

Universidad Autónoma de Baja California

Facultad de Ingeniería de Mexicali



“Relocalización de eventos sísmicos aplicando el método de *Source-Specific Station Terms* para el Valle de Mexicali y las Sierras Peninsulares de Baja California, México”

TESIS

que como requisito parcial para obtener el título de
INGENIERO CIVIL, presenta:

PAULA ALONDRA SALAZAR GUERRERO

Director de Tesis:

DR. ERIK ESTEBAN RAMÍREZ RAMOS

Mexicali, Baja California, México 2024

Resumen

El norte de Baja California se caracteriza por su alta actividad sísmica, producto de una compleja tectónica generada por la interacción entre las placas del Pacífico y Norteamericana. Por ello, ha sido crucial estudiar la sismicidad a través del tiempo, lo cual se ha logrado gracias al monitoreo realizado por la Red Sísmica del Noroeste de México (RESNOM) desde finales de 1970. La red ha evolucionado en instrumentación como en el manejo de la información. En sus inicios, esta contaba con una cobertura de estaciones limitada, principalmente para el área del Valle de Mexicali. Por consecuencia y aunado a las variaciones laterales de velocidad excluidas de los modelos, las incertidumbres en los parámetros hipocentrales de los sismos tienden a ser mayores en las primeras décadas de operación. Asimismo, dado el cambio en los softwares utilizados para localizar eventos sísmicos, el catálogo se encuentra actualmente en distintos formatos. Para abordar ambas cuestiones, se llevó a cabo la uniformización y relocalización del catálogo de RESNOM de 1979 al 2020. Lo primero se logró mediante la conversión de los archivos en su formato original al formato Nórdico (SeisAn; Havskov *et al.*, 2020) utilizando scripts escritos en Python a la par de un extenso trabajo de control de calidad que consistió en la adecuación de los archivos (en su formato original) para poder ser leídos por los scripts de conversión de manera automática. Para llevar a cabo la relocalización, se implementó el método de relocalización relativa *Source-Specific Station Terms* (Richards-Dinger y Shearer, 2000). Las relocalizaciones muestran una mejora en la distribución hipocentral ilustrando patrones más claros y una menor cantidad de eventos fijados a cierta profundidad. Aunque se puede apreciar una dispersión epicentral, comparado con el catálogo original, por lo general, los epicentros relocalizados siguen alineándose a los sistemas de falla principales. Tras el análisis de perfiles sísmicos relocalizados, en el sistema de fallas Cucapah-Indiviso se aprecia un incremento en la sismicidad entre 2010 y 2016, derivada de la ocurrencia del sismo del 4 de abril de 2010 El Mayor-Cucapah M_w 7.2. En dicho perfil se aprecia un patrón en el extremo sur de la falla Indiviso entre 5 y 15 km de profundidad aproximadamente; este se podría atribuir a una estructura geológica, sin embargo, se requeriría de un análisis más detallado utilizando otros estudios geofísicos y geológicos para determinarlo.

Palabras clave: Sismicidad, relocalización, catalogo sísmico, Valle de Mexicali, Sierras Peninsulares de Baja California, método *Source-Specific Station Terms*.

Abstract

The north of Baja California is characterized by its high seismic activity, product of the complex tectonic and geologic settings generated by the relative movement between the Pacific and North American plates. Therefore, it has been crucial to study seismicity over time, which has been achieved thanks to the monitoring carried out by the Red Sísmica del Noroeste de México (RESNOM) since the late 1970s. The network has evolved both in instrumentation and in data management. In its beginnings, station coverage was limited, especially for the Mexicali Valley. This, in addition to unmodeled lateral velocity variations, has resulted in considerable uncertainties in earthquake hypocentral parameters for the first decades of the records. Likewise, given the change in the software used to locate seismic events, the catalog is available in different formats. To address both issues, the standardization and relocation of the RESNOM catalog from 1979 to 2020 were carried out. The first was achieved by converting the files in their original format to a Nordic format (SeisAn; Havskov *et al.*, 2020) using Python scripts as well as thorough quality control process that consisted of adapting the files (in their original format) to be read by the conversion scripts automatically. To relocate the events, the relative relocation method Source-Specific Station Terms (Richards-Dinger and Shearer, 2000) was implemented. The relocated seismic events show an improved hypocentral distribution that illustrate clearer patterns as well as a decrease in fixed depths. Although an epicentral dispersion can be seen, compared to the original catalog, generally, the relocated events still align with the main fault systems. After the analysis of relocated seismic profiles, the Cucapah-Indiviso systems shows an increased seismicity between 2010 and 2016, derived from the occurrence of the April 4, 2010, El Mayor-Cucapah earthquake Mw 7.2. Herein, a pattern can be observed at the southern end of the Indiviso fault at depths between 5 and 15 km approximately; this could be attributed to a geological structure, however a more detailed analysis using other geophysical and geological studies would be required to determine it.

Keywords: Seismicity, relocation, seismic catalog, Mexicali Valley, Peninsular Ranges of Baja California, Source-Specific Station Terms method.

Dedicatoria

Para mi hija Mila, que este trabajo sea una inspiración para ti y evidencie la importancia de la participación de las mujeres en la ciencia.

Agradecimientos

Al Dr. Erik Ramírez por confiar en mí y darme la oportunidad de trabajar a su lado y aprender de su experiencia.

La fuente de datos para la realización de esta tesis provino de la Red Sísmica del Noroeste de México.

Al Dr. José Antonio Vidal Villegas por facilitar la obtención de la Información necesaria para la realización de esta tesis.

A la Universidad Autónoma de Baja California por brindar apoyo económico para solventar mis gastos durante este proyecto mediante la Beca de Investigación (2022-1) y la beca de Tesis de Licenciatura (2022-2).

Al Instituto de Ingeniería por brindarme un espacio de trabajo y las herramientas necesarias para llevarlo a cabo.

Al United States Geological Survey (USGS) por permitir mi participación en el “Taller de Pronóstico de Réplicas”.

Al EarthScope Consortium por otorgar la beca “EarthScope International Undergraduate Internship Program”.

A la Dra. Concepción Carreón por brindarme siempre su apoyo e integrarme a la comunidad del Instituto de Ingeniería desde el inicio.

A mi madre Rosalba Guerrero por darme su apoyo incondicional en cada aspecto de la vida para lograr desarrollarme personal y profesionalmente.

A mi padre Rubén Salazar por siempre creer en mí y abrirme las puertas a un sinfín de oportunidades; este un fruto más de tu esfuerzo y sacrificio en vida.

A Manlio Ávila por su paciencia, apoyo y amor incondicional a lo largo de mi desarrollo profesional.

Contenido

Resumen.....	ii
Abstract.....	iv
Dedicatoria.....	v
Agradecimientos.....	vi
Lista de figuras.....	x
Lista de tablas.....	xii
1. Introducción.....	1
1.1 Relocalización de sismos.....	1
1.1.1 Descomposición hipocentroidal.....	2
1.1.2 <i>HypoDD</i>	3
1.1.3 Source-Specific Station Terms.....	4
1.1.4 Source-Specific Station Terms Shrinking Box.....	5
1.2 Tectónica y sismicidad en Baja California, México.....	7
1.2.1 El Valle de Mexicali.....	9
1.2.2 Las Sierras Peninsulares de Baja California.....	10
1.3 Catálogo de RESNOM.....	11
2. Metodología.....	13
2.1 Formato de datos.....	13
2.2. Relocalización.....	20
3. Resultados y discusión.....	21
3.1 Nuevo catálogo.....	21
3.2 Perfiles.....	24
3.2.1 Centro de dispersión de Cerro Prieto.....	24
3.2.2 Cucapah-Indiviso.....	26

3.2.3 Dixieland	30
3.2.4 Imperial	31
3.2.5 San Miguel-Vallecitos	32
4. Conclusiones.....	33
5. Bibliografía	35
6. Anexos	42
Anexo I: Elementos de cada tipo de archivo	42
Anexo II: Descripción del archivo phaselist.....	48
Anexo III: Scripts de conversión	50
A. RESNOM.....	50
B. Hypo71	59
Anexo IV: Scripts para creación de archivos de entrada.....	71
A. Eventlist	71
B. Phaselist.....	76
C. Stationlist.....	87

Lista de figuras

Figura 1. (A) Histogramas obtenidos de Lin <i>et al.</i> (2007). (B) Perfiles 1-1' y 2-2' de la figura (B), obtenidos de Lin (2018)	6
Figura 2. (A) Mapa tectónico del norte de Baja California. (B) Sismicidad para el nBC reportada por RESNOM de 1979 a 2020.....	8
Figura 3. Distribución de estaciones que fueron utilizadas para la localización de sismos de: (A) 1979 a 1990, (B) 1990 a 2000, (C) 2000 a 2010 y (D) 2010 a 2020.....	12
Figura 4. Ejemplo de los formatos de los archivos de tiempos de arribo.....	14
Figura 5. Ejemplos de los archivos de entrada: eventlist, phaselist y stationlist.....	17
Figura 6. Modelos de velocidad utilizados para el cálculo de las relocalizaciones. En azul se representa el modelo de Castro <i>et al.</i> (2011) y en rojo el modelo Sierra97 de Frez <i>et al.</i> (2000).....	19
Figura 7. Eventos sísmicos de 1979 al 2020 relocalizados utilizando la técnica SSST.	22
Figura 8. Histogramas de profundidad para el catálogo de RESNOM para A) SPBC y B) MV y para las relocalizaciones C) SPBC y D) MV.....	23
Figura 9. Sismicidad en el centro de dispersión de Cerro Prieto (B-B') de 1979 a 2020 dada por (A) catálogo de RESNOM y (B) relocalizaciones utilizando SSST.....	25
Figura 10. Perfiles de eventos registrados de 1979 a 2020 del perfil Cucapah-Indiviso (C-C') de (A) RESNOM y (B) relocalizaciones utilizando SSST	28
Figura 11. Perfil de localizaciones del sistema Cucapah-Indiviso C-C; (Figura 10) utilizando SSST (A) el perfil de los eventos ocurridos entre 1979 a 2009, (B) los eventos ocurridos entre 2010 a 2016 y (C) eventos ocurridos entre 2017 a 2020.	29

Figura 12. Sismicidad para la falla (inferida) Dixieland D-D'. (A) Localizaciones del catálogo de RESNOM y (B) localizaciones obtenidas implementando SSST 30

Figura 13. Perfiles de la falla Imperial E-E' (A) localizaciones de RESNOM y (B) localizaciones obtenidas con SSST 32

Figura 14. Localizaciones de eventos para el sistema San Miguel-Vallecitos F-F'; (A) Localizaciones de RESNOM y (B) localizaciones obtenidas con SSST 33

Lista de tablas

Tabla 1. Modelo de velocidad “2ca” para el Valle de Mexicali	20
Tabla 2. Modelo de velocidad “Sierra97” para las Sierras Peninsulares de Baja California	20

1. Introducción

1.1 Relocalización de sismos

La localización de eventos sísmicos ha sido un tema de amplia discusión entre los sismólogos, quienes buscan constantemente aprovechar las nuevas tecnologías para lograrlo. La localización de sismos consiste en determinar un hipocentro (latitud, longitud y profundidad) y un tiempo de origen, así como el epicentro (proyección vertical del hipocentro a la superficie). Para ello, se comparan los tiempos de viaje observados (tiempo que tardan las ondas sísmicas en viajar desde su origen hasta la estación que la registra) registrados con los tiempos de viaje calculados de las ondas P y S para cada estación y se estiman los residuales (diferencia entre el tiempo observado y el calculado). Los tiempos de viaje calculados se obtienen utilizando modelos de velocidad para la región, mientras que los tiempos de viaje registrados se obtienen de las lecturas de arribos P y S de las estaciones sísmicas. Los residuales se minimizan mediante métodos estadísticos como los mínimos cuadrados, suma de diferencias, entre otros, hasta encontrar la localización más adecuada (Shearer, 2019). Sin embargo, existen diversos factores que sesgan estas localizaciones; algunas de las causas que generan estos sesgos son, por ejemplo, el suponer que las incertidumbres en los tiempos de viaje no pueden ser correlacionados entre estaciones distintas o excluir de los modelos de velocidad las heterogeneidades laterales presentes en la corteza, siendo el último el más común (Richards-Dinger y Shearer, 2000).

Existen métodos que buscan mitigar el efecto de estas heterogeneidades en los cálculos de las localizaciones empleando procesos iterativos que minimicen los residuales de un

grupo o cúmulo de eventos. Este análisis parte de que el valor de los residuales en los tiempos de viaje de eventos cercanos es constante para una misma estación ya que atraviesan aproximadamente la misma estructura de la corteza terrestre (Waldhauser, 2001).

De esta manera, se pueden analizar grupos de eventos y buscar correcciones en los tiempos de viaje. Estas correcciones pueden ser estáticas, es decir, que solo hay una corrección en el tiempo de viaje por estación, como en el método de *Static Station Terms* (Richards-Dinger y Shearer, 2000). Por otro lado, las correcciones pueden variar en función de la posición de la fuente (hipocentro), como es el caso de los métodos *Source-Specific Station Terms* (Richards-Dinger y Shearer, 2000) y *Source-Specific Station Terms Shrinking Box* (Lin y Shearer, 2005). Como resultado, se obtienen localizaciones relativas más precisas, a diferencia del cálculo de la localización absoluta (calculada o especificada dentro de un sistema geográfico fijo y una base de tiempo fija) de cada evento. Si la localización inicial, es decir aquella dada por un catálogo de eventos sísmicos, es cercana a la localización real, los resultados convergen rápidamente (Shearer, 2019). Existen también otras metodologías que se pueden seguir para relocalizar sismos, a continuación, se describirán algunas de ellas aunado a las mencionadas anteriormente.

1.1.1 Descomposición hipocentroidal

Este método fue propuesto y utilizado por Jordan y Sverdrup (1981) para localizar grupos de eventos en el océano. El método utiliza un enfoque similar al de localización del evento maestro y consiste en relocalizar eventos dentro de un cúmulo respecto al centroide de

este. La precisión de las relocalizaciones depende de qué tan acertada sea la localización del evento maestro. Este método fue aplicado por Lin y Shearer (2005) en datos sintéticos y concluyeron que, para un cúmulo de eventos, los resultados obtenidos son muy similares a los obtenidos utilizando *SSST* y *HypoDD* (Lin y Shearer, 2005).

1.1.2 *HypoDD*

HypoDD (Waldhauser, 2001) es una técnica que utiliza el algoritmo de doble diferencia (DD; Waldhauser y Ellsworth, 2000) para relocalizar eventos sísmicos. La técnica de DD considera que la trayectoria de los rayos (vector normal al frente de onda) entre dos sismos van a ser similares durante casi todo el trayecto considerando algunos factores: En primer lugar, la distancia entre los hipocentros de dos eventos provenientes de una misma región es menor a la distancia entre cualquiera de los eventos a una estación en común (Got *et al.*, 1994); en segundo lugar, se asume que la escala de las variaciones en las velocidades de onda es muy grande a comparación de la distancia hipocentral entre los eventos, así como la distancia entre cualquiera de estos eventos a una estación en común. *HypoDD* analiza el catálogo para derivar las diferencias en los tiempos de viaje (*double difference residuals*) de pares de eventos para una estación y posteriormente minimizar estos residuales mediante el método de mínimos cuadrados. Este método resulta útil para relocalizar eventos que se encuentren cercanos entre sí. Waldhauser y Ellsworth (2000) probaron este método relocalizando dos cúmulos de sismos a lo largo del norte de la falla de Hayward en California, EE. UU. Con los resultados concluyeron que aplicando el método *HypoDD* se disminuye la dispersión

presente en las localizaciones del catálogo original y provee imágenes más claras de la sismicidad.

1.1.3 Source-Specific Station Terms

Para este proyecto de investigación se utilizó el método *Source-Specific Station Terms* (SSST; Richards-Dinger y Shearer, 2000) para relocalizar los eventos sísmicos registrados en el norte de Baja California (nBC) desde 1979 hasta el 2020. Este método consiste en un proceso iterativo que requiere los tiempos de arribo de las fases P y S para calcular correcciones para cada par fuente-estación. Estas correcciones se obtienen calculando la mediana ponderada de los residuales (tiempo observado menos el calculado) de un número de eventos dentro de una esfera de un radio r_{\max} alrededor de un evento objetivo. Para este método, cada estación obtiene una función de corrección que a su vez varía en función de la posición de la fuente (Lin y Shearer, 2005; Richards-Dinger y Shearer, 2000). Con los SSST calculados se realizan las relocalizaciones y se itera hasta satisfacer la medida de convergencia deseada; para ello se asigna un umbral de valores (usualmente menor a 0.1 s) para los residuales y se establece el número de iteraciones (el valor por defecto son 10 iteraciones, aunque puede converger en menos, ej. Lin *et al.*, 2007). SSST es un método práctico y eficiente que mejora las localizaciones relativas de los sismos en regiones en las cuales existan variaciones laterales significativas de velocidad de corteza. Asimismo, los SSST calculados se ajustan a la densidad de los eventos, tanto en áreas grandes con escasa sismicidad, como en áreas pequeñas de sismicidad densa (Richards-Dinger y Shearer, 2000), por lo cual se

considera que se adapta bien a las condiciones de la sismicidad del norte de Baja California, la cual se explicará más adelante.

1.1.4 Source-Specific Station Terms Shrinking Box

El método *Source-Specific Station Terms Shrinking Box* (Lin *et al.*, 2007) es una versión mejorada del método SSST, introducida por Lin y Shearer (2005). Este método obtiene las funciones de corrección por estación (al igual que las del método Source-Specific Station Terms) con la diferencia que va reduciendo el radio de la esfera alrededor de un evento objetivo (r_{\max}) entre la primera y última iteración. Los autores aplicaron este método en una serie de datos sintéticos y determinaron que los resultados obtenidos son similares a los obtenidos utilizando el método de *Hypocentroidal Decomposition* (Jordan y Sverdrup, 1981).

En la Figura 1.A obtenida de (Lin *et al.*, 2007), se muestran los histogramas de distribución de profundidades para 6 catálogos de eventos del Sur de California; en ellos se comparan diversos métodos de relocalización como SSST, DD, correlación cruzada de forma de ondas, entre otros. En esta se observa que cuando se utiliza el método *Source Specific Station Terms* (Richards-Dinger y Shearer, 2000) se genera una curva mejor distribuida donde los eventos ocurren más frecuentemente entre ~0 y 15 km. La Figura 1.B consiste en una serie de perfiles de la sismicidad relocalizada en la falla de San Andrés (cerca de Parkfield, California) utilizando los métodos anteriormente mencionados; aquí se puede apreciar que implementando SSST los hipocentros tienden a constreñirse y formar patrones más claros en las secciones transversales (1-1' y 2-2') que los reportados en el catálogo del *Northern California Earthquake Data Center*

(NCEDC; Northern California Earthquake Data Center y UC Berkeley Seismological Laboratory, 2014).

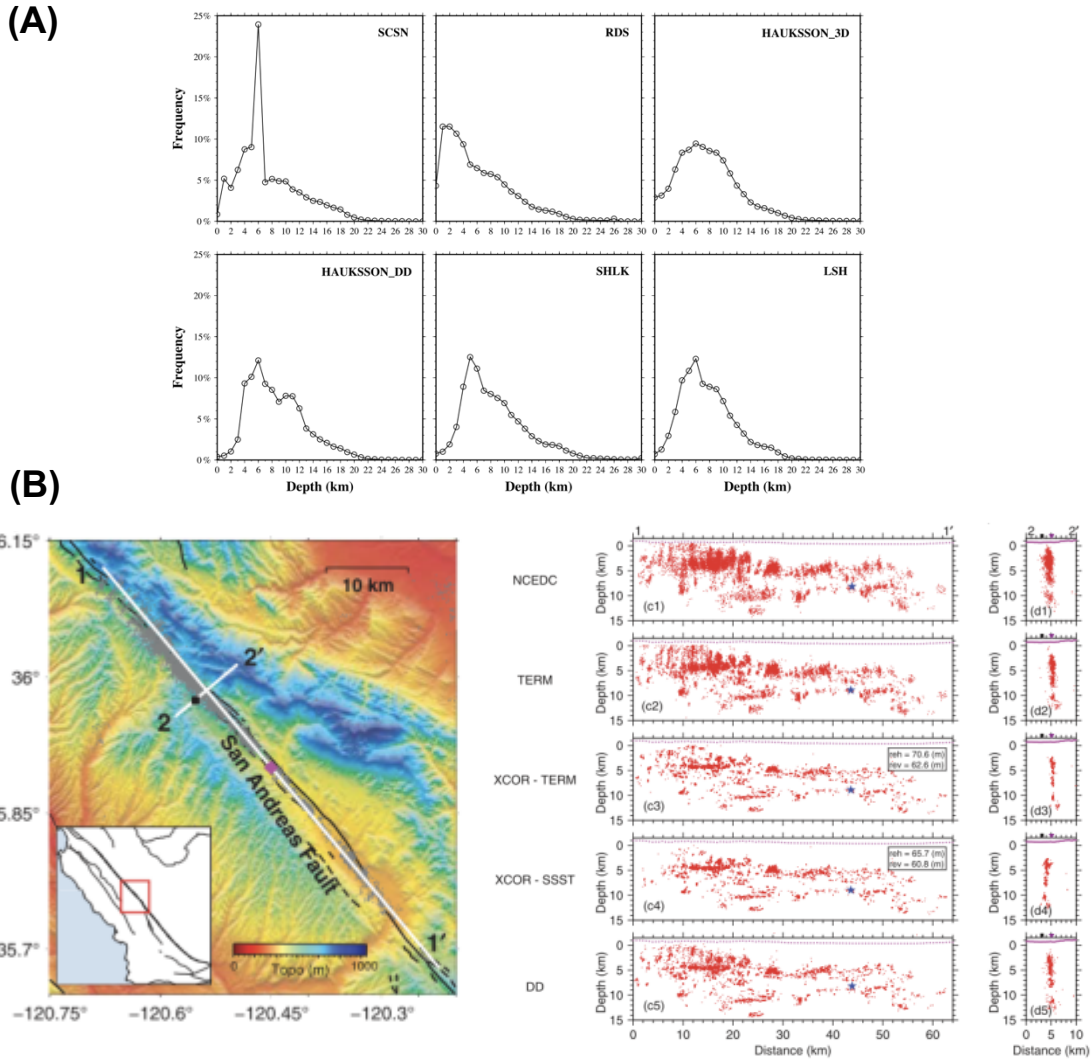


Figura 1. (A) Histogramas obtenidos de Lin *et al.* (2007) que muestran la distribución de las profundidades de 6 catálogos del Sur de California: SCSN, catálogo de la Red Sísmica del Sur de California utilizando un modelo 1D; RDS, eventos ocurridos de 1981 a 1998 utilizando el método SSST y un modelo 1D (Richards-Dinger y Shearer, 2000); HAUKSSON_3D, localizaciones de eventos de 1981 a 2000 utilizando un modelo 3D del Sur de California (Hauksson, 2000); HAUKSSON_DD, localizaciones de eventos ocurridos de 1984 a 2002 utilizando el método de doble diferencia y correlación cruzada de forma onda (Hauksson y Shearer, 2005); SHLK, localizaciones de sismos de 1984 a

2002 obtenidas utilizando el método SSST 1D, análisis de cúmulos de eventos similares y correlación cruzada de forma de ondas (Shearer *et al.*, 2005); LSH, localizaciones de eventos de 1981 a 2005 utilizando análisis de cúmulos, correlación cruzada de forma de ondas y tiempos diferenciales (Lin *et al.*, 2007). Los perfiles 1-1' y 2-2' de la figura (B), obtenidos de Lin (2018), muestran los resultados obtenidos relocalizando la sismicidad de la falla de San Andrés utilizando diversos métodos: NCEDC, localizaciones del catálogo del Northern California Earthquake Data Center; TERM, resultados implementando SSST con el cálculo previo de correcciones por estación; XCOR-TERM, correlación cruzada de forma de ondas con correcciones por estación; XCOR-SSST, correlación cruzada de forma de ondas sin correcciones por estación; DD, método de dobles diferencias.

Por otra parte, si lo que se desea es mejorar las localizaciones absolutas, se deben tomar en cuenta las variaciones laterales de velocidad utilizando un modelo de velocidad 3D (Richards-Dinger y Shearer, 2000). Actualmente, esto se puede lograr combinando el método SSST *Shrinking Box* junto al algoritmo de trazado de rayos 3D SIMULPS (Evans *et al.*, 1994; Lin *et al.*, 2007) o utilizando técnicas como GrowClust3D (Trugman *et al.*, 2023). Para ello se requeriría el análisis de las formas de onda.

1.2 Tectónica y sismicidad en Baja California, México

El norte de Baja California (nBC) se conforma principalmente por dos ambientes geológicos distintos: las Sierras Peninsulares de Baja California (SPBC) y el Valle de Mexicali (Lomnitz *et al.*, 1970), divididas por el Escarpe Principal del Golfo (Stock y Hodges, 1990). Las SPBC son de composición granítica mientras que el Valle de Mexicali (VM) consiste en una cuenca sedimentaria subdividida en dos cuencas: la Laguna Salada y el Valle de Mexicali.

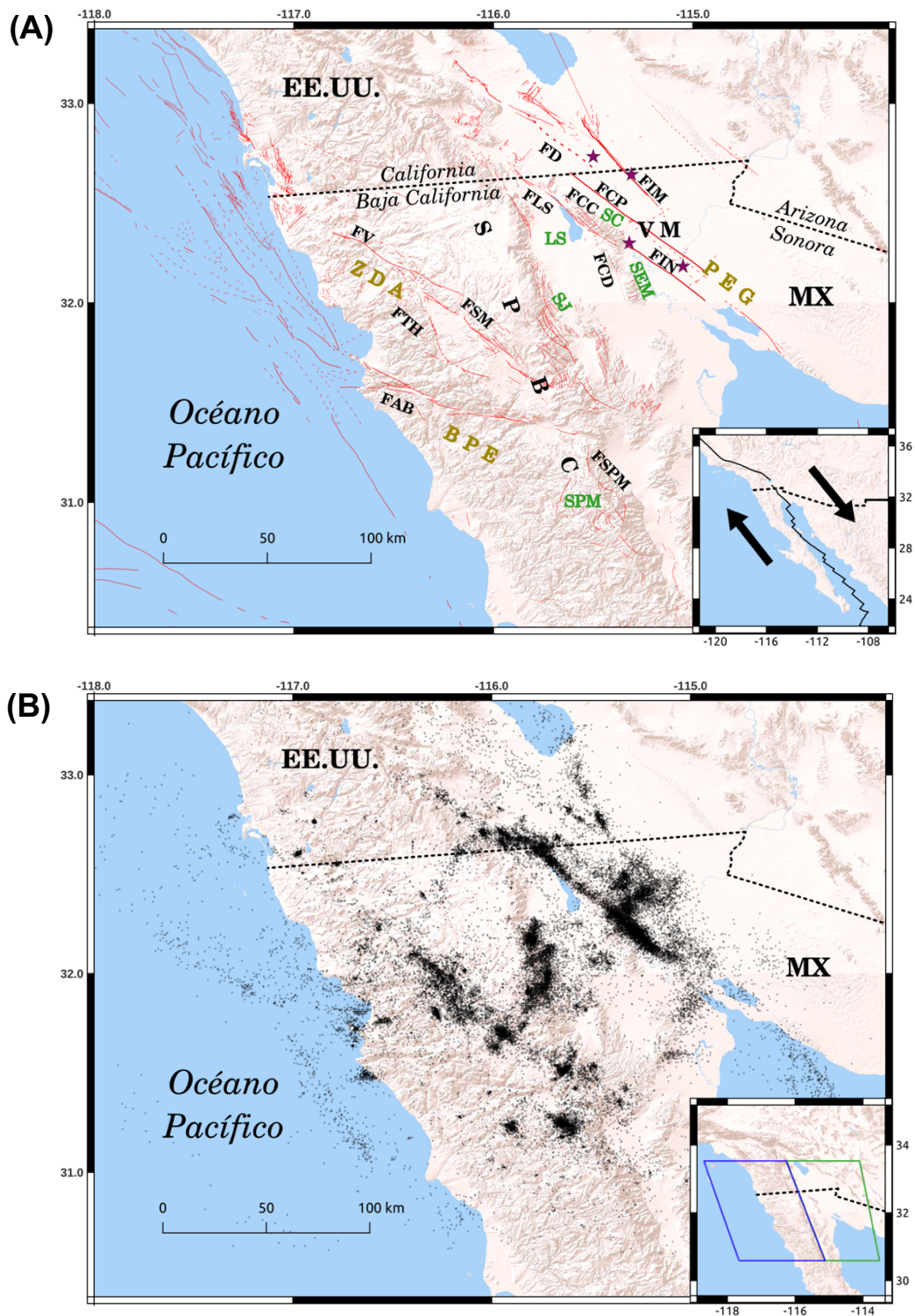


Figura 2. (A) Mapa tectónico del norte de Baja California. Las fallas se representan con las líneas rojas: FIM, falla Imperial; FD, falla Dixieland (inferida); FCC, falla Cucapah; FCP, falla Cerro Prieto; FLS, falla Laguna Salada; FCD, falla Cañada de David; FIN, falla

Indiviso; FV, falla Vallecitos; FSM, falla San Miguel; FTH, falla Tres Hermanos; FAB, falla Agua Blanca; FSPM, falla San Pedro Mártir. Las cuencas, sierras, provincias y zonas principales se indican de la siguiente manera: SEM, Sierra El Mayor; SC, Sierra Cucapah; LS, Laguna Salada; SPM, San Pedro Mártir. SJ, Sierra Juárez; PEG, Provincia Extensional del Golfo; ZDA, Zona de Deformación Activa; BPE, Bloque Peninsular Estable. Los sismos históricos se indican con las estrellas moradas. El mapa del recuadro inferior derecho muestra el límite de las placas del Pacífico y Norteamericana, así como la dirección del movimiento entre ambas. **(B)** Sismicidad para el nBC reportada por RESNOM de 1979 a 2020. El mapa del recuadro inferior derecho muestra la delimitación de los polígonos (para la separación de la sismicidad) de las SPBC y el VM, en color azul y verde, respectivamente.

1.2.1 El Valle de Mexicali

La cuenca del Valle de Mexicali está rellena por sedimentos provenientes del Río Colorado (5-6 km, Pelayo *et al.*, 1991; Ramírez *et al.*, 2021) y en ella se pueden encontrar importantes fallas como Imperial, falla Algodones, falla Cerro Prieto, falla Cucapah y falla Indiviso, esta última observada debido a la ocurrencia del sismo de 4 de abril de 2010 El Mayor-Cucapah M_w 7.2 (Ramírez-Ramos *et al.*, 2015). La extensión sur de la falla de San Andrés, San Jacinto y Elsinore en California, EE. UU., en conjunto con las fallas Imperial y Cerro Prieto, son responsables del 86% del movimiento total (49 ± 3 mm/año) entre las placas del Pacífico y Norteamérica (Bennett *et al.*, 1996). La mayoría de los sismos $M > 6$ ocurren en el sistema de fallas Imperial-Cerro Prieto, mismas que se localizan dentro del VM y se desplazan a velocidades de 35 ± 2 mm/año y 42 ± 1 mm/año respectivamente (Bennett *et al.*, 1996). Ejemplos de sismos históricos en el VM son: El Centro 1940 ML = 7.1, Valle Imperial 1979 ML = 6.9, secuencia de la Ciénega de Santa Clara 1934 ML = 6.5 y 7.1, 1980 Victoria ML = 6.1 y el Mayor-Cucapah M_w = 7.2 (Figura 2A),

este último causó daños estructurales considerables a las infraestructuras de la ciudad de Mexicali (Meneses y Kleinfelder, 2010; Nava y Glowacka, 1999).

El Valle Mexicali se delimita al oeste por las sierras Cucapah y El Mayor; ambas cadenas montañosas resultan del levantamiento del basamento Mesozoico y son rodeadas por las fallas de la Laguna Salada y Cañada de David (Fletcher y Spelz, 2009; Ramírez-Ramos et al., 2015).

La Laguna Salada es un semigraben controlado por una falla del mismo nombre ubicada al este, así como por la falla de la Cañada David. La falla Laguna Salada es una falla de alto ángulo (buzamiento $> 45^\circ$, The American Geological Institute, 1984), con un rumbo hacia el noroeste y un corte oblicuo normal-dextral, mientras que la falla de la Cañada David se caracteriza por ser una falla por desprendimiento de bajo ángulo (buzamiento $< 45^\circ$, The American Geological Institute, 1984; Fletcher y Spelz, 2009). Al este de la Laguna Salada se encuentran también las cadenas montañosas Sierra Cucapah y Sierra El Mayor, mientras que al oeste está delimitada por la Sierra Juárez (Figura 2).

1.2.2 Las Sierras Peninsulares de Baja California

Las SPBC son, en su mayoría, de composición granítica, y exhiben una diversa y compleja geología; estas han experimentado procesos geológicos significativos incluyendo orogenia asociada a magmatismo de arco durante la era Mesozoica, así como fallamiento transformante asociado a un rift continental durante la era Cenozoica (Lewis et al., 2001). Actualmente, se distinguen por sus diversos sistemas de falla de deslizamiento

asociadas a una tectónica compleja. Las SPBC se pueden dividir en dos dominios estructurales distintos: Zona de Deformación Activa y el Bloque Peninsular Estable (Lewis *et al.*, 2001). La Zona de Deformación Activa incluye las fallas Elsinore, San Jacinto y Agua Blanca. El Bloque Peninsular Estable está separado de la Provincia Extensional del Golfo por el Escarpe Principal del Golfo, y está controlado por el sistema de fallas de San Pedro Mártir al sur y por la falla Agua Blanca al norte. Algunos de los eventos históricos más representativos para las SPBC son: el sismo ocurrido en 1963 en la intersección de las fallas San Miguel y Vallecitos $M_L = 5.8$; una secuencia de 6 sismos que ocurrieron en la falla de San Miguel en 1954 y 1956 $M_L > 6$; el sismo en la falla Vallecitos 1949 $M = 5.7$ (Frez *et al.*, 2000; Stock *et al.*, 2022).

Estas complejas estructuras geológicas y su comportamiento se manifiestan a través de la actividad sísmica en las SPBC y VM, misma que se ha registrado y estudiado a lo largo de las últimas cuatro décadas por la Red Sísmica del Noroeste de México (RESNOM; CICESE, 1980), así como algunas estaciones de la Red Sísmica del Sur de California y del Servicio Sismológico Nacional.

1.3 Catálogo de RESNOM

Los datos sísmicos utilizados para las relocalizaciones en el presente trabajo se obtuvieron del catálogo de RESNOM que actualmente forma parte de la Red Sismológica del Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE, 1980). Esta red opera para monitorear la sismicidad del noroeste de México (Vidal-Villegas *et al.*, 2018). La red comenzó el monitoreo con sismómetros de periodo corto y posteriormente evolucionó a banda ancha, acelerómetros y GPS. Puesto que en

sus inicios la red contaba con una instrumentación limitada, la información se complementaba con las observaciones de estaciones del sur de California cercanas a la frontera. En la Figura 3 se muestran las estaciones sísmicas que aportaron la información para localizar los eventos de RESNOM de 1979 a 2020. Cabe señalar que algunas estaciones funcionaron de manera temporal y no se encuentran actualmente activas. En la figura se observa también cómo en la última década hubo un aumento considerable de estaciones, motivado principalmente tras el sismo EMC.

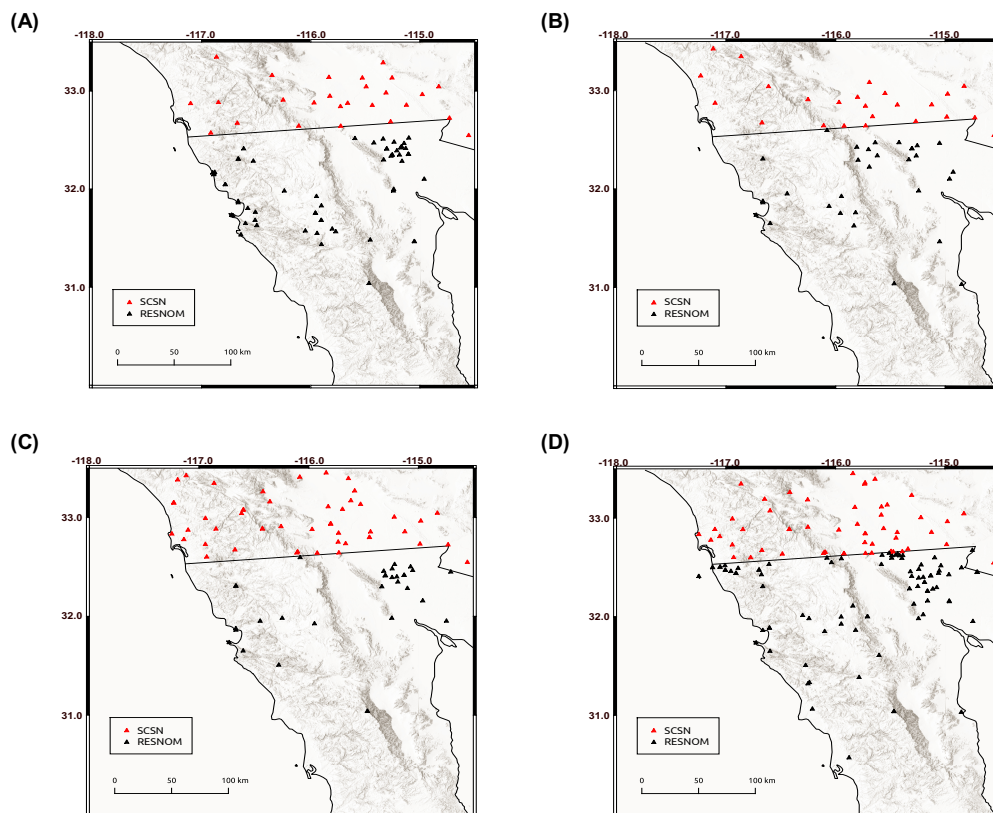


Figura 3. Distribución de estaciones cuyos registros fueron utilizados para la localización de sismos de: **(A)** 1979 a 1990, **(B)** 1990 a 2000, **(C)** 2000 a 2010 y **(D)** 2010 a 2020. Los triángulos en color negro son estaciones nacionales mientras que los triángulos en color rojo son estaciones de la Red del Sur de California (SCSN; California Institute of Technology y United States Geological Survey, 1926).

El área de análisis se encuentra dentro del recuadro con vértices en las coordenadas: -118.6947, 33.5302; -116.2621, 33.5302; -117.6575, 30.5864; -113.5063, 30.5854 (Figura 2.B). El catálogo analizado de RESNOM (1979 al 2020) consiste en un total de 37,591 eventos (Figura 2.B) registrados en el norte de Baja California.

2. Metodología

El primer paso fue recopilar el catálogo de los eventos sísmicos de 1979 al 2020 de la Red Sísmica del Noroeste de México (RESNOM). La información recolectada consiste en los tiempos de arribo de las fases P y S de 37,591 eventos registrados por la red. A continuación, el procedimiento de relocalización se dividió en dos partes:

- i. Se organizó y se dio formato a toda la información: localización hipocentral, tiempo de origen, tiempos de arribo y estaciones utilizadas.
- ii. Se estimaron las nuevas localizaciones implementando el método *Source-Specific Station Terms* (Richards-Dinger y Shearer, 2000).

2.1 Formato de datos

Los datos obtenidos (archivos de tiempos de arribo) del catálogo de RESNOM se encontraban en tres diferentes formatos (Figura 4), por lo cual fue necesario organizarlos de manera que éstos quedaran en formato único. Los eventos de 1979 a 1990 se encontraban en formato propio de RESNOM (Vidal Villegas y Munguía Orozco, 1991), los archivos de 1991 al 2002 se encontraban en formato *hypo71* (Lee y Lahr, 1972) y por

último los archivos del 2003 al 2020 se encontraban en formato Nórdico (SEISAN; Havskov *et al.*, 2020) el cual se utiliza actualmente por la red.

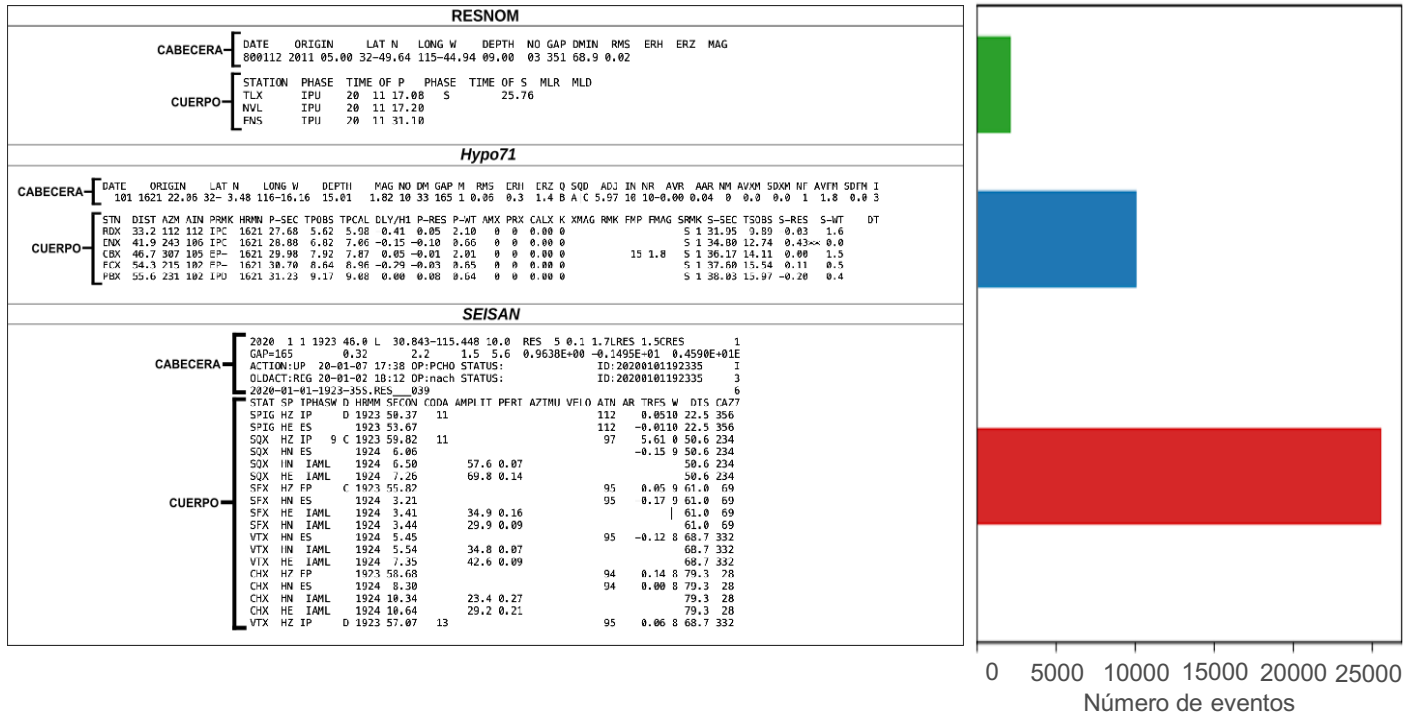


Figura 4. Ejemplo de los formatos de los archivos de tiempos de arribo que contienen la información del evento y su localización (cabecera); los tiempos de arribo y amplitudes de las componentes de las estaciones (cuerpo). En el panel superior se encuentra un ejemplo de archivo en formato RESNOM. En el panel intermedio se encuentra un ejemplo de archivo en formato hypo71. En el panel inferior se encuentra un ejemplo de archivo en formato Nórdico. La gráfica muestra la cantidad de eventos registrados por cada uno de los formatos.

El catálogo original de RESNOM consiste en listas mensuales de cada evento registrado por la red. La configuración de la información para cada evento en las listas varía dependiendo su formato, pero todos constan de una cabecera y cuerpo (Figura 4; Anexo I). La primera, contiene información como fecha, tiempo de origen, coordenadas

epicentrales, profundidad, entre otros. El segundo contiene las lecturas de las fases y sus componentes (tiempos de viaje, residual, amplitud, amplitud máxima, etc.).

Para poder uniformizar el catálogo fue necesario convertir los archivos RESNOM y *hypo71* a formato Nórdico; la información recopilada fue sometida a un control de calidad, es decir, el proceso de adecuación de los archivos (en su formato original) para poder ser leídos por los scripts de conversión de manera automática. Durante este proceso fue necesario revisar aquellos archivos que fallaban al ser convertidos y realizar las correcciones correspondientes, tales como agregar o quitar espacios en blanco, cambiar letras por números o viceversa, entre otros. Aunque los formatos son similares, se requirieron scripts de conversión distintos (Anexo III). Para ambos casos se escribió un script principal y uno auxiliar; el script auxiliar extrajo la información base: Coordenadas, distancias, localización, tiempos de origen y las fases. De esta forma, el objetivo del script principal fue reacomodar la información recopilada del script auxiliar de acuerdo con la estructura formato Nórdico.

Una vez convertidos todos los archivos al formato Nórdico, se realizaron diversas revisiones para corroborar que la información de los archivos creados coincidiera con la original y, en caso de necesitarlo, hacer las correcciones pertinentes tanto en los archivos como los scripts de conversión utilizados. Con lo anterior, se logró unificar el catálogo de 1979 a 2020 al formato Nórdico.

A continuación, se tomaron los eventos individuales y se escribieron tres scripts en Python. El primer script (Anexo IV.A) extrajo información de la cabecera de cada evento (número de identificación, tiempo de origen, epicentro y magnitud) para crear el archivo

eventlist: lista de eventos. El segundo script, unió todos los archivos ya en formato Nórdico, para generar el archivo *phaselist* (Anexo IV.B); este archivo consiste en una lista con todos los eventos (desde 1979 hasta el 2020) cada uno de ellos con su cabecera y cuerpo (arribos de fases P y S); este se describe a mayor detalle en el Anexo II. Es importante mencionar que la información de las fases debe incluir el código de la red sísmica, nombre de estación, identificación de la fase leída, distancia fuente-estación y el tiempo de viaje observado para cada lectura. Por último, con el tercer script se realizó una lista las estaciones que registraron cada fase de cada evento (Anexo IV.C); para obtener las coordenadas de cada una de las estaciones fue necesario consultar el catálogo de RESNOM digitalizado y en PDF, buscar en boletines históricos, consultar con técnicos de RESNOM y buscar en la página de SCEDC (<https://scedc.caltech.edu/>). Una vez obtenidas las coordenadas para cada estación, se utilizó el sistema de información geográfica Google Earth para obtener las elevaciones correspondientes de cada estación. De esta manera, se logró generar el archivo *stationlist*: listado de estaciones. La estructura de los archivos generados se puede observar en la Figura 5.

phaselist										
26572	1999	12	05	20	38	46.18	32.51733	-115.30566	0.1700	2.22
BC CPX	P	2.4200	11.000							
BC CPX	S	50.950	11.000							
BC RDX	P	14.020	87.400							
BC RDX	S	10.900	87.400							
BC CHX	P	18.750	118.40							
BC CHX	S	19.550	118.40							
BC PGX	P	19.670	123.80							
BC PGX	S	20.880	123.80							
BC CBX	P	21.440	129.70							
BC CBX	S	23.030	129.70							
26573	1999	12	06	08	53	04.55	31.21183	-115.92866	14.010	2.54
BC SPX	P	7.7000	48.000							
BC SPX	S	18.020	48.000							
BC ECX	P	12.750	80.500							
BC ECX	S	27.130	80.500							
BC RDX	P	12.480	81.400							
BC RDX	S	26.680	81.400							
BC CHX	P	14.300	88.300							
BC CHX	S	29.270	88.300							
BC PBX	P	15.250	95.900							
BC PBX	S	31.400	95.900							
BC ENX	P	15.880	102.00							
BC ENX	S	32.550	102.00							
BC CBX	P	21.430	140.60							
stationlist										
AGC	37.86130	-122.43000	-0.220							
AMS	33.14140	-115.25493	-0.146							
B11	32.16183	-116.88383	-0.377							
BAR	32.68005	-116.67215	-0.529							
BMA	32.17267	-116.89833	-0.083							
BON	32.69450	-115.26850	-0.014							
CAL	32.29283	-116.52550	-0.372							
CAT	31.60000	-115.80317	-1.058							
CBK	32.91579	-116.25226	-0.355							
CBX	32.31310	-116.66400	-1.238							
CCS	31.86817	-116.66383	-0.037							
CH2	33.29624	-115.33693	-0.353							
CHH	32.48667	-115.23667	-0.008							
CHP	34.80833	-119.01167	-1.607							
CIP	31.81083	-116.57667	-0.073							
eventlist										
51914	2011	01	02	03	22	27.300	32.18200	-115.28100	6.0000	2.60
51915	2011	01	02	04	25	16.300	32.01300	-115.02200	8.0000	2.50
51916	2011	01	02	16	32	43.300	32.20200	-115.27600	6.0000	2.20
51918	2011	01	02	19	55	32.900	32.15800	-115.23000	8.0000	2.70
51919	2011	01	03	03	09	7.9000	32.36800	-115.24200	6.0000	1.70
51920	2011	01	03	07	26	36.800	32.21500	-115.27700	5.0000	2.60
51921	2011	01	03	10	42	0.9000	32.29200	-115.36500	5.0000	2.30
51922	2011	01	03	11	26	56.300	31.83900	-115.10500	9.0000	2.20
51923	2011	01	03	12	23	20.500	32.34500	-115.34600	6.0000	2.20
51926	2011	01	04	00	44	28.600	31.99700	-115.04300	6.0000	3.10
51927	2011	01	04	20	24	16.400	32.16300	-115.23800	9.0000	2.80

Figura 5. Ejemplos de los archivos de entrada: eventlist, phaselist y stationlist. El primer panel muestra dos ejemplos de eventos en el archivo *phaselist* para los cuales se registraron 10 y 17 fases, respectivamente; el segundo panel muestra 15 estaciones del archivo *stationlist* con sus coordenadas geográficas y elevaciones; en el tercer panel se muestran 11 eventos dentro del archivo *eventlist*.

Con el objetivo de delimitar dos polígonos (Figura 2.B): uno para el Valle de Mexicali (VM) y otro para las Sierras Peninsulares de Baja California (SPBC), se escribieron dos scripts en Python los cuales se utilizaron para separar el *eventlist* en las regiones correspondientes. Tanto el archivo *phaselist* como *stationlist* no requirieron separación.

Para realizar las relocalizaciones fue necesario generar las tablas de tiempo de viaje para el VM y SPBC utilizando los códigos *vzfillin*, y *deptable* (Richards-Dinger y Shearer, 2000). Con *vzfillin* (Lin y Shearer, 2006) se remuestrearon los modelos de velocidad (z , V_p , V_s) a intervalos de 1 km con una profundidad máxima de 30 km. Debido a la diferencia en las características de la corteza para las regiones VM y SPBC, fueron requeridos dos modelos de velocidad (Figura 6 y Tabla 1 y 2): i) Para la región de SPBC se utilizó el modelo de velocidad de 5 capas “sierra97” de Frez *et al.* (2000), el cual es una versión modificada del modelo de Nava y Brune (1982); este se recalculó incorporando a los tiempos de arribo originales una velocidad somera de V_p 4.6 km/s para la zona de falla de San Miguel. Para las SPBC se probó también el modelo de Nava y Brune (1982) el cual propone 3 capas, un grosor de corteza de ~42 km y una velocidad P de 8.0 km/s para el manto superior. ii) Para la región de VM se utilizó el modelo “2ca” de Castro *et al.* (2011) de 13 capas; este fue adaptado del modelo de Fuis *et al.* (1984) derivado de un análisis de refracción sísmica para el Valle Imperial tomando como referencia estudios anteriores (McMechan y Mooney, 1980). No obstante, se realizaron pruebas de relocalizaciones utilizando otros modelos de velocidad como el modelo “er” de Ramírez Ramos *et al.* (2015) y “fm” de Fabriol y Munguía (1997). El primero, consiste en un modelo de 5 capas, con una velocidad de 7.7 km/s para el manto superior (Ramírez-Ramos *et al.*, 2015). El modelo fm propone 7 capas, un grosor de corteza de ~42 km y una velocidad P de 7.8 km/s para el manto superior. Las diferencias entre los modelos de velocidad para las SPBC y el VM radican principalmente en las profundidades y el número de capas; esto se debe a la diferencia en las características geológicas de ambas

regiones como el grosor de la corteza y las variaciones en la composición mineralógica de esta.

Por último, con *do.tt* se calcularon las tablas de tiempo de viaje en función de la distancia y profundidad de la fuente para ambas regiones (MV y PRBC). Dado que se utiliza un modelo 1D, los valores son los obtenidos de la interpolación entre valores adyacentes de los modelos de onda P de cada región (Figura 6).

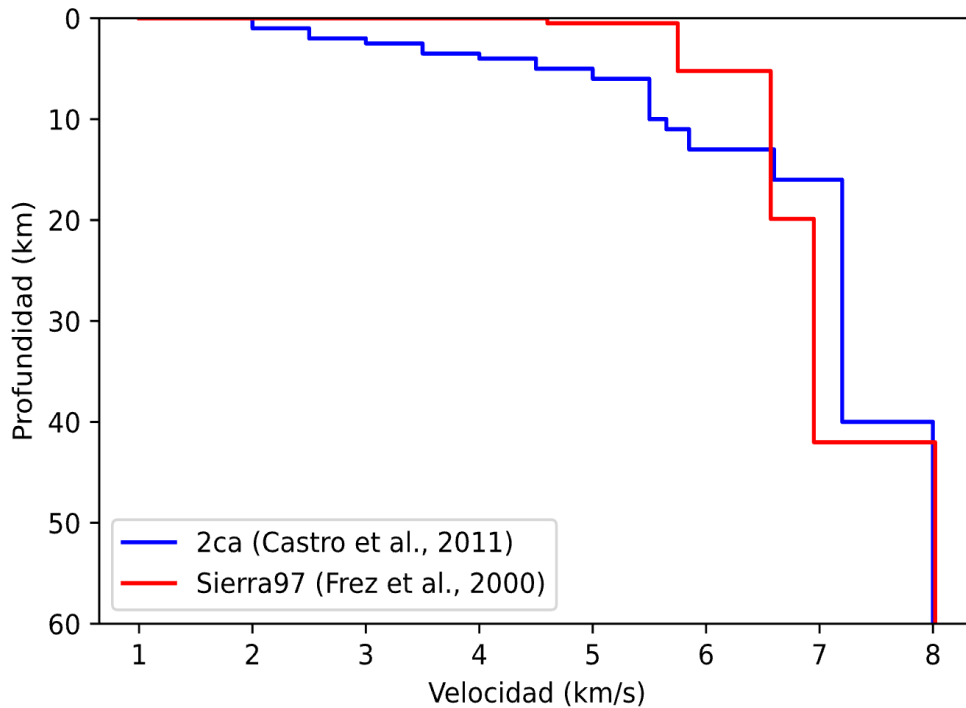


Figura 6. Modelos de velocidad utilizados para el cálculo de las relocalizaciones. En azul se representa el modelo de Castro *et al.* (2011) y en rojo el modelo Sierra97 de Frez *et al.* (2000).

Tabla 1

2ca	
Profundidad (km)	V_p (km/s)
0.0-1.0	2.00
1.0-2.0	2.50
2.0-2.50	3.00
2.50-3.50	3.50
3.50-4.00	4.00
4.00-5.00	4.50
5.00-6.00	5.00
6.00-10.00	5.50
10.00-11.00	5.65
11.00-13.00	5.85
13.00-16.00	6.60
16.00-40.00	7.20
40.00-	8.00

Tabla 2

Sierra97	
Profundidad (km)	V_p (km/s)
0.00	4.60
0.50	5.75
5.23	6.57
19.88	6.95
42.02	8.02

2.2. Relocalización

Para relocalizar el catálogo de RESNOM (1979 al 2020), se utilizó el paquete de programas *XCORLOC* (SSST; Lin, 2018). Para las relocalizaciones, se crearon dos proyectos separados: el del VM y el de las SPBC (Figura 2B). Esto se realizó debido a las diferencias de la corteza para ambas regiones, como se mencionó anteriormente.

Previo a la relocalización, se ajustaron los siguientes parámetros: se seleccionó la norma L3, híbrida entre las normas L1 y L2, como medida de ajuste para el modelo debido a su robustez; esta le da un menor peso a los puntos o valores que se alejen o salgan de la tendencia (Lin, 2018). Se ajustó también el número de marcado de fases máximos en el archivo de inc.common, así como el número máximo de eventos, entre otros. Para cada región se cambió el rango máximo de distancia epicentral, desde 50 km hasta 140 km en intervalos de 10 km para determinar cuál mostraba una mejor distribución espacial de los eventos.

3. Resultados y discusión

3.1 Nuevo catálogo

Una de las principales aportaciones de este trabajo fue la unificación del catálogo de eventos sísmicos para el nBC en el formato Nórdico (SEISAN; Havskov *et al.*, 2020). El formato Nórdico proporciona una estructura ordenada para almacenar y procesar datos sísmicos de manera eficiente; esto representa una oportunidad para desarrollar nuevos proyectos de relocalización implementando otros métodos y técnicas disponibles (Havskov *et al.*, 2020). Este formato es utilizado por RESNOM al presente en la localización de sismos, por la cual la unificación del catálogo en este formato resulta beneficiosa.

De las relocalizaciones surgió un nuevo catálogo con un total de 37,591 eventos, cuyos epicentros se pueden observar en la Figura 7. En la figura, se puede apreciar una dispersión epicentral, comparado con el catálogo original (Figura 2B), pero por lo general, los epicentros siguen alineándose a los sistemas de falla discutidos en la sección 1.2.

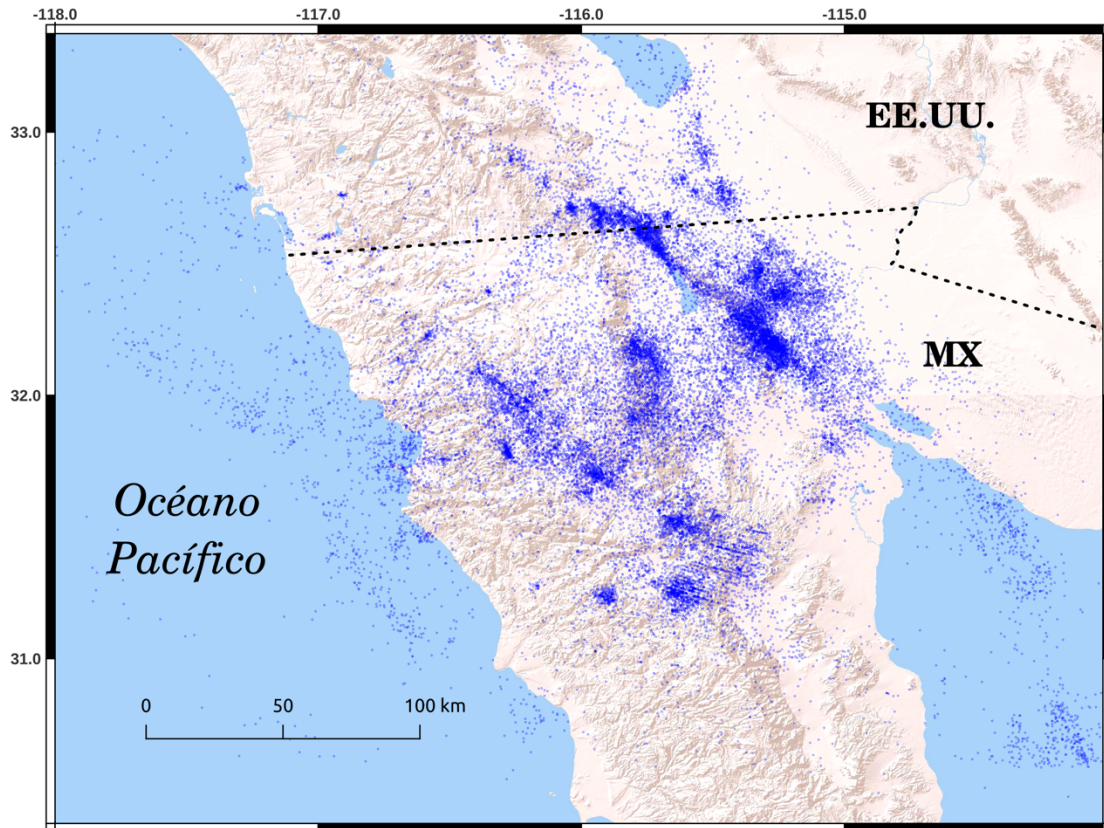


Figura 7. Eventos sísmicos registrados en el nBC de 1979 al 2020 relocalizados utilizando la técnica SSST.

Para el análisis hipocentral de las nuevas localizaciones, se realizaron los histogramas de profundidad (Figura 8); en estos se puede observar la distribución de las profundidades de las SPBC y MV, tanto para el catálogo de RESNOM (Figura 8.A y 8.B)

como para las relocalizaciones utilizando SSST (Figura 8.C y 8.D). Comparando ambos resultados, se percibe una mejora en la distribución. Esto se debe a que la mayoría de las profundidades del catálogo original están fijas a ciertos valores (entre 5 y 10 km aproximadamente), y en este trabajo, las profundidades quedaron libres al momento de relocalizar todos los eventos. Por otra parte, el aumento en las localizaciones registradas a una profundidad de 0 km podría ser un artificio del programa, especialmente en la región de las SPBC, probablemente debido a la menor densidad de estaciones sísmicas comparadas con las del VM.

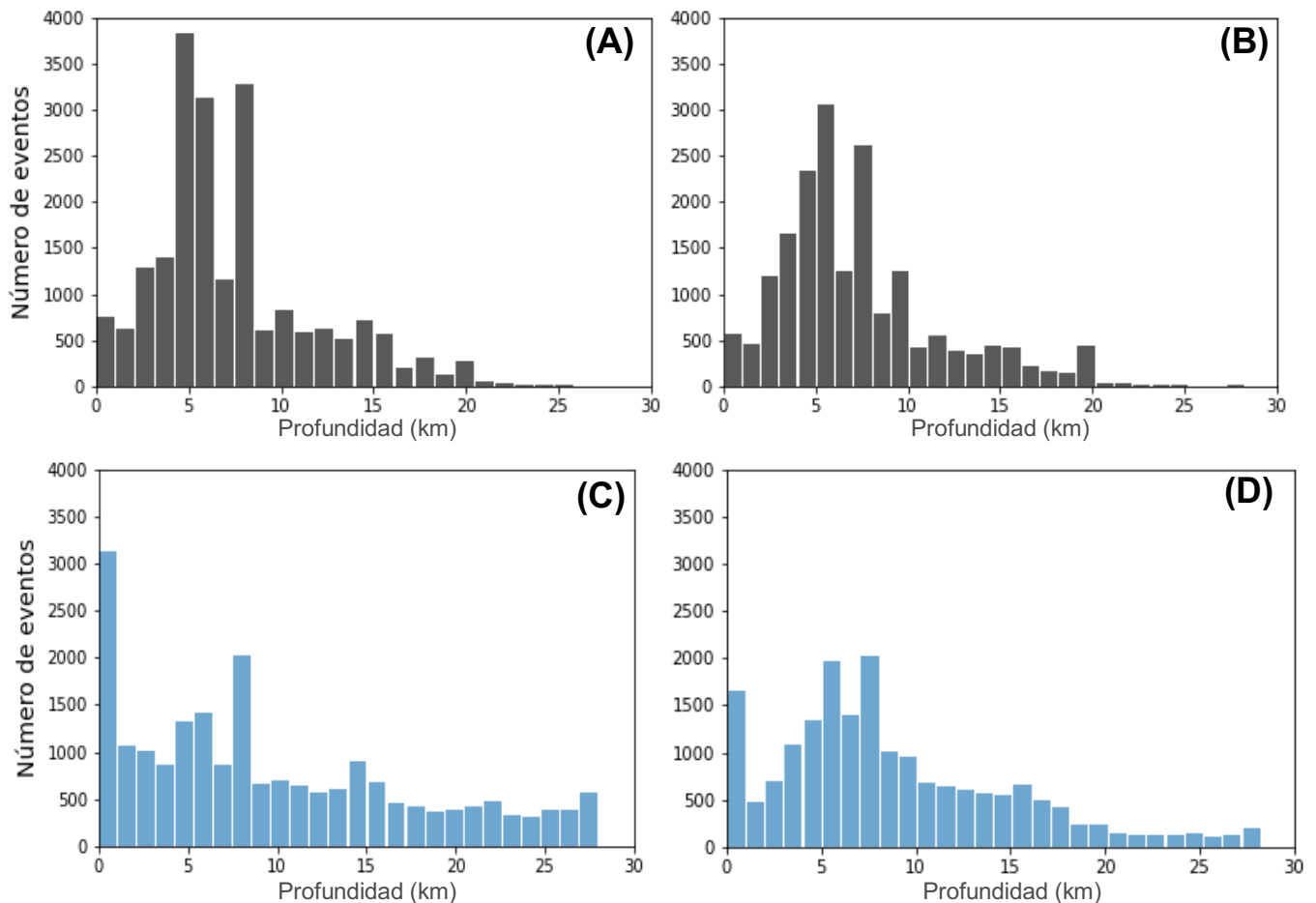


Figura 8. Histogramas de profundidad para el catálogo de RESNOM para **A)** SPBC y **B)** MV y para las relocalizaciones **C)** SPBC y **D)** MV.

3.2 Perfiles

En esta sección se presentan, describen y analizan los perfiles de algunas zonas de falla localizadas en el nBC, tanto para los eventos originales como para los relocalizados. En éstos se pueden observar las distribuciones de las profundidades y las diferencias principales entre ambas localizaciones.

3.2.1 Centro de dispersión de Cerro Prieto

El volcán Cerro Prieto se localiza dentro del VM y aquí se puede encontrar una falla de rumbo lateral derecho con el mismo nombre. Esta se extiende desde el volcán hasta la cuenca Wagner en el Golfo de California. La sismicidad presente, principalmente al norte de la falla, exhibe mecanismos caracterizados por movimientos normales y oblicuo-normales, los cuales coinciden con la naturaleza extensional general de la cuenca *pull-apart* (Fabriol y Munguía, 1997) y que dan origen al Centro de Dispersión de Cerro Prieto (CDCP). El CDCP se encuentra entre los extremos noroeste y sureste de las fallas Cerro Prieto e Imperial (Figura 2), respectivamente, dentro de una cuenca que contiene depósitos sedimentarios del Río Colorado. Asimismo, se distingue por una rápida deformación geodésica, alto flujo de calor, una elevada actividad sísmica y vulcanismo (Nava y Glowacka, 1999). Al norte de la falla Cerro Prieto se encuentra el Campo Geotérmico Cerro Prieto operado desde 1973 por la Comisión Federal de Electricidad. En la Figura 9 se presentan los perfiles de sismicidad del CDCP. La sismicidad reportada por RESNOM se concentra a profundidades entre 5 km y 10 km principalmente (Figura 9A), y en los sismos relocalizados estas presentan una mejora en la distribución (Figura 9B) manteniendo el mismo rango de profundidades. Lo anterior debido a que las

profundidades ya no están fijas como en las localizaciones originales: líneas horizontales indicadas con una flecha blanca en la gráfica inferior de la Figura 9.A. Se observa sismicidad a través de todo el perfil, sin embargo, se distingue una mayor concentración de eventos al este del CDCP. Esto coincide con lo registrado por Fabriol y Munguía (1997) en un estudio donde se analizó la sismicidad (de 1994 a 1995) en la zona de Cerro Prieto, donde se determinó que los hipocentros se encuentran principalmente en la porción norte de la falla Cerro Prieto y dentro del Campo Geotérmico, con profundidades entre 1 y 6 km.

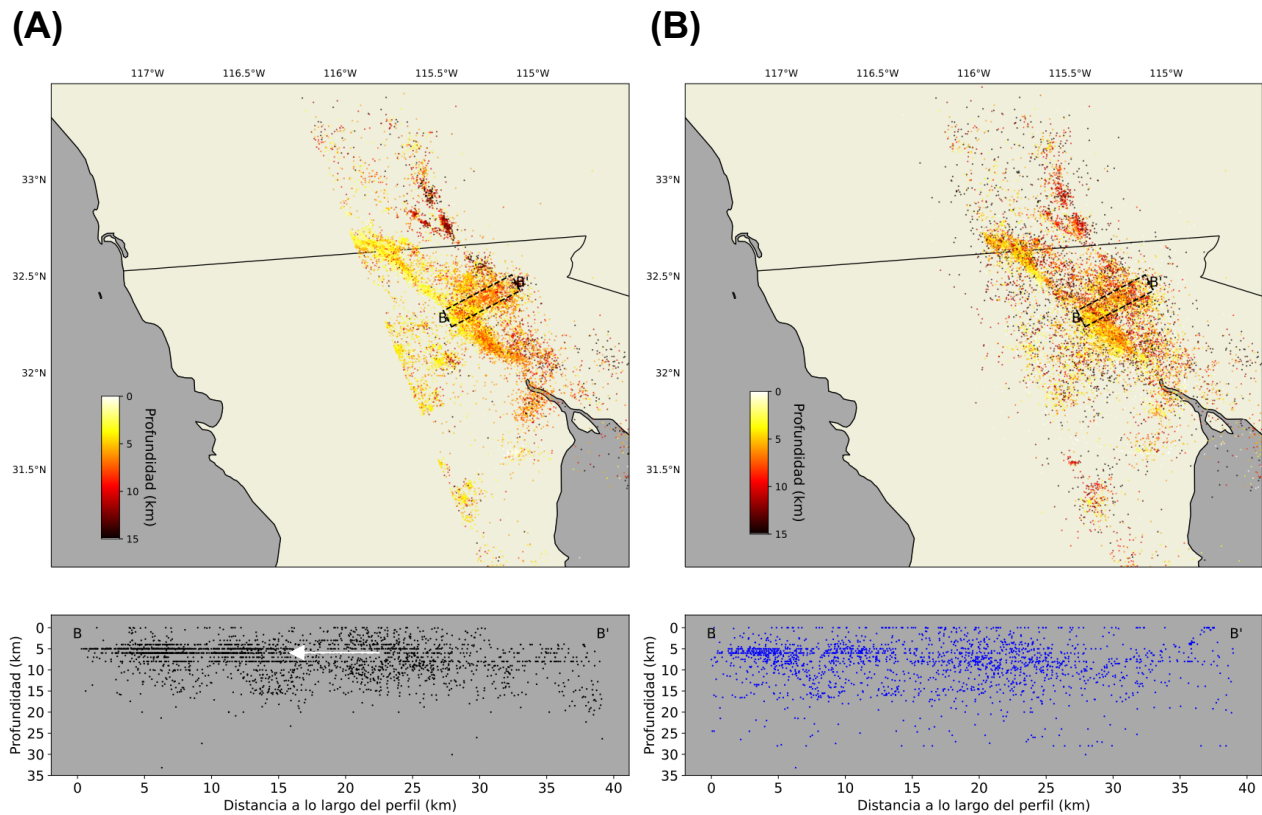


Figura 9. Sismicidad en el centro de dispersión de Cerro Prieto ($B-B'$) de 1979 a 2020 dada por **(A)** catálogo de RESNOM y **(B)** relocalizaciones utilizando SSST. La flecha blanca indica cómo se aprecian los eventos con profundidades fijas.

3.2.2 Cucapah-Indiviso

La Sierra Cucapah está dominada por fracturas de rumbo NW-SE localizadas al oeste de la sierra; la continuidad de la sierra se ve interrumpida por múltiples fallas paralelas como las fallas Cucapah, Borrego y Pescadores (Rosales *et al.*, 2012). La ocurrencia del sismo del 2010 M_w 7.2 no solo evidenció la tectónica transtensional de la región, sino que reveló conexiones entre los diversos sistemas de falla presentes (Hauksson *et al.*, 2011); con la secuencia del Mayor-Cucapah se logró distinguir la falla Indiviso, sepultada bajo la gran capa de sedimentos del VM (Hauksson *et al.*, 2011). La sierra El Mayor se localiza al sur de la sierra Cucapah y ambas surgen del levantamiento ocasionado por la interacción entre las fallas de desplazamiento lateral y las fallas normales presentes.

En la Figura 10.A y 10.B se muestran los sismos del catálogo de RESNOM desde 1979 hasta el 2020 y los sismos relocalizados utilizando SSST, respectivamente. En el perfil de las localizaciones del catálogo de RESNOM se puede observar que las profundidades se distribuyen entre 0 y 15 km y se observan dos cúmulos: uno al norte del perfil (Sierra Cucapah) y uno al sur (paralelo a Sierra El Mayor). La mayor parte de los sismos se encuentran a profundidades relativamente superficiales: entre 5 y 15 km para ambos catálogos. Sin embargo, se puede observar que las profundidades están fijadas para los eventos del catálogo original y se distinguen como líneas rectas en ciertos valores (0 a 10 km) indicadas con la flecha blanca en la Figura 10.A. Las profundidades de las relocalizaciones muestran una distribución similar, en la cual se aprecian dos cúmulos de sismicidad: uno al norte de la falla que se extiende 0 y 30 km y otro al sur entre 60 y 90 km a lo largo del perfil C-C'. Se distingue un patrón entre ~2 a 16 km de profundidad; la sismicidad se distribuye desde la superficie de la parte sur de la falla Indiviso (al sur del

perfil) y va aumentando su profundidad hacia el norte del perfil. Este patrón podría deberse a una estructura geológica que, debido a su geometría, podría deberse a una falla lístrica. En este sentido, un estudio de sismica de reflexión de Chanes-Martínez *et al.* (2014) caracterizó diversos perfiles del Delta del Río Colorado, en el cual se encontraron fallas contiguas a la falla Indiviso; sin embargo, el estudio solo penetra hasta ~5 km de profundidad por lo que no se puede concluir si se trata de estas fallas. Para ello, se requeriría de un estudio más detallado utilizando otras metodologías geofísicas.

Un análisis temporal de esta zona evidenció un incremento en la actividad sísmica entre 2010 y 2016 (Figura 11.B). La gran cantidad de réplicas del sismo El Mayor-Cucapah de 2010 registradas en este periodo ilustra la baja sismicidad en décadas anteriores (Figura 11.A) y posteriores (Figura 11.C), incluso contando más estaciones sísmicas instaladas a partir del 2012 (Vidal-Villegas *et al.*, 2018). Lo anterior evidencia también por qué la falla Indiviso no había sido identificada previo al sismo EMC de 2010.

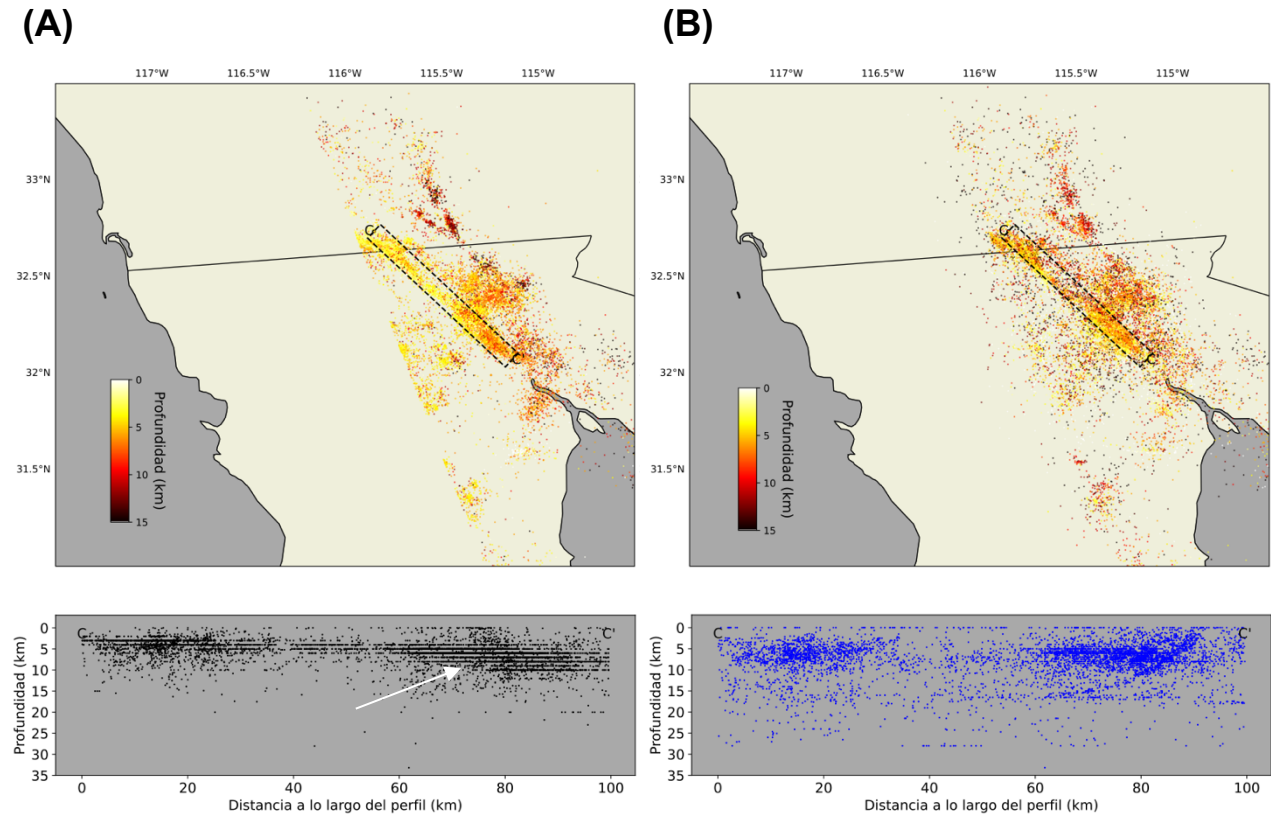


Figura 10. Perfiles de eventos registrados de 1979 a 2020 del perfil Cucapah-Indiviso (C-C') de **(A)** RESNOM y **(B)** relocalizaciones utilizando SSST. La flecha blanca indica un ejemplo del alineamiento horizontal debido a la fijación de las profundidades.

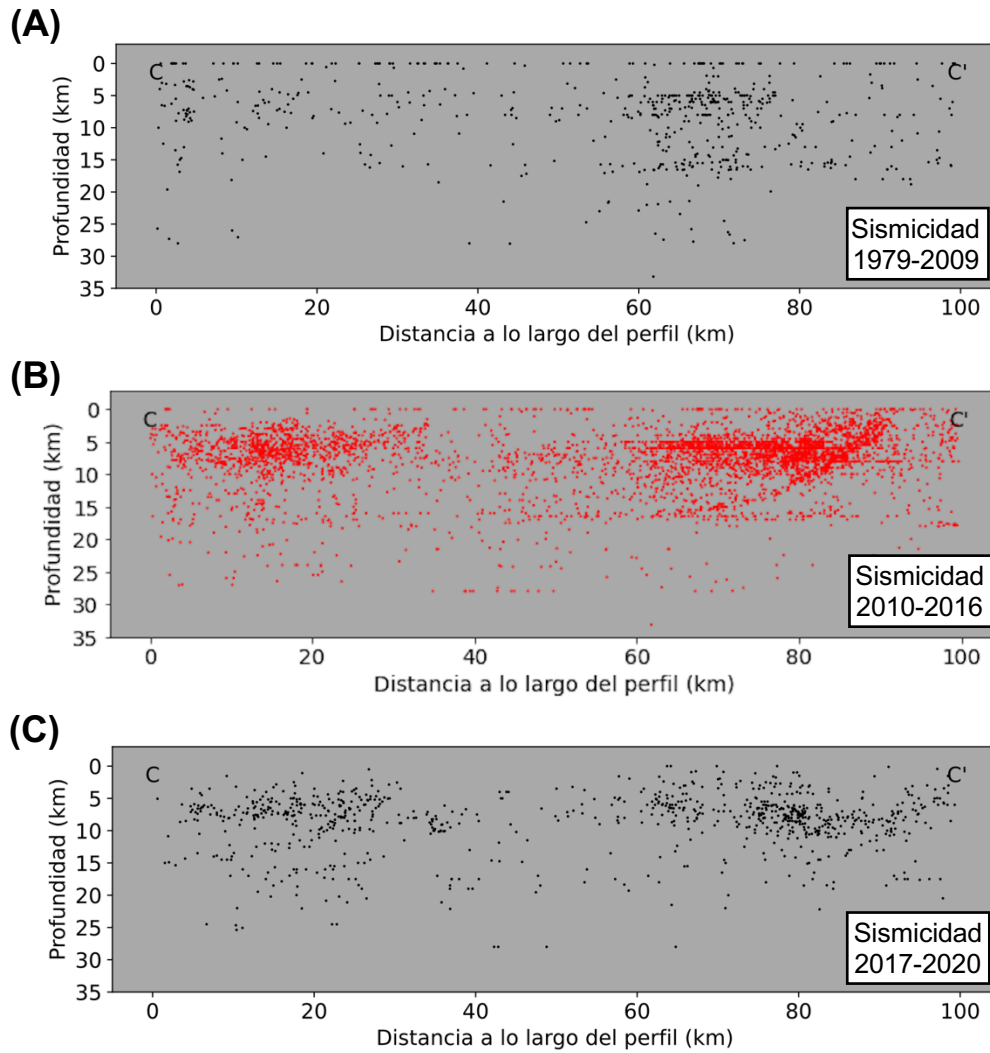


Figura 11. Perfil de localizaciones del sistema Cucapah-Indiviso C-C utilizando SSST. (A) Eventos ocurridos entre 1979 a 2009, (B) los eventos ocurridos entre 2010 a 2016 y (C) eventos ocurridos entre 2017 a 2020.

3.2.3 Dixieland

Dixieland es una región/poblado del Valle Imperial, CA. Aquí se encuentra una falla cuya ruptura no es visible en la superficie. La falla se localiza al sur del Valle Imperial justo al norte de la frontera mexicana, y se infiere derivado del alineamiento de la sismicidad de esa región (Stock *et al.*, 2022). Como se puede observar en la Figura 12, los epicentros se concentran mayoritariamente al centro y sur del perfil (cercano a la frontera). La distribución hipocentral se encuentra principalmente entre los 5 y 20 km, tanto en las localizaciones de RESNOM, como en las relocalizaciones (SSST).

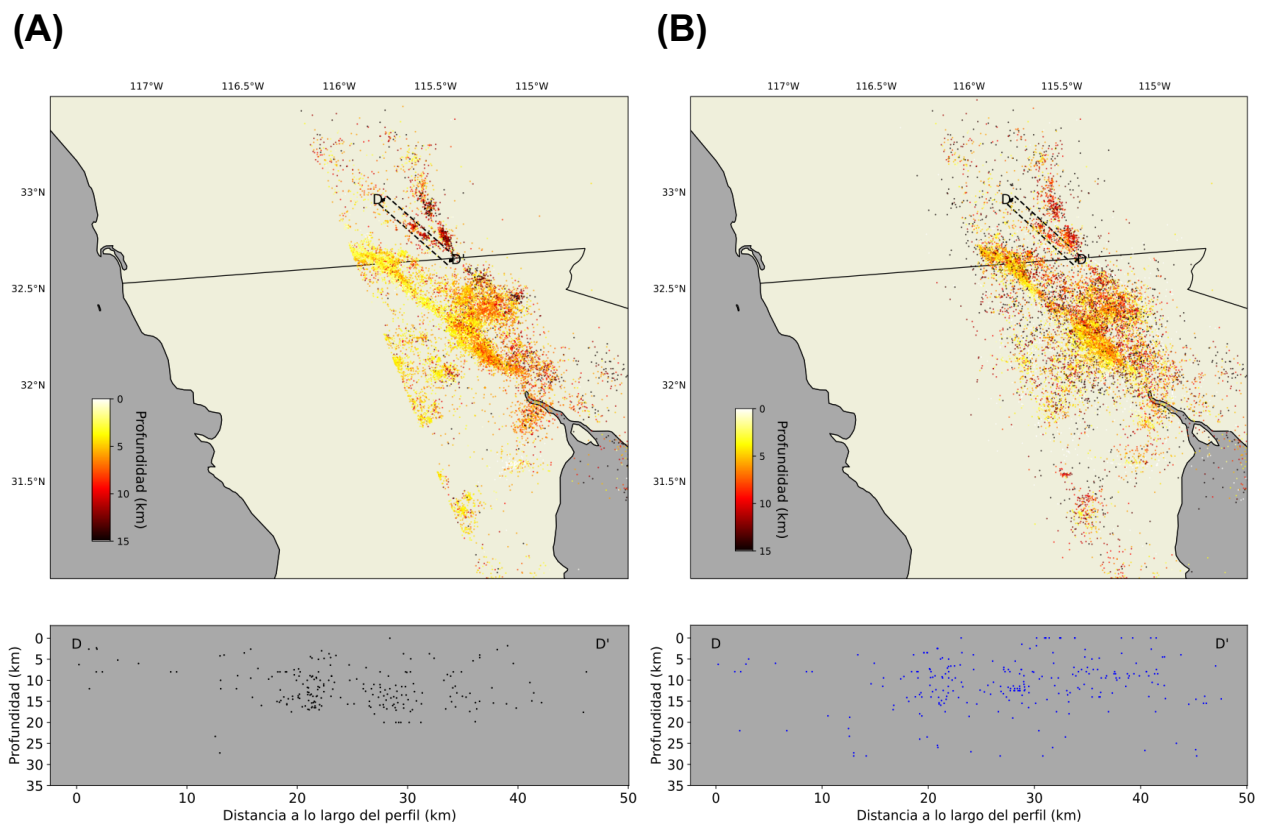


Figura 12. Sismicidad para la falla (inferida) Dixieland D-D'. **(A)** Localizaciones del catálogo de RESNOM y **(B)** localizaciones obtenidas implementando SSST.

3.2.4 Imperial

La falla Imperial es un elemento muy importante en la tectónica regional. Esta falla se extiende desde el sur de la falla de San Andrés hasta el Centro de Dispersión Cerro Prieto y se localiza cercana a comunidades del condado Imperial como El Centro, Calexico, Brawley, Holtville, Imperial y Westmorland, así como en el este de la ciudad de Mexicali y poblados del VM. La falla Imperial se caracteriza por su movimiento lateral derecho, el cual resulta de los esfuerzos cortantes producidos por el movimiento principal entre las placas del Pacífico y de Norteamérica. En el perfil E-E' (Figura 13) se puede apreciar la distribución de los epicentros tanto para las localizaciones de RESNOM (Figura 13.A), como para las localizaciones obtenidas con SSST (Figura 13.B), así como sus respectivas profundidades. En las profundidades del catálogo de RESNOM se observan líneas horizontales correspondientes a las profundidades fijadas, mientras que en las relocalizaciones estas no muestran restricciones. En ambos perfiles, la sismicidad se aprecia en cúmulos entre 10 y 15 km de profundidad tanto al norte como al sur de la frontera, coincidiendo con la sismicidad reportada en análisis previos (Stock *et al.*, 2022). La concentración de eventos en ambos cúmulos es de particular relevancia ya que su cercanía a las comunidades anteriormente mencionadas representa un riesgo importante para éstas, además de su potencial para generar sismos de grandes magnitudes (por ejemplo, el sismo de 1979 ML = 6.9).

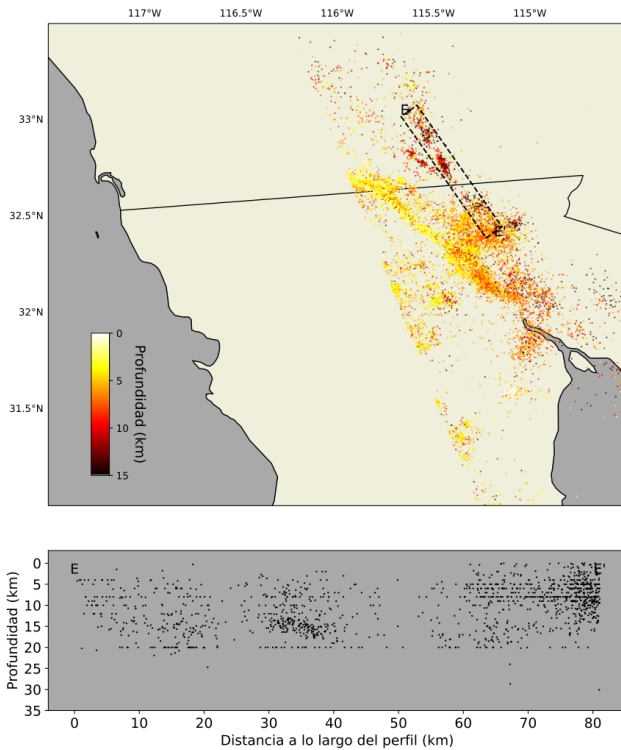
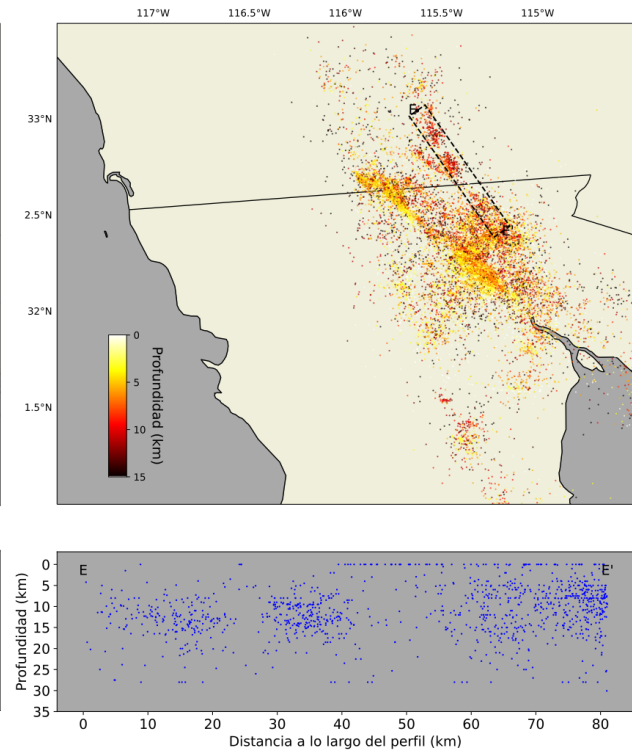
(A)**(B)**

Figura 13. Perfiles de la falla Imperial E-E' **(A)** localizaciones de RESNOM y **(B)** localizaciones obtenidas con SSST.

3.2.5 San Miguel-Vallecitos

Este sistema está compuesto por las fallas San Miguel y Vallecitos, el sistema acomoda gran parte del movimiento entre las placas del Pacífico y Norteamérica (5 ± 2 mm/año, Frez *et al.*, 2000) y es la estructura más sísmicamente activa de las SPBC (Stock *et al.*, 2022). En los perfiles de la Figura 14 se pueden apreciar las profundidades fijadas como líneas horizontales entre $\sim 5-8$ km en la Figura 14.A; por otra parte, en la Figura 14.B se puede notar que este alineamiento persiste en los eventos relocados. Esto podría

deberse al modelo utilizado, ya que la velocidad se mantiene constante a esas profundidades, así como a la poca densidad de estaciones sísmicas.

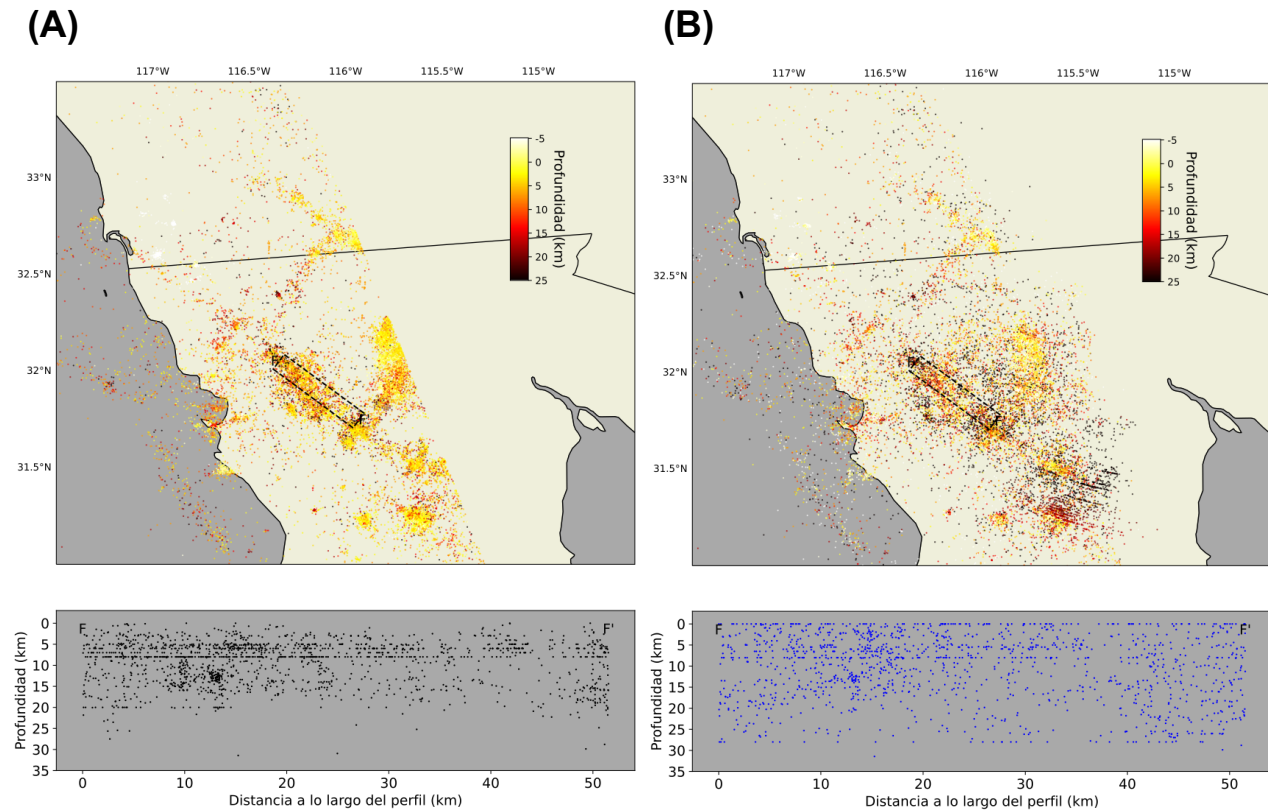


Figura 14. Localizaciones de eventos para el sistema San Miguel-Vallecitos F-F'; **(A)** Localizaciones de RESNOM y **(B)** localizaciones obtenidas con SSST.

4. Conclusiones

La unificación del catálogo sísmico de RESNOM es un gran avance para el análisis y la caracterización de la sismicidad en el nBC. El catálogo unificado facilitará el procesamiento de los datos en futuras investigaciones y permite agregar los registros más recientes de la red, ya que es el formato utilizado actualmente. A pesar de la dispersión presente de los epicentros resultado de la relocalización respecto al catálogo

de RESNOM, existe una mejora en la distribución de las profundidades de la sismicidad en el nuevo catálogo. Esto lo confirma la comparación entre perfiles; en estos se pueden apreciar líneas horizontales a ciertas profundidades para las localizaciones originales, las cuales se atribuyen a la fijación de estas a ciertos valores, mientras que en la sismicidad relocalizada se aprecia una mejora en la distribución.

Se registró un aumento en la sismicidad del 2010 al 2016 en el VM asociada a las réplicas del sismo del 2010 El Mayor-Cucapah $M_w=7.2$, en el sistema Cucapah-Indiviso. En este sistema también se puede apreciar un patrón en las localizaciones hipocentrales el cual se podría atribuir a una estructura geológica adyacente a la falla indiviso; sin embargo, se requeriría un estudio más detallado utilizando otros métodos geofísicos para poder determinarlo.

Debido a la compleja tectónica del nBC y antecedentes de sismos de gran magnitud, desarrollar un modelo de velocidad 3D resultaría beneficioso para una mejor relocalización de los eventos del catálogo de RESNOM. Para ello se pueden utilizar los archivos generados eventlist, stationlist y phaselist, además del catálogo en formato Nórdico.

5. Bibliografía

- Bennett, R. A., Rodi, W., y Reilinger, R. E. (1996). Global Positioning System constraints on fault slip rates in southern California and northern Baja, Mexico. In *Journal of Geophysical Research* (Vol. 101, Issue B10).
- California Institute of Technology, y United States Geological Survey. (1926). *Southern California Seismic Network*. International Federation of Digital Seismograph Networks.
- Castro, R. R., Acosta, J. G., Wong, V. M., Pérez-Vertti, A., Mendoza, A., y Inzunza, L. (2011). Short note location of aftershocks of the 4 April 2010 m w 7.2 El mayor-Cucapah earthquake of Baja California, Mexico. *Bulletin of the Seismological Society of America*, 101(6), 3072–3080. <https://doi.org/10.1785/0120110112>
- Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE). (1980). Red Sísmica del Noroeste de México. In *International Federation of Digital Seismograph Networks*. <https://doi.org/https://doi.org/10.7914/SN/BC>
- Chanes-Martínez, J. J., González-Escobar, M., Suárez-Vidal, F., y Gallardo-Mata, C. G. (2014). Structural Geometry of a Sector of the Colorado River Delta, Baja California, Mexico, Based on Seismic Reflections. *Pure and Applied Geophysics*, 171(7), 1107–1127. <https://doi.org/10.1007/s00024-013-0729-z>
- Evans, J. R., Eberhart-Phillips, D., y Thurber, C. H. (1994). *User's Manual for SIMULPS 12 For Imaging VP and vp lvs : A Derivative of the "Thurber" Tomographic Inversion SIMUL3 for Local Earthquakes and Explosions*.
- Fabriol, H., y Munguía, L. (1997). Seismic activity at the Cerro Prieto geothermal area (Mexico) from August 1994 to December 1995, and its relationship with tectonics

and fluid exploitation. *Geophysical Research Letters*, 24(14), 1807–1810.

<https://doi.org/10.1029/97GL01669>

Fletcher, J. M., y Spelz, R. M. (2009). Patterns of Quaternary deformation and rupture propagation associated with an active low-angle normal fault, Laguna Salada, Mexico: Evidence of a rolling hinge? *Geosphere*, 5(4), 385–407.

<https://doi.org/10.1130/GES00206.1>

Frez, J., González, J. J., Acosta, J. G., Nava, F. A., Méndez, I., Carlos, J., García-Arthur, R. E., y Alvarez, M. (2000). A detailed microseismicity study and current stress regime in the peninsular ranges of northern Baja California, Mexico: The Ojos Negros region. *Bulletin of the Seismological Society of America*, 90(5), 1133–1142. <https://doi.org/10.1785/0119990164>

Fuis, G. S., Mooney, W. D., Healy, J. H., McMechan, G. A., y Lutter, W. J. (1984). A Seismic Refraction Survey of the Imperial Valley Region, California. In *Journal of Geophysical Research* (Vol. 89, Issue B2).

Got, J. L., Fréchet, J., y Klein, F. W. (1994). Deep fault plane geometry inferred from multiplet relative relocation beneath the south flank of Kilauea. *Journal of Geophysical Research*, 99(B8). <https://doi.org/10.1029/94jb00577>

Hauksson, E. (2000). Crustal structure and seismicity distribution adjacent to the Pacific and North America plate boundary in southern California. *Journal of Geophysical Research: Solid Earth*, 105(B6), 13875–13903.

<https://doi.org/10.1029/2000jb900016>

Hauksson, E., y Shearer, P. (2005). Southern California hypocenter relocation with waveform cross-correlation, part 1: Results using the double-difference method.

Bulletin of the Seismological Society of America, 95(3), 896–903.

<https://doi.org/10.1785/0120040167>

Hauksson, E., Stock, J., Hutton, K., Yang, W., Vidal-Villegas, J. A., y Kanamori, H.

(2011). The 2010 Mw 7.2 El Mayor-Cucapah earthquake sequence, Baja California, Mexico and Southernmost California, USA: Active seismotectonics along the Mexican pacific margin. *Pure and Applied Geophysics*, 168(8–9), 1255–1277.

<https://doi.org/10.1007/s00024-010-0209-7>

Havskov, J., Voss, P. H., y Ottemöller, L. (2020). Seismological observatory software:

30 yr of seisan. *Seismological Research Letters*, 91(3), 1846–1852.

<https://doi.org/10.1785/0220190313>

Jordan, T. H., y Sverdrup, K. A. (1981). Teleseismic location techniques and their appli-

cation to earthquake clusters in the South-Central Pacific. *Bulletin of the Seismological Society of America*, 71(4), 1105–1130. <http://www.bssaonline.org/content/71/4/1105.abstract>

Lee, W. H. K., y Lahr J.C. (1972). HYPO71: a computer pro- gram for determining hypo- center, magnitude, and first motion pattern of local earthquakes. In *U.S. Geological Survey*.

Lewis, J. L., Day, S. M., Magistrale, H., Castro, R. R., Astiz, L., Rebollar, C., Eakins, J.,

Vernon, F. L., y Brune, J. N. (2001). Crustal thickness of the Peninsular Ranges and Gulf Extensional Province in the Californias. *Journal of Geophysical Research: Solid Earth*, 106(B7), 13599–13611. <https://doi.org/10.1029/2001jb000178>

- Lin, G. (2018). The source-specific station term and waveform cross-correlation earthquake location package and its applications to California and New Zealand. *Seismological Research Letters*, 89(5), 1877–1885. <https://doi.org/10.1785/0220180108>
- Lin, G., y Shearer, P. (2005). Tests of relative earthquake location techniques using synthetic data. *Journal of Geophysical Research: Solid Earth*, 110(4), 1–14. <https://doi.org/10.1029/2004JB003380>
- Lin, G., y Shearer, P. (2006). The COMPLOC earthquake location package. *Seismological Research Letters*, 77(4), 440–444. <https://doi.org/10.1785/gssrl.77.4.440>
- Lin, G., Shearer, P. M., y Hauksson, E. (2007). Applying a three-dimensional velocity model, waveform cross correlation, and cluster analysis to locate southern California seismicity from 1981 to 2005. *Journal of Geophysical Research: Solid Earth*, 112(12), 1–14. <https://doi.org/10.1029/2007JB004986>
- Lomnitz, C., Mooser, F., Allen, C. R., Brune, J. N., y Thatcher, W. (1970). *Seismicity and Tectonics of the Northern Gulf of California Region, Mexico. Preliminary Results.*
- McMechan, G. A., y Mooney, W. D. (1980). Asymptotic Ray Theory and Synthetic Seismograms for Laterally Varying Structures: Theory and Application to the Imperial Valley, California. In *Bulletin of the Seismological Society of America* (Vol. 70, Issue 6). <http://pubs.geoscienceworld.org/ssa/bssa/article-pdf/70/6/2021/5329011/bssa0700062021.pdf>
- Meneses, J., y Kleinfelder. (2010). *The El Mayor Cucapah, Baja California Earthquake April 4, 2010.* www.eeri.org

Nava, F. A., y Glowacka, E. (1999). *Fault-slip Triggering, Healing, and Viscoelastic Afterworking in Sediments in the Mexicali-Imperial Valley* (Vol. 156).

Nava, F., y Brune, J. N. (1982). An earthquake-explosion reversed refraction line in the Peninsular Ranges of southern California and Baja California Norte. *Bulletin of the Seismological Society of America*, 72(4), 1195–1206.

<https://doi.org/10.1785/BSSA0720041195>

Northern California Earthquake Data Center, y UC Berkeley Seismological Laboratory. (2014). *NCEDC*.

Pelayo, A., Razo, A., Gutierrez, L. C. A., Arellano, F., Espinoza, J. M., y Quijano, J. L. (1991). Main geothermal fields of Mexico; Cerro Prieto geothermal field, Baja California. In *The Geology of North America* (Vol. 3). Mexico The Geological Society of America. http://pubs.geoscienceworld.org/books/book/chapter-pdf/3733909/9780813754703_ch5.pdf

Ramírez, E. E., Bataille, K., Vidal-Villegas, J. A., Stock, J. M., y Ramírez-Hernández, J. (2021). Moho Depth of Northern Baja California, Mexico, From Teleseismic Receiver Functions. *Earth and Space Science*, 8(6).

<https://doi.org/10.1029/2020EA001463>

Ramírez-Ramos, E. E., Vidal-Villegas, A., González-Fernández, A., y Stock, J. M. (2015). A crustal velocity model for the Southern Mexicali Valley, Baja California, Mexico. *Seismological Research Letters*, 86(1), 181–191.

<https://doi.org/10.1785/0220140007>

- Richards-Dinger, K. B., y Shearer, P. M. (2000). Earthquake locations in southern California obtained using source-specific station terms. *Journal of Geophysical Research: Solid Earth*, 105(B5), 10939–10960. <https://doi.org/10.1029/2000jb900014>
- Rosales, J. Á., Camacho Hernández, J. M., Ilitch, V., Federico, G., Macías Valdez, G., y Carrillo, D. H. (2012). *Preliminary geologic features of the new geothermal zone Cucapah at the Mexicali Valley, BC, Mexico* (Vol. 25, Issue 2).
- Shearer, P., Hauksson, E., y Lin, G. (2005). Southern California hypocenter relocation with waveform cross-correlation, part 2: Results using source-specific station terms and cluster analysis. *Bulletin of the Seismological Society of America*, 95(3), 904–915. <https://doi.org/10.1785/0120040168>
- Shearer, P. M. (2019). *Introduction to Seismology* (3rd ed.). Cambridge University Press. <https://doi.org/10.1017/9781316877111>
- Stock, J. M., y Hodges, K. V. (1990). Miocene to Recent structural development of an extensional accommodation zone, northeastern Baja California, Mexico. In *Journal of Structural Geology* (Vol. 12, Issue 3).
- Stock, J. M., Ross, Z., y Ramos, E. R. (2022). *Seismotectonics of the California-Mexico Border Region*.
- The American Geological Institute. (1984). *Dictionary of Geological Terms* (R. L. Bates y J. A. Jackson, Eds.; 3rd ed.). Anchor Books.
- Trugman, D. T., Chamberlain, C. J., Savvaidis, A., y Lomax, A. (2023). GrowClust3D.jl: A Julia Package for the Relative Relocation of Earthquake Hypocenters Using 3D Velocity Models. *Seismological Research Letters*, 94(1), 443–456. <https://doi.org/10.1785/0220220193>

Vidal Villegas, J. A., y Munguía Orozco, L. (1991). Boletín de Información Sísmica. *Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)*.

Vidal-Villegas, J. A., Munguía, L., González-Ortega, J. A., Nuñez-Leal, M. A., Ramírez, E., Mendoza, L., Castro, R. R., y Wong, V. (2018). The northwest Mexico seismic network: Real-time seismic monitoring in northern Baja California and northwestern Sonora, Mexico. *Seismological Research Letters*, 89(2A), 324–337.

<https://doi.org/10.1785/0220170183>

Waldhauser, F. (2001). hypoDD: A computer program to compute double-difference earthquake locations (2001). *USGS Open File Rep*, 1–113.

Waldhauser, F., y Ellsworth, W. L. (2000). A Double-difference Earthquake location algorithm: Method and application to the Northern Hayward Fault, California. *Bulletin of the Seismological Society of America*, 90(6), 1353–1368.

<https://doi.org/10.1785/0120000006>

6. Anexos

Anexo I: Elementos de cada tipo de archivo

A. Formato RESNOM

RESNOM												
CABECERA	DATE	ORIGIN	LAT N	LONG W	DEPTH	NO	GAP	DMIN	RMS	ERH	ERZ	MAG
	800112	2011 05.00	32-49.64	115-44.94	09.00	03	351	68.9	0.02			
CUERPO	STATION	PHASE	TIME OF P	PHASE	TIME OF S	MLR	MLD					
	TLX	IPU	20 11 17.08	S	25.76							
	NVL	IPU	20 11 17.20									
	FNS	TPIJ	20 11 31.10									

Cabecera:

DATE: Año, mes y día del evento

ORIGIN: Hora, minuto y segundos del tiempo de origen (UTM, tiempo meridiano cero).

LAT N: Latitud (en grados y minutos).

LONG W: Longitud (en grados y minutos).

DEPTH: Profundidad en km (una "*" después del valor indica que es una solución con profundidad fija).

NO: Número de lecturas empleadas en la solución.

GAP: Separación azimutal (en grados) más larga entre las estaciones.

DMIN: Distancia epicentral (en km) de la estación más cercana.

RMS: Error cuadrático medio de los residuales de tiempo (en s).

ERH: Error estándar del epicentro (en km).

ERZ: Error estándar de la profundidad (en km). Si el espacio está en blanco, significa que no existen suficientes datos para calcularlo.

MAG: Magnitud

Cuerpo:

STATION: Nombre de la estación.

PHASE TIME OF P: Tiempo de arribo de fase P (I, impulsiva; E, emergente; polaridad se representa con + o -, respectivamente).

PHASE TIME OF S: Tiempo de arribo de fase S.

B. Formato Hypo71

		Hypo71																											
CABECERA	DATE	ORIGIN	LAT N	LONG W	DEPTH	MAG NO	DM	GAP	M	RMS	ERH	ERZ	Q	SQD	ADJ	IN	NR	AVR	AAR	NM	AVXM	SDXM	NF	AVFM	SDFM	I			
		101	1621	22.06	32- 3.48	116-16.16	15.01	1.82	10	33	165	1	0.06	0.3	1.4	B	A	C	5.97	10	10-0.00	0.04	0	0.0	0.0	1	1.8	0.0	3
CUERPO	STN	DIST	AZM	AIN	PRMK	HRMN	P-SEC	TPQBS	TPCAL	DLY/H1	P-RES	P-WT	AMX	PRX	CALX	K	XMAG	RMK	FMP	FMAG	SRMK	S-SEC	TSQBS	S-RES	S-WT	DT			
	RDX	33.2	112	112	IPC	1621	27.68	5.62	5.98	0.41	0.05	2.10	0	0	0.00	0					S	1	31.95	0.89	0.03	1.6			
	CNX	41.9	243	106	IPC	1621	28.88	6.82	7.06	-0.15	-0.10	0.65	0	0	0.00	0					S	1	34.00	12.74	0.43	0.0			
	CBX	46.7	307	105	EP-	1621	29.98	7.92	7.87	0.05	-0.01	2.01	0	0	0.00	0			15	1.8	S	1	36.17	14.11	0.00	1.5			
	FCX	54.3	215	102	EP-	1621	30.70	8.64	8.96	-0.29	-0.03	0.65	0	0	0.00	0					S	1	37.60	15.54	0.11	0.5			
PBX	55.6	231	102	LPD	1621	31.23	9.17	9.08	0.00	0.08	0.64	0	0	0.00	0					S	1	38.03	15.97	-0.20	0.4				

Cabecera:

N0: Número de fases leídas P y S con ponderaciones finales mayores a 0.1.

DM: Distancia a la estación más cercana (en km).

GAP: Separación azimutal (en grados) más larga entre las estaciones.

M: 1.

RMS: Error cuadrático medio de los residuales de tiempo (en s).

ERH: Error horizontal (en km).

ERZ: Error vertical (en km).

Q: Calidad del hipocentro:

Q	Epicentro	Profundidad Focal
A	Excelente	Muy bueno
B	Muy bueno	Bueno
C	Bueno	Pobre
D	Pobre	Pobre

SQD: Clasificación de QS y dD (no utilizable).

ADJ: Último ajuste de hipocentro en km. Normalmente, este es igual a Q o menor que 0.005.

IN: Código de instrucción.

NR: Número de lecturas, incluido las no utilizados, para determinar el hipocentro.

AVR: Promedio de los residuales (en s).

AAR: Promedio de los residuales absolutos (en s).

NM: Número de lecturas de estación disponibles para calcular la magnitud de amplitud máxima.

AVXM: Promedio de XMAG (magnitud de la amplitud máxima calculada) de las estaciones disponibles.

SDXM: Desviación estándar de XMAG (magnitud de la amplitud máxima calculada) de las estaciones disponibles.

NF: Número de lecturas de estaciones disponibles para calcular la magnitud de F-P (FMAG).

AVFM: Promedio de FMAG de las estaciones disponibles.

SDFM: Desviación estándar de FMAG de las estaciones disponibles.

I: Número de iteraciones realizadas para llegar a un valor hipocentral final.

Cuerpo:

STN: Nombre de la estación.

DIST: Distancia epicentral (en km).

AZM: Ángulo azimutal entre el epicentro y la estación calculado desde el norte (en grados).

AIN: Ángulo de incidencia respecto a la vertical descendente.

PRMK: Fase registrada (I, impulsiva; E, emergente; polaridad se representa con + o -).

HRMN: Hora y minuto del tiempo de arribo de los datos de entrada.

P-SEC: La porción de los segundos del tiempo de la onda P.

TPOBS: Tiempos de viaje observados de P (en s). $TPOBS = T + DT - ORG$, donde T = P-tiempo de arribo, DT = corrección de tiempo, and ORG = tiempo de origen.

TPCAL: Tiempo de viaje calculado (en s).

DLY/H1: Si se utiliza un *Delay Model* para la estación, DLY significa el retraso de esta (en s), obtenido del archivo de entrada de estaciones. Si se utiliza un modelo *First Layer*, entonces H1 significa el espesor de la primera capa (en km) en la estación.

P-RES: Residual del tiempo de arribo de P en segundos. Si hay "***" significa que en la ponderación de Jeffrey de tiempo de arribo de P no es confiable.

P-WT: Ponderación utilización utilizado en la solución del hipocentro para los arribos de P, es una combinación de control de calidad.

AMX: Amplitud máxima en mm.

PRX: Periodo de amplitud máxima (en s).

CALX: Calibración utilizada para calcular XMAG (en mm).

K: Número de sistema de la estación.

XMAG: Magnitud de la amplitud máxima calculada.

RMK: Remark de los datos de entrada

FMP: F-P en segundos de los datos de entrada.

FMAG: Magnitud de F-P calculada de F-P y DIST.

SRMK: Registro de fase S.

S-SEC: La porción de segundos del tiempo arribo de S (después del origen).

TSOBS: Tiempos de viaje observados de S (en s); TPOBS.

S-RES: Residual del arribo de S (en s).

S-WT: Ponderación utilizada en la solución del hipocentro para los arribos S.

DT: Corrección de tiempo para la estación (en s). Se utiliza para corregir todas las estaciones a la misma base de tiempo.

Línea 1	
1	Libre
2 a 5	Año
6	Libre
7 a 8	Mes
9 a 10	Día
11	Comúnmente en blanco, F si el tiempo de origen es fijo
12 a 13	Hora
14 a 15	Minutos
16	Libre
17 a 20	Segundos
21	Indicador de modelo de localización
22	Indicador de distancia
23	ID de evento
24 a 30	Latitud
31 a 38	Longitud
39 a 43	Profundidad
44	Indicador de profundidad
45	Indicador de localización
46 a 48	Agencia encargada de reportar el hipocentro
49 a 51	Número de estaciones utilizadas
52 a 55	RMS de los residuales de tiempo
56 a 59	Magnitud No. 1
60	Tipo de magnitud
61 a 63	Agencia encargada de reportar la magnitud
64 a 67	Magnitud No. 2
68	Tipo de magnitud
69 a 71	Agencia encargada de reportar la magnitud
72 a 75	Magnitud No. 3
76	Tipo de magnitud
77 a 79	Agencia encargada de reportar la magnitud
80	Tipo de línea

Línea 6
Nombre del archivo

Línea 7	
1	Libre
2 a 6	Nombre de la estación
7	Tipo de instrumento
8	Componente
9	Libre o calidad
10	Indicador de calidad
11 a 14	ID de fase
15	Ponderación
16	Libre o indicador de marcado de fase automático
17	Primer movimiento o polaridad (C o D)
18	Libre
19 a 20	Hora (puede ser hasta 48 para indicar siguiente día)
21 a 22	Minutos
23 a 28	Segundos
29	Libre
30 a 33	Duración (en segundos)
34 a 40	Amplitud (en nm, nm/s, nm/s ² o counts)
41	Libre
42 a 45	Periodo (segundos)
46	Libre
47 a 51	Back azimuth (en grados)
52	Libre
53 a 56	Velocidad de fase (en km/s)
57 a 60	Ángulo de incidencia (en grados)
61 a 63	Residual del back azimuth
64 a 68	Residual de tiempo de viaje
69 a 70	Ponderación
71 a 75	Distancia epicentral (en km)
76	Libre
77 a 79	Azimut en origen
80	Tipo de línea

Línea E	
1	Libre
2 a 5	Texto "GAP="
6 a 8	Gap*
9	Libre
10	Indicador del programa de localización. En blanco para versiones actualizadas.
11	Libre
12 a 14	Agencia
15 a 20	Error de tiempo de origen
25 a 30	Error de latitud (y)
31 a 32	Libre
33 a 38	Error de longitud (x; en km)
39 a 43	Error de profundidad (z; en km)
44 a 55	Covarianza (x,y)
56 a 67	Covarianza (x,z)
68 a79	Covarianza (y,z)
80	Tipo de línea

Línea I	
1	Libre
2 a 8	Texto auxiliar para el indicador de acción
9 a 11	Última acción realizada
12	Libre
13 a 26	Fecha y hora de la última acción
27	Libre
28 a 30	Texto auxiliar para operador
36 a 42	Texto auxiliar para estatus
43 a 56	Registro FLAGS, no definido aún
57	Libre
58 a 60	Texto auxiliar para el ID
61 a 74	ID, año a segundo
75	Si es d, esto indica que un nuevo archivo ID se creó y este tenía uno o más segundos de diferencia a un ID existente
76	Indica si el ID se ha sincronizado con el tiempo de origen. Si está en blanco significa que no se ha sincronizado, S significa sincronizado
77 a 79	Libre
80	Tipo de línea

*Gap: separación máxima (en grados) entre dos estaciones adyacentes, medida desde el epicentro de un sismo. Proporciona información importante sobre la distribución de las estaciones de una red.

Anexo II: Descripción del archivo phaselist

El archivo phaselist comprende todas las fases registradas de 1979 a 2020 en el catálogo de RESNOM. Cada evento consiste en una cabecera y un cuerpo, como se muestra a continuación:

		52059	2011	01	20	11	22	35.400	32.65500	-115.83300	4.0000	1.80
BC	YUH	P	36.990		8.4000							
BC	SGL	P	36.990		10.000							
BC	WES	P	37.860		15.000							
BC	WES	S	40.680		15.000							
BC	ERP	P	38.680		18.700							
BC	COK	P	39.380		23.800							
BC	IBP	P	39.530		24.300							
BC	IBP	S	42.710		24.300							
BC	BAR	P	47.980		78.700							
BC	CBX	P	49.610		86.700							
BC	CBX	S	0.0300		86.700							
BC	SDR	P	52.120		104.00							
BC	SPIG	P	3.6300		182.00							

.

.

.

La cabecera indica lo siguiente:

- ID del evento
- Año
- Mes
- Día
- Hora
- Minuto
- Segundo
- Latitud
- Longitud
- Profundidad en km.
- Magnitud

El cuerpo indica lo siguiente:

- Localización de la estación (para identificar si esta se encuentra en el estado de Baja California o si es estación de una red estadounidense).
- Nombre de la estación
- Fase registrada
- Para fase P: TPOBS, valor que indica tiempos de viaje observados de P, en segundos. Para fase S: S-SEC, valor que indica la porción de segundos del tiempo arribo de S (segundos después del origen).
- DIST, es decir, distancia epicentral en km.

Anexo III: Scripts de conversión

A. RESNOM

Script principal

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 15 17:51:43 2022

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR
"""
import glob
# from collections import OrderedDict
# import decimal
from hypoaux import h_time_hypo71, h_loc_hypo71, h_info_hypo71, p_phase,
s_phase
# decimal.getcontext().rounding = decimal.ROUND_DOWN

files=sorted(glob.glob('resnom/*.hyp'))

for file in files:
# file='1990-06-30-23-46-53.30.hyp'
# file= '1981-09-18-11-51-02.10.hyp'
    print(file)
    agency='RES' # Three-letter agency code
    mag_ty='L' # Type of magnitude L=ML, b=mb, B=mB, s=Ms, S=MS, W=MW,
G=MbLg
    YYYY, MM, DD, hh, mm, ss, num = h_time_hypo71(file)
    lat, lon, dep, fix= h_loc_hypo71(file)
    ns, rms, mag1, mag2, gap, erh, erz =h_info_hypo71(file)

    o_filename=str(DD)+'-'+str(hh)+str(mm)+'-
'+str(num)+'L.S'+str(YYYY)+str(MM)
    #Header of Nordic file
    with open('faltantes/'+o_filename,'w') as out_file:
        # Header 1st line
        out_file.write(' '+str(YYYY)+' '+str(MM)+str(DD)+'
'+str(hh)+str(mm)+' '+str(ss)+
' L'+lat+lon+dep+fix+' '+agency+ns+' '+rms+'
'+mag1+mag_ty+agency+' '+mag2+mag_ty+agency+' 1\n')
        out_file.write(' GAP='+gap+' '+erz+' '+erh+' '+erh+'
E\n')
        out_file.write(' ACTION:REG 23-01-01 01:01 OP:erik STATUS:
ID:'+
YYYY+MM+DD+hh+mm+ss[0:2]+' I\n')
```

```

        out_file.write(' STAT SP IPHASW D HRMN SECON CODA AMPLIT PERI
AZIMU VELO AIN AR TRES W DIS CAZ7\n')
    # Header of phase section
    idd=[]
    index=1
    f=open(file,'r')
    for line in f:
        if 'STATION' in line:
            idd.append(index)
            index+=1
    f.close()

    for row in range(int(idd[0])-1,int(idd[0])+100):
        f=open(file,'r')
        line=f.readlines()[row+1]
        # print(line)
        if line==' \n':
            break
        if line=='\n':
            break
        else:
            p_phase(file, line, out_file)
            if line[31:32]=='S':
                s_phase(file, line, out_file)

```

Script auxiliar

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 9 13:43:40 2022

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""

def ev_coordinates(file):
    # Extract latitude and longitude of each file/event.
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[23:24]
    LO=qline[32:33]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()

```

```

f=open(file,'r')
info=f.readlines()[1]
lat=int(info[18:20])
lat_min=float(info[21:26])
lat=la_si*(lat+(lat_min/60))

lon=int(info[27:30])
lon_min=float(info[31:36])
lon=lo_si*(lon+(lon_min/60))
f.close()
return lat, lon

def back_azimuth(stcn, evla, evlo):
    import numpy as np
    from obspy.geodetics.base import gps2dist_azimuth

    file='stlist.in'
    stlis=np.loadtxt(file, usecols=(1,), dtype=str)
    latitudes=np.loadtxt(file, usecols=(2,), dtype=float)
    longitudes=np.loadtxt(file, usecols=(3,), dtype=float)

    j=0
    for i in stlis:
        if i==stcn:
            break
        else:
            j=j+1
    stla=latitudes[j]
    stlo=longitudes[j]
    caz=gps2dist_azimuth(evla, evlo, stla,stlo, a=6378137.0,
f=0.0033528106647474805)

    return caz

def evst_dist(stcn, evla, evlo):
    import numpy as np
    import geopy.distance

    file='stlist.in'
    stlis=np.loadtxt(file, usecols=(1,), dtype=str)
    latitudes=np.loadtxt(file, usecols=(2,), dtype=float)
    longitudes=np.loadtxt(file, usecols=(3,), dtype=float)

    j=0
    for i in stlis:
        i=i.ljust(5)
        if i==stcn:
            break
        else:
            j=j+1
    stla=latitudes[j]
    stlo=longitudes[j]
    evstdis=geopy.distance.geodesic([evla, evlo], [stla,stlo]).km

```

```

return evstdis

def stime_correction(hour, minutes, seconds):
    if seconds>=60:
        seconds=seconds-60
        minutes=minutes+1
        if minutes>=60:
            minutes=minutes-60
            hour=hour+1
    hour=str(hour).rjust(2,' ')
    minutes=str(minutes).rjust(2,' ')
    seconds=str(seconds)[0:5]
    return hour, minutes, seconds

def h_time_hypo71(file):    # Reading Origin Time
    f=open(file,'r')
    info=f.readlines()[1]

    #date
    date=info.split()[0]
    YYYY='19'+date[0:2]
    MM=date[2:4]
    DD=date[4:6]
    #time
    hh=int(info[7:9])
    mm=int(info[9:11])
    hh=str(hh).rjust(2,'0')
    mm=str(mm).rjust(2,'0')

    #segundos
    num=int(info[12:14])
    num=str(num).rjust(2,'0')
    den=int(info[15:17])
    den=str(den)[0].ljust(2,'0')
    sec=num+'.'+den

    return YYYY, MM, DD, hh, mm, sec, num

def h_loc_hypo71(file): # Reading Location information
    # Latitude and Longitude #####
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[23:24]
    LO=qline[32:33]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()

```

```

f=open(file,'r')
info=f.readlines()[1]
lat=int(info[18:20])
lat_min=float(info[21:26])
lat=la_si*(lat+(lat_min/60))
lat=round(lat,3)
lat=str(format(lat,'0.3f'))

lon=int(info[27:30])
lon_min=float(info[31:36])
lon=lo_si*(lon+(lon_min/60))
lon=round(lon,3)
lon=str(format(lon,'.3f'))

# Empty spaces corrections
if len(lat)==5:
    lat=' '+lat
if len(lat)==6:
    lat=' '+lat
if len(lat)==7:
    lat=' '+lat

if len(lon)==5:
    lon=' '+lon
if len(lon)==6:
    lon=' '+lon
if len(lon)==7:
    lon=' '+lon

# Depth #####
dep=float(info[36:42])
dep=round(dep,1)
dep=str(format(dep,'.1f'))

if len(dep)==3:
    dep=' '+dep
if len(dep)==4:
    dep=' '+dep

# Fixed case #####
fix=str(info[42:43])
if (fix=='*'):
    fix='F'
else:
    fix=' '
f.close()
return lat, lon, dep, fix

def h_info_hypo71(file): #Reading additinoal information
f=open(file,'r')
info=f.readlines()[1]
# Number of stations #####
idd=[]
index=1

```

```

f=open(file,'r')
for line in f:
    if 'STATION' in line:
        idd.append(index)
        index+=1
f.close()

stcns=[]
for p in range(int(idd[0]),int(idd[0])+100):
    f=open(file,'r')
    line=f.readlines()[p]
    if line==' \n':
        break
    if line=='\n':
        break
    else:
        stcns.append(line[0:5])

number_stations=[]
for item in stcns:
    if item not in number_stations:
        number_stations.append(item)
ns=format(len(number_stations),'3.0f')
ns=str(ns)

# RMS #####
rms=float(info[55:60])
rms=str(round(rms,1))

# Magnitude #####
mag1=info[72:75]
if mag1==' ':
    mag1==' '
if mag1==' ':
    mag1==' '
else:
    mag1=float(mag1)
    mag1=str(round(mag1,1))
mag2=mag1

# GAP #####
gap=float(info[47:50])
gap=format(gap,'3.0f')
gap=str(gap)
#if len(gap)==1:
#    gap=' '+gap
#if len(gap)==2:
#    gap=' '+gap

# Errors #####
erh=info[61:65]
if erh==' ':
    erh=' '
else:

```

```

        erh=float(erh)
        erh=str(format(erh,'6.1f'))
    erz=info[66:70]
    if erz==' ':
        erz=' '
    else:
        erz=float(erz)
        erz=str(format(erz,'6.1f'))

    return ns, rms, mag1, mag2, gap, erh, erz

def p_phase(file, line, out_file):
    import numpy as np

    # Header info needed.
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[23:24]
    LO=qline[32:33]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()
    f=open(file,'r')
    info=f.readlines()[1]
    lat=int(info[18:20])
    lat_min=float(info[21:26])
    lat=la_si*(lat+(lat_min/60))

    lon=int(info[27:30])
    lon_min=float(info[31:36])
    lon=lo_si*(lon+(lon_min/60))
    f.close()

    # print(line)

    string=line[0:5] #station name
    stcn=string.strip()
    stcn=stcn.ljust(5)
    qind=line[9:10] #E or I
    phase=line[10:11] #P-phase lecture
    wp=line[12:13] #Weight of phase: empty or 0, 1, 2, 3, 4
    pol=line[11:12] #Polarity
    if pol=='+' :
        polar='C'
    if pol=='C' :
        polar='C'
    if pol=='U' :
        polar='C'

```

```

if pol=='-':
    polar='D'
if pol=='D':
    polar='D'
if pol=='N':
    polar='N'
if pol==' ':
    polar=' '

hour=line[16:18] #Hour
min=line[20:22] #Minute
sec=line[23:28] #Seconds
coda=' ' # Add coda time if have it
amp=0 #Maximum amplitude (nm)
if amp==0.0:
    amp=' '
else:
    amp=str(format(amp, '7.1f'))
# per=float(line[70:73]) #Period of max amplitude
per=0
if per==0.0:
    per=' '
else:
    per=str(format(per, '4.0f'))

velo=' ' # Phase Velocity Km/s (add value if you have it)
ainc=' ' #Angle of incidence
ar=' ' #BAZ residual (add value if you have it)
tres=' '
wei='10'
evla, evlo=ev_coordinates(file)

AZ=back_azimuth(str(string.strip()),evla, evlo) ##Azimuth and
BackAzimuth at station
baz=round(float(AZ[1]))
baz=str(format(baz, '3.0f'))
caz=round(float(AZ[2]))
caz=str(format(caz, '3.0f'))

dist=float(evst_dist(stcn, evla, evlo)) #Distance Source to Station
dist=str(format(dist, '4.0f'))

out_file.write(' '+stcn+'SZ '+qind+phase+' '+wp+' '+polar+'
'+hour+min+' '+
    sec+' '+coda+' '+amp+' '+per+' '+baz+' '+velo+' '+ainc+' '+ar+'
'+tres+
    wei+' '+dist+' '+caz+' \n')

def s_phase(file, line, out_file):
    import numpy as np

    # Header info needed.
    f=open(file, 'r')
    qline=f.readlines()[0]

```

```

LA=qline[23:24]
LO=qline[32:33]
if LA=='N':
    la_si=1
if LA=='S':
    la_si=-1
if LO=='E':
    lo_si=1
if LO=='W':
    lo_si=-1
f.close()

f=open(file,'r')
info=f.readlines()[1]
lat=int(info[18:20])
lat_min=float(info[21:26])
lat=la_si*(lat+(lat_min/60))

lon=int(info[27:30])
lon_min=float(info[31:36])
lon=lo_si*(lon+(lon_min/60))
f.close()

string=line[0:5] #station name
stcn=string.strip()
stcn=stcn.ljust(5)
qind=line[30:31] #E or I
phase=line[31:32] #S-phase lecture
wp=line[33:34] #Weight of phase: empty or 0, 1, 2, 3, 4
pol=line[32:33] #Polarity
if pol=='+' :
    polar='C'
if pol=='U' :
    polar='C'
if pol=='C' :
    polar='C'
if pol=='-' :
    polar='D'
if pol=='D' :
    polar='D'
if pol==' ' :
    polar=' '

hh=line[16:18] #Hour
mm=line[20:22] #Minute
ss=float(line[40:45])
hour, min, sec=stime_correction(int(hh), int(mm), ss)
sec=sec.rjust(5, ' ')

coda=' ' # Add coda time if have it
amp=0 #Maximum amplitude (nm)
if amp==0.0:
    amp=' '
else:

```

```

        amp=str(format(amp, '7.1f'))
# per=float(line[70:73]) #Period of max amplitude
per=0
if per==0.0:
    per='    '
else:
    per=str(format(per, '4.0f'))

velo='    ' # Phase Velocity Km/s (add value if you have it)
ainc='    ' #Angle of incidence
ar='    ' #BAZ residual (add value if you have it)
tres='    '
wei='10'
evla, evlo=ev_coordinates(file)

AZ=back_azimuth(str(string.strip()),evla, evlo) ##Azimuth and
BackAzimuth at station
baz=round(float(AZ[1]))
baz=str(format(baz, '3.0f'))
caz=round(float(AZ[2]))
caz=str(format(caz, '3.0f'))

dist=float(evst_dist(stcn, evla, evlo)) #Distance Source to Station
dist=str(format(dist, '4.0f'))

    out_file.write(' '+stcn+'SZ '+qind+phase+'    '+wp+' '+polar+'
'+hour+min+' '+
        sec+' '+codat+' '+amp+' '+per+'    '+baz+' '+velo+' '+ainc+' '+ar+'
'+tres+
        wei+' '+dist+' '+caz+' \n')

```

B. Hypo71

Script principal

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 15 17:51:43 2022

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""
import glob
# from collections import OrderedDict
# import decimal

```

```

from hypo71aux import h_time_hypo71, h_loc_hypo71, h_info_hypo71, p_phase,
s_phase
# decimal.getcontext().rounding = decimal.ROUND_DOWN

files=sorted(glob.glob('hypo71/err2/*.hyp'))

for file in files:
    print(file)
    agency='RES' # Three-letter agency code
    mag_ty='L' # Type of magnitude L=ML, b=mb, B=mB, s=Ms, S=MS, W=MW,
G=MbLg
    YYYY, MM, DD, hh, mm, ss, num= h_time_hypo71(file)
    lat, lon, dep, fix= h_loc_hypo71(file)
    ns, rms, mag1, mag2, gap, erh, erz =h_info_hypo71(file)

    o_filename=str(DD)+'-'+str(hh)+str(mm)+'-
'+str(num)+'L.S'+19'+str(YYYY)+str(MM)
    #Header of Nordic file
    with open('extra/'+o_filename,'w') as out_file:
        # Header 1st line
        out_file.write(' '+19'+str(YYYY)+' '+str(MM)+str(DD)+'
'+str(hh)+str(mm)+' '+str(ss)+
' L'+lat+lon+dep+fix+' '+agency+ns+' '+rms+'
'+mag1+mag_ty+agency+
' '+mag2+mag_ty+agency+' 1\n')
        out_file.write(' GAP='+gap+' '+erz+' '+erh+' '+erh+'
E\n')
        out_file.write(' STAT SP IPHASW D HRMN SECON CODA AMPLIT PERI AZIMU
VELO AIN AR TRES W DIS CAZ7\n')

    # Header of phase section
    idd=[]
    index=1
    f=open(file,'r')
    for line in f:
        if 'STN' in line:
            idd.append(index)
            index+=1
    f.close()

    for row in range(int(idd[0])-1,int(idd[0])+100):
        f=open(file,'r')
        line=f.readlines()[row+1]
        # print(line)
        if line==' \n':

```

```

        break
    if line=='\n':
        break
    else:
        p_phase(file, line, out_file)
        if line[100:101]=='S':
            s_phase(file, line, out_file)

```

Scrip auxiliar

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Nov  9 13:43:40 2022

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""

def ev_coordinates(file):
    # Extract latitude and longitude of each file/event.
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[24:25]
    LO=qline[34:35]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()
    f=open(file,'r')
    info=f.readlines()[1]
    lat=int(info[19:21])
    lat_min=float(info[22:27])
    lat=la_si*(lat+(lat_min/60))

    lon=int(info[28:31])
    lon_min=float(info[32:37])
    lon=lo_si*(lon+(lon_min/60))
    f.close()

```

```

    return lat, lon

def back_azimuth(stcn, evla, evlo):
    import numpy as np
    from obspy.geodetics.base import gps2dist_azimuth

    file='stlist.in'
    stlis=np.loadtxt(file, usecols=(1,), dtype=str)
    latitudes=np.loadtxt(file, usecols=(2,), dtype=float)
    longitudes=np.loadtxt(file, usecols=(3,), dtype=float)

    j=0
    for i in stlis:
        if i==stcn:
            break
        else:
            j=j+1
    stla=latitudes[j]
    stlo=longitudes[j]
    caz=gps2dist_azimuth(evla, evlo, stla,stlo, a=6378137.0,
f=0.0033528106647474805)

    return caz

def stime_correction(hour, minutes, seconds):
    if seconds>=60:
        seconds=seconds-60
        minutes=minutes+1
        if minutes>=60:
            minutes=minutes-60
            hour=hour+1
    hour=str(hour).rjust(2,' ')
    minutes=str(minutes).rjust(2,' ')
    seconds=str(seconds)[0:5]
    return hour, minutes, seconds

def h_time_hypo71(file): # Reading Origin Time
    f=open(file,'r')
    info=f.readlines()[1]

    #date
    date=info.split()[0]
    YYYY=date[0:2]
    MM=date[2:4]
    DD=date[4:7]
    #time
    hh=int(info[8:10])

```

```

mm=int(info[10:13])
hh=str(hh).rjust(2,'0')
mm=str(mm).rjust(2,'0')

#segundos
num=int(info[13:15])
num=str(num).rjust(2,'0')
den=int(info[16:18])
den=str(den)[0]#.ljust(2,'0')
sec=num+'.'+den

return YYYY, MM, DD, hh, mm, sec, num

def h_loc_hypo71(file): # Reading Location information
#   Latitude and Longitude #####
f=open(file,'r')
qline=f.readlines()[0]
LA=qline[24:25]
LO=qline[34:35]
if LA=='N':
    la_si=1
if LA=='S':
    la_si=-1
if LO=='E':
    lo_si=1
if LO=='W':
    lo_si=-1

f=open(file,'r')
info=f.readlines()[1]
lat=int(info[19:21])
lat_min=float(info[22:27])
lat=la_si*(lat+(lat_min/60))
lat=round(lat,3)
lat=str(format(lat,'0.3f'))

lon=int(info[28:31])
lon_min=float(info[32:37])
lon=lo_si*(lon+(lon_min/60))
lon=round(lon,3)
lon=str(format(lon,'.3f'))

# Empty spaces corrections
if len(lat)==5:
    lat=' '+lat
if len(lat)==6:
    lat=' '+lat

```

```

if len(lat)==7:
    lat=' '+lat

if len(lon)==5:
    lon=' '+lon
if len(lon)==6:
    lon=' '+lon
if len(lon)==7:
    lon=' '+lon

# Depth #####
dep=float(info[39:44])
dep=round(dep,1)
dep=str(format(dep, '.1f'))

if len(dep)==3:
    dep=' '+dep
if len(dep)==4:
    dep=' '+dep

# Fixed case #####
fix=str(info[44:45])
if (fix=='*'):
    fix='F'
else:
    fix=' '

return lat, lon, dep, fix

def h_info_hypo71(file): #Reading additinoal information
f=open(file,'r')
info=f.readlines()[1]

# Number of stations #####
idd=[]
index=1
f=open(file,'r')
for line in f:
    if 'STN' in line:
        idd.append(index)
        index+=1
f.close()

stcns=[]
for p in range(int(idd[0]),int(idd[0])+100):
    f=open(file,'r')
    line=f.readlines()[p]

```

```

    if line==' \n':
        break
    if line=='\n':
        break
    else:
        stcns.append(line[1:5])

number_stations=[]
for item in stcns:
    if item not in number_stations:
        number_stations.append(item)
ns=format(len(number_stations),'3.0f')
ns=str(ns)

# RMS #####
rms=float(info[64:68])
rms=str(round(rms,1))

# Magnitude #####
mag1=info[48:51]
if mag1==' ':
    mag1=''
else:
    mag1=float(info[47:51])
    mag1=str(round(mag1,1))
mag2=mag1

# GAP #####
gap=float(info[58:61])
gap=format(gap,'3.0f')
gap=str(gap)
# if len(gap)==1:
#     gap=' '+gap
# if len(gap)==2:
#     gap=' '+gap

# Errors #####
erh=info[69:73]
if erh==' ':
    erh=' '
else:
    erh=float(erh)
    erh=str(format(erh,'6.1f'))
erz=info[74:78]
if erz==' ':
    erz=' '
else:

```

```

    erz=float(erz)
    erz=str(format(erz,'6.1f'))

return ns, rms, mag1, mag2, gap, erh, erz

def p_phase(file, line, out_file):
    import numpy as np

    # Header info needed.
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[24:25]
    LO=qline[34:35]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()
    f=open(file,'r')
    info=f.readlines()[1]
    lat=int(info[19:21])
    lat_min=float(info[22:27])
    lat=la_si*(lat+(lat_min/60))

    lon=int(info[28:31])
    lon_min=float(info[32:37])
    lon=lo_si*(lon+(lon_min/60))
    f.close()

    # for p in range(int(idd[0]),int(idd[0])+100):
    # f=open(file,'r')
    # line=f.readlines()[row+1]

    # print(line)

    string=line[1:5] #station name
    stcn=string.strip()
    stcn=stcn.ljust(5)
    qind=line[20:21] #E or I
    phase=line[21:22] #P-phase lecture
    wp=line[23:24] #Weight of phase: empty or 0, 1, 2, 3, 4
    pol=line[22:23] #Polarity

```

```

if pol=='+' :
    polar='C'
if pol=='C' :
    polar='C'
if pol=='U' :
    polar='C'
if pol=='-' :
    polar='D'
if pol=='D' :
    polar='D'
if pol=='N' :
    polar=' '
if pol==' ' :
    polar=' '
##SI HAY ASTERISCOS modificar##
hour=line[25:27] #Hour
min=line[27:29] #Minute
sec=line[30:35] #Seconds
coda=' ' # Add coda time if have it
amp=float(line[66:69])*1000000 #Maximum amplitude (nm)
if amp==0.0:
    amp=' '
else:
    amp=str(format(amp, '7.1f'))
per=float(line[70:73]) #Period of max amplitude
if per==0.0:
    per=' '
else:
    per=str(format(per, '4.0f'))

baz=float(line[12:15]) ##Back Azimuth
baz=str(format(baz, '3.0f'))

velo=' ' # Phase Velocity Km/s (add value if you have it)
ainc=float(line[16:19]) #Angle of incidence
ainc=str(format(ainc, '3.0f'))
ar=' ' #BAZ residual (add value if you have it)

TRES=str(line[54:59]) #Travel-time residual
tres=TRES.strip()
if len(tres)==4:
    tres=tres.ljust(4, '0')
if len(tres)==5:
    tres=round(float(tres), 1)
    tres=str(tres)

ttweight=line[59:61]

```

```

if ttweight=='**':
    wei='00'
if ttweight==' ':
    wei='10'

dist=float(line[6:11]) #Distance Source to Station
dist=str(format(dist,'4.0f'))

azi=line[12:15] #Azimuth at souce

# print(baz)
lat, lon=ev_coordinates(file)
caz=back_azimuth(str(string.strip()),lat, lon) ##Azimuth at station
caz=round(float(caz[2]))
caz=str(format(caz, '3.0f'))

    out_file.write(' '+stcn+'SZ '+qind+phase+' '+wp+' '+polar+'
'+hour+min+' '+
    sec+' '+coda+' '+amp+' '+per+' '+baz+' '+velo+' '+ainc+' '+ar+'
'+tres+
    wei+' '+dist+' '+caz+' \n')

def s_phase(file, line, out_file):
    import numpy as np

    # Header info needed.
    f=open(file,'r')
    qline=f.readlines()[0]
    LA=qline[24:25]
    LO=qline[34:35]
    if LA=='N':
        la_si=1
    if LA=='S':
        la_si=-1
    if LO=='E':
        lo_si=1
    if LO=='W':
        lo_si=-1
    f.close()

    f=open(file,'r')
    info=f.readlines()[1]
    lat=int(info[19:21])
    lat_min=float(info[22:27])
    lat=la_si*(lat+(lat_min/60))

    lon=int(info[28:31])

```

```

lon_min=float(info[32:37])
lon=lo_si*(lon+(lon_min/60))
f.close()

# for p in range(int(idd[0]),int(idd[0])+100):
# f=open(file,'r')
# line=f.readlines()[row+1]
# print(line)

string=line[1:5] #station name
stcn=string.strip()
stcn=stcn.ljust(5)
qind=line[99:100] #E or I
phase=line[100:101] #S-phase lecture
wp=line[102:103] #Weight of phase: empty or 0, 1, 2, 3, 4
pol=line[101:102] #Polarity
if pol=='+' :
    polar='C'
if pol=='U' :
    polar='C'
if pol=='C' :
    polar='C'
if pol=='-' :
    polar='D'
if pol=='D' :
    polar='D'
if pol==' ' :
    polar=' '

#Origin Time
f=open(file,'r')
info=f.readlines()[1]
hh=int(info[8:10])
mm=int(info[10:13])
hh=str(hh).rjust(2,'0')
mm=str(mm).rjust(2,'0')
ss=float(info[13:18])
s_ss=float(line[104:109])
ss=ss+s_ss
hour, min, sec=stime_correction(int(hh), int(mm), ss)
sec=str(format(float(sec), '2.2f'))
sec=str(sec).ljust(5,'0')
# print(len(sec))
f.close()

coda=' ' # Add coda time if have it

```

```

amp=float(line[66:69])*1000000 #Maximum amplitude (nm)
if amp==0.0:
    amp='      '
else:
    amp=str(format(amp, '7.1f'))
per=float(line[70:73]) #Period of max amplitude
if per==0.0:
    per='      '
else:
    per=str(format(per,'4.0f'))

baz=float(line[12:15]) ##Back Azimuth
baz=str(format(baz, '3.0f'))

velo='      ' # Phase Velocity Km/s (add value if you have it)
ainc=float(line[16:19]) #Angle of incidence
ainc=str(format(ainc, '3.0f'))
ar='      ' #BAZ residual (add value if you have it)

TRES=str(line[116:121]) #Travel-time residual
tres=TRES.strip()
if len(tres)==4:
    tres=tres.ljust(4,'0')
if len(tres)==5:
    tres=round(float(tres),1)
    tres=str(tres)

wei='10'
dist=float(line[6:11]) #Distance Source to Station
dist=str(format(dist,'4.0f'))

azi=line[12:15] #Azimuth at souce

# print(baz)
lat, lon=ev_coordinates(file)
caz=back_azimuth(str(string.strip()),lat, lon) ##Azimuth at station
caz=round(float(caz[2]))
caz=str(format(caz, '3.0f'))

    out_file.write('      '+stcn+'SZ      '+qind+phase+'      '+wp+'      '+polar+'
'+hour+min+' '+
    sec+' '+codat+' '+amp+' '+per+'      '+baz+' '+velo+' '+ainc+' '+ar+'
'+tres+
    wei+' '+dist+' '+caz+' \n')

```

Anexo IV: Scripts para creación de archivos de entrada

A. Eventlist

RESNOM

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
Created on Tue Mar 15 11:11:15 2022
@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""

import glob

def name_file(file,idd):
    f=open(file,'r')
    info=f.readlines()[1]

    #date
    year=info[0:2]
    YYYY='19'+str(year)
    print(YYYY)

    MM=info[2:4]
    if MM[0]==' ':
        MM='0'+MM[1]
    DD=info[4:6]

    #time
    hh=int(info[7:9])
    mm=int(info[9:11])
    hh=str(hh).rjust(2,'0')
    mm=str(mm).rjust(2,'0')

    #segundos
    num=int(info[12:14])
    num=str(num).rjust(2,'0')
    den=int(info[15:17])
    den=str(den).ljust(3,'0')
    sec=num+'.'+den
    print(sec)

    #latitud
    lat_deg=int(info[18:20])
    lat_sec=float(info[21:26])
    lat=lat_deg+(lat_sec/60)
    nlat=str(lat).ljust(8,'0')[0:8]
```

```

print(nlat)

#longitud
lon_deg=int(info[27:30])
lon_sec=float(info[31:36])
lon=(lon_deg+(lon_sec/60))*-1 #West
nlon=str(lon).ljust(10,'0')[0:10]

#profundidad
depth=float(info[37:42])
ndepth=str(depth).ljust(6,'0')[0:6]

#magnitud
mag=str(info[72:75])
if mag==' ':
    mag=0.00
nmag=str(mag).ljust(4,'0')[0:4]

# name=str(YYYY+'-'+MM+'-'+DD+'-'+hh+'-'+mm+'-'+sec+'.hyp')
name=str(" "+str(idd)+" "+str(YYYY)+" "+str(MM)+" "+str(DD)+"
"+str(hh)
        +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
        +" "+str(nmag)+"\n")
return name

files=glob.glob('separated_hypo/*.hyp') ##### EDITAR
#####
files=sorted(files)

idd=10000
with open('hypo_evlist.in','w') as evlis_file:
    for fil in range(len(files)):
        name=name_file(files[fil],idd)
        print(name)
        evlis_file.write(name)
        idd=idd+1

```

Hypo71

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 15 11:11:15 2022
@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""
import glob
def name_file(file,idd,year):

```

```

# for i in idd:
f=open(file,'r')
info=f.readlines()[1]

#date
#y=info[1:3]
#print(y)
# if y>='93':
#     YYYY='19'+y
# elif y==' ':
#     YYYY='2000'
# else:
#     YYYY='200'+y[1]
YYYY=str(year)
print(YYYY)

MM=info[3:5]
if MM[0]==' ':
    MM='0'+MM[1]
DD=info[5:7]

#time
hh=int(info[8:10])
mm=int(info[10:13])
hh=str(hh).rjust(2,'0')
mm=str(mm).rjust(2,'0')

#segundos
num=int(info[13:15])
num=str(num).rjust(2,'0')
den=int(info[16:18])
den=str(den).ljust(3,'0')
sec=num+'.'+den

#latitud
lat_deg=int(info[19:21])
lat_sec=float(info[22:27])
lat=lat_deg+(lat_sec/60)
nlat=str(lat).ljust(8,'0')[0:8]

#longitud
lon_deg=int(info[28:31])
lon_sec=float(info[32:37])
lon=(lon_deg+(lon_sec/60))*-1 #West
nlon=str(lon).ljust(10,'0')[0:10]

#profundidad
depth=float(info[39:44])
ndepth=str(depth).ljust(6,'0')[0:6]

#magnitud
mag=str(info[47:51])
if mag==' ':
    mag=0.00

```

```

nmag=str(mag).ljust(4,'0')[0:4]

name=str(YYYY+'-'+MM+'-'+DD+'-'+hh+'-'+mm+'-'+sec+'.hyp')
name=str("      "+str(idd)+" "+str(YYYY)+" "+str(MM)+" "+str(DD)+"
"+str(hh)
        +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
        +" "+str(nmag)+"\n")
return name

files=glob.glob('hypo71_separado/*.hyp') ##### EDITAR
#####
files=sorted(files)
idd=12091
with open('hypo71_evlist.in','w') as evlis_file:
    for fil in range(len(files)):
        year=int(files[fil][16:20])
        name=name_file(files[fil],idd,year)
        print(name)
        evlis_file.write(name)
        idd=idd+1

```

Nórdico

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 15 11:11:15 2022
@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""
import glob

def name_file(file, year):
    # for i in idd:
    f=open(file,'r')
    info=f.readlines()[0]

    #date
    # y=info[1:3]
    # print(y)
    # if y>='93':
    #     YYYY='19'+y
    # elif y==' ':
    #     YYYY='2000'
    # else:
    #     YYYY='200'+y[1]
    YYYY=str(year)

```

```

MM=info[6:8]
if MM[0]==' ':
    MM='0'+MM[1]
DD=info[8:10]
if DD[0]==' ':
    DD='0'+DD[1]

#time
hh=int(info[11:13])
mm=int(info[13:15])
hh=str(hh).rjust(2,'0')
mm=str(mm).rjust(2,'0')

#segundos
num=int(info[16:18])
num=str(num).rjust(2,'0')
den=int(info[19:20])
den=str(den).ljust(3,'0')
sec=num+'.'+den

#latitud
lat=float(info[24:30])
nlat=str(lat).ljust(8,'0')[0:8]

#longitud
lon=float(info[30:38])
nlon=str(lon).ljust(10,'0')[0:10]

#profundidad
depth=float(info[39:43])
ndepth=str(depth).ljust(6,'0')[0:6]

#magnitud
mag=(info[56:59])
print(mag)
# if mag==' ':
#     mag=0.00
nmag=str(mag).ljust(4,'0')[0:4]
print(nmag)
#Fixed case
# fix=str(info[43:44])
# if fix=='F' or fix=='f':
#     name=str(YYYY+'-'+MM+'-'+DD+'-'+hh+'-'+mm+'-'+sec+'.hyp')
# name=str("      "+str(YYYY)+" "+str(MM)+" "+str(DD)+" "+str(hh)
#         +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
#         +" "+str(nmag)+"\n")
name=str("      "+str(idd)+" "+str(YYYY)+" "+str(MM)+" "+str(DD)+"
"+str(hh)
        +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
        +" "+str(nmag)+"\n")
print(name)

```

```

# else:
#     name=' '
return name

files=glob.glob('acc_sfiles/20*') ##### EDITAR #####
files=sorted(files)

idd=22138
with open('event_list_nordic.out','w') as evlis_file:
    for fil in files:
        year=int(fil[11:15])
        name=name_file(fil, year)
        # if name!=' ':
            # print(name)
        evlis_file.write(name)
        idd=idd+1

```

B. Phaselist

RESNOM

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 15 11:11:15 2022

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""

import glob
import datetime as dt
import geopy.distance

def name_file(file,idd):
    f=open(file,'r')
    info=f.readlines()[1]

    #date
    year=info[0:2]
    YYYY='19'+str(year)

    MM=info[2:4]
    if MM[0]==' ':
        MM='0'+MM[1]
    DD=info[4:6]

    #time
    hh=int(info[7:9])

```

```

mm=int(info[9:11])
hh=str(hh).rjust(2,'0')
mm=str(mm).rjust(2,'0')

#segundos
num=int(info[12:14])
num=str(num).rjust(2,'0')
den=int(info[15:17])
den=str(den).ljust(2,'0')
sec=num+'.'+den

#latitud
lat_deg=int(info[18:20])
lat_sec=float(info[21:26])
lat=lat_deg+(lat_sec/60)
nlat=str(lat).ljust(8,'0')[0:8]

#longitud
lon_deg=int(info[27:30])
lon_sec=float(info[31:36])
lon=(lon_deg+(lon_sec/60))*-1 #West
nlon=str(lon).ljust(10,'0')[0:10]

#profundidad
depth=float(info[37:42])
ndepth=str(depth).ljust(6,'0')[0:6]

#magnitud
mag=str(info[72:75])
if mag==' ':
    mag=0.00
nmag=str(mag).ljust(4,'0')[0:4]

# name=str(YYYY+'-'+MM+'-'+DD+'-'+hh+'-'+mm+'-'+sec+'.hyp')
name=str(" "+str(idd)+" "+str(YYYY)+" "+str(MM)+" "+str(DD)+"
"+str(hh)
        +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
        +" "+str(nmag)+"\n")
return name

def distance(lat,lon,stcn):
#funcion que dado lat y lon de un epicentro y stcn a analizar, arroja
#la distancia entre evento-estacion (revisando la lista de estaciones)

    coor1=(lat, lon)
    # coor2=(LAT,LON)
    # dist=geopy.distance.geodesic(coor1, coor2).km
    dist=10
    return dist

```

```

def time_check(d,ho,mi,se):
    if se >= 60.0: #If seconds is bigger than 60
        se=se-60.0
        mi=mi+1
    if mi >= 60.0:
        mi=mi-60
        ho=ho+1

    if ho >= 24:
        ho=ho-24
        d=d+1
    se = str(se).ljust(5,'0')
    return se, mi, ho, d

def total_seconds(self):
    """Total seconds in the duration."""
    return ((self.days * 86400 + self.seconds) * 10**6 +
            self.microseconds) / 10**6

def phase_p(line, time):
    stcn=str(line[0:5]).strip()
    stcn=stcn.ljust(5)

    p=str(line[10:11])
    # print(p)
    hhmmss=str(str(line[16:18])+'-'+str(line[20:22])+'-'+str(line[23:28]))
    # print(hhmmss)
    return stcn, p, hhmmss

def phase_s(line):
    stcn=str(line[0:5]).strip()
    stcn=stcn.ljust(5)

    dist=10

    print(line[:])
    # print(line[:])
    # print(line[:])
    # hhmmss=str(str(line[25:27])+'-'+str(line[27:29])+'-'
'+str(line[104:109]))
    s=str(line[31:32])

    #segundos despues del tiempo de origen
    if str(line[40:45]) == ' ':
        sec='EE.EE'
    else:
        num=int(line[40:42])
        num=str(num).rjust(2,'0')
        den=int(line[43:45])
        den=str(den).rjust(2,'0')
        sec=num+'.'+den
        # print(sec)

    return stcn, s, dist, sec

```

```

files=glob.glob('separated_hypo/*.hyp') ##### EDITAR
#####
files=sorted(files)

idd=10000

with open('hypo_phaselist.in','w') as evlis_file:
    for fil in range(len(files)):
        #Cabecera de cada evento
        name=name_file(files[fil],idd)
        year=int(name[10:15])
        # print(year)
        #origin time
        if int(name[31:33])<10:
            ot=dt.datetime(int(name[10:15]), int(name[16:18]),
int(name[19:21]),
                                int(name[22:24]), int(name[25:27]),
int(name[28:30]),
                                int(name[31:33])*10000)
        else:
            ot=dt.datetime(int(name[10:15]), int(name[16:18]),
int(name[19:21]),
                                int(name[22:24]), int(name[25:27]),
int(name[28:30]),
                                int(name[31:33])*10000)
        print(name)
        evlis_file.write(name)

#fases leidas
f=open(files[fil],'r')
lines=f.readlines()
for p in range(4, 100):
    f=open(files[fil],'r')
    line=f.readlines()[p] #Extrac data from here

    if line.strip()=='':
        break
    if line.strip()=='\n':
        break
    if line=='\n':
        break

    #Extract p-phase info
    if line[39:41]=='*': #Condicionante de cuando no se toma
en cuenta la fase desechada
        break
    else:
        stcn, p, hhmms=phase_p(line,1)
        dist=distance(name[35:43], name[45:55], stcn)
        #time check of ORIGIN TIME

```

```

        se, mi, ho,
d=time_check(int(name[19:21]),int(hhmmss[0:2].strip()),
              int(hhmmss[3:5].strip()),
              float(hhmmss[6:11].strip()))

        sen, sed=str(se).split('.')

        #P-wave picks (with posible time correction)
        pt=dt.datetime(year,int(name[16:18]),int(d),
                       int(ho),int(mi),int(sen),
int(sed[0:2])*10000)
        ptd=(pt - ot).total_seconds()

        phase=str('BC '+stcn+' '+p+'
'+str(ptd).ljust(6,'0')+' '+str(dist)+'\n')
        evlis_file.write(phase)

        #S-wave picks (with time correction)
        if str(line[31:32])=='S':
            stcn, s, dist, sec=phase_s(line)

            if sec=='EE.EE':
                break
            else:
                #time check
                # se, mi, ho,
d=time_check(int(name[19:21]),int(hhmmss[0:2].strip()),
              int(hhmmss[3:5].strip()),
              float(hhmmss[6:11].strip()))
                # sen, sed=str(se).split('.')
                # st=dt.datetime(year,int(name[16:18]),int(d),
                #                 int(ho),int(mi),int(sen),
int(sed[0:2])*10000)
                # std=(st - ot).total_seconds()

                phase=str('BC '+stcn+' '+s+'
'+str(sec).ljust(6,'0')+' '+str(dist)+'\n')
                evlis_file.write(phase)
                # if std < 0:
                #     print(std)
                #     print(name)

            idd=idd+1

```

Hypo71

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

```

Created on Tue Mar 15 11:11:15 2022

```
@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR
```

```
"""
```

```
import glob
import datetime as dt

def name_file(file,idd, year):
    # for i in idd:
    f=open(file,'r')
    info=f.readlines()[1]

    #date
    YYYY=year
    # y=info[1:3]
    # if y>='93':
    #     YYYY='19'+y
    # elif y==' ':
    #     YYYY='2000'
    # else:
    #     YYYY='200'+y[1]

    MM=info[3:5]
    if MM[0]==' ':
        MM='0'+MM[1]
    DD=info[5:7]

    #time
    hh=int(info[8:10])
    mm=int(info[10:13])
    # print(hh)
    # print(mm)
    hh=str(hh).rjust(2,'0')
    mm=str(mm).rjust(2,'0')

    #segundos
    num=int(info[13:15])
    num=str(num).rjust(2,'0')
    den=int(info[16:18])
    den=str(den).rjust(2,'0')
    sec=num+'.'+den
    # print(sec)
    #latitud
    lat_deg=int(info[19:21])
    lat_sec=float(info[22:27])
    lat=lat_deg+(lat_sec/60)
    nlat=str(lat).ljust(8,'0')[0:8]

    #longitud
    lon_deg=int(info[28:31])
```

```

lon_sec=float(info[32:37])
lon=(lon_deg+(lon_sec/60))*-1 #West
nlon=str(lon).ljust(10,'0')[0:10]

#profundidad
depth=float(info[39:44])
ndepth=str(depth).ljust(6,'0')[0:6]

#magnitud
mag=str(info[47:51])
if mag==' ':
    mag=0.00
nmag=str(mag).ljust(4,'0')[0:4]
name=str(YYYY+'-'+MM+'-'+DD+'-'+hh+'-'+mm+'-'+sec+'.hyp')
name=str(" "+str(idd)+" "+str(YYYY)+" "+str(MM)+" "+str(DD)+"
"+str(hh)
        +" "+str(mm)+" "+str(sec)+" "+str(nlat)+" "+str(nlon)+"
"+str(ndepth)
        +" "+str(nmag)+"\n")

return name

def time_check(d,ho,mi,se):
    if se >= 60.0: #If seconds is bigger than 60
        se=se-60.0
        mi=mi+1
    if mi >= 60.0:
        mi=mi-60
        ho=ho+1

    if ho >= 24:
        ho=ho-24
        d=d+1
    se = str(se).ljust(5,'0')
    return se, mi, ho, d

def total_seconds(self):
    """Total seconds in the duration."""
    return ((self.days * 86400 + self.seconds) * 10**6 +
            self.microseconds) / 10**6

def phase_p(line, time):
    stcn=str(line[0:5]).strip()
    stcn=stcn.ljust(5)

    dist=str(line[6:11]).strip()
    dist=dist.ljust(6,'0')

    p=str(line[21:22])

    hhmmss=str(str(line[25:27])+'-'+str(line[27:29])+'-'+str(line[30:35]))
    return stcn, p, dist, hhmmss

def phase_s(line):

```

```

stcn=str(line[0:5]).strip()
stcn=stcn.ljust(5)

dist=str(line[6:11]).strip()
dist=dist.ljust(6,'0')

s=str(line[100:101])
# hhmss=str(str(line[25:27])+'-'+str(line[27:29])+'-'
'+str(line[104:109]))
#segundos despues del tiempo de origen
num=int(line[104:106])
num=str(num).rjust(2,'0')
den=int(line[107:109])
den=str(den).rjust(2,'0')
sec=num+'.'+den

return stcn, s, dist, sec

files=glob.glob('hypo71_separado/*.hyp') ##### EDITAR
#####
files=sorted(files)

idd=12091

with open('hypo71_phaselist.in','w') as evlis_file:
    for fil in range(len(files)):
        #Cabecera de cada evento

        year=int(files[fil][-26:-22]) #problemas con 19* a 20* así que
mejor del archivo
        name=name_file(files[fil],idd,str(year))

        #origin time
        if int(name[31:33])<10:
            ot=dt.datetime(year, int(name[16:18]), int(name[19:21]),
                int(name[22:24]), int(name[25:27]),
int(name[28:30]),
                int(name[31:33])*10000)
        else:
            ot=dt.datetime(year, int(name[16:18]), int(name[19:21]),
                int(name[22:24]), int(name[25:27]),
int(name[28:30]),
                int(name[31:33])*10000)

        print(name)
        evlis_file.write(name)

        #fases leidas
        f=open(files[fil],'r')
        lines=f.readlines()
        for p in range(4, 100):
            f=open(files[fil],'r')

```

```

line=f.readlines()[p] #Extrac data from here
print(line)
if line.strip()==' ':
    break
if line.strip()=='\n':
    break
if line=='\n':
    break

#Extract p-phase info
if line[39:41]=='**': #Condicionante de cuando no se toma
en cuenta la fase desechada
    pass
else:
    stcn, p, dist, hhmss=phase_p(line,1)

#time check of ORIGIN TIME
se, mi, ho,
d=time_check(int(name[19:21]),int(hhmss[0:2].strip()),
int(hhmss[3:5].strip()),
float(hhmss[6:11].strip()))

sen, sed=str(se).split('.')

#P-wave picks (with posible time correction)
pt=dt.datetime(year,int(name[16:18]),int(d),
int(ho),int(mi),int(sen),
int(sed[0:2])*10000)
ptd=(pt - ot).total_seconds()

phase=str('BC '+stcn+' '+p+'
'+str(ptd).ljust(6,'0')+' '+dist+'\n')
print(phase)
evlis_file.write(phase)

#S-wave picks (with time correction)

if str(line[100:101])=='S':
    stcn, s, dist, sec=phase_s(line)

    if line[121:123]=='**':
        pass
    else:
        #time check
        # se, mi, ho,
d=time_check(int(name[19:21]),int(hhmss[0:2].strip()),
#
int(hhmss[3:5].strip()),
#
float(hhmss[6:11].strip()))
# sen, sed=str(se).split('.')
# st=dt.datetime(year,int(name[16:18]),int(d),

```

```

int(sed[0:2])*10000)          #           int(ho),int(mi),int(sen),
                             # std=(st - ot).total_seconds()

                             phase=str('BC '+stcn+' '+s+'
'+str(sec).ljust(6,'0')+' '+dist+'\n')
                             evlis_file.write(phase)
                             # if std < 0:
                             #     print(std)
                             #     print(name)

idd=idd+1

```

Nórdico

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Aug 29 12:22:01 2021

@author: erikestebanramirezramos
#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

"""
import glob
import numbers
from PyNordicRW import Read_Nordic

archz=glob.glob("acc_sfiles/20*")
archz=sorted(archz)

nid10=22137
with open("nordic_phaselist.in","w") as file:
    for i in range(len(archz)):
        print(archz[i])
        nordicfile=Read_Nordic(archz[i])
        ##### First element of file
        for i in nordicfile:
            id=str(i)
            ide=nordicfile[id]

        ##### First elements of Header #####
        header=ide["HEADER"]
        # print(header)

        ##### Date #####
        date=header["L1"]
        year=date["Year"]
        month=date["Month"]

```

```

day=date["Day"]
hour=date["Hour"]
minute=date["Min"]
second=date["Sec"]

##### Location #####
lat=date["Lat"]
lon=date["Lon"]
depth=date["Dep"]
mag=date["Mag1"]

##### Re-order all the numbers to adapt to fortran-like format used by ssst
# id10=int(str(int(float(id)*10))[-10:]) #Last 10 digits of id to fit in fortran format
# nid10=str(id10).zfill(10)
nmonth=str(month).zfill(2)
nday=str(day).zfill(2)
nhour=str(hour).zfill(2)
nminute=str(minute).zfill(2)
nsecond=str(second).ljust(6, '0')

nlat=str(lat).ljust(8, '0')
nlon=str(lon).ljust(10, '0')
ndepth=str(depth).ljust(6, '0')
nmag=str(mag).ljust(4, '0')

nid10=nid10+1
file.write("      "+str(nid10)+" "+str(year)+" "+str(nmonth)+"
"+str(nday)+" "+str(nhour)
          +" "+str(nminute)+" "+str(nsecond)+" "+str(nlat)+"
"+str(nlon)+" "+
          str(ndepth)+" "+str(nmag)+"\n")

##### Phases #####
phase=ide["PHASE"]
# print(phase)
stcns=[]
for s in phase: #List of stations used in location of Resnom
event
    stcns.append(s)

# phases=[]
for f in stcns:
    phases=phase[f] #P and S dictionary
    for a in phases: #P and S loop
        arrival=phases[a]
        for ns in arrival:
            content=arrival[ns]
            STA=content["STA"] #Station code
            INST=content["INST"] #Type of instrument (H: Broad
band, S: Short period)

```

```

COMP=content["COMP"] #Component
PH=content["PH"] #Phase pick
HOUR=content["Hour"] #Hour of pick
MIN=content["Min"] #Minute of pick
SEC=content["Sec"] #Sec of pick
DIS=content["DIS"] #Distance
if isinstance(DIS, numbers.Integral):
    DIS=float(DIS)

NSTA=str(STA).ljust(5," ")
NSEC=str(SEC).ljust(6,'0')

if DIS == 0:
    print(DIS+'erik')
else:
    NDIS=str(DIS).ljust(6,'0')
    file.write("BC "+str(NSTA)+" "+str(PH)+"
"+str(NSEC)+
                " "+str(NDIS)+"\n")

```

C. Stationlist

```

#MODIFICADO POR KARLA FLORES
#MODIFICADO POR PAULA SALAZAR

import glob
import numpy as np

files=glob.glob('files/*.hyp')
files=sorted(files)

stations=[]
with open("resnom_station_list.dat","w") as final:
    for file in files:
        for file in files:
            print(file)
            stcns=np.loadtxt(file, usecols=(1,), dtype=str) #Station list
            for number in range(len(stcns)):
                f=open(file)
                info=f.readlines()[number]
                j=info.split()[1]
                print(j)
                if j not in stations:
                    stations.append(j)
                    final.write(info)

```

*Este script se adaptó para los distintos formatos