de onda**

Uno de los objetivos de los métodos de detección automática de sismos, es encontrar formas de ondas muy parecidas. El método de correlación cruzada de formas de onda fue diseñado con esto en mente, evaluando la similitud entre una forma de onda continua y eventos de interés. Este método realiza detecciones a partir de una forma de onda en particular, es decir un “*template*”, el cual puede ser simulado o empírico (Withers, 1998). Para después detectar señales con esa misma forma o las más parecidas en registros continuos largos. El método mejora la sensibilidad de las detecciones al calcular la similitud entre formas de onda. Este es un método muy efectivo al encontrar formas de onda similares en registros con mucho ruido o en registros que se adquirieron en lugares con una tasa ruido-sígnal baja.

Este método realiza una búsqueda exhaustiva de “una forma de onda en todo el registro”. Esto lo hace calculando el coeficiente de correlación (C.C.), de un *template* de forma de onda con ventanas de tiempo sucesivas provenientes de un registro sísmico.

Por lo general se calcula el C.C. normalizado, debido a que es importante detectar señales con la misma forma sin importar las amplitudes. Pueden existir señales parecidas, pero con amplitudes diferentes. De esta manera se obtienen resultados de C.C. entre 1 y -1; siendo 1 para señales idénticas, 0 para señales con ninguna relación y -1 para señales idénticas, pero con signos opuestos.

El método es bastante efectivo y con muchas aplicaciones, no obstante, es un método limitado por los *templates*, los cuales son escogidos en función al objetivo de la investigación. Esto debido a que, al realizar detecciones, nada más identificará como detecciones válidas a aquellos eventos que pasen cierto umbral de detección. Lo que es una ventaja y una desventaja al mismo tiempo, dado que, si se hace una búsqueda con el objetivo de encontrar una familia de señales en específico, el método de correlación cruzada lo logrará. Sin embargo, si se buscan familias o fuentes diferentes de eventos al *template*, el método de correlación cruzada no será efectivo.

## **Sismología en volcanes**

Para Schmincke (2004), un volcán es una etapa en el flujo de materia y energía del interior de la Tierra a la superficie. Este cuenta con un sistema magma-volcán, al igual que incluye un complejo sistema de raíces que yace debajo de él. Por otro lado, Zobin (2017), define un volcán como una ventila o chimenea, que conecta una reserva de magma, ubicado en las profundidades de la corteza de la Tierra con la superficie terrestre. La composición interna de un volcán, la de sus magmas y algunos de sus procesos son conocidos gracias a la expulsión de material desde el interior y también por estudios sísmicos realizados en volcanes.

La sismología volcánica, de acuerdo con Zobin (2017), estudia las señales sísmicas que se originan a partir de volcanes y de su actividad. Estudia la interacción entre procesos volcánicos y procesos sismotectónicos.

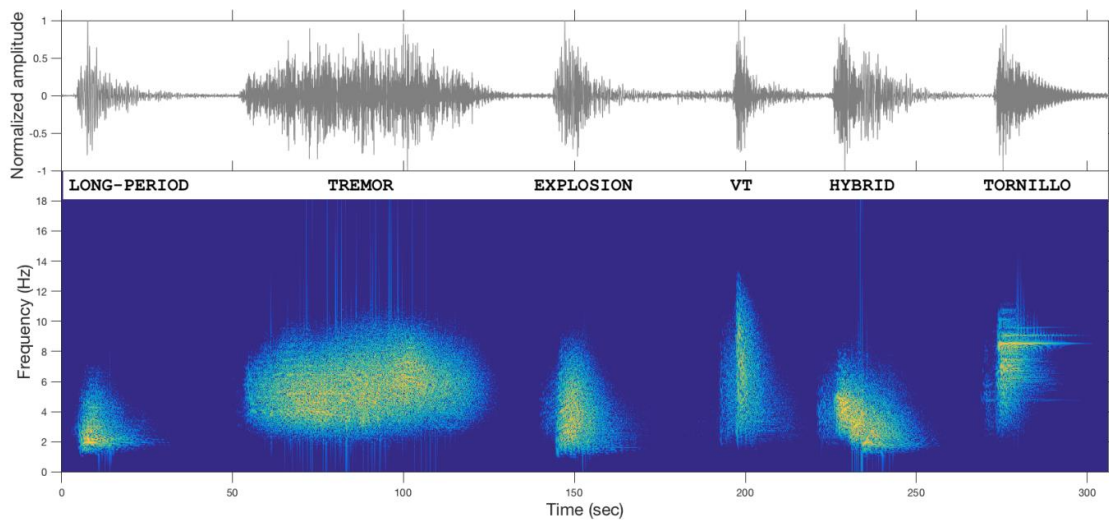
El sismólogo japonés Fusakichi Omori (1912), definió a un terremoto volcánico como una perturbación sísmica, provocada por la acción directa de una fuerza volcánica, o una cuyo origen se encuentra debajo, o adyacente a un volcán. Es importante entender los procesos internos del volcán, su funcionamiento como sistema y sus riesgos a la población, teniendo en cuenta que existen poblaciones que se ubican en cercanía a estos volcanes, por lo que los volcanes representan un riesgo.

Una de las herramientas principales con la que se monitorean los volcanes es su actividad sísmica. Generalmente un aumento la actividad sísmica en las cercanías de un volcán ocurre en respuesta a un aumento de la actividad volcánica (subida de gases, aumento de la producción de gases, aumento de la deformación, etc.). Por este motivo es de gran importancia, mantener una red de sismómetros de banda ancha para tener una mejor perspectiva de la naturaleza y comportamiento de las señales sísmicas causadas por la actividad volcánica.

Se han realizado diferentes clasificaciones con lo que respecta a volcanes, entre ellas se puede resaltar: tipo de erupción; tipo de terremoto volcánico, forma, etc. Sin embargo, para objetivos de este trabajo se hablará de la clasificación de las señales sísmicas que genera un volcán.

## Clasificación de señales en volcanes

Dentro de un volcán existe un gran número de procesos e interacciones donde cada uno de ellos produce una señal diferente. Una de las clasificaciones de señales provenientes de un volcán pertenece a Chouet (Nature, 1996), donde menciona algunas de las diferentes firmas sísmicas provenientes de un volcán por: “periodo largo” (LP), “Volcano-Tectónicos” (VT), “Híbridos”, “Tremores” o “periodo muy largo” (VLP). Por su parte, Malfante et al (2018 y 2020), clasifica a seis familias de eventos que se pueden asociar a volcanes (*Figura 8*):



*Figura 8. Clasificación de señales provenientes de un volcán. En la imagen se incluye la forma de onda y el espectrograma de señales sísmicas provenientes del volcán Ubina. Imagen tomada de “Malfante et al. (2018). Machine Learning for Volcano-Seismic Signals: Challenges and Perspectives. IEEE Signal Processing Magazine, Institute of Electrical and Electronics Engineers, 35 (2), pp.20 - 30.”*

En la *Figura 8* se muestran las formas de onda, así como su representación espectral de las señales siguientes:

- **Periodo Largo (LP):** son eventos que se caracterizan por tener un bajo contenido espectral (<6Hz), por lo general este es angosto. Tienen un arribo de onda P claro, aunque el arribo de onda S no es tan claro y en ocasiones, la onda S ni siquiera se puede identificar. A veces estos periodos llegan a ser muy largos y en ese caso se habla de señales de periodo muy largo o VLP (>0.44Hz (Chouet, 2003)).
- **Tremores:** tienen una amplitud sostenida, que puede durar desde minutos hasta días, y un rango de frecuencia ancho, en algunos volcanes su origen

se debe a la superposición de eventos LP (McNutt, 1992). Algunos tremores son armónicos con una sola frecuencia predominante.

- **Explosiones:** son eventos que son generados por expulsiones súbitas de magma, cenizas y gas, su contenido espectral es intermedio. Debido a que se trata de un evento súbito y con causas complejas, es muy raro que existan dos explosiones que se parezcan entre sí.
- **Volcano-Tectónicos (VT):** estos eventos se caracterizan por tener amplitudes grandes en la onda P y en la onda S y por tener un alto contenido espectral, alcanzan altas frecuencias, muy similar a los sismos tectónicos. El objetivo de esta tesis es hacer detecciones de esta familia.
- **Híbridos:** presenta características de VT y de LP, tanto en forma de onda, como en contenido espectral (mezcla de al menos dos frecuencias diferentes).
- **Tornillos:** se caracterizan por tener una forma de onda similar a la de un tornillo, tienen un contenido espectral alto y coda larga. Igual en su forma espectral parece tener forma de tornillo. Se pueden considerar como un tipo específico de evento LP.

Se pueden encontrar otros tipos de eventos con diferentes formas de onda y firmas espectrográficas asociadas a otros eventos volcánicos, pero estos eventos no son encontrados regularmente en todos los volcanes y se asocian a volcanes específicos. En el volcán Popocatepetl, se espera encontrar familias de VT's, LP's, explosivos y tremores. Esto de acuerdo con reportes del Centro Nacional de Prevención de Desastres (CENAPRED).

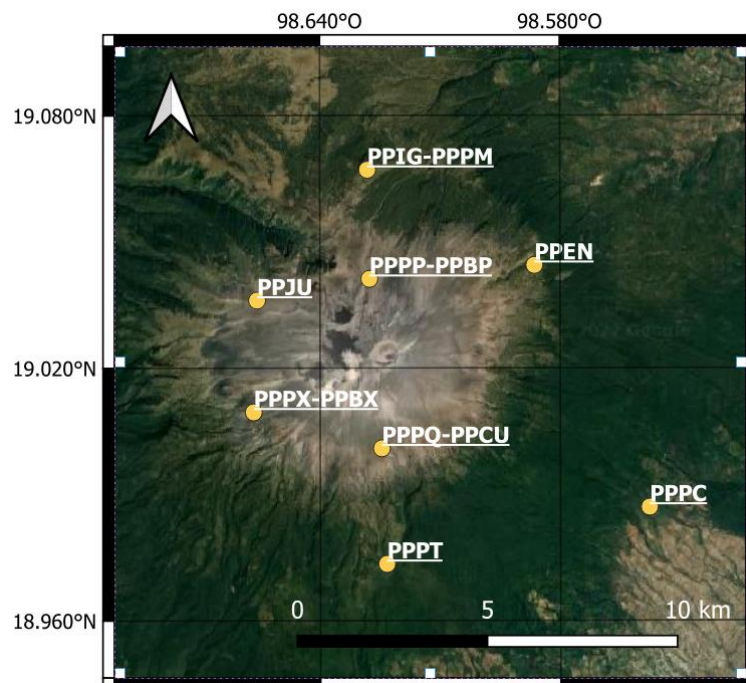


## **CAPÍTULO 3: ACERCA DE LOS DATOS Y LAS ESTACIONES**

### **Estaciones Alrededor del Popocatépetl**

Para la realización de este trabajo se utilizaron datos de las estaciones que se encargan de monitorear la actividad sísmica del volcán Popocatépetl. El volcán Popocatépetl, es un volcán activo con una elevación promedio de 5,452 m.s.n.m. Se localiza cerca de una de las zonas más densamente pobladas del mundo, con alrededor de 25 millones de personas viviendo en un radio de 80 kilómetros. Debido a esto el volcán representa un gran riesgo. A finales de 1994, el volcán Popocatépetl comenzó a presentar actividad explosiva, después de casi 70 años de quietud (De la Cruz y Siebe, 1997). Por lo que desde esa fecha, ha sido prioridad realizar un monitoreo constante del mismo, con el objetivo de alertar a toda la población en la zona de riesgo.

El volcán actualmente es monitoreado por el CENAPRED. A finales de 2018, se renovaron las estaciones de monitoreo sísmico en el volcán Popocatépetl, las cuales se encuentran ubicadas tal como se muestra en la *Figura 9* (las coordenadas se muestran en la *Tabla 1*).



*Figura 9. Mapa de estaciones de la red de monitoreo volcánico del Popocatépetl en el año 2018.*

Nombre	Código	Latitud [°]	Longitud [°]	Elevación promedio [m]
Canario	PPPP-PPBP	19.0411	-98.6276	4305.5
Chipiquixtle	PPPX-PPBX	19.0094	-98.6566	3978.8
Colibrí	PPPC	18.9869	-98.5577	2669.2
Cuervos	PPPQ-PPCU	19.0009	-98.6246	4200
Encinos	PPEN	19.0443	-98.5864	3655
Juncos	PPJU	19.0360	-98.6557	4206
Tetexcaloc	PPPT	18.9734	-98.6233	3330
Tlamacas	PPIG-PPPM	19.0670	-98.6281	3980

Tabla 1. Coordenadas de estaciones de la red de monitoreo volcánico del Popocatepetl en el año 2018.

Para este trabajo se revisaron los datos de la estación de monitoreo sísmico Canario, a la cual nos referiremos como PPPP. Se revisará un periodo de fechas, abarcando desde mediados del mes de noviembre del 2018 hasta el fin de ese año. Este periodo representa fechas en las que el volcán Popocatepetl presentó un periodo de actividad intensa, coincidiendo con el monitoreo hecho por CENAPRED, donde en ese periodo se emitió el semáforo de alerta volcánica color amarillo por: actividad explosiva, lluvias de ceniza y posibilidad de flujos (CENAPRED, 2018).

### Disponibilidad de datos

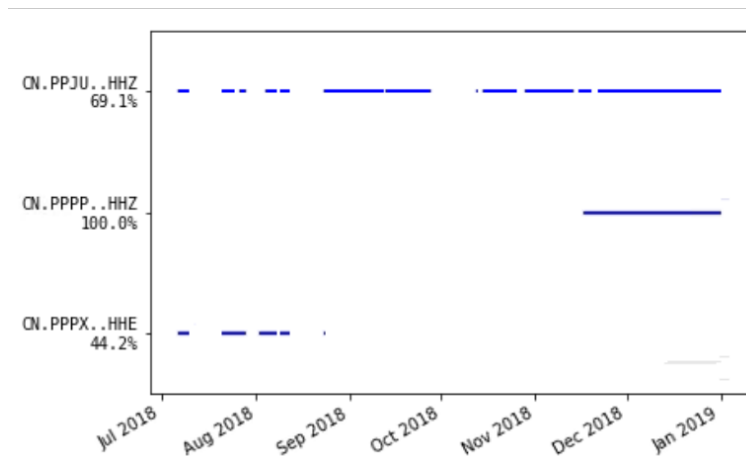


Figura 10. Calendario de días con disponibilidad de datos.

En la *Figura 10* se muestra la disponibilidad de los datos en las estaciones: Canarios (PPPP), Juncos (PPJU) y Chipiquixtle (PPPX). Se puede ver que mientras que la estación PPJU, es la que abarca mayor cantidad de tiempo, tiene una disponibilidad de un 69.1%. Es decir, la base de datos no es continua, al menos no en su totalidad. La estación PPPP, es la que cuenta con la mayor continuidad abarcando desde mediados de noviembre y todo diciembre, debido a esto se decidió utilizar esta estación. Y finalmente la estación PPPX es la que menor disponibilidad de datos tiene, por lo que no resulta viable para utilizar FAST.

Por estas razones para en este trabajo se decidió utilizar la base de datos de la estación Canario PPPP. Además, es la estación más cercana al cráter del volcán. Es una estación que presenta alrededor de 45 días de datos continuos, que tal como se ve en la *Figura 10*, se encuentra en su 100%, la base de datos abarca desde el 16 de noviembre (16:00:40 en horario universal, UTC) hasta el 1 de enero de 2019 (00:00:00 UTC), a diferencia de las otras bases de datos disponibles. Para esta estación se utilizarán los datos obtenidos por las tres componentes: HHZ, que indica el movimiento vertical del sismómetro (arriba a abajo); HHE, que indica uno de los movimientos horizontales del sismómetro (este a oeste); y HHN, que indica el otro movimiento horizontal (norte a sur).

## **Instrumentación en las estaciones**

La base de datos se obtuvo de estaciones con la siguiente instrumentación:

Los sismómetros de banda-ancha son sensores de velocidad, de marca *Nanometrics*, y de modelo *Trillium Post-Hole 120s* (Figura 11). Este modelo de acuerdo con el fabricante (*Nanometrics*), es ideal para estudiar sismos locales, regionales y telesísmicos, debido a su curva de respuesta la cual es lo suficientemente amplia para detectar una gran gama de frecuencias. Fueron enterrados a profundidades entre 1 y 5 metros.



Figura 11. (*Nanometrics Trillium Post Hole. Obtenida de nanometrics.ca* )

Este instrumento tiene las siguientes características, de acuerdo con el fabricante:

- Rango de frecuencia: 120 s – 150 Hz.
- Bajo ruido propio.
- Consumo bajo de energía.
- Fácil de instalar y de utilizar.

El instrumento cuenta con un rango de frecuencias amplio, esto es porque se trata de un sismómetro de banda ancha, el cual permite registrar bajas frecuencias como altas frecuencias. Por esta habilidad son instrumentos útiles en el monitoreo de volcanes, ya que se pueden detectar eventos de bajas frecuencias como de altas. El sismómetro *Trillium 120s Post Hole*, y *Güralp*, cuentan con un rango amplio de frecuencias, lo que es visible en sus curvas de respuesta, tanto a la frecuencia, como en fase (Figura 12).

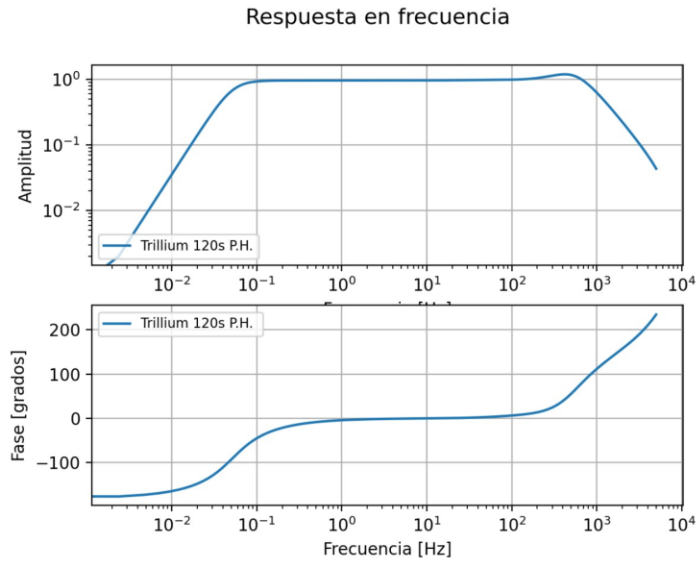


Figura 12. Curva de respuesta en velocidad (amplitud y fase).

En la *Figura 12* se muestra la curva de respuesta del sismómetro y del sistema de adquisición en amplitud y en fase en función de la frecuencia para el *Trillium 120s Post Hole*, obtenida a partir de los polos y ceros del instrumento (captor sísmico y sistema de adquisición). Los seis polos y once ceros están dados en la *Tabla 2*. Se ha usado la función `paz_2_freq_response` de la paquetería `ObsPy`, la cual trabaja con frecuencias angulares, por lo que los valores de la siguiente tabla es necesario multiplicarlos por  $2\pi$ , si se quiere utilizar esta función.

<b>Parámetro</b>	<b>Valores</b>
<b>Ceros [m/s]</b>	0
	0
	-31.63
	-160
	-350
	-3177
<b>Polos [m/s]</b>	$-0.036614 \pm 0.037059i$
	-32.55
	-142
	$-364 \pm 404i$
	-1260
	$-4900 \pm 5204i$
	$-7100 \pm 1700i$
<b>Constante [MV*s/m]</b>	$1.2488239 \times 10^{15}$

Tabla 2. Polos y Ceros de sismómetro Trillium 120s Posthole. Valores obtenidos de “Merchant, B et al. (2017). Next Generation Qualification: Nanometrics T120PH Seismometer Evaluation. United States. <https://doi.org/10.2172/1402551>”

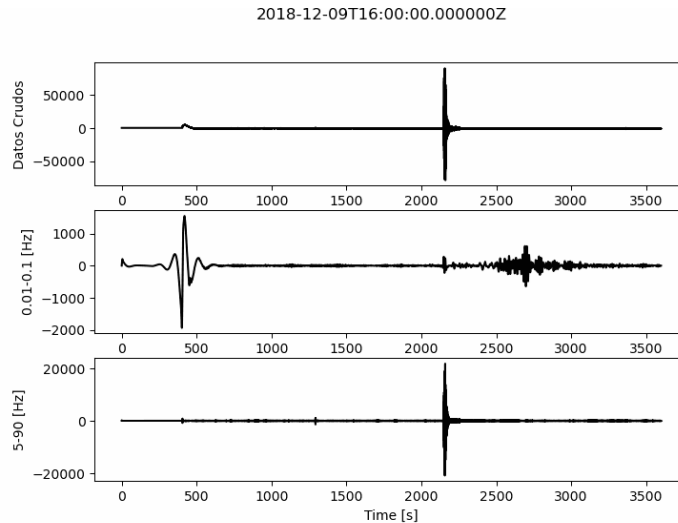
La respuesta es plana en amplitud entre aproximadamente 0.1 Hz y 200 Hz, lo cual se aproxima a las especificaciones del fabricante. A su vez se tiene una respuesta plana lo suficientemente larga, por lo que a partir del sistema de polos y ceros se confirma que el sismómetro es de banda ancha.

En los instrumentos modernos, se intenta en medida de lo posible, obtener una fase cercana a 0 en una banda ancha de frecuencias, para evitar desfases cuando uno filtra los datos. Se ve esta parte de fase nula entre 0.1 Hz y 100 Hz.

La estación Colibrí (PPPC), fue la única estación que no fue modernizada, y sigue funcionando con un sismómetro de banda ancha de la marca Güralp.

La importancia de usar estaciones de banda ancha se puede observar en la *Figura 13*, donde se muestra un ejemplo de una hora de datos correspondiente a la componente vertical de la estación PPPP, del día 9 de diciembre del 2018. Esta hora se muestra sin filtrar (gráfico superior), con un filtro pasabandas entre 0.01 y 0.1 Hz (gráfico de en medio), con el objetivo de observar eventos lejanos; y otro pasabandas entre 5 y 90 Hz (gráfico inferior), donde se ven sismos pequeños cercanos al volcán. Para este trabajo, se quiere encontrar señales cercanas al

volcán usando datos obtenidos en la estación más cercana al cráter (PPPP), por lo que el filtro se realizará con bandas en frecuencias intermedias, tal como se mencionará más adelante.



*Figura 13. Ejemplo de una hora de datos de la estación PPPP del día 9 de diciembre del 2018. Arriba: Datos Crudos. En medio: Filtrado con una banda de bajas frecuencias. Abajo: Filtrado con una banda de altas frecuencias.*

Es posible realizar análisis como el anterior debido a que, como se mencionó previamente, el sismómetro *Trillium 120s Post Hole*, y *Güralp*, cuentan con un rango amplio de frecuencias, lo que permite el registro de una amplia gama de eventos.

El sistema de adquisición fue de la marca Nanometrics, de modelo CENTAUR. Se utilizó una frecuencia de muestreo de 200 Hz para poder registrar pequeños sismos VT's que son de alta frecuencia por ser pequeños y cercanos al volcán.

Adicionalmente, es de gran importancia mencionar que las estaciones son de alimentación solar. Específicamente usan Baterías de 12 VDC con un arreglo de paneles solares. Se puede observar en la *Figura 14*, un ejemplo de helicorder del día 09/12/2018 en la estación PPPP. Hay periodos donde no se registra información, posiblemente debido a que la batería de esa estación se haya agotado, durante la noche (el corte de energía empieza alrededor de las 04:00 UTC, es decir a las 22:00 en horario local y termina 12 horas después alrededor de las 16:00 UTC es decir las 10:00 en horario local, variando mucho por día).

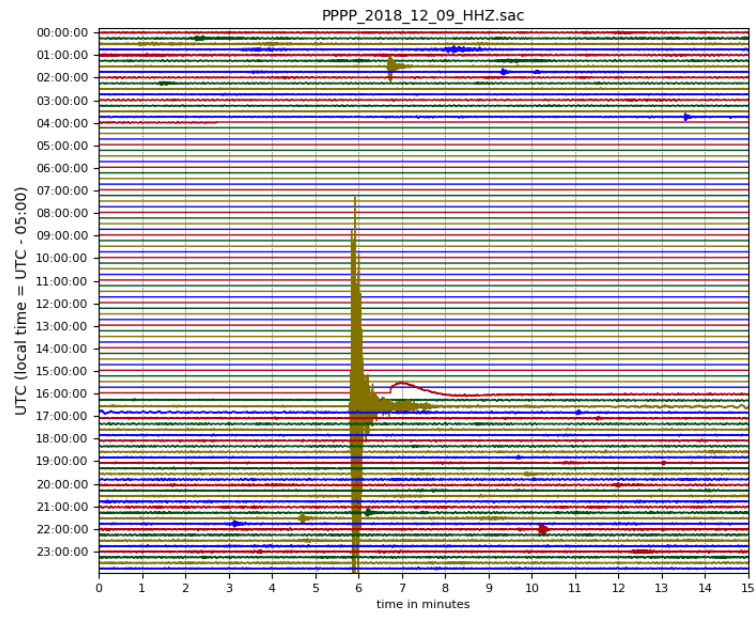
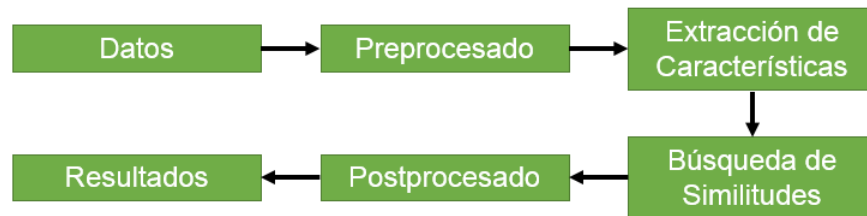


Figura 14. Helicorder para la estación PPPP del día 9 de diciembre del 2018.



## **CAPÍTULO 4: METODOLOGÍA**

FAST o por sus siglas en inglés “*Fingerprint and Similarity Thresholding*” es un software de detección no supervisada de terremotos. Es una adaptación del funcionamiento del programa “*Waveprint*” (Baluja, 2008), el cual a partir de recibir segundos de una canción puede identificar la canción, siempre y cuando este dentro de una base de datos. FAST, recibe como datos de entrada señales sísmicas continuas de tres componentes, provenientes de una o de múltiples estaciones dentro de una red. Obteniendo como producto final una lista de marcas de tiempo con posibles detecciones. En la *Figura 15* se muestra un diagrama de flujo con las etapas del programa.



*Figura 15. Diagrama de flujo de FAST.*

Entre las ventajas principales de FAST se incluyen: 1) su portabilidad, es decir es posible implementarlo en cualquier sistema compatible, ya que dentro de la paquetería se encuentra todo lo necesario para usarlo; 2) que es escalable, ya que se puede implementar con cualquier base de datos de cualquier red, sin importar la duración; 3) su no dependencia de catálogos de eventos ya conocidos; 4) FAST puede funcionar con bases de datos de las que no se tengan *templates*; y 5) su rápida velocidad de ejecución, ya que permite analizar rápido bases de datos de semanas, meses, incluso años en solamente algunas horas de procesamiento para encontrar terremotos similares.

Por otro lado, FAST detecta sismos similares, pero solamente si hubo varias señales similares. Si un evento que sólo ocurre una vez en la base de datos este no será detectado por FAST. A su vez, para poder hacer uso de FAST se necesita tener una gran cantidad de memoria en la computadora. Al trabajar con bases de datos de años, este requisito puede presentar dificultades para el operador. Otra de

sus limitaciones es que como resultado se obtiene una lista de señales similares, no obstante, esta es una lista de señales candidatas y hace que sea necesario realizar más procesos para determinar los eventos reales, esto se discutirá más adelante. De igual manera, otra de sus desventajas es que no realiza la detección de fases, como lo son la onda P y la onda S, por lo que no es útil para realizar localización de eventos.

En este capítulo se tiene como objetivo hacer una reseña del funcionamiento del programa FAST paso a paso. Para información más detallada, se pueden referir a la guía de usuario, o a los artículos de: Baluja (2008), Yoon (2015), Beroza (2017). Ya que estos guían de mejor manera en cómo funciona el programa y entran a detalle en el significado de cada parámetro.

### **Comparación entre las metodologías anteriores con FAST**

Ya que previamente se revisaron las metodologías de: STA/LTA y Correlación Cruzada de Formas de Onda; se puede ver que FAST tiene varias ventajas con respecto a los otros métodos, entre las cuales se incluyen:

- Es importante dejar en claro que, no existe una metodología de detección que sea óptima en todas las condiciones.
- Los tres métodos hacen detecciones con base en un parámetro de la señal. STA/LTA, lo hace con energía; Template Matching, por correlación de las señales. FAST es similar en este aspecto ya que igual realiza detecciones tomando como parámetro la energía espectral de la señal.
- STA/LTA, es un método efectivo y es fácil de implementar. Sin embargo, representa una opción poco conveniente debido a que es necesario manejar parámetros difíciles de establecer correctamente, además de las desventajas siguientes: algunas detecciones se pueden omitir y otras detecciones pueden ser falsas.
- Correlación cruzada de formas de onda, es más difícil de implementar porque es un requisito conocer *a priori* los *templates* que se van a utilizar. La metodología FAST es diferente, porque no necesita de *templates* previamente obtenidos, detección no supervisada.

## **Requisitos para utilizar FAST**

FAST es un software escrito en Python y C++ y fue diseñado para ejecutarse en núcleos de Linux, por lo que no es compatible con las arquitecturas de Windows y de macOS. El código se beneficia al ser ejecutado en computadoras con mayor memoria y CPU's.

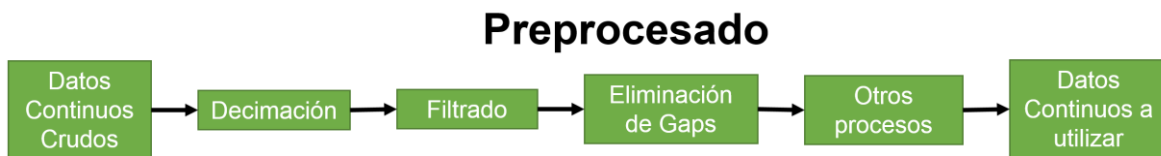
Para ejecutar FAST, se utilizó una computadora con las siguientes especificaciones:

- Procesador Intel Core i7-6700k S-1151 con 8 núcleos a 4.0GHz.
- Con una tarjeta madre ASUS Z170 Pro Gaming con procesadores Intel de sexta generación.
- Memoria de 64GB de RAM y 10TB de disco duro.

Estas especificaciones son adecuadas para ejecutar de manera correcta FAST. Debido a que se generan millones de huellas binarias. Es recomendable tener acceso a un sistema con buena capacidad de almacenamiento.

## **Preprocesado**

El preprocesado consiste en los pasos que se muestran en el diagrama de flujo contenido en la *Figura 16*:



*Figura 16. Diagrama de flujo de preprocesado de datos*

Algunos de estos pasos en realidad son opcionales, sin embargo, FAST se beneficia de su aplicación. Con la decimación es posible reducir la frecuencia de muestreo con la que se va a trabajar, a partir de un valor llamado factor de decimación. Algunos parámetros como la longitud de ventana y el intervalo de ventaneo son controlados por el número de muestras. Por lo que, si la señal continua cuenta con una tasa de muestreo alto, estas ventanas e intervalos serán más pequeños, y como se explicará más adelante, esto llevará a que FAST tomé

más tiempo de ejecución. En este trabajo se toma un factor de decimación de 2, con el objetivo de reducir la frecuencia de muestreo de 200 Hz a 100 Hz, teniendo la frecuencia de Nyquist (la cual representa la máxima frecuencia para poder reconstruir la señal), en los 50 Hz.

Se ha hablado de la importancia de las bandas de muestreo, ya que ciertos tipos de señales se presentarán en frecuencias diferentes. Por esta razón es importante obtener una banda de frecuencias en donde se encuentren las señales objetivo. Por lo que es necesario utilizar un filtro tipo pasa-bandas, tomando en cuenta las ondas que se quieren detectar. En este trabajo se utiliza una banda de frecuencias que va de 2-20 Hz. Es posible utilizar este rango de valores debido a que se encuentran por debajo de la frecuencia de Nyquist. FAST cuenta con un script que mediante funciones escritas con ayuda de la paquetería *ObsPy*, realiza, tanto la decimación, como el filtrado.

FAST requiere que se eliminen los espacios vacíos, o llamados de otra manera "*gaps*". Debido a que estos significan un periodo donde el sismómetro se encontraba apagado por problemas de energía (baterías) o que hubo un problema de transmisión de los datos (a veces faltan microsegundos o segundos). Estos espacios vacíos no pueden tener valores iguales a 0, debido a que, si hay muchos valores continuos igual a 0, se pueden confundir con los momentos en los que el sismómetro obtiene el valor de 0 por sí sólo y así generar falsas detecciones. Esto hace que se obtenga un mayor número de falsas detecciones, lo que abruma al programa y en consecuencia hace que tarde más tiempo en acabar de ejecutarse. Por lo que es necesario remover los valores continuos de 0's. FAST incluye un script que permite llenar esos espacios llenos con 0 de ruido no correlacionado, el cual no influye en las detecciones que hace el programa.

De igual manera, después de hacer los tres pasos anteriores, puede que sea necesario: eliminar glitches (errores instrumentales), hacer despiking (quitar picos de la señal) u otras operaciones con la señal. Se recomienda realizar estas operaciones como sea conveniente. Sin embargo, para obtener mejores resultados sólo son necesarias las primeras tres operaciones.

## **Extracción de características**

Después del preprocesado, sigue el paso de extracción de características. En pocas palabras esta etapa de FAST, recibe datos continuos, a los cuales segmenta en ventanas que se pueden sobrelapar entre ellas con un pequeño espacio de ventaneo entre ellas. Para obtener como salida una huella binaria por cada ventana que se genera, tal como se muestra en la *Figura 17*.



*Figura 17. Esquema general de etapa de extracción de características. Adaptada de la guía de usuario de FAST.*

El objetivo de la etapa de extracción de características es transformar a la forma de onda, de forma que se obtenga una representación única sin modificar los aspectos de la forma de onda que están asociados a ella. Esto es asociable a la manera en la que cada persona tiene una huella dactilar única con la que a partir de ciertos datos biométricos un sistema puede determinar la identidad de una persona. Los pasos que sigue FAST para obtener la “huella dactilar” de una forma de onda son los mostrados en la *Figura 18*.

## Extracción de características

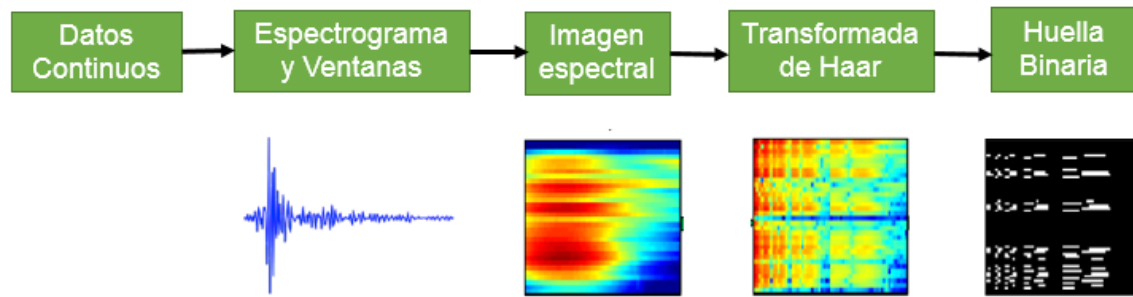


Figura 18. Esquema detallado de etapa de extracción de características. Adaptado de guía de usuario de FAST.

FAST realiza un ventaneo a la señal, dicho de otra forma, segmenta en ventanas de tiempo determinadas por el usuario. Para esto, se utiliza un retardo (lag) o un tiempo entre ventanas igual determinado por el usuario. Estas ventanas de tiempo tienen que ser del tiempo adecuado con respecto a las señales que se quieran detectar. En otras palabras, no tiene que ser muy corta, porque de esta manera la onda P y la onda S podrían no ser detectadas; ni muy larga, porque de otra manera se agregaría ruido.

Se tienen que utilizar estos parámetros para todas las estaciones y para todas las componentes. El paso de generación de ventanas es de los más importantes del programa, debido a que el número de ventanas será igual al número de imágenes espectrales. A su vez, es el mismo que el número de huellas binarias. Lo que hace que el número de ventanas sea directamente proporcional con el tiempo de ejecución y con la precisión del programa. Por lo que, ventanas cortas con un lag entre ventanas corto, harán que el programa sea más preciso con las detecciones teniendo como costo un tiempo de ejecución más largo. La manera de realizar el cálculo del número de huellas que serán generadas es el siguiente:

$$\#huellas = \left[ \frac{\#muestras - (\text{ancho de ventana} - \text{lag})}{\text{lag}} \right]$$

Para el caso de este trabajo, se tomó mes y medio de datos. En este trabajo se utilizó un valor de duración de ventana de 12.4 segundos y un lag de ventana de 1 segundo. Recordando que después de la operación de decimación, la frecuencia

de muestreo es de 100 Hz. Se tiene entonces un aproximado de 3,916,840,000 muestras por componente, en toda la base de datos. Considerando que se tomaron ventanas de 1,240 muestras y 100 muestras para lag, se obtuvieron alrededor de 3,916,786 huellas binarias por componente.

Una vez que se obtienen las ventanas, se procede a obtener la imagen espectral de una de ellas. FAST realiza esta operación aplicando una función taper de ventana de Hamming, que consiste en una función cuyo objetivo es suavizar la señal. Después se calcula la transformada de Fourier, mediante la Transformada de Fourier de Tiempo Corto (STFT) de cada ventana. La STFT es conveniente debido a que se toman ventanas de la señal, por lo que el contenido espectral de cada una es diferente, haciendo el cálculo de la frecuencia con ese segmento de tiempo. A diferencia de la transformada de Fourier convencional que calcula la frecuencia usando todo el tiempo de la señal (Kehtarnavaz, 2008).

Una vez calculada la STFT, se obtiene la potencia de cada ventana, esto se hace elevando al cuadrado la amplitud de la STFT compleja de la ventana. En este paso se hace un remuestreo a la imagen espectral. El valor de remuestreo tiene que ser el de una potencia de 2, esto se hace para un cálculo posterior. Yoon (2015) destaca que el valor de la potencia calculada para cada imagen espectral va a ser diferente en todos los casos, sin embargo, los valores de potencia más grandes van a corresponder a terremotos. Igual menciona que este criterio se ha utilizado para hacer detecciones, no obstante, FAST hace todavía otros cálculos, que serán mencionados a lo largo del capítulo, para obtener detecciones.

Se emplea la transformada de ondícula de Haar, la cual es una herramienta que se encarga de obtener una imagen con menor resolución a partir de una imagen original, conservando coeficientes que mantienen información de la imagen original y a partir de los cuales es posible reconstruirla (Stollnitz, 1995). Esto se hace en cada imagen espectral, para remuestrear nuevamente la señal obteniendo una imagen espectral con dimensiones de  $remuestreo1 * remuestreo2$ . Donde el primer remuestreo representa la longitud en muestras de la imagen espectral y el segundo representa el ancho en muestras de la imagen espectral.

Una ondícula, es una herramienta matemática que descompone los datos jerárquicamente en forma de promedio general en niveles sucesivos de detalle al mostrar desviaciones de la forma general.

El objetivo de hacer la transformada de ondícula de Haar es obtener la representación de la ondícula de la ventana. Esto facilita la compresión de los datos al mismo tiempo que se mantiene robusta a perturbaciones por ruido, cumpliendo con el objetivo de esta etapa. Otra de las razones por la cual se usa la transformada de ondícula es que, a diferencia de la transformada de Fourier, que tiene funciones localizables sólo en el dominio de la frecuencia. La transformada de Haar tiene funciones localizables tanto en el dominio de la frecuencia como en el dominio del tiempo. Lo que significa que se obtienen más coeficientes para poder describir señales muy impulsivas como terremotos.

Una vez obtenida la representación de ondícula de cada imagen espectral el siguiente paso es comprimir más los datos, para reducir el tamaño de almacenamiento de la imagen espectral. Lo que se hace al seleccionar coeficientes de ondícula, tomando un porcentaje pequeño, y descartando los demás. Dado que la mayor parte de la señal es ruido, se espera que los coeficientes correspondientes a terremotos se desvíen a los del promedio. Por lo que se escoge a los k-número de coeficientes más grandes. El número total de coeficientes va a ser igual a  $remuestreo1 * remuestreo2$ . Los coeficientes son colocados en una matriz de dimensiones  $M \times N$  donde M es el número total de coeficientes y N es el número total de huellas binarias.

En cada renglón N de la matriz se va a calcular una puntuación para cada coeficiente M relacionada al promedio y a la desviación estándar del renglón. Una vez calculada la puntuación para cada coeficiente por renglón se escoge un porcentaje del total de coeficientes que serán los k-coeficientes con la mayor puntuación, sin importar que el coeficiente asociado tenga valores positivos o negativos. La guía de usuario recomienda utilizar como valor k, un 15-30% del número total de coeficientes.



Una vez que se obtienen los k-coeficientes, el resto de los coeficientes se convierte en cero y los k-coeficientes obtienen un valor de +1 o -1 dependiendo si son positivos o negativos respectivamente. Guardar el signo del valor en lugar del valor, permite comprimir aún más los datos sin degradarlos con ruido.

Finalmente utilizando algoritmos de *Locality Sensitive Hashing* o “LSH”, los cuales son métodos ampliamente utilizados para búsquedas de vecino cercano en bases de datos con alta dimensionalidad, se obtiene una huella binaria dispersa única para cada imagen espectral, que consiste en una secuencia de unos y ceros. Se minimiza la cantidad de bits requeridos ya que cada huella binaria usa el doble de bits que coeficientes de Haar.

Después de obtener todas las huellas binarias de la base de datos, FAST calcula un valor de índice global, el cual es una forma de hacer referencia en donde se encuentran en el tiempo cada huella, para utilizarlo en la siguiente etapa.

### **Búsqueda de similitudes**

El proceso de búsqueda de similitudes realiza una búsqueda aproximada, tomando como parámetro de búsqueda una forma de onda, reducida a huella binaria como resultado de la etapa anterior. Después ingresa a una base de datos con todas las huellas binarias disponibles y obtiene como resultado una huella binaria similar, es decir una forma de onda similar.

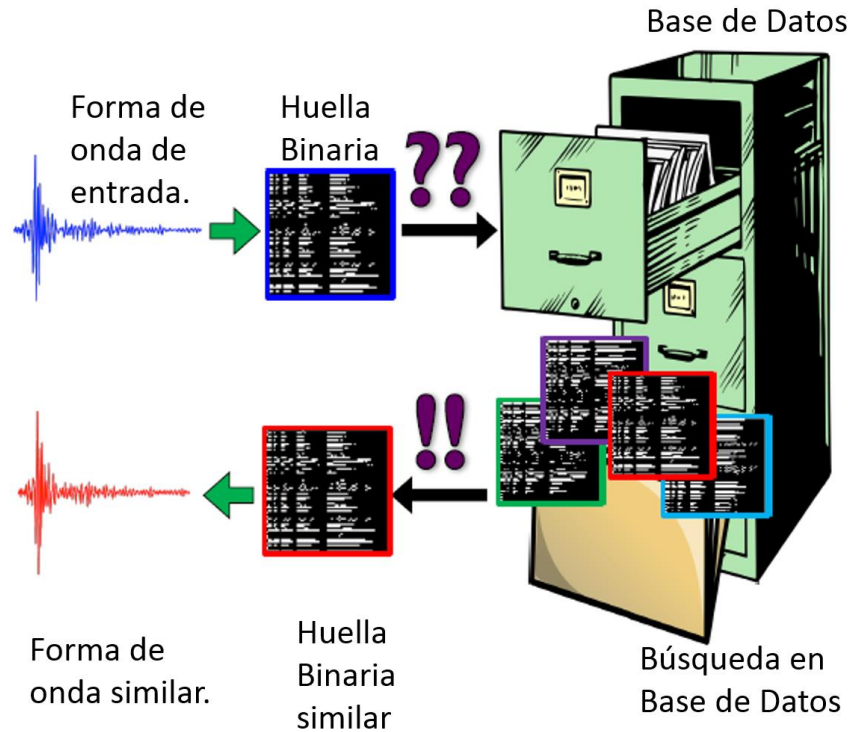


Figura 19. Esquema general de etapa de búsqueda de similitudes. Adaptada de la guía de usuario de FAST.

Es posible explicar esta etapa como una búsqueda en un archivero (Figura 19), donde se va a buscar un archivo similar a un archivo que se tiene en mano. Pero sólo buscando en ciertas carpetas donde podría encontrarse el archivo, en lugar de buscar en todas carpetas que existen. Sin embargo, la explicación tiene más detalles. El propósito de la Figura 20 es realizar un esquema de la manera de trabajar de esta etapa del programa incluyendo a las operaciones que se realizan:

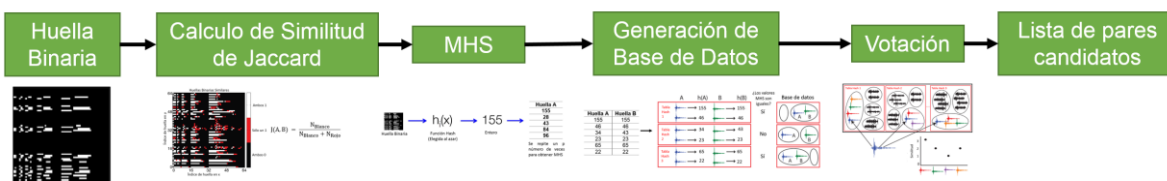


Figura 20. Esquema detallado de etapa de búsqueda de similitudes. Adaptada de la guía de usuario de FAST.

Para empezar a hablar de la búsqueda de similitudes, es necesario hablar del hasheo. Este consiste en operaciones que sirven para añadir, buscar o quitar objetos en bases de datos, esto hecho de una manera eficiente. FAST, en esta etapa vuelve a utilizar técnicas de hasheo LSH. El método de LSH es útil para esta etapa, debido a que al buscar conjuntos de pares similares omite comparar los pares

que no son similares. Retomando la *Figura 19*, y la analogía establecida con el archivero, FAST organiza las huellas binarias similares en carpetas, por lo que se realizará la búsqueda de la huella deseada sólo en esas carpetas, en lugar de hacerla en toda la base de datos.

El primer paso que se realiza después de la extracción de características es que se agrupan las huellas binarias que tienen la mayor probabilidad de ser similares, dentro del mismo grupo. La manera en la que se mide esta similitud es con la Similitud de Jaccard, la cual toma dos huellas binarias similares y las sobrelapa, comparando los valores binarios de la huella. La operación se realiza de la siguiente manera:

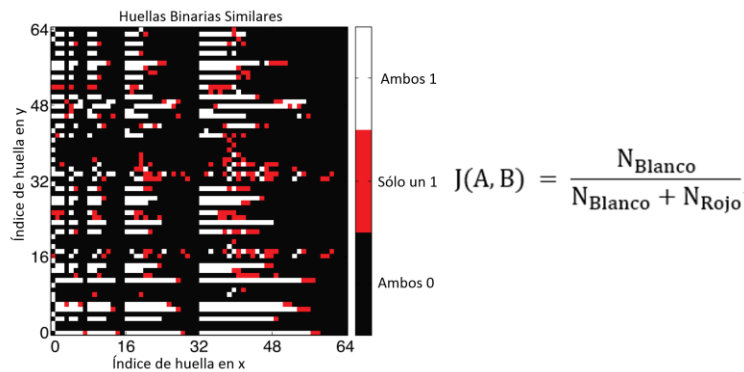


Figura 21. Cálculo de la similitud de Jaccard. Adaptada de la guía de usuario de FAST.

donde en la fórmula de la similitud de Jaccard, el numerador es el número de elementos donde en ambas huellas hay 1 (blanco), dividido entre el número de elementos donde en ambas sea 1 (blanco) más el número de elementos donde en sólo 1 elemento haya un 1 (rojo). Los espacios en negro corresponden a donde en ambas huellas hay 0, como se puede observar en la *Figura 21*.

Un paso intermedio es asociar una función aleatoria Min-Hash  $h(x)$ , donde  $x$  corresponde a la huella binaria, con el objetivo de obtener un valor entero asociado a cada huella. Este proceso se hace un “p” número de veces, para obtener la Firma de Min-Hash (MHS), de cada huella binaria. En la *Figura 22* se ve un ejemplo de esta operación tomando el valor de “p” como 5.

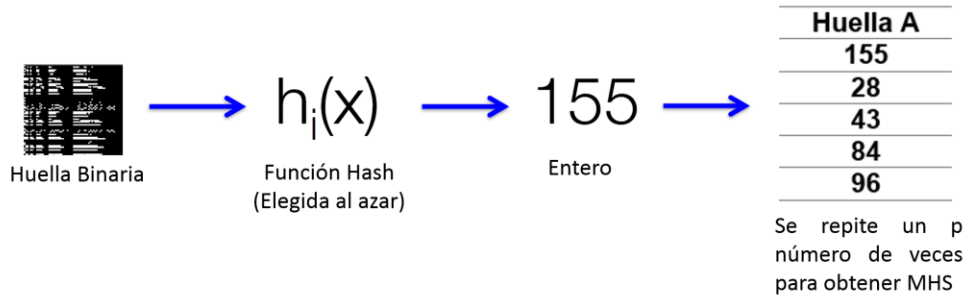


Figura 22. Generación de MHS con  $p = 5$ . Adaptada de la guía de usuario de FAST.

El número “p” o el número de objetos en MHS es un parámetro que es calculado indirectamente. Más adelante se explicará la manera en la que se obtiene este número. La razón por la que se hace una MHS por huella es para poder compararlas posteriormente, ya que una de las propiedades de LSH que tiene Min-Hash, establece que a partir de la probabilidad de que existan dos MHS similares se puede usar para estimar la similitud de Jaccard. Es decir,

$$Pr[h(A) = h(B)] = J(A, B).$$

Es importante señalar que esta relación conserva la similitud entre la huella A y la huella B de una manera probabilística. Esta operación se realiza al contar los enteros que sean idénticos entre las huellas A y B, como se mostrará en la *Figura 23*. Posteriormente dividiendo este número entre “p”, por lo que esta estimación mejorará a medida que el número “p” incremente (Yoon, 2015).

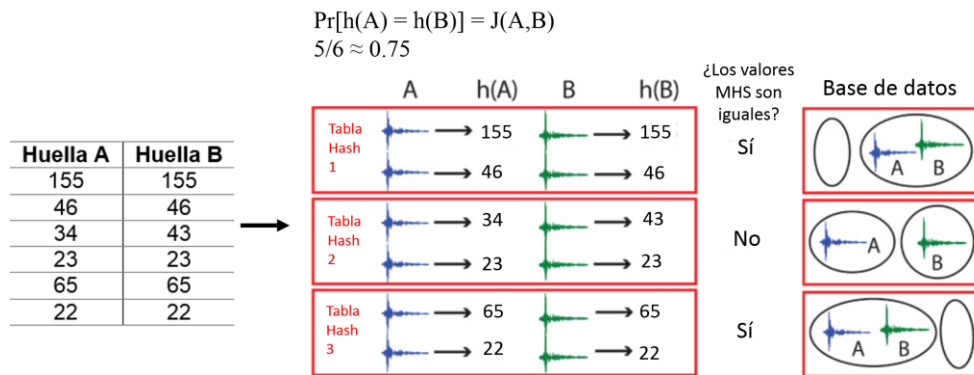
Posteriormente, LSH decide como insertar cada huella dentro de la base de datos. Esto lo realiza comparando los enteros dentro de MHS por cada huella binaria similar, además de servir para realizar la estimación de la similitud de Jaccard, esto sirve para que LSH asigne a cada huella a un grupo de hash, correspondiente al parámetro de funciones por tabla hash, o número “r”. Dentro de la *Figura 23* este número representa a las carpetas dentro del nivel del archivero, retomando la analogía del archivero de la *Figura 19*, ya que es donde se almacenarán las referencias a las huellas.

El número “r” es el más sensible dentro del programa, ya que influye en la eficacia de las detecciones al igual que en el tiempo de ejecución. La guía de usuario establece qué a valores más grandes, hay más detecciones perdidas, menos

detecciones falsas y un tiempo de ejecución menor. Sin embargo, a valores más pequeños ocurre lo contrario, es decir, menos detecciones perdidas, más detecciones falsas y un tiempo de ejecución mayor. Por lo que el número  $r$  se escoge con base en el tamaño en tiempo de la base de datos. Para bases de datos de días y semanas se recomiendan valores bajos, mientras que para bases de datos de años se recomiendan valores altos.

El número de tablas de hash, o número “ $b$ ” es un parámetro que es puesto por el usuario que almacena los grupos propuestos por el número “ $r$ ”, se podría decir que son los niveles del archivero de la *Figura 19*.

La manera en la que se calcula el número  $p$  es realizar la multiplicación del número “ $r$ ” por el número “ $b$ ”. En la *Figura 23* se mostrará un ejemplo tomando estos valores como parámetros:  $r = 2$ ,  $b = 3$  y  $p = r * b = 6$  y considerando una similitud de Jaccard de 0.75. Para ilustrar cómo realiza FAST esta clasificación de huellas similares:

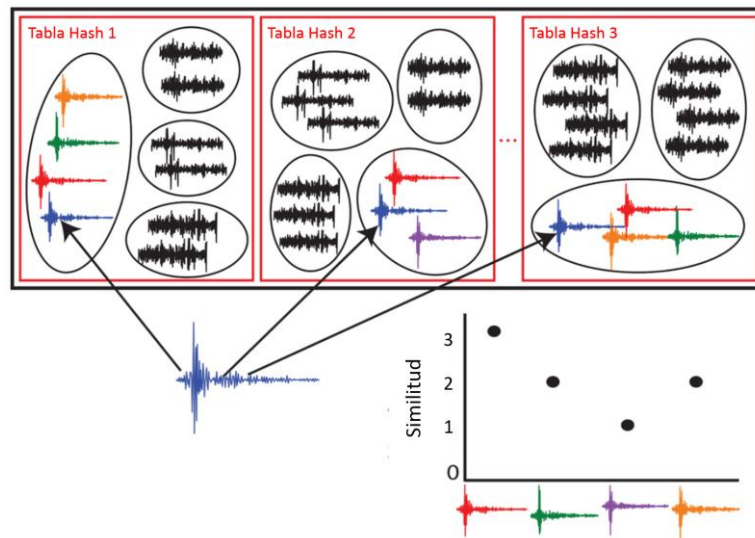


*Figura 23. Ejemplo de la manera en que LSH genera la base de datos. Con los parámetros:  $p = 6$ ,  $r = 2$  y  $b = 3$  y considerando una similitud de Jaccard de 0.75. Adaptada de la guía de usuario de FAST.*

De esta manera se comprueba de una manera probabilística que, si las huellas se parecen, es muy probable que sean agrupadas en el mismo lugar. Posteriormente se hace esto para cada una de las huellas de forma que, al final, todas y cada una de las 3,916,786 huellas binarias se encuentren representadas dentro de un grupo en todas las tablas. Es importante remarcar que estos pasos no almacenan a la huella en sí, almacenan una referencia numérica a la huella. Así es como LSH, produce múltiples agrupaciones de huellas dentro de los grupos.

Esta base de datos representa una manera fácil y eficiente de encontrar huellas binarias, así que, para poder encontrar sismos similares dentro de una base de datos continua. FAST utiliza a cada huella como un parámetro de búsqueda, para poder encontrar a todas las huellas que son similares a ella. Para cada huella que se va a buscar, se determina su grupo de hash usando el procedimiento marcado por la *Figura 23*. Esto para conseguir todas las referencias contenidas en los grupos, haciendo parejas, donde el primer miembro es la huella que se busca y el otro es una huella similar del mismo grupo en la base de datos.

El tiempo de ejecución de esta operación dependerá del número de objetos por grupo, debido a esto es recomendable tener pocos objetos por grupo para así ignorar las referencias a otros grupos (Yoon, 2015). Esta búsqueda se realiza por tabla, por lo que, es necesario un parámetro que indique cuantas parejas son encontradas en el mismo grupo por tabla dentro de la base de datos. Este parámetro es el número “v”, conocido como número de votos o Similitud FAST. El número “v” sirve como un umbral de detección para detecciones hechas con sólo una estación, en la guía recomiendan ponerlo con valores bajos primero y subirlo adecuadamente. Por lo que la salida de esta etapa es una lista de pares similares o en este caso de pares candidatos. En la *Figura 24* se muestra como realiza el programa este proceso:



*Figura 24. Búsqueda de pares similares dentro de la base de datos. Adaptada de la guía de usuario de FAST.*

A partir de una huella de búsqueda (forma de onda azul), se busca por cada tabla el grupo donde se encuentra el par más similar. Para cada huella en la base de datos. Se cuenta el número de tablas en donde aparezca la huella de búsqueda y una huella similar se encuentren dentro del mismo grupo. Será un par candidato únicamente si, el número de tablas en donde este par se encuentra en el mismo grupo supera o es igual al valor de  $v$ . Al final se regresa una lista de todos los pares candidatos dentro de la base de datos.

## **Postprocesado**

Por último, es necesario convertir la lista de pares candidatos de huellas similares a una lista de tiempos de detección. Existen diferentes rutinas dentro de FAST para realizar operaciones adicionales sobre la lista de pares candidatos obtenidos. Estas son principalmente para organizar las detecciones de manera que sea posible leer los resultados de manera ordenada, para el caso de este proyecto, sólo se utilizaron estas rutinas. La primera de estas rutinas es un script que consiste en acomodar las detecciones en una lista que incluye los siguientes valores: 1) el número de huella binaria correspondiente a la detección; 2) la cantidad de muestras que contienen a la detección (dL); 3) el número de pares que contienen a la detección (ndets); 4) el valor de similitud máxima de la detección (peaksum) y; 5) la suma de los valores de similitud de la detección (volume). Esta lista no contiene al tiempo de detección directamente. Por lo que, se utilizó otra rutina que convierte el número de huella a la hora en la que comienza la huella. Finalmente obteniendo como resultado la lista de detecciones.

Adicionalmente, FAST incluye scripts que permiten realizar búsquedas de señales similares sobre una red de estaciones, mediante un umbral de detección. Es decir, si un número de estaciones dentro de la red hizo la detección, se regresa una lista de detecciones con todos estos eventos que superan el umbral. Posteriormente, existen otros scripts que realizan filtrado de detecciones duplicadas en la red o si hay detecciones que se traslapen en tiempo.



## **CAPÍTULO 5: RESULTADOS**

Se ejecutó FAST con los parámetros mostrados en la *Tabla 3* (para realizar la extracción de características), y en la *Tabla 4* (para realizar la búsqueda de similitudes):

Para Huella Binaria:

<b>Parámetro de entrada</b>	<b>Valor</b>
<b>Tasa de muestreo</b>	100 [Hz]
<b>Longitud de ventana en serie de tiempo para espectrograma</b>	600 muestras
<b>Lag de ventana en serie de tiempo para espectrograma</b>	20 muestras
<b>Longitud de ventana espectral (largo de imagen espectral)</b>	32 muestras
<b>Lag de ventana espectral</b>	5 muestras
<b>Duración de ventana espectral</b>	12.4[s]
<b>k-número de coeficientes de ondícula para conservar</b>	200
<b>Ancho para imagen espectral</b>	32 muestras

*Tabla 3. Parámetros de entrada para extracción de características.*

Para Búsqueda de Similitudes:

<b>Parámetro LSH</b>	<b>Valor</b>
<b>Número de tablas (b)</b>	100
<b>Número de funciones Hash por tabla (r)</b>	6
<b>Número de elementos MSH (p)</b>	600
<b>Número de votos (v)</b>	10

*Tabla 4. Parámetros de LSH para búsqueda de similitudes.*

El programa con estos parámetros hizo un total de 2475 detecciones, lo que le llevó al programa un poco menos de dos horas en total, vale la pena remarcar que como se mencionó en el capítulo anterior este tiempo depende de: el número total de huellas y del número r. Ya que, si incrementan el número total de huellas o disminuye el valor del número r, FAST tomará más tiempo en ejecutarse.

Podrá parecer un número alto de detecciones, sin embargo, tomando en cuenta que son aproximadamente 45 días de registro, se tendría un promedio de 55 detecciones por día. Además, en las siguientes figuras (*Figuras 25 – 30*), se mostrarán algunas explicaciones con respecto a las detecciones. Estas figuras se hicieron utilizando el software libre llamado SWARM.

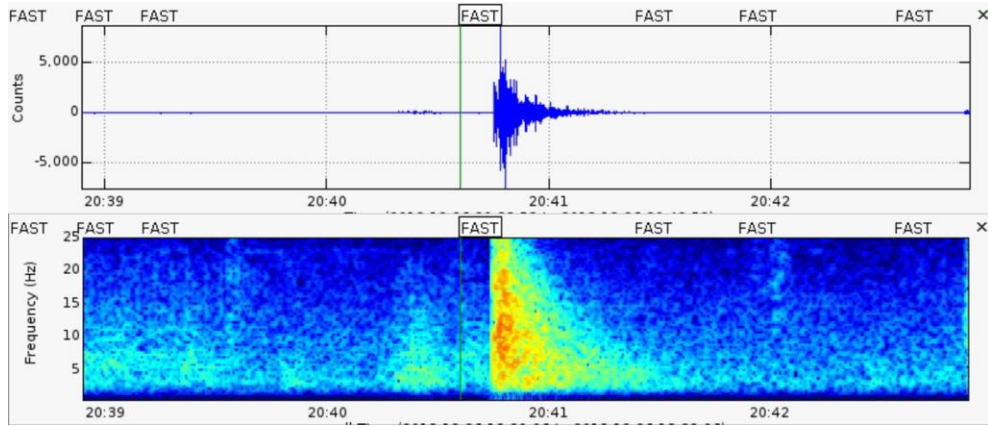


Figura 25: Ejemplo de detección de VT del día 26/12/2018, hecha con FAST.

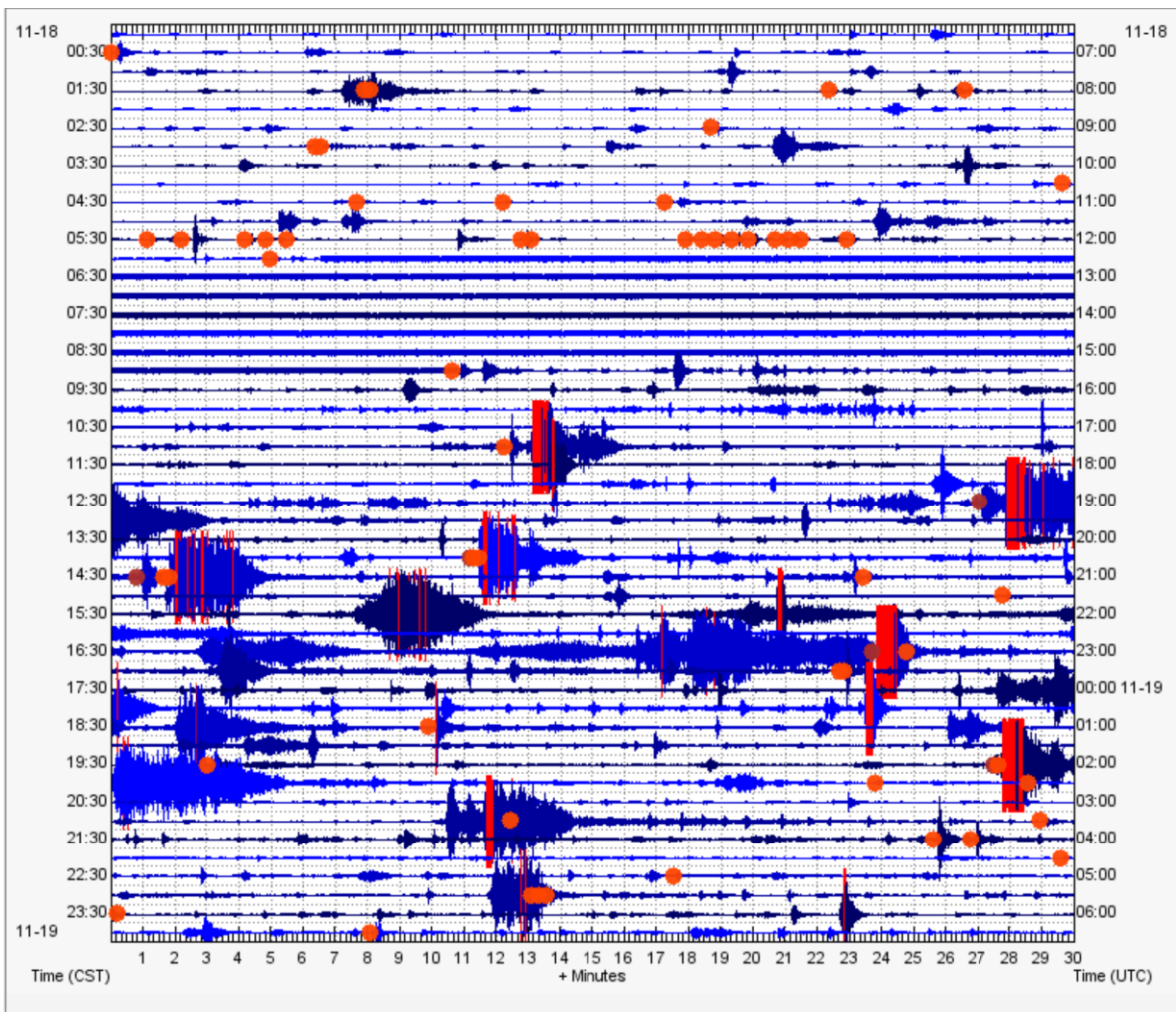


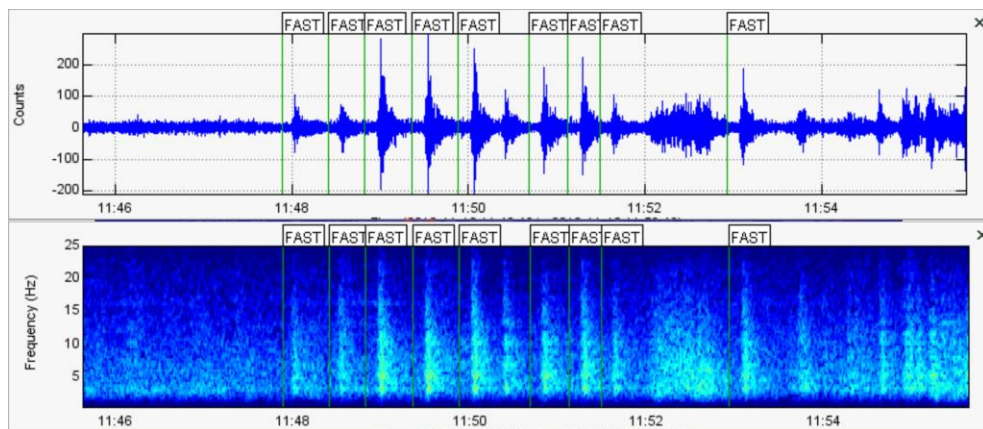
Figura 26. Ejemplo de detecciones hechas con FAST, donde los puntos naranjas son las detecciones realizadas por el programa y los puntos rojos son los reportados por CENAPRED.

Se observan señales grandes que no fueron detectadas, lo que es normal porque estas no corresponden a VT's (muchas veces son explosiones del volcán,

las cuales presentan un comportamiento que no se repite), se buscan detecciones que sean como la mostrada en la *Figura 25*. Por su lado, la *Figura 26* muestra un ejemplo de un día de detecciones (18/11/2018).

Las detecciones son pocas y corresponden a los VT's, que son pocos en el Popocatépetl. Eso es muy satisfactorio porque en la mayoría de los casos no hace detecciones que no sean de origen por alguna perturbación (lo que hubiera hecho un método como STA/LTA). Esto debido a que, como se explicó previamente, de la representación de ondícula de la ventana, sólo se elige a los coeficientes con valores más altos, que en la mayoría de los casos representan a sismos. Eso es un gran logro de FAST, debido a que encuentra a los VT's del volcán, ya que estos presentan contenidos espectrales similares por lo que las huellas generadas por estos eventos son similares.

Dentro de la *Figura 26*, alrededor de las 05:48 horario local (etiquetas de la izquierda), se puede observar una cadena de detecciones que al verla de cerca (*Figura 27*), enseña una cadena de detecciones de VT's. Este es un ejemplo del poder de hacer detecciones de señales similares usando FAST.



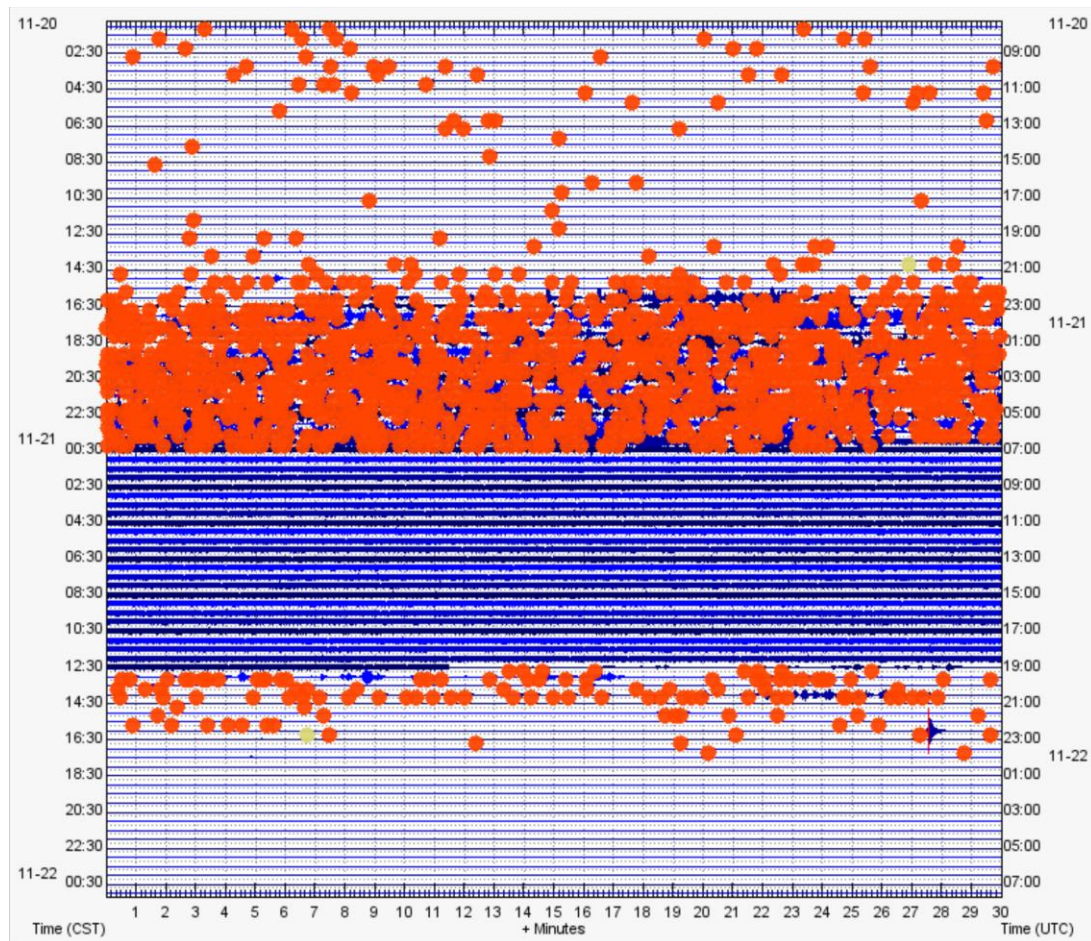
*Figura 27. Ejemplo de detecciones de VT's hechas con FAST.*

Estas detecciones no fueron reportadas por el CENAPRED, sin embargo, su alto contenido espectral y forma de onda, nos muestra que es posible que estas señales sean eventos asociados con el volcán, en específico VT's.

Se mencionó previamente que las detecciones hechas con un software no reemplazan a las hechas por un analista, más si le ayudan con esta tarea. FAST

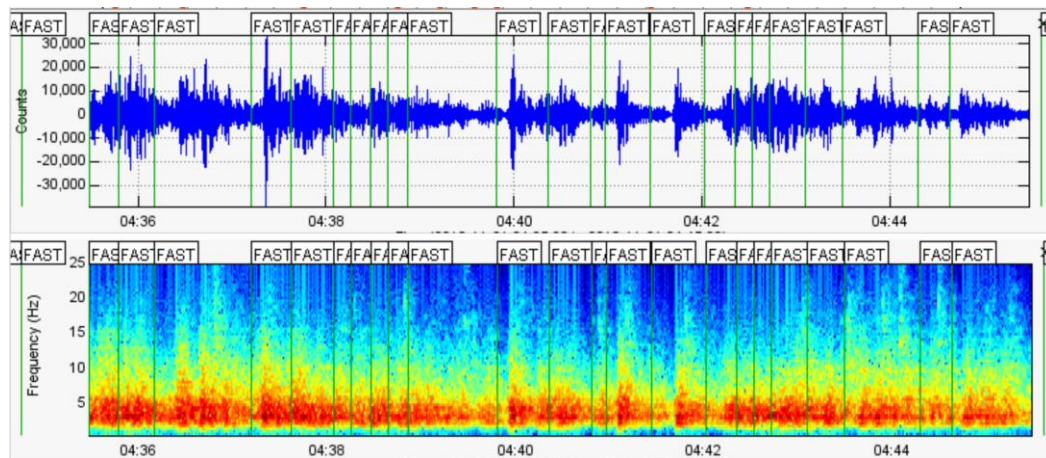
como software es capaz de realizar detecciones dentro de eventos como tremores. En el siguiente ejemplo (*Figura 28*) se mostrará esta habilidad de FAST, de realizar detecciones que se le puedan escapar al analista o a otros métodos de detección.

En la *Figura 28* se muestran las detecciones del día 20/11/2018, este día tiene muchas detecciones debido a que ese día ocurrió un tremor largo de duración de 1441 minutos (CENAPRED, 2018). Como el contenido espectral dentro de un tremor es similar, se hicieron detecciones. Principalmente sobre la etapa de mayor intensidad, que fue desde las 16:07 (horario local) hasta las 05:07 del día siguiente. Lamentablemente, en el registro de la estación PPPP, se le agotó la energía al sensor, por lo que no se pudo observar el fin de la alta intensidad del tremor en las detecciones. En la *Figura 28* el período donde se quedó sin energía el sensor es representado por ruido.



*Figura 28. Ejemplo de detecciones del día 20/11/2018 hechas con FAST. Donde los puntos naranjas representan a las detecciones hechas con FAST y el punto amarilla el inicio y final del tremor reportado por CENAPRED.*

La Figura 28 muestra como FAST realiza detecciones sobre todo el tremor, esto principalmente debido a que durante un evento de tipo tremor, el contenido espectral es sostenido por una cantidad larga de tiempo. Dadas las características de FAST, esto significa que se detectarían señales similares a lo largo de todo el tremor. En el caso de la *Figura 29* se muestra un ejemplo de las detecciones realizadas sobre el tremor, en estas se puede observar unos VT's contenidos dentro del tremor.



*Figura 29. Ejemplo de detecciones del día 20/11/2018 hechas con FAST.*

En la *Figura 29* se muestra como FAST realiza detecciones de VT's contenidos dentro del tremor. Por lo que si FAST es utilizado con datos monitoreo sísmico de un volcán ayuda al analista a realizar detecciones dentro de los eventos de tremor, lo que mejoraría los pronósticos del comportamiento de un volcán durante un tremor largo. De igual manera FAST sirve como indicador de la duración aproximada de estos eventos, incluyendo si tienen etapas de actividad intensa, ya que como se ve en la *Figura 28*, el ritmo con el cual FAST hace detecciones incrementa después de las 16:07(hora local), tal como se ve en la *Figura 30*.

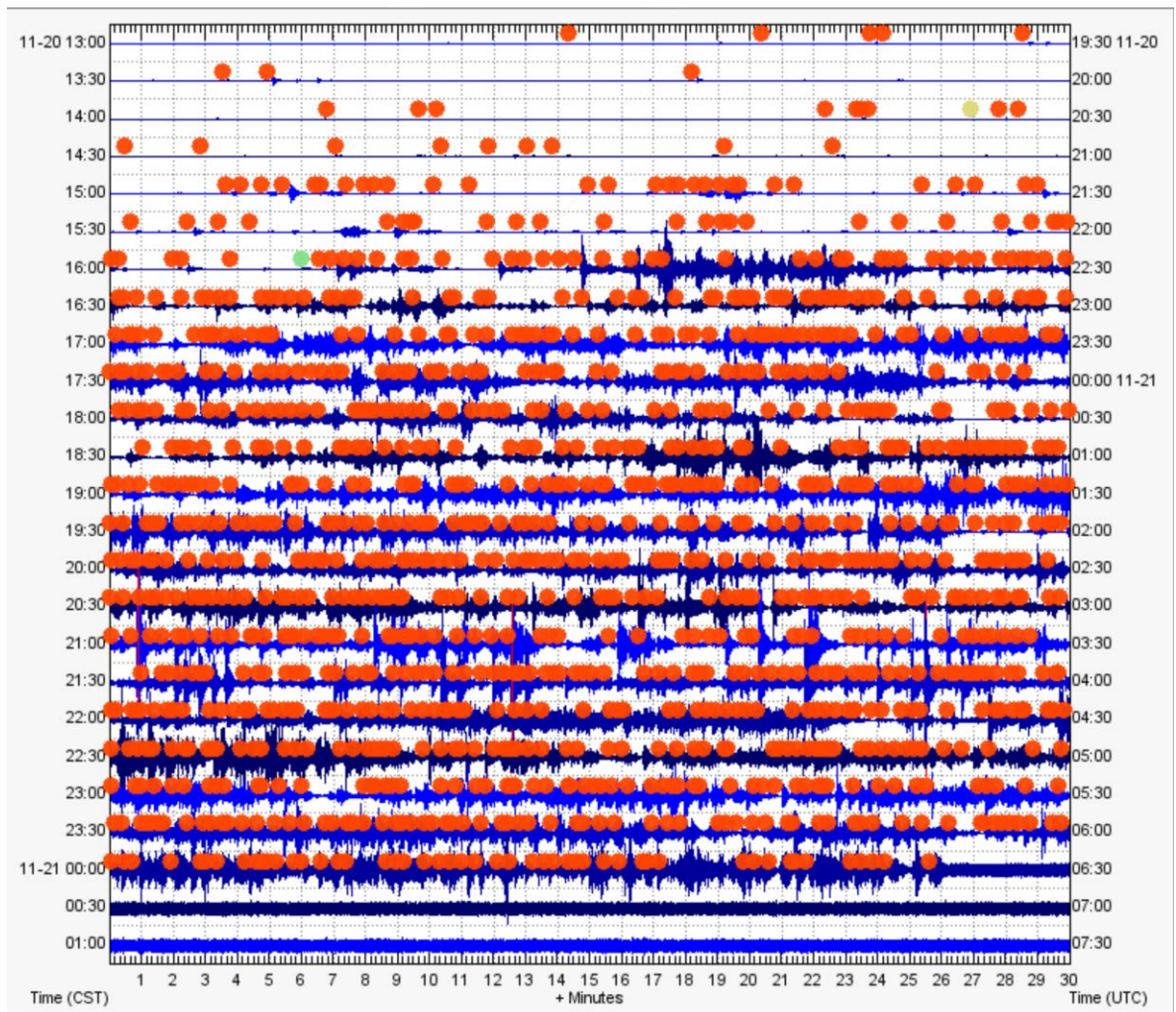


Figura 30. Detecciones sobre la actividad de mayor intensidad en el tremor del 20/11/2018 hechas con FAST. Donde los puntos naranjas representan las detecciones de FAST, el punto amarillo el inicio del tremor y el punto verde el inicio del periodo de mayor actividad.

Se discutirá en el capítulo 6 maneras de cómo mejorar el programa para detectar las otras señales volcánicas y sobre cómo se pueden clasificar de manera automática.

## **CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES**

La metodología FAST, representa una gran área de oportunidad para la detección automática. El programa fue inicialmente desarrollado para detectar sismos tectónicos. Debido a las características del funcionamiento de FAST, este trabajo propone usarlo para detectar sismos volcánicos. Como es un problema muy complejo, lo que se hizo en primera instancia fue detectar de entre todos los tipos de sismos volcánicos (VLP's, LP's, tremores, explosiones, híbridos, etc.) solamente los VT's (es decir las señales más parecidas a la meta inicial por la cual fue hecho FAST). Esa parte funciona bien, se han detectado VT's con un número mínimo de falsas detecciones (ruido), mientras que las señales que no fueron detectadas se debieron a que el programa observó que no existían suficientes señales similares. Esto se puede corregir de dos maneras: 1) disminuyendo el valor del número  $r$ , 2) disminuyendo el número  $v$ . Sin embargo, esto aumentaría el número de falsas detecciones.

Debido a que el algoritmo LSH que utiliza FAST está basado en métodos probabilísticos, con los cuales se puede reducir un segmento de forma de onda hasta llegar a representarla como un arreglo de enteros. Este arreglo, al ser comparado con otro arreglo de enteros, estima la similitud entre dos formas de onda. Se puede considerar otra manera de calcular la similitud entre las huellas, de manera que se obtengan mejores resultados. Partiendo después de la etapa de Extracción de Características, podría funcionar como alternativa al método de Búsqueda de Similitudes. Calcular el coeficiente de correlación entre huellas binarias, así como si fuesen imágenes. Esta alternativa presenta un problema, ya que en esta etapa trabajaría con las huellas binarias, en lugar de trabajar con sus MHS. Lo que hace que FAST no consuma más recursos que los que debe. Sin embargo, al hacer esto, confía mucho en la probabilidad de que la similitud de Jaccard será estimada correctamente en todos los casos.

El que FAST no realice clasificación de familias de eventos, da lugar a que en el futuro se implemente un algoritmo que, a partir de características o elementos de las huellas binarias, sea posible asignar etiquetas a las detecciones. Así logrando

una clasificación por familias de eventos. Una posible manera de hacerlo podría ser calculando el coeficiente de correlación usando huellas binarias, respectivas a cada familia de eventos como *templates* y considerando un umbral de detección, asignarle la etiqueta correspondiente. Otra posibilidad cae en implementar redes neuronales, al construir modelos con bases de datos de entrenamiento con las huellas binarias, para posteriormente hacer la clasificación de familias de eventos. Esta vez con una base de datos real y de esta manera poder evaluar su eficiencia.

De igual manera, una de las maneras para asegurar un mayor número de detecciones positivas es aplicar el programa sobre una red sísmica. Esto funciona con uno de los scripts dentro de postprocesado. En el cual mediante las detecciones que se hacen en cada estación se establece un umbral de modo que, si un número mayor de estaciones al umbral detectan al evento, FAST lo declara detección. Por lo que se podría implementar de manera que funcione sobre una red de estaciones alrededor del Popocatepetl. Con el objetivo de mejorar el monitoreo y los resultados del programa, ya que, al tener acceso a los datos de toda la red, se podría solucionar el problema de la falta de datos por cortes de energía.

FAST, puede realizar detección de eventos volcanotectónicos en una estación de monitoreo volcánico y por lo tanto dentro de una red. De manera aceptable y es posible realizar detecciones que se le pudieron escapar a un analista, de esta forma ayudándole con su trabajo. Sin embargo, el método no tiene la habilidad de clasificar estas detecciones dentro de las familias por sí mismo. Esta habilidad sigue siendo del analista, no obstante, por la manera en la que FAST realiza detecciones, se tiene la confianza que sólo se van a detectar eventos que sus características permitan la existencia de eventos similares.



## **BIBLIOGRAFÍA**

### **Libros:**

Kehtarnavaz, N. (2008). Digital Signal Processing System Design (Second Edition). Elsevier.

Schmincke, H.-U. (2004). Volcanism. Springer.

Ziemer et al. (1998). Signals & Systems. Continuous and Discrete (Fourth Edition). Prentice Hall.

Zobin, V. M. (2017). Introduction to volcanic seismology (Third edition). Elsevier.

### **Artículos:**

Allen, Rex. (1982). Automatic phase pickers: Their present use and future prospects. Bulletin of the Seismological Society of America 1982; 72 (6B): S225–S242. doi: <https://doi.org/10.1785/BSSA07206B0225>

Baluja et al. (2008). Waveprint: efficient wavelet-based audio fingerprinting. Pattern Recognition, 41(11), 3467–3480. <https://doi-org.pbidi.unam.mx:2443/10.1016/j.patcog.2008.05.006>

Beroza et al. (2017). Seismicity during the initial stages of the Guy-Greenbrier, Arkansas, earthquake sequence. Journal of Geophysical Research: Solid Earth, 122, 9253–9274. <https://doi.org/10.1002/2017JB014946>

Beyreuther et al. (2010). ObsPy: A Python Toolbox for Seismology, Seismological Research Letters, 81 (3), pp 530-533. [http://www.seismosoc.org/publications/SRL/SRL\\_81/srl\\_81-3\\_es/](http://www.seismosoc.org/publications/SRL/SRL_81/srl_81-3_es/)

Chouet, B. (1996). Long-Period volcano seismicity: its source and use in eruption forecasting. Nature. (1996) Vol 380, pp 309-315

Chouet, B. (2003). Volcano Seismology. Pure Applied Geophysics. 160 (2003), pp 739-788.

Cortés et al. (2021). Practical Volcano-Independent Recognition of Seismic Events: VULCAN.ears Project. Frontiers in Earth Science. Volume 8. Article 616676. doi: [10.3389/feart.2020.616676](https://doi.org/10.3389/feart.2020.616676)

De la Cruz-Reyna, S. y Siebe, C. (1997). The giant Popocatepetl stirs. Nature 388, 227 (1997). <https://doi.org/10.1038/40749>

Gibbons et al. (2006). The detection of low magnitude seismic events using array-based waveform correlation. Geophysical Journal International, 165(1), 149–166. <https://doi-org.pbidi.unam.mx:2443/10.1111/j.1365-246X.2006.02865.x>

Legrand et al. (2021) The influence of fluids in the unusually high-rate seismicity in the Ometepec segment of the Mexican subduction zone. Geophysical Journal International, 226, pp. 524-535. doi: [10.1093/gji/ggab106](https://doi.org/10.1093/gji/ggab106)

Malfante et al. (2018). Machine Learning for Volcano-Seismic Signals: Challenges and Perspectives. IEEE Signal Processing Magazine, Institute of Electrical and Electronics Engineers, 35 (2), pp.20 - 30. doi:[10.1109/MSP.2017.2779166](https://doi.org/10.1109/MSP.2017.2779166)

Malfante et al. (2020). A machine learning approach for automatic classification of volcanic seismicity at La Soufriere Volcano, Guadeloupe. *Journal of Volcanology and Geothermal Research*, <https://doi.org/10.1016/j.jvolgeores.2020.107151>

McNutt, S. R. (2005). Volcanic Seismology. *Annual Review of Earth & Planetary Sciences*, 33(1), 461-C-3. <https://doi-org.pbidi.unam.mx:2443/10.1146/annurev.earth.33.092203.122459>

Merchant, B et al. (2017). Next Generation Qualification: Nanometrics T120PH Seismometer Evaluation. United States. <https://doi.org/10.2172/1402551>

Omori, F., 1912. The eruptions and earthquakes of the Asama-yama. *Bulletin of the Imperial Earthquake Investigation Committee* 6, 1-147.

Stollnitz et al. (1995). Wavelets for computer graphics: a primer.1. *IEEE Computer Graphics and Applications*, 15(3), 76–84. <https://doi-org.pbidi.unam.mx:2443/10.1109/38.376616>

Trnkoczy, A. (1999): Understanding and parameter setting of STA/LTA trigger algorithm. - In: Bormann, P. (Ed.), *New Manual of Seismological Observatory Practice 2 (NMSOP-2)*, Potsdam: Deutsches GeoForschungsZentrum GFZ, 1-20. [https://doi.org/0.2312/GFZ.NMSOP-2\\_IS\\_8.1](https://doi.org/0.2312/GFZ.NMSOP-2_IS_8.1)

Withers et al. (1998). A comparison of select trigger algorithms for automated global seismic phase and event detection. *Bulletin of the Seismological Society of America* 1998; 88 (1): 95–106. doi: <https://doi.org/10.1785/BSSA0880010095>

Yoon et al. (2015). Earthquake detection through computationally efficient similarity search. *Science Advances*, e1501057, 1. <https://doi.org/10.1126/sciadv.1501057>

#### **Recursos:**

CENAPRED. Reportes del monitoreo de CENAPRED al volcán Popocatépetl. Del 16/11/2018 al 01/01/2019. Disponibles en: <https://www.cenapred.unam.mx/reportesVolcanGobMX/>

Equipo de desarrollo de ObsPy. Guía de usuario de ObsPy. Disponible en: <https://docs.obspy.org/tutorial/index.html>

Equipo de desarrollo de SWARM, USGS. Guía de usuario de SWARM. Disponible en: <https://volcanoes.usgs.gov/software/swarm/documentation.shtml>

Nanometrics. Nanometrics. Trillium 120 s Posthole Seismometer. Disponible en: <https://www.nanometrics.ca/products/seismometers/trillium-120-posthole>.

Yoon et al (2015). Guía de usuario de FAST. Disponible en: <https://github.com/claraeyoon/FAST>