**Proyecto: GeoSismic3D: Modelado e impresión 3D del subsuelo para aplicaciones petroleras**

**Manual de Usuario**

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.

**Autor:**

**Juan Esteban Diaz Delgado Estudiante de Ingeniería Electrónica Universidad Surcolombiana**

**2026**

Contenido

[Introducción 3](#_Toc221226785)

[Softwares utilizados 3](#_Toc221226786)

[Librerías necesarias 3](#_Toc221226787)

[Adquisición de Datos 3](#_Toc221226788)

[Código Python 4](#_Toc221226789)

[Fase 1 4](#_Toc221226790)

[Explicación del código 4](#_Toc221226791)

[Fase 2 6](#_Toc221226792)

[Explicación del código 6](#_Toc221226793)

[Fase 3 7](#_Toc221226794)

[Explicación del código 8](#_Toc221226795)

[Fase 4 10](#_Toc221226796)

[Explicación del código 10](#_Toc221226797)

[Fase 5 11](#_Toc221226798)

[Explicación del código 11](#_Toc221226799)

[Fase 6 11](#_Toc221226800)

[Visualización de datos por Blender 12](#_Toc221226801)

# Introducción

El presente documento fue realizado con el objetivo de demostrar de forma práctica y teórica una aplicación de la exploración de hidrocarburos con base al estudio de subsuelos las cuales permiten identificar estructuras favorables para la formación y extracción de combustibles fósiles como el petróleo y el gas, mediante datos sísmicos los cuales los cuales se buscar representar mediante modelos bidimensionales y tridimensionales con el fin practico de imprimir esta información digital en modelos físicos 3D o 2D facilitando la comprensión espacial de las estructuras geológicas.

# Softwares utilizados

**Visual Studio Code (Python):** Se utilizo como herramienta para el procesamiento de datos de tipo .segy para el filtrado de ruido, interpolación, extracción de capas (horizontales), conversión de amplitud/profundidad y generación de mallas.

**Blender:** Se utilizopara visualizarlos modelos 3D debido a su uso de software libre y de código abierto.

# Librerías necesarias

* Segyio
* Numpy
* Matplotlib
* numpy-stl

# Adquisición de Datos

Se procedió a descargar los datos por medio de **SEG Open Seismic Datasets** en el cual se encuentra un gran repertorio de datos sísmicos de pozos petroleros para uso académico, se procedió a descargar el paquete **Dutch F3 seismic dataset** el cual tiene un peso aproximado de 1GB, el F3 es un bloque sísmico real de la offshore del Mar del Norte (North Sea) en los Países Bajos. El cual posee un conjunto de datos 3D de reflexión sísmica.

Se descargo un archivo tipo zip el cual al descomprimirse tiene un archivo .segy los cuales son un estándar de datos sísmicos muy usados en geofísica y exploración petrolera.

Un archivo SEGY guarda:

* Encabezado textual: información general del proyecto.
* Encabezado binario: parámetros técnicos (frecuencia de muestreo, formato de datos, etc.).
* Trazas sísmicas: cada traza es una señal registrada por un sensor (geófono), mostrando cómo viaja

# Código Python

## Fase 1

Se procede a crear un archivo llamado segy.py con el fin de abrir el archivo .segy

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**Imagen N°1.** Código para abrir el archivo .segy

### Explicación del código

import segyio

Se utiliza esta librería para leer y escribir archivos sísmicos SEG-Y el cual es un formato estándar para almacenar datos sísmicos en 2D y 3D.

import numpy as np

Se utiliza esta librería para trabajar con arreglos numéricos debido a que los datos sísmicos son matrices gigantes de números o amplitudes.

import matplotlib.pyplot as plt

Se utiliza esta librería para visualizar los datos.

archivo = "nombre y ubicación del archivo" # se cambia el nombre si es diferente

es la variable que guarda la ruta del archivo.

with segyio.open(archivo, "r", ignore\_geometry=True) as f:

Esta línea abre y cierra el archivo correctamente en modo de lectura e ignora todas a geometría (coordenadas X,Y reales) debido a que solo se quiere exportar los datos aunque ahora no es necesario.

print("Número de trazas:", f.tracecount)

Esta línea imprime el numero total de trazas sísmicas.

    print("Número de muestras por traza:", f.samples.size)

Imprime el numero de muestras por traza

    traza = f.trace[0]

Lee la primera traza

plt.plot(traza)

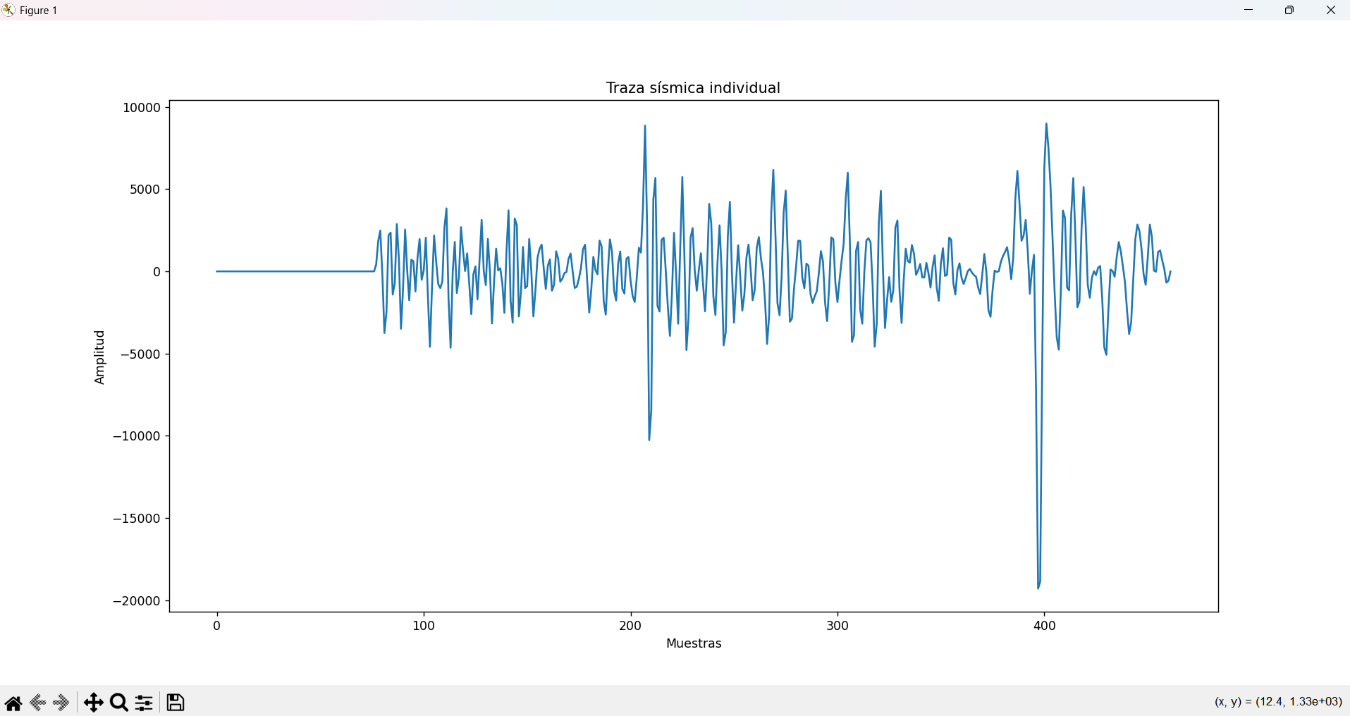
plt.title("Traza sísmica individual")

plt.xlabel("Muestras")

plt.ylabel("Amplitud")

plt.show()

Grafica la traza



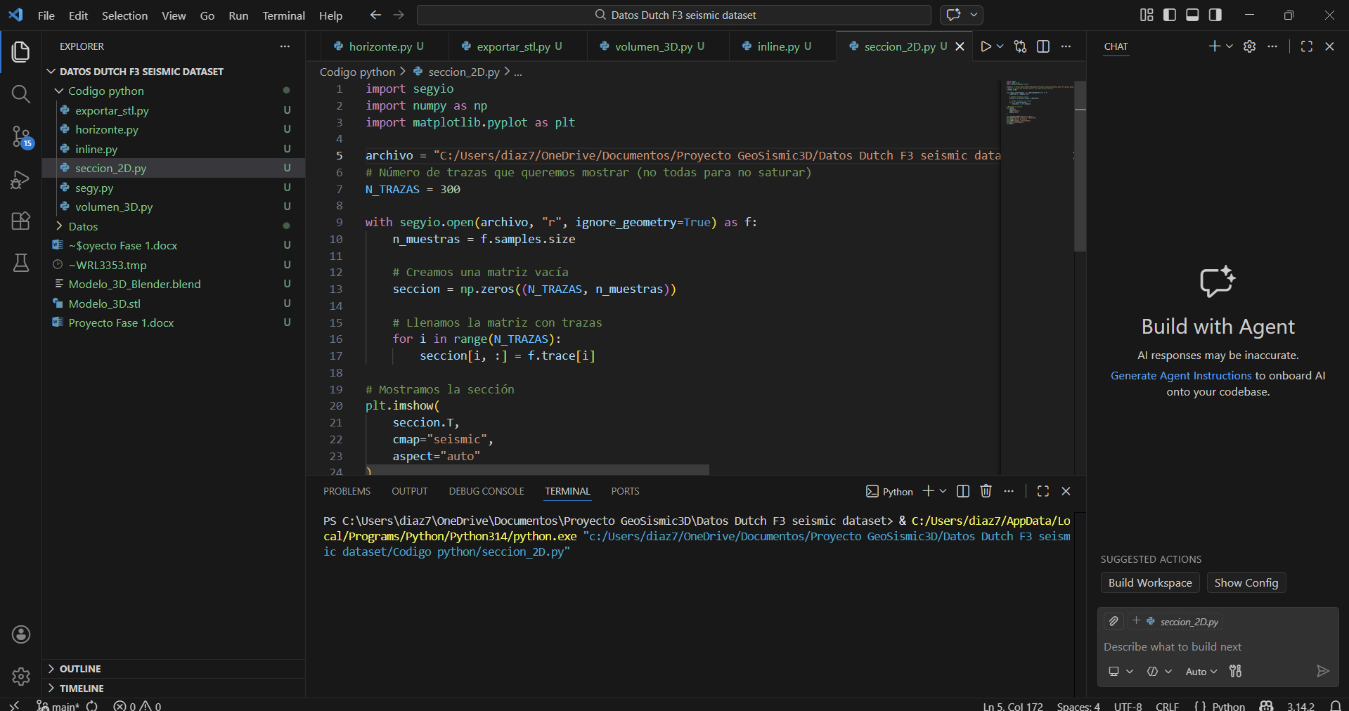
**Imagen N°2**. Grafica de la traza

## Fase 2

Se procede a construir el modelo una sección sísmica en 2D, es como cortar el pastel verticalmente y se procede mirar el trozo de lado.

Cada línea de color puede representar:

* una capa de roca
* un cambio de material
* un límite entre formaciones



+

**Imagen N°3**. Código en Python

### Explicación del código

N\_TRAZAS = 300

Utilizamos una pequeña parte solo 300/ 619101

seccion = np.zeros((N\_TRAZAS, n\_muestras))

se procede a crear una tabla vacía:

* Filas son las trazas
* Columnas es la profundidad

seccion[i, :] = f.trace[i]

Se procede a copiar toda la traza y a colocar la fila correcta

plt.imshow(seccion.T, cmap="seismic")

Se procede a convertir números en colores:

* rojo / azul → amplitud positiva / negativa
* continuo → capas
* rupturas → fallas

invert\_yaxis()

Es el estándar.

* arriba = superficie
* abajo = profundidad

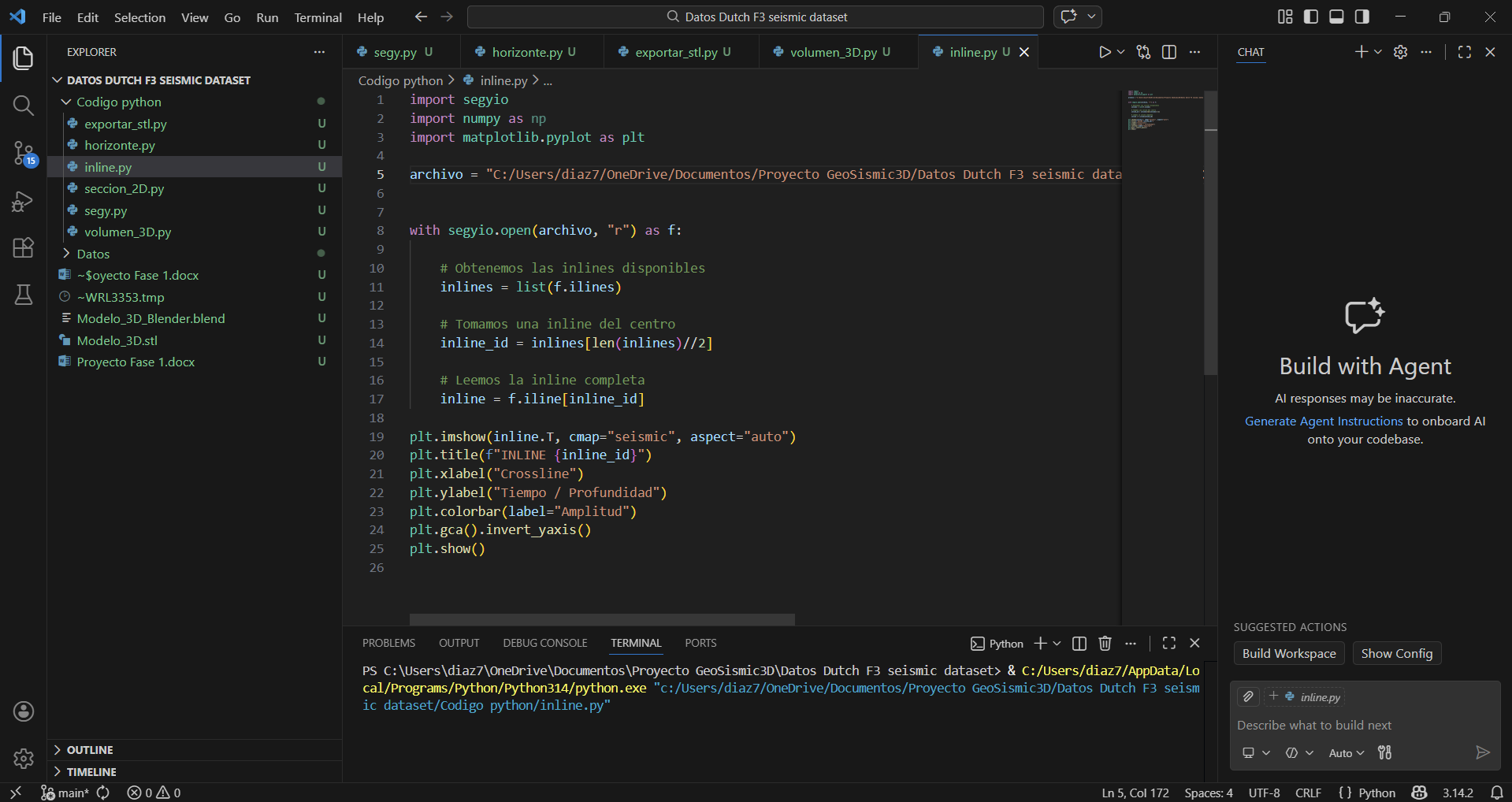
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Imagen N°4**. Grafica de la traza

## Fase 3

Se procede a realizar un corte 2D en una dirección del terreno por ejemplo norte-sur.



**Imagen N°5**. Código Python

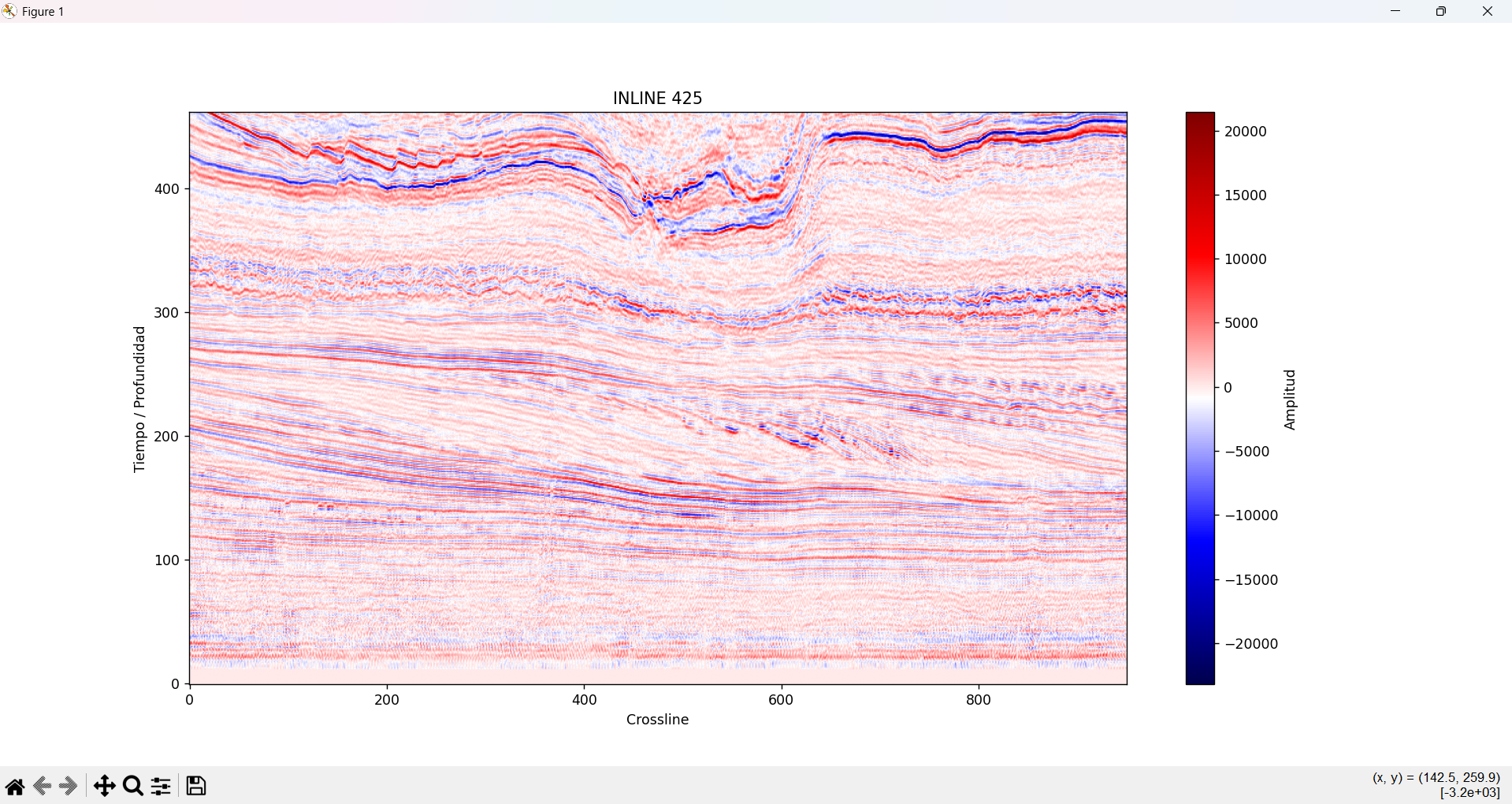
### Explicación del código

f.ilines

Se obtienen las líneas inline del volumen

inline = f.iline[inline\_id]

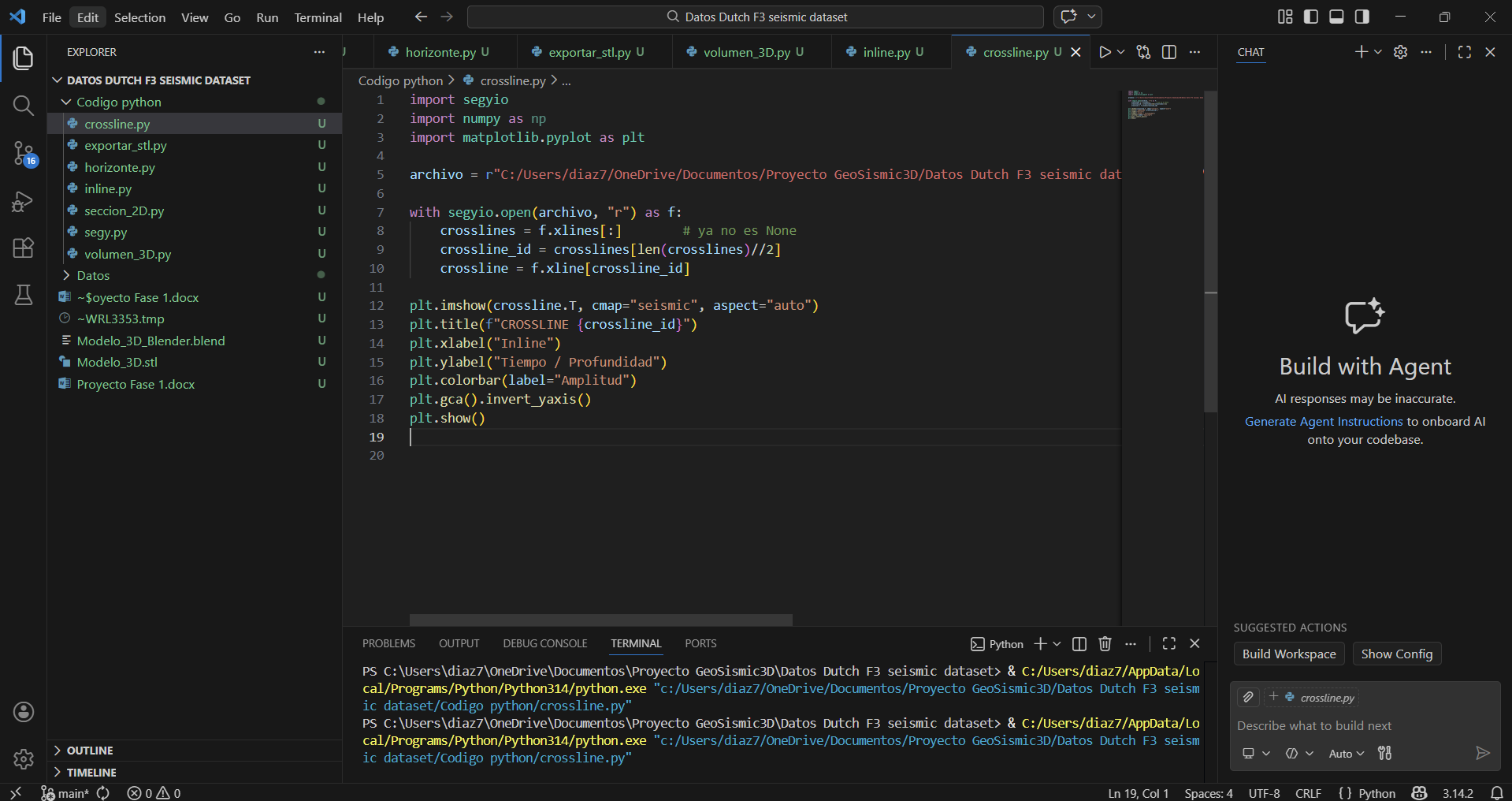
Se procede a extraer una rebanada del volumen en una dirección fija.

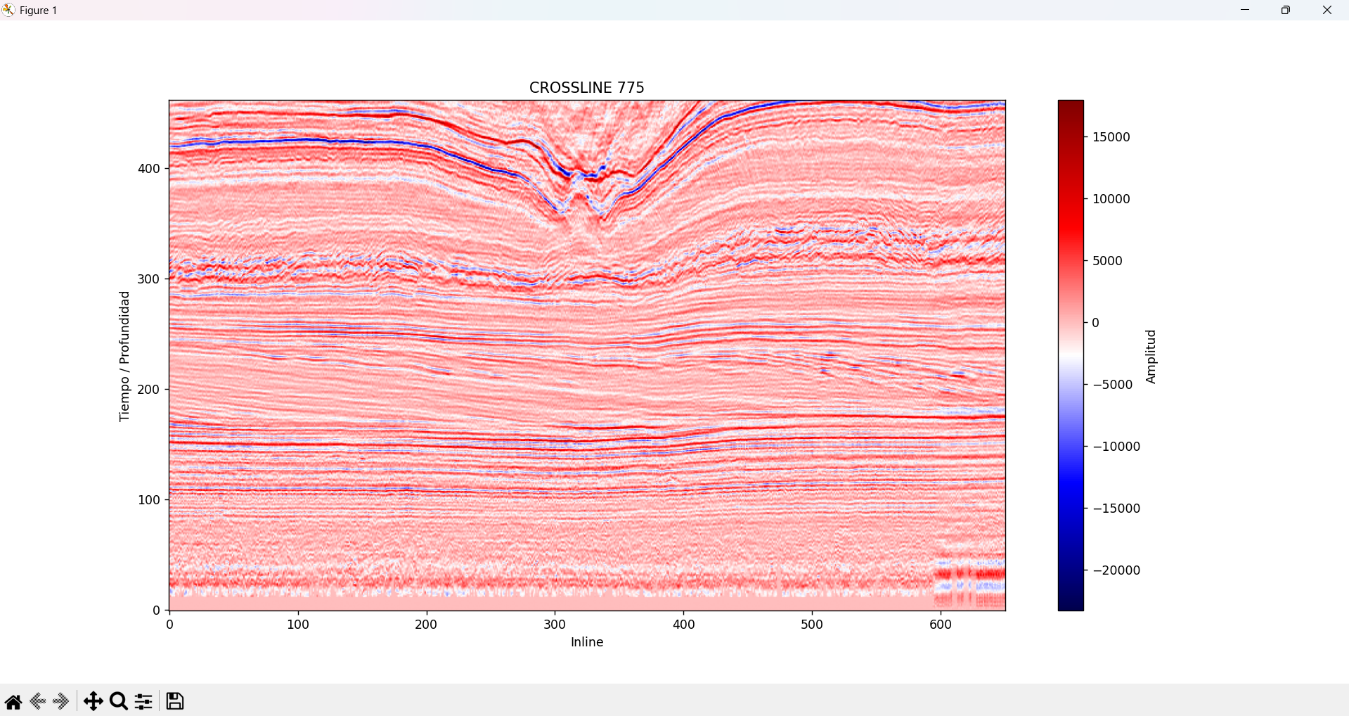


**Imagen N°6**. Grafica Inline

**Opcional CROSSLINE**

Un CROSSLINE es un corte 2D perpendicular a la inline.



**Imagen N°7**. Código Python

**Imagen N°8**. Imagen CROSSLINE

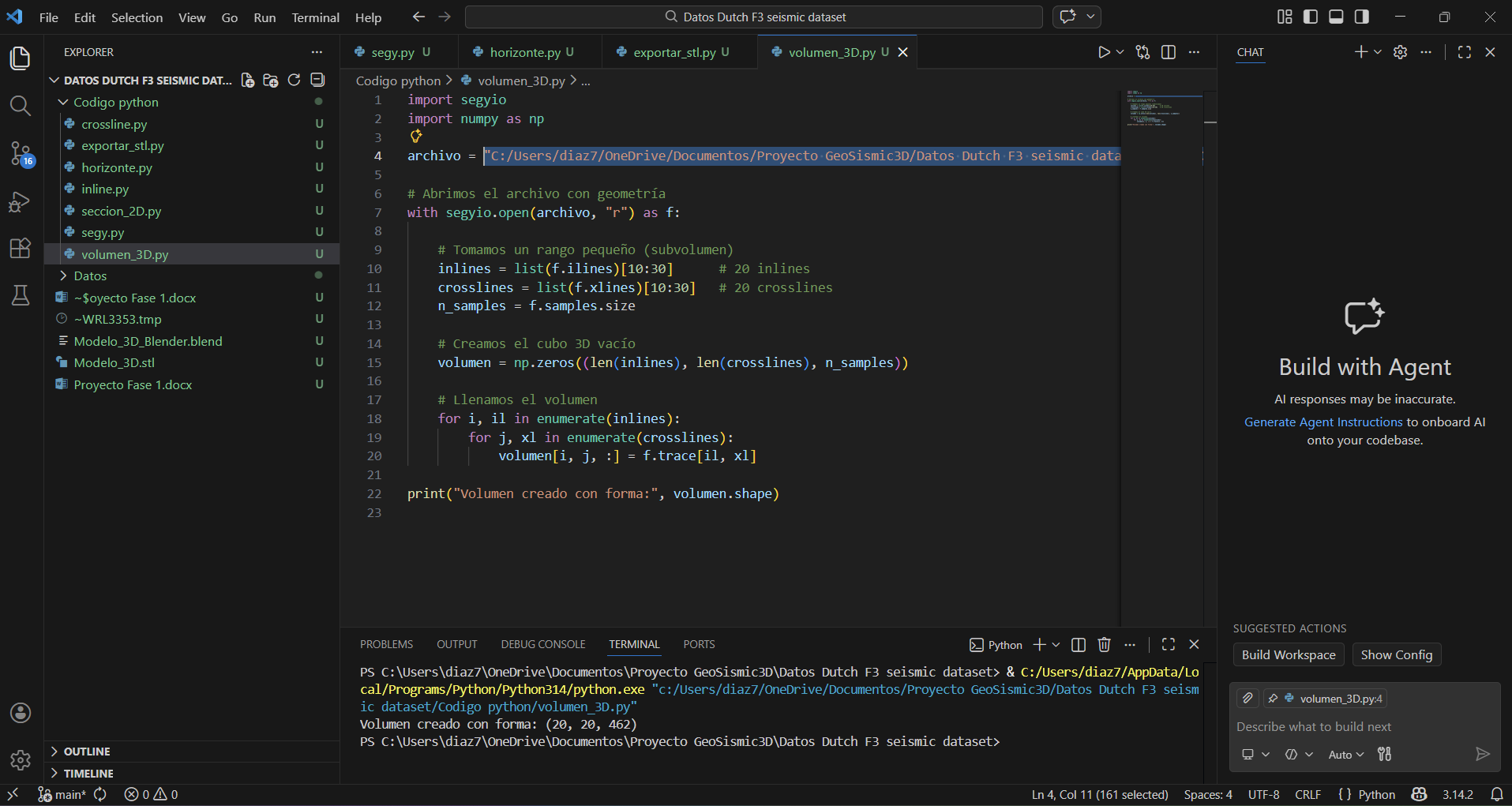
## Fase 4

Se procede a construir el volumen solido en 3D

* Una **inline** es una hoja
* Una **crossline** es otra hoja perpendicular
* Muchas hojas apiladas forman un **bloque**

Matemáticamente así seria el volumen

volumen[inline, crossline, profundidad]



**Imagen N°9**. Código Python

### Explicación del código

inlines = list(f.ilines)[10:30]

Tomamos solo:

* 20 líneas del centro del levantamiento

crosslines = list(f.xlines)[10:30]

20 cortes perpendiculares

volumen = np.zeros((20, 20, profundidad))

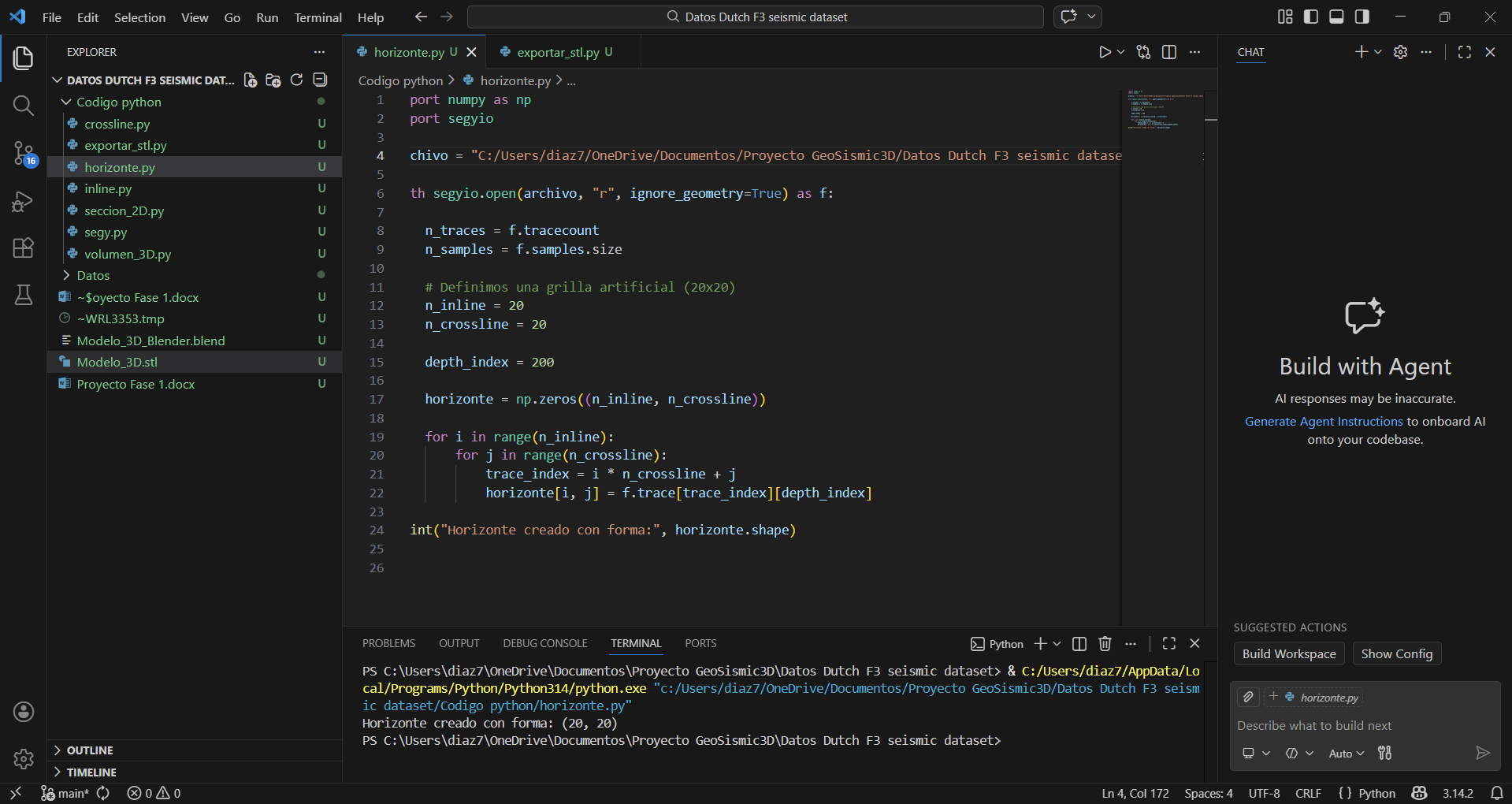
se procede a crear un cubo con profundidad real.

f.trace[il, xl]

se procede a navegar el subsuelo en 3D

## Fase 5

Se procede a extraer una rebanada Horizonte



**Imagen N°10**. Código Python

### Explicación del código

ignore\_geometry=True

Se procede a olvidar los mapas para que solo de los datos crudos

trace\_index = i \* n\_crossline + j

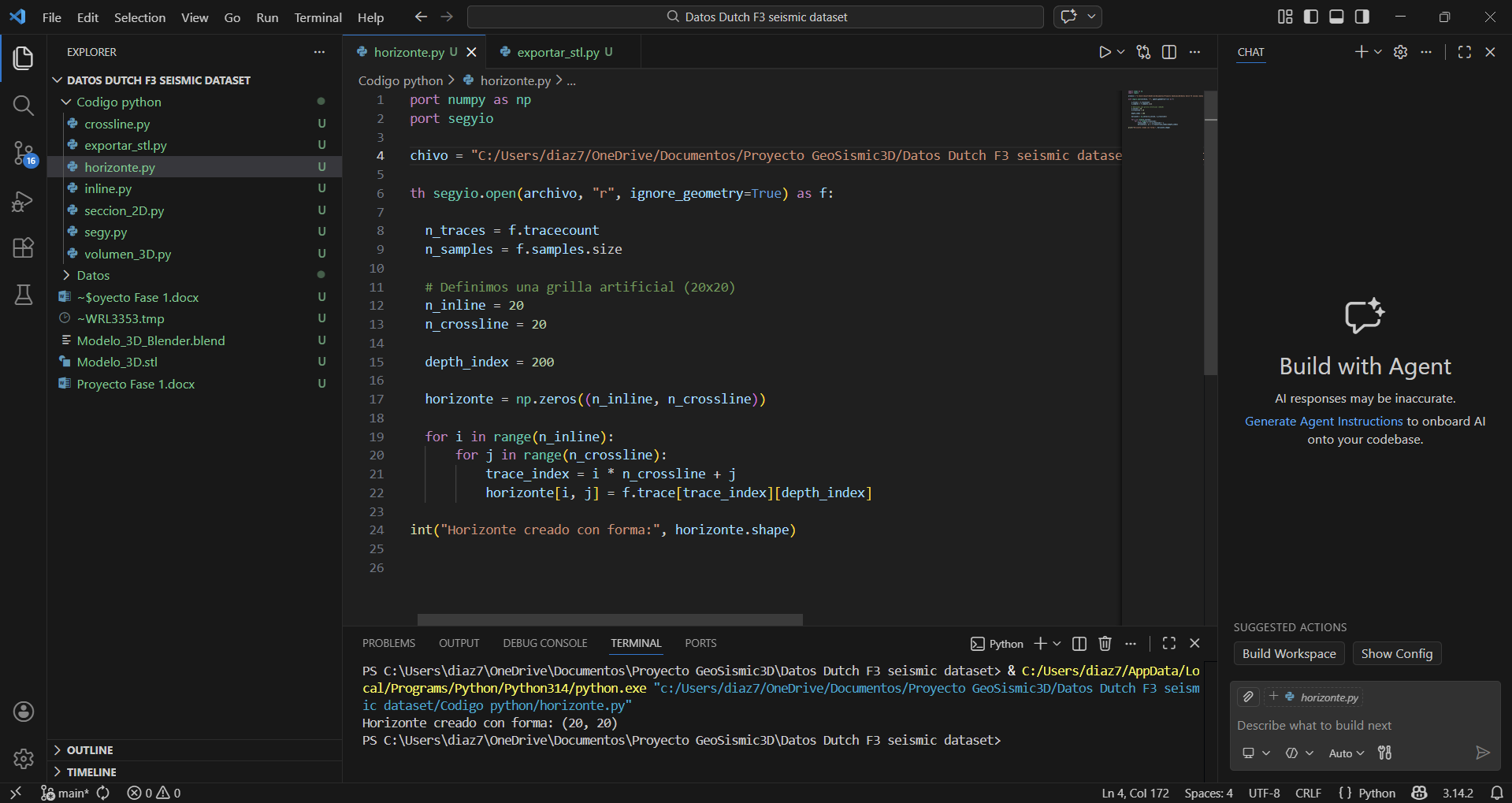
Se procede a recorrer las trazas en orden simulando una malla regular.

depth\_index = 200

Se procede a escoger una profundidad fija por lo tanto se convierte en un horizonte sísmico.

## Fase 6

Se procede a convertir el horizonte en una malla 3D en STL



# Visualización de datos por Blender

Una ves que se obtenga el archivo .stl se procede a utilizar blender para visualizar el modelo 3D.

