

Seminario de investigación acerca de los Conceptos básicos de Python y primer acercamiento al uso de GPRPy para el procesamiento, visualización e interpretación de datos de GPR.

Autores

Jesús Oswaldo Arias Gamarra, Dairon Arley Galeano Bustos, Juan David Ochoa y

Fabian Leonardo Suarez Ortiz

Trabajo de grado para optar al título de Geólogo

Director:

Rocío del Pilar Bernal Olaya

Geóloga PhD

Codirector

Angi Viviana Aparicio Guevara

Geóloga

Universidad Industrial de Santander

Facultad de Ingenierías Fisicoquímicas

Escuela de Geología

Bucaramanga

2021

### **Agradecimientos**

“A Dios, por haber sido mi guía y fortaleza en todos los momentos de mi vida.

A mi familia, por haberme brindado su amor, paciencia y comprensión.

A la profesora Rocio Bernal, por habernos brindado su tiempo, acompañamiento y apoyo incondicional durante este proceso.

Al PhD. Jheyston Omar Serrano y al Geo. Álvaro Manrique por haber compartido los datos para la realización del proyecto”

Jesus Arias

“ Primero a Dios porque me dio el regalo más grande del mundo que es mi familia y me permitió poder cumplir este sueño.

Segundo a mi mamá Andrea Bustos y a mi papá Wilmar Galeano porque sin su apoyo no hubiera conseguido ser la persona que soy y obtener este título, y a mis hermanos por su constante compañía.

Tercero a la profesora Rocío Bernal por su gran ayuda, disposición y compromiso como directora para el trabajo realizado y su trato como una amiga que estuvo siempre en las situaciones más difíciles.

Y por último y no menos importante a mis amigos que en momentos duros durante esta etapa de estudiante con una palabra de aliento o con diferentes acciones me motivaron a continuar.”

Dairon Galeano

“A mis padres, quiénes con su esfuerzo y dedicación , lograron brindarme los recursos para mi estudio, y hoy ven su fruto, dar cosechas.

Juan Ochoa

“Gracias a 143bikes, a parchaos freir y azar, a vestimenti, a mi familia, a los profes y a los compañeros, a Dios, y a la universidad.”

Fabian Suarez

## **Introducción**

El presente trabajo se realizó bajo la modalidad de seminario de investigación o seminario alemán, durante el primer y segundo semestre del 2020. A través de la discusión durante 13 sesiones, se aprendieron conceptos básicos del lenguaje de programación Python, aspectos generales del funcionamiento del método geofísico GPR, el funcionamiento del módulo GPRpy y metodologías de interpretación geológica en la escala de observación de la información de GPR.

El presente informe contiene un valioso aporte para la Escuela de Geología, a través de un resumen de las discusiones sostenidas, y los resultados del procesamiento e interpretación de datos de GPR. Adquiridos por estudiantes de la maestría de Geofísica y del pregrado de Geología de la Universidad Industrial de Santander, durante el 2018, en dos sectores con contextos geológicos contrastantes: 1.) en el municipio de Mogotes, Santander, cuyo casco urbano reposa sobre un abanico aluvial, rasgo geológico estudiado con GPR exitosamente por (Franke, Hornung, & Hinderer, 2015), de interés como posible reservorio de agua en la región (Sanchez Pinto, 2020) y 2.) en Bayunca, Cartagena en la zona aledaña a un volcán de lodo de 150 metros de diámetro.

Las referencias consultadas para desarrollar el seminario incluyen el libro How to Think Like a Computer Scientist: Learning with Python 3 (Wentworth P. et al, 2020), y diversos artículos de investigación con casos de estudio similares a los tratados en este seminario, toda esta bibliografía se encuentra especificada al final de este documento. Este material bibliográfico fue discutido a manera de presentaciones lideradas por un relator, resumidas en actas que aparecen como material anexo a esta entrega.

## Tabla de Contenido

Introducción .....	3
1 Fundamentos de Python .....	18
1.1 La forma del programa.....	18
1.1.1 Reglas para establecer una variable .....	20
1.1.2 Orden jerárquico en las operaciones aritméticas .....	21
1.2 Flujo del programa.....	23
1.2.1 Iterador for .....	24
1.3 Funciones .....	26
1.3.1 Funciones que requieren argumentos.....	28
1.4 Tipos de datos .....	28
1.4.1 Cadenas (Strings) .....	29
1.4.2 Tuplas.....	31
1.4.3 Listas .....	32
1.4.4 Diccionarios .....	36
1.5 Numpy.....	37
1.5.1 Funciones contenidas en Numpy .....	38
1.6 Archivos .....	39
1.6.1 Directorios.....	40

1.6.2	Módulos matemáticos .....	41
1.7	GPRPy.....	42
1.7.1	La interfaz de usuario (GUI) y la funcionalidad del módulo.....	43
1.7.2	Herramientas del GPRPy .....	44
2	Fundamentos del GPR (Radar de Penetración Terrestre) .....	50
2.1	Fundamento físico del GPR .....	51
2.1.1	Velocidad de onda.....	51
2.1.2	Reflexión y transmisión de ondas de radio .....	53
2.1.3	Refracción de ondas de radio .....	54
2.1.4	Refracción crítica .....	55
2.1.5	Polarización eléctrica .....	59
2.1.6	Permitividad dieléctrica ( $\epsilon$ ).....	60
2.2	Propiedades dieléctricas de minerales formadores de rocas .....	60
2.3	Adquisición GPR .....	62
2.3.1	Tipos de arreglos en sondeos GPR .....	63
2.4	Procesamiento GPR .....	65
2.4.1	Corrección de tiempo zero .....	65
2.4.2	Análisis de velocidad .....	65
2.4.3	Filtros .....	67

2.4.4	Herramientas de Ganancia .....	69
2.4.5	Migración .....	70
2.4.6	Correcciones de elevación o topográficas.....	71
3	Casos de Estudio Desarrollados en el Seminario.....	72
3.1	Caso de Estudio 1: Mogotes Santander .....	74
3.1.1	Localización de la línea de estudio .....	74
3.1.2	Resumen Geológico .....	76
3.1.3	Geología local .....	76
3.2	Caso de Estudio 2: Bayunca, Bolívar .....	79
3.2.1	Localización de la línea de estudio .....	79
3.2.2	Resumen geológico .....	81
3.2.3	Geología local .....	83
4	Procesamiento e Interpretación de Datos De GPR .....	84
4.1	Caso de Estudio 1: Abanico aluvial de Mogotes, Santander .....	84
4.1.1	Procesamiento .....	86
4.1.2	Identificación y definición de Radarfacies .....	93
4.1.3	Interpretación .....	98
4.2	Caso de Estudio 2: Volcán de lodo en Bayunca, Cartagena .....	98
4.2.1	Procesamiento .....	100

4.2.2	Identificación y definición de radarfacies .....	104
4.2.3	Interpretación .....	105
5	Conclusiones y Recomendaciones .....	106
	Referencias Bibliográficas .....	108

### Lista de Tablas

Tabla 1. Palabras clave en código Python. ....	21
Tabla 2. Orden jerárquico de las operaciones aritméticas. ....	22
Tabla 3. Propiedades electromagnéticas de diferentes materiales. ....	53
Tabla 4. Propiedades eléctricas (Glover, 2015). ....	56
Tabla 5. Valores de permeabilidad magnética ( $\mu$ ) para algunos minerales (Zhdanov M. , 2018).58	
Tabla 6. Permitividades dieléctricas de distintos materiales (Zhdanov M. , 2018). ....	61



### Lista de Figuras

Figura 1. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	19
Figura 2. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	19
Figura 3. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	20
Figura 4. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	21
Figura 5. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	22
Figura 6. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	23
Figura 7. Interfaz Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	24
Figura 8. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	24
Figura 9. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	25
Figura 10. Representación espiral, interfaz gráfica, How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	25

Figura 11. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	26
Figura 12. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	27
Figura 13. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	27
Figura 14. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	28
Figura 15. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	28
Figura 16. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	29
Figura 17. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	29
Figura 18. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	30
Figura 19. Código de Python en Visual Studio Code.	30
Figura 20. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	31
Figura 21. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	31

Figura 22. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	31
Figura 23. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	31
Figura 24. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	32
Figura 25. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	32
Figura 26. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	32
Figura 27. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	33
Figura 28. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	33
Figura 29. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	34
Figura 30. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	34
Figura 31, Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	34
Figura 32. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	35

Figura 33. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	35
Figura 34. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	36
Figura 35. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	36
Figura 36. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	37
Figura 37. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	37
Figura 38. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	37
Figura 39. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	38
Figura 40. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	38
Figura 41. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	38
Figura 42. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	39
Figura 43. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	39

Figura 44. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	40
Figura 45. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	40
Figura 46. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	41
Figura 47. Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).	41
Figura 48. Código de Python en Visual Code Studio.	42
Figura 49. Interfaz GPRPy.	44
Figura 50. Perfil antes y después de aplicar la herramienta adj profile.	45
Figura 51. Perfil GPR antes y después de aplicar el filtro Dewow.	46
Figura 52. Perfil GPR antes y después de aplicar el filtro Smooth.	47
Figura 53. Perfil GPR aplicando la ganancia AGC.	48
Figura 54. Perfil GPR aplicando la corrección de topografía.	49
Figura 55. Ley de Snell, tomada de <a href="http://www.britannica.com/science/light/Reflection-and-refraction">www.britannica.com/science/light/Reflection-and-refraction</a> .	54
Figura 56. Ángulo crítico, tomado de <a href="http://gpg.geosci.xyz/content/GPR/GPR_fundamental_principles.html">gpg.geosci.xyz/content/GPR/GPR_fundamental_principles.html</a> .	55
Figura 57. Polarización eléctrica (Fano, 2019).	60
Figura 58. Alcance de adquisición de las frecuencias típicas usadas en el GPR dependiendo el tipo de material. (P., Annan, s.f.).	62

Figura 59. Aplicaciones GPR y frecuencias adecuadas (P., Annan, s.f.).	63
Figura 60. Arreglo zero offset, recuperada de <a href="https://gpg.geosci.xyz/content/GPR/index.html">https://gpg.geosci.xyz/content/GPR/index.html</a> .	64
Figura 61. Arreglo CMP, recuperada de <a href="https://gpg.geosci.xyz/content/GPR/index.html">https://gpg.geosci.xyz/content/GPR/index.html</a> .	64
Figura 62. Ejemplo de cómo determinar la velocidad a partir de la función hiperbólica (Cassidy & Jol, 2009).	67
Figura 63. Ganancia AGC para un perfil GPR (Cassidy & Jol, 2009).	70
Figura 64. Ejemplo de corrección por elevación topográfica (Cassidy & Jol, 2009).	72
Figura 65. Equipo de adquisición GPR utilizado, recuperada de <a href="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSmsr6_JeFyiYHcfH2XkDpyCFr352MWIupNw&amp;usqp=CAU">https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSmsr6_JeFyiYHcfH2XkDpyCFr352MWIupNw&amp;usqp=CAU</a> .	73
Figura 66. Ubicación de la línea número 1, Google Earth.	75
Figura 67. Línea número 1, imagen radargrama, GPRPy.	76
Figura 68. 1) Mapa topográfico de Colombia ilustrado. 2) Mapa geológico de la plancha 136.	78
Figura 69. Ubicación de la línea número 2, Google Earth.	80
Figura 70. Diagrama de flujo usado para el procesamiento de datos GPR.	85
Figura 71. Línea GPR sin procesamiento.	86
Figura 72. Línea GPR con la corrección de tiempo cero.	87
Figura 73. Línea GPR con la aplicación del filtro Dewow.	88
Figura 74. Línea GPR con la aplicación del filtro smooth y la herramienta AGC.	89
Figura 75. Línea GPR con la hipérbola seleccionada para determinar la velocidad.	89
Figura 76. Línea GPR con el cambio de unidades de profundidad.	90
Figura 77. Foto del afloramiento ilustrando las litofacies interpretadas.	91

Figura 78. Fotografía de afloramiento. ....	91
Figura 79. Línea GPR donde se evidencia la superficie de erosión.....	93
Figura 80. Radarfacies definidas por los intérpretes.....	94
Figura 81. Radar facie 1.....	94
Figura 82. Radar facie 2.....	95
Figura 83. Radar facie 3.....	95
Figura 84. Radar facie 4.....	96
Figura 85. Radar facie 5.....	97
Figura 86. Radar facie 6.....	98
Figura 87. Diagrama de flujo para el procesamiento de los datos GPR. ....	99
Figura 88. Corrección tiempo cero. ....	100
Figura 89. Línea de GPR con la aplicación del filtro Dewow. ....	101
Figura 90. Línea GPR con el filtro smooth.....	102
Figura 91. Análisis de velocidad mediante hipérbolas. (0.08 ns/m).....	103
Figura 92. Línea GPR con el cambio del dominio de profundidad. ....	103
Figura 93. Perfil trabajado con las radar facies definidas por los intérpretes. ....	104

## Resumen

**Título:** Informe Final del Seminario de investigación acerca de los Conceptos básicos de Python y primer acercamiento al uso de GPRPy para el procesamiento, visualización e interpretación de datos de GPR.

**Autores:** Fabián Leonardo Suarez Ortiz, Juan David Ochoa Barajas, Jesús Oswaldo Arias Gamarra, Dairon Arley Galeano Bustos \*\*

**Palabras clave:** GPR, Python, GPRPy, volcanes de lodo, radarfacie, litofacie.

### Descripción:

El trabajo realizado se hizo por medio de la modalidad de seminario de investigación cuyo objetivo fue el de establecer un conocimiento básico sobre el radar de penetración terrestre (GPR) y Python, para incursionar en el procesamiento e interpretación geológica de datos preexistentes adquiridos en Mogotes, Santander y Bayunca, Cartagena con la herramienta GSSI de 400MHz disponible en la Universidad.

En este informe se presenta un resumen de 14 sesiones de discusión de bibliografía sobre los conceptos básicos de GPR, Python y GPRPy, además del resultado del tratamiento de las imágenes disponibles GPR aplicando el conocimiento adquirido. Para realizar este tratamiento se propone un flujo de procesamiento de las imágenes que incluye los filtros Dewow y Smooth junto con las herramientas Seth Zero Time y AGC; y una metodología de interpretación que parte de la identificación de distintos grupos de radarfacies y de litofacies. Los resultados del estudio permitieron generar una interpretación geológica de los datos en la escala de estudio del GPR y demostrar la utilidad de la herramienta, hasta ahora usada para un estudio geológico en la Escuela de Geología.

---

\*Trabajo de Grado

\*\*Facultad de Ingenierías Físicoquímicas, Escuela de Geología UIS. Director: Rocio Del Pilar Bernal Olaya, PhD. Geología.



### **Abstract**

**Title:** Final Report of the Research Seminar on Python's basic concepts and GPRPy's first approach for the processing, visualization, and interpretation of GPR data. \*

**Authors:** Dairon Arley Galeano Bustos, Jesús Oswaldo Arias Gamarra, Juan David Ochoa Barajas, Fabián Leonardo Suarez Ortiz. \*\*

**Key Words:** GPR, Python, mud volcanoes, alluvial fan, radarfacie, lithofacie.

### **Description:**

The project was done through the modality of a research seminar whose objective was to establish a basic knowledge about ground penetrating radar (GPR) and Python, to enter into the geological processing and interpretation of pre-existing data acquired in Mogotes, Santander and Bayunca, Cartagena with the 400MHz GSSI tool which was available at the University.

This report presents a summary of 14 discussion sessions of bibliography on the basic concepts of GPR, Python and GPRPy, in addition to the result of the treatment of the available GPR images applying the knowledge acquired. To carry out this treatment, an image processing flow is proposed that includes the Dewow and Smooth filters together with the Seth Zero Time and AGC tools; and an interpretation methodology that starts from the identification of different groups of radarfacies and lithofacies. The results of the study made it possible to generate a geological interpretation of the data on the GPR study scale and to demonstrate the usefulness of the tool, until now used for a geological study at the School of Geology.

\*Bachelor thesis

\*\*Faculty of Physics & Chemical Engineering. Geology Department. Director: Rocio del Pilar Bernal Olaya, PhD Geologist.

## 1 Fundamentos de Python

Python es un lenguaje de programación creado por el informático holandés Guido van Rossum. Con este lenguaje puede crearse un programa, que es un resumen de secuencias de instrucciones que especifican cómo realizar una operación, ya sea la representación gráfica de una ecuación matemática o resolver un sistema de ecuaciones.

Este se encuentra catalogado como un lenguaje de alto nivel, que se diferencia de sus opuestos (lenguajes de bajo nivel) debido a su mayor complejidad, siendo estos últimos tan básicos que son ejecutados en códigos simples, como por ejemplo secuencias de códigos binarios (“0” y “1”). Casi todos los programas están escritos en lenguajes de alto nivel debido a que son mucho más fáciles de programar, siendo su poca extensión, la facilidad para leerlos y el menor tiempo para escribirlos algunas de sus cualidades. También cabe resaltar la facilidad para reproducir sus programas en diferentes ordenadores sin necesidad de inducir grandes modificaciones. (Wentworth P. et al, 2020).

### 1.1 La forma del programa

Python es un lenguaje de alto nivel, así como C++, PHP y Java., que se usa para crear programas. Un programa se define como una serie de indicaciones para obtener un cómputo, el cual puede ser matemático o de texto. Los programas cuentan con ciertas instrucciones generales para cada tipo de lenguaje, entre las que se encuentran *input* (obtención de datos de una fuente), *output* (mostrar los datos de un archivo), y *math* (gestionar operaciones matemáticas). En la escritura del programa, la tarea de depuración es la que más tiempo toma. Esta se define como el proceso de revisión de las instrucciones, y de identificación de errores en la escritura de los

programas, proceso que se enfoca principalmente en tres tipos diferentes de errores: Errores de sintaxis (estructura y reglas del programa que se deben respetar para ser ejecutados), errores de tiempo de ejecución (ocurren justo después de haber ejecutado el programa) y errores de semántica (errores que se ejecutan correctamente pero no entregan el resultado esperado).

Un valor en el lenguaje de programación se representa como una letra o un número, dependiendo del tipo de programa considerado. Estos valores se clasifican en tipos de datos como por ejemplo enteros (cualquier número entero: 1, 2, 3...), cadenas ("Hello, World) o decimales (cualquier número decimal: 1.1; 1.2; 1.3).

Figura 1.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> type("Hello, World!")
<class 'str'>
>>> type(17)
<class 'int'>
```

En contraste, una variable se refiere a un nombre o código al cual se le asigna un valor determinado para ser utilizado en la ejecución de un programa. Como lo vemos en la figura 2 en la que tenemos 3 variables (*message*, *n* y *pi*) con distintos tipos de valores (cadenas, enteros y decimales).

Figura 2.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> message = "What's up, Doc?"
>>> n = 17
>>> pi = 3.14159
```

### 1.1.1 Reglas para establecer una variable

Las variables pueden ser establecidas en forma de números o letras, pero si comienzan con números tienen que ser anteceditas con un guion bajo (\_), el cual también puede ser usado entre las palabras para servir como espacio en variables que tengan más de una palabra, lo cual sirve, pero no es muy recomendado, porque puede generar confusión al momento de nombrar las variables.

Figura 3.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> message
'What's up, Doc?'
>>> n
17
>>> pi
3.14159
```

Existen ciertas palabras que deben omitirse al momento de establecer o asignar una variable, esto debido a que estas mismas representan cierto tipo de funciones. Ejemplos de estas palabras aparecen en la Tabla 1.

Tabla 1.

Palabras clave en código Python.

<b>and</b>	<b>as</b>	<b>assert</b>	<b>break</b>	<b>class</b>	<b>continue</b>
<b>def</b>	del	elif	else	except	exec
<b>finally</b>	for	from	global	if	import
<b>in</b>	is	lambda	nonlocal	not	or
<b>pass</b>	raise	return	try	while	whit
<b>yield</b>	True	False	None		

La combinación entre variables y funciones da origen a las expresiones, por ejemplo, la combinación de variables (*minutes*, *hours*) junto con enteros para hacer la conversión de minutos a horas que muestra la figura 4.

Figura 4.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> minutes = 645
>>> hours = minutes / 60
>>> hours
10.75
```

### 1.1.2 Orden jerárquico en las operaciones aritméticas

Al igual que en las operaciones matemáticas que realizamos a diario, en el lenguaje de programación existe un orden cuando dos o más operaciones evalúan una expresión de forma simultánea. El orden jerárquico de las operaciones aritméticas se muestra en la Tabla 2.

Tabla 2.

*Orden jerárquico de las operaciones aritméticas.*

Prioridad	Operador	Significado	Ejemplo
1	( )	Paréntesis	$3 * (2 + 4)$
2	**	Exponenciación	$2 ** 3 = 8$
3	*	Multiplicación	$2 * 4 + 5 = 13$
	/	División	$8 / 2 + 3 = 7$
4	+	Suma	$2 + 4 = 6$
	-	Resta	$3 - 1 = 2$

Es necesario mencionar el operador Input, que requiere que el usuario ingrese un valor para dar resultado a la expresión propuesta. Un ejemplo aparece en la figura 5, en la que se requiere ingresar un número que defina un radio determinado para determinar el valor del área de un círculo.

Figura 5.

*Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```

1 response = input("What is your radius? ")
2 r = float(response)
3 area = 3.14159 * r**2
4 print("The area is ", area)

```

## 1.2 Flujo del programa

Esencialmente la forma en la que el lenguaje trabaja consiste en la introducción de una función o variable, la cual puede estar acompañada de varios estamentos que modifiquen ciertos parámetros que se haya ingresado. Todo esto puede desencadenar ya sea en una secuencia de iteraciones que nos pueden ayudar a resolver una incógnita matemática, o dar un resultado tan simple como una gráfica que obedezca dichas instrucciones.

El ejemplo que se muestra usa un programa interno que se maneja en Python llamado *Turtle*. Este módulo se obtiene por medio de la función *import* como se muestra en la figura 6.

Figura 6.

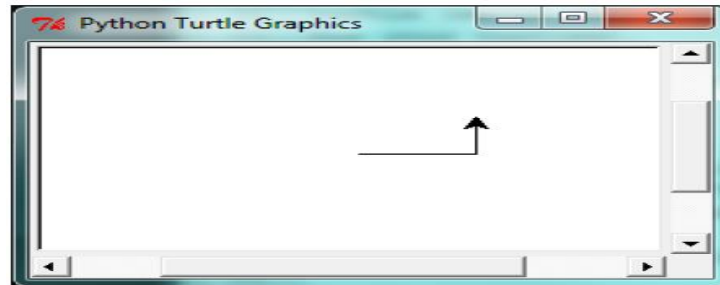
Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1  import turtle                # Allows us to use turtles
2  window = turtle.Screen()     # Creates a playground for turtles
3  alex = turtle.Turtle()       # Create a turtle, assign to alex
4
5  alex.forward(50)             # Tell alex to move forward by 50 units
6  alex.left(90)                # Tell alex to turn by 90 degrees
7  alex.forward(30)             # Complete the second side of a rectangle
8
9  window.mainloop()            # Wait for user to close window
```

La función abrirá una interfaz, figura 7, en la que aparece una flecha cuyo color, tamaño y grosor pueden modificarse; adicionalmente la flecha puede ejecutar una serie de movimientos a partir del uso de comandos, con los que se dibujan polígonos de distintas formas.

Figura 7.

Interfaz Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).



### 1.2.1 Iterador for

El módulo *for* permite iterar una expresión un número determinado de veces, haciendo los códigos en el que se repitan varias veces los mismos comandos más cortos. Un ejemplo de este iterador se observa en la figura 8 en la que se asigna cada valor (en este caso de tipo *string*) de una lista (*friend*) a una variable llamada *invite*.

Figura 8.

Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).

```
1 for friend in ["Joe", "Zoe", "Zuki", "Thandi", "Paris"]:
2     invite = "Hi " + friend + ". Please come to my party!"
3     print(invite)
4 # more code can follow here ...
```

```
Hi Joe. Please come to my party!
Hi Zoe. Please come to my party!
Hi Zuki. Please come to my party!
Hi Thandi. Please come to my party!
Hi Paris. Please come to my party!
```

También se muestra un ejemplo de la iteración en el módulo *Turtle*, en la figura 9, en la que luego de definir modificaciones visuales al ícono donde se creará el dibujo, tales como color tanto



del lienzo como de la figura, al igual que su forma, se usa el módulo iterativo *for* para representar una espiral mostrada en la figura 10.

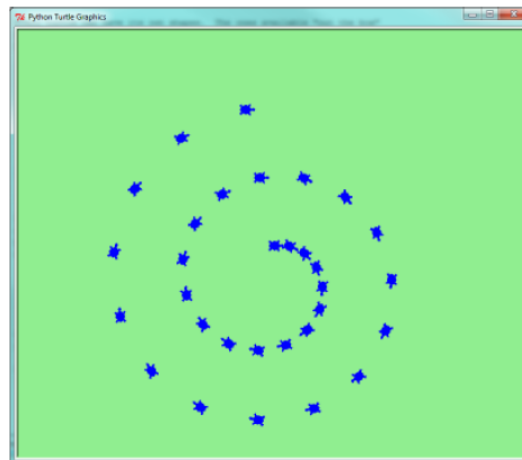
Figura 9.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
1 import turtle
2 window = turtle.Screen()
3 window.bgcolor("lightgreen")
4 tess = turtle.Turtle()
5 tess.shape("turtle")
6 tess.color("blue")
7
8 tess.penup()           # This is new
9 size = 20
10 for _ in range(30):
11     tess.stamp()       # Leave an impression on the canvas
12     size = size + 3    # Increase the size on every iteration
13     tess.forward(size) # Move tess along
14     tess.right(24)     # ... and turn her
15
16 window.mainloop()
```

Figura 10.

Representación espiral, interfaz gráfica, *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).



### 1.3 Funciones

En el lenguaje de Python una función representa uno o un conjunto de códigos que están destinados a desarrollar una tarea. Estas se rigen por una estructura de sintaxis que incluye el nombre (*NAME*), los parámetros que van a afectar (*PARAMETERS*) y los estamentos (*STATEMENTS*) o las instrucciones para que la función se desarrolle, figura 11.

Figura 11.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
def <NAME> ( <PARAMETERS> ) :  
    <STATEMENTS>
```

Un ejemplo de esta estructura se muestra usando el módulo *Turtle*. El nombre de la función es “*draw\_multicolor\_square*” con los parámetros “*animal*” y “*size*”, en los que los estamentos de estos parámetros cambiarán la forma en que se ejecuta. Figura 12.

Figura 12.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```

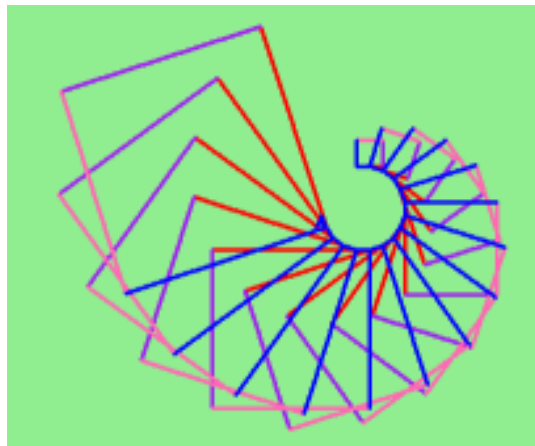
1  import turtle
2
3  def draw_multicolor_square(animal, size):
4      """Make animal draw a multi-color square of given size."""
5      for color in ["red", "purple", "hotpink", "blue"]:
6          animal.color(color)
7          animal.forward(size)
8          animal.left(90)
9
10 window = turtle.Screen()           # Set up the window and its attributes
11 window.bgcolor("lightgreen")
12
13 tess = turtle.Turtle()              # Create tess and set some attributes
14 tess.pensize(3)
15
16 size = 20                           # Size of the smallest square
17 for _ in range(15):
18     draw_multicolor_square(tess, size)
19     size += 10                      # Increase the size for next time
20     tess.forward(10)                # Move tess along a little
21     tess.right(18)                  # and give her some turn
22
23 window.mainloop()

```

La gráfica resultante muestra una serie de cuadros que van disminuyendo de tamaño formando una espiral, figura 13.

Figura 13.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).



### 1.3.1 Funciones que requieren argumentos

Muchas funciones preestablecidas solo requieren que se definan valores para ejecutarlas. Es el caso de la función “abs” que requiere un número para determinar su valor absoluto. Figura 14.

Figura 14.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> abs(5)
5
>>> abs(-5)
5
```

La función exponencial “pow” requiere un número, y un exponente, en este orden, para determinar un resultado. Figura 15.

Figura 15.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> pow(2, 3)
8
>>> pow(7, 4)
2401
```

## 1.4 Tipos de datos

Existen 4 tipos principales de datos dentro de los que se encuentran: *Strings*, tuplas, listas y diccionarios.

### 1.4.1 Cadenas (Strings)

Cada cadena se define como una secuencia de caracteres (números, letras) que representan un texto. Tiene sus propios atributos, minúsculas o mayúsculas, por ejemplo, y métodos que modifican la estructura interna de la cadena, “nombre del atributo.upper ()” que en este caso cambian a letra mayúscula cada letra de la cadena original. Figura 16.

Figura 16.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> our_string = "Hello, World!"
>>> all_caps = our_string.upper()
>>> all_caps
'HELLO, WORLD!'
```

A estas cadenas pueden aplicarse funciones, como “len”, que nos permite obtener el número de caracteres que tiene la cadena analizada. Figura 17.

Figura 17.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> word = "banana"
>>> len(word)
6
```

El iterador “for” puede ser usado en las cadenas también. Por ejemplo, en el caso en el que usáramos un sufijo común entre varios prefijos para formar diferentes palabras, como por ejemplo el sufijo común “ack” y el prefijo JKMPUQ que creará las palabras Jack, Kack, Mack, Pack, Ouack y Quack. La forma de uso del iterador se muestra en la figura 18.

Figura 18.

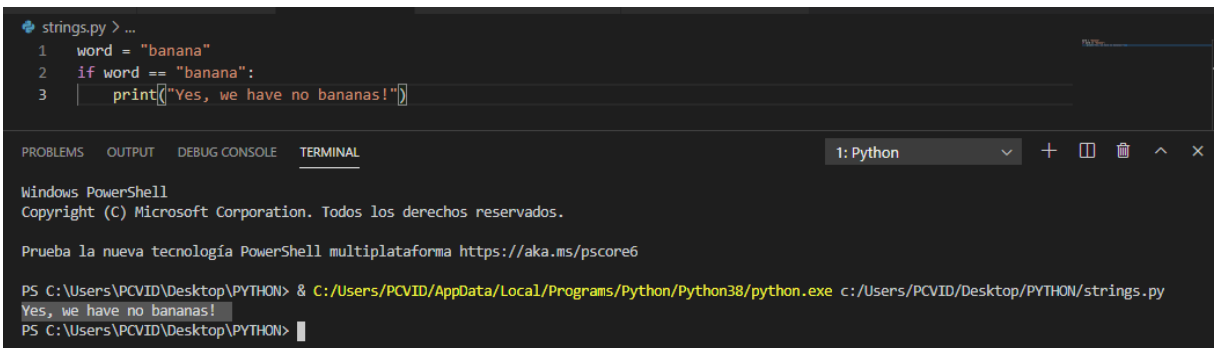
Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1 prefixes = "JKLMNOPQ"
2 suffix = "ack"
3
4 for p in prefixes:
5     print(p + suffix)
```

Las cadenas pueden compararse entre sí, usando un contador de caracteres. Un ejemplo se observa en la figura 19, en el que se define un código que permitirá comparar el valor de la función Word que es “*banana*” con otras funciones.

Figura 19.

Código de Python en Visual Studio Code.



```
strings.py > ...
1 word = "banana"
2 if word == "banana":
3     print("Yes, we have no bananas!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

PS C:\Users\PCVID\Desktop\PYTHON> & C:\Users\PCVID\AppData\Local\Programs\Python\Python38\python.exe c:/Users/PCVID/Desktop/PYTHON/strings.py  
Yes, we have no bananas!  
PS C:\Users\PCVID\Desktop\PYTHON>

Una de las características principales de las cadenas, a diferencia de otro tipo de valores, es que estos no se pueden modificar una vez han sido establecidos. El error mostrado en la figura 20 prueba que la función que pretende cambiar un carácter al valor de la variable *greeting* no puede hacerlo. Sin embargo, una cadena puede concatenarse con otras, o puede disgregarse usando el método “*split*”. La figura 21 muestra la separación palabra por palabra de la cadena que contiene la frase “*Well I never did said Alice*”.

Figura 20.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1 greeting = "Hello, world!"
2 greeting[0] = 'J'           # ERROR!
3 print(greeting)
```

Figura 21.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> phrase = "Well I never did said Alice"
>>> words = phrase.split()
>>> words
['Well', 'I', 'never', 'did', 'said', 'Alice']
```

### 1.4.2 Tuplas

Las tuplas son un tipo de datos similar a una lista, ya que agrupan un grupo de variables, pero que difieren de estas porque no pueden modificarse luego de crearlas. Estas son utilizadas para agrupar datos como dígitos o caracteres. Figura 22.

Figura 22.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> julia = ("Julia", "Roberts", 1967, "Duplicidad", 2009, "Actriz",
...         "Atlanta, Georgia")
```

Para asignar los valores de las tuplas se usa una función, figura 23, que los agrupa o desagrupa, como se muestra en la figura 24.

Figura 23.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> bob = ("Bob", 19, "CS")
>>> (name, age, studies) = bob    # tuple unpacking
>>> name
```

Figura 24.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
'Bob'  
>>> age  
19  
>>> studies  
'CS'
```

Las tuplas se usan para integrar el número deseado de valores en una misma función, característica que las hace útiles en la definición de los parámetros que se operan en ecuaciones, como, por ejemplo, cuando se busca hallar las dimensiones de un objeto utilizando la circunferencia y el área. Figura 25.

Figura 25.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1 def circle_stats(r):  
2     """ Return (circumference, area) of a circle of radius r """  
3     circumference = 2 * math.pi * r  
4     area = math.pi * r * r  
5     return (circumference, area)
```

### 1.4.3 Listas

Las listas se definen como un conjunto de valores llamados ítems o elementos que pueden ser de diferentes tipos, como cadenas o dígitos. Figura 26.

Figura 26.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1 >>> vocabulary = ["apple", "cheese", "dog"]  
2 >>> numbers = [17, 123]  
3 >>> an_empty_list = []  
4 >>> print(vocabulary, numbers, an_empty_list)  
5 ["apple", "cheese", "dog"] [17, 123] []
```



Cada elemento de una lista puede usarse independientemente, a partir de su posición en la lista. Estas posiciones están numeradas empezando con el 0 de izquierda a derecha. Figura 27.

Figura 27.

*Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> numbers[0]
17
```

Al igual que con los strings podemos aplicar operadores booleanos (de falso o verdadero “True, False”) para determinar la existencia o no de un valor dentro de la lista. La figura 28 muestra un ejemplo, con la lista “*horsemen*” en el que se encuentran 4 valores (en este caso cadenas). Los operadores booleanos permiten verificar la presencia o no de los elementos de interés, en este caso cadenas de interés.

Figura 28.

*Código Python How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> horsemen = ["war", "famine", "pestilence", "death"]
>>> "pestilence" in horsemen
True
>>> "debauchery" in horsemen
False
>>> "debauchery" not in horsemen
True
```

Cuando los valores dentro de las listas son dígitos pueden operarse como lo muestra el ejemplo de la figura 29.

Figura 29.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> first_list = [1, 2, 3]
>>> second_list = [4, 5, 6]
>>> both_lists = first_list + second_list
>>> both_lists
[1, 2, 3, 4, 5, 6]
```

```
>>> [0] * 4
[0, 0, 0, 0]
>>> [1, 2, 3] * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

A diferencia de las cadenas en las listas pueden modificarse los elementos, agregando un operador usando la posición de los elementos en la lista, cuya numeración empieza con 0 de izquierda a derecha. Figura 30.

Figura 30.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> fruit = ["banana", "apple", "quince"]
>>> fruit[0] = "pear"
>>> fruit[2] = "orange"
>>> fruit
['pear', 'apple', 'orange']
```

Cuando dos o más listas contienen el mismo tipo de valores, se denominan “aliased”. Cuando dos listas tienen elementos comunes, pueden modificarse juntamente con una instrucción común. Figura 31.

Figura 31.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> a = [1, 2, 3]
>>> b = a
>>> a is b
True
```

En las listas, el iterador *for* también funciona, como se muestra en la figura 32, en el que permite iterar la variable “*friends*”, usando la función “*friend*” para usar el iterador.

Figura 32.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
1 friends = ["Joe", "Zoe", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]
2 for friend in friends:
3     print(friend)
```

Las variables que contengan una cadena (string ) únicamente, se pueden separar en forma de listas. La figura 33 muestra un ejemplo usando un operador estandarizado `nombre.split()` que separa una cadena denominada “song” en diferentes palabras creando una nueva variable.

Figura 33.

Código Python *How to Think Like a Computer Scientist. Wentworth, P., et al (2011).*

```
>>> song = "The rain in Spain..."
>>> words = song.split()
>>> words
['The', 'rain', 'in', 'Spain...']
```

**1.4.3.1 Matrices.** Las listas también pueden ser útiles para representar matrices, por ejemplo “mx” que contiene tres listas de dígitos dentro de una misma lista. Figura 34.

Figura 34.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
>>> mx = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

#### 1.4.4 Diccionarios

Los diccionarios son un tipo de almacenamiento de datos que nos permite guardar cualquier tipo de cadena (*string*), lista o dígito dentro de una variable. La notación de los diccionarios es las llaves { }. En la figura 35, se muestra un diccionario de frutas con un valor numérico asignado a cada fruta, a manera de inventario de inventario de supermercado.

Figura 35.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> inventory = {"apples": 430, "bananas": 312, "oranges": 525, "pears": 217}
>>> print(inventory)
{'pears': 217, 'apples': 430, 'oranges': 525, 'bananas': 312}
```

Los diccionarios permiten ejecutar distintas operaciones, eliminar y agregar valores. La figura 36 muestra un ejemplo, usando el diccionario de la figura 35 removiendo el elemento “bananas”.

Figura 36.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> del inventory["bananas"]
>>> print(inventory)
{'apples': 430, 'oranges': 525, 'pears': 217}
```

En los diccionarios también puede usarse el iterador for. Las figuras 37 y 38 muestra un ejemplo, en el que se usa for para iterar un método pre-establecido en Python “nombre.keys()”.

Figura 37.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
1 for key in english_spanish.keys(): # The order of the k's is not defined
2     print("Got key", key, "which maps to value", english_spanish[key])
3
4 keys = list(english_spanish.keys())
5 print(keys)
```

Dando como resultado 3 líneas de iteraciones que contienen valores del mismo diccionario.

Figura 38.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
Got key three which maps to value tres
Got key two which maps to value dos
Got key one which maps to value uno
['three', 'two', 'one']
```

## 1.5 Numpy

Las operaciones que ejecuta Python con valores de texto y numéricos expresados de manera sencilla, no realizan procesos matemáticos. Figura 39. Es por esto que debe usarse en vez la librería Numpy, que establece y ejecuta operaciones matemáticas. Figura 40.

Figura 39.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> a = [2, 3, 8]
>>> 2 * a
[2, 3, 8, 2, 3, 8]
```

Figura 40.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> import numpy as np
>>> a = np.array([2, 3, 8])
>>> 2.1 * a
array([ 4.2,  6.3, 16.8])
```

### 1.5.1 Funciones contenidas en Numpy

La función `np.array()` permite transformar los valores contenidos en una lista en valores matemáticos operables. Figura 41.

Figura 41.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> b = np.array([
    [2, 3, 8],
    [4, 5, 6],
])
>>> b[1]
array([4, 5, 6])
>>> b[1][2]
6
```

Usando la lógica matemática pueden crearse filtros, figura 42, en el que descartamos los valores numéricos dentro de esta lista que sean mayores a 200.

Figura 42.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> a = np.array([230, 10, 284, 39, 76])
>>> cutoff = 200
>>> a[a > cutoff] = 0
>>> a
np.array([0, 10, 0, 39, 76])
```

La figura 43 muestra un ejemplo en el que se multiplican dos matrices, una 3x2 y la otra 1x2.

Figura 43.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> a = np.array([
    [0, 1],
    [2, 3],
    [4, 5],
    ])
>>> b = np.array([10, 100])
>>> a * b
array([[ 0, 100],
       [ 20, 300],
       [ 40, 500]])
```

## 1.6 Archivos

Los códigos que se escriben en Python son generalmente guardados en lugares de almacenamiento “no – volátiles” (mantienen los datos aun cuando el ordenador no se encuentra encendido – usb, CD, etc) denominados archivos. Los módulos, tal como el módulo de *Turtle*, son archivos que contienen líneas de estamentos o códigos para ser usadas en otros programas.

Para crear un archivo, la función tiene dos argumentos (línea 1): el primero siendo el nombre del archivo (test.txt) y el segundo la forma en que ejecutamos dicho archivo, siendo “w” el código que indica que el archivo se encuentra en proceso de escritura. El siguiente ejemplo, figura 44, aplica la función a una variable “myfile”.

Figura 44.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
1 with open("test.txt", "w") as myfile:
2     myfile.write("My first file written from Python\n")
3     myfile.write("-----\n")
4     myfile.write("Hello, world!\n")
```

### 1.6.1 Directorios

En el momento en que se crea un archivo ya sea abriéndolo o escribiéndolo; se debe establecer en el directorio en donde se ejecutará el programa. De igual forma, al abrir un archivo de tipo lectura, Python busca la ruta del mismo directorio, esto quiere decir que para abrir un archivo en otra carpeta tendríamos que especificar la ruta (archivos o carpetas para llegar al directorio). Figura 45.

Figura 45.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> wordsfile = open("/usr/share/dict/words", "r")
>>> wordlist = wordsfile.readlines()
>>> print(wordlist[:6])
['\n', 'A\n', 'A's\n', 'AOL\n', 'AOL's\n', 'Aachen\n']
```

Mediante la función *urlretrieve* (línea 6) podemos descargar cualquier tipo de contenido que sea compatible con Python desde la web ingresando la dirección URL (línea 3). Figura 46.



Figura 46.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
1 import urllib.request
2
3 url = "http://xml.resource.org/public/rfc/txt/rfc793.txt"
4 destination_filename = "rfc793.txt"
5
6 urllib.request.urlretrieve(url, destination_filename)
```

### 1.6.2 Módulos matemáticos

El módulo *math* contiene códigos de funciones matemáticas, o constantes matemáticas tal como el número pi ( $\pi$ ) o Euler. Figura 47.

Figura 47.

Código Python *How to Think Like a Computer Scientist*. Wentworth, P., et al (2011).

```
>>> import math
>>> math.pi                # Constant pi
3.141592653589793
>>> math.e                  # Constant natural log base
2.718281828459045
>>> math.sqrt(2.0)          # Square root function
1.4142135623730951
>>> math.radians(90)        # Convert 90 degrees to radians
1.5707963267948966
>>> math.sin(math.radians(90)) # Find sin of 90 degrees
1.0
>>> math.asin(1.0) * 2      # Double the arcsin of 1.0 to get pi
3.141592653589793
```

## Aplicación geológica

Dentro del desarrollo del seminario, elaboramos códigos que nos permitieran poder involucrar la geología a la programación. Como en la figura 48, en la cual vemos un código básico para la determinación de la dureza de un material según la escala de Mohs.

Figura 48.

Código de Python en Visual Code Studio.

```
1 def dureza (talco, yeso, calcita, fluorita, apatito, plagioclasa, cuarzo, topacio, corindon, diamante) :
2     if escala == 1 :
3         return talco
4     if escala == 2 :
5         return yeso
6     if escala == 3 :
7         return calcita
8     if escala == 4 :
9         return fluorita
10    if escala == 5 :
11        return apatito
12    if escala == 6 :
13        return plagioclasa
14    if escala == 7 :
15        return cuarzo
16    if escala == 8 :
17        return topacio
18    if escala == 9 :
19        return corindon
20    if escala == 10 :
21        return diamante
22
23
24    escala = int( input ( "¿cuál es el valor de dureza del mineral?"))
25    gp = dureza("talco", "yeso", "calcita", "fluorita", "apatito", "plagioclasa", "cuarzo", "topacio", "corindon", "diamante")
26
27    print("el mineral en la escala de mohs es ", gp)
28
```

## 1.7 GPRPy

Para el procesamiento y posterior interpretación de los datos GPR adquiridos es necesario el uso de un software especializado.

GPRPy es un software para procesar, visualizar e interpretar datos adquiridos con el radar de penetración terrestre, escrito en Python. Es de código abierto, compatible con una variedad de sistemas de GPR, incluyendo GSSI. El procesamiento y el trazado de datos se pueden realizar mediante interfaces gráficas de usuario, lo que facilita el aprendizaje del software y el compartir de los códigos como parte de una publicación, para garantizar una investigación reproducible. GPRPy permite el procesamiento y visualización de datos de perfil, análisis de velocidad, interpolación de cubos de datos 3D a partir de datos de perfil e interpolación 3D para interfaces visibles en múltiples perfiles (Plattner, 2020).

El procesamiento de datos de GPRPy está optimizado para reducir la velocidad y el consumo de memoria, mediante el uso de estructuras de datos y funciones del módulo Numpy de Python. Las dos GUI (interfaz gráfica de usuario) tienen la capacidad de crear automáticamente scripts de Python, almacenando internamente los pasos de procesamiento utilizados en la GUI. Los scripts de Python generados automáticamente, junto con los datos sin procesar, se pueden compartir para permitir a los investigadores reproducir los pasos de procesamiento y visualizar los resultados generados durante el proceso de investigación. GPRPy está alojado en un sitio web de desarrollo de software colaborativo que figura en la sección "Disponibilidad de datos y materiales". Esto permite a la comunidad utilizar GPRPy como código base para implementar sus propios procedimientos de procesamiento de datos y hacerlos disponibles instantáneamente. Los archivos producidos tienen las terminaciones en GPR (archivos. dt1), GSSI (archivos. dzt).

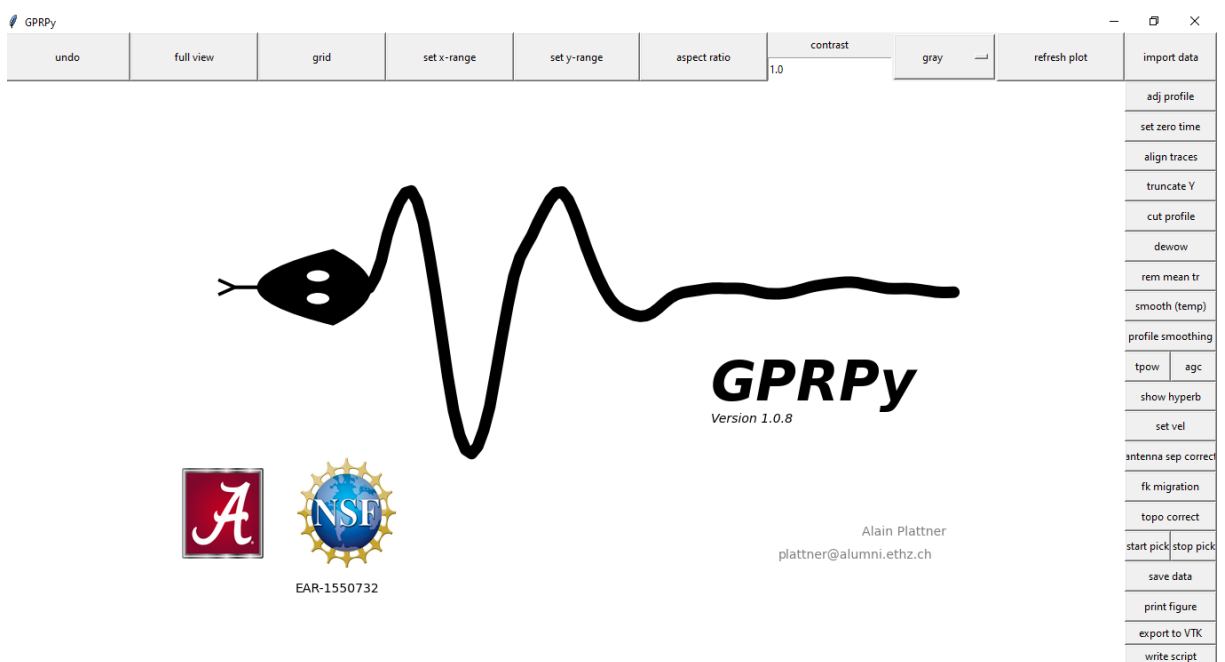
### **1.7.1 La interfaz de usuario (GUI) y la funcionalidad del módulo**

El GPRPy tiene dos interfaces para el procesamiento de datos de los radar-gramas y otra para el análisis de velocidad.

En este informe se hace énfasis en la GUI de procesamiento de datos, que es la utilizada con los datos disponibles, esta incluye filtros como herramientas para el mejoramiento de la calidad de visualización, conversión a profundidad a partir del análisis de hipérbolas, corrección del cero en las líneas que permiten corregir la posición vertical de las trazas de los datos si existe topografía.

Figura 49.

Interfaz GPRPy.



### 1.7.2 Herramientas del GPRPy

Las herramientas de visualización, estas están ubicadas en la parte superior de la interfaz gráfica, sólo modifican la apariencia de los datos, no su contenido. Entre estas herramientas se encuentran:

- undo: (atrás) Devuelve una acción realizada en el programa, incluyendo las acciones de procesamiento.

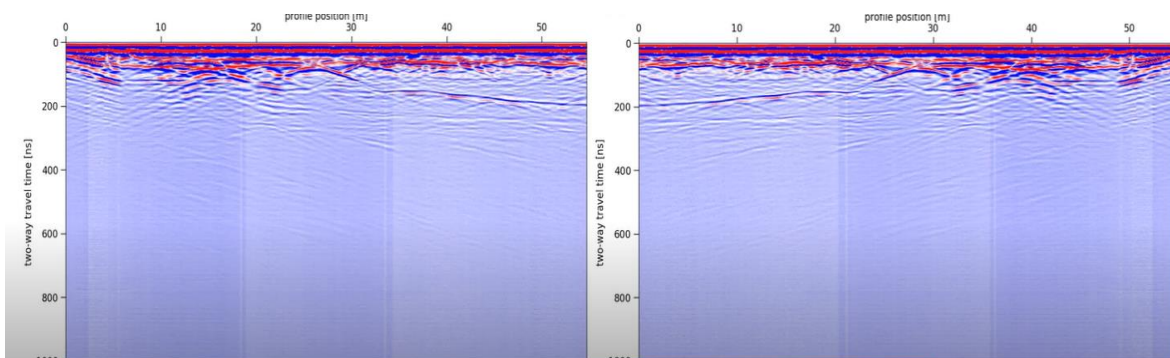
- full view: Permite observar las trazas de la línea completas.
- grid: Añade una grilla de profundidad y distancia horizontal a la imagen.
- set x-range o set y-range: Establece el rango de valores que aparece en los ejes de la imagen.
- contrast: Permite cambiar el contraste de tonos tanto para la escala de grises como para el rojo-azul.
- bwr o gray: Determina los tonos de la imagen de dos opciones: rojo y azul o en escala de grises.
- refresh plot: Esta opción actualiza la imagen luego de que se aplique una herramienta.
- import data: Carga los datos para observar en el programa.

Las herramientas de procesamiento modifican los datos, estas se encuentran ubicadas en el costado derecho de la interfaz gráfica del programa. Entre ellas se incluyen:

- adj profile: Esta herramienta permite cambiar el orden lateral de aparición de las trazas, de toda la línea o de un segmento especificado. El efecto es de reflejo. Figura 50.

*Figura 50.*

*Perfil antes y después de aplicar la herramienta adj profile.*

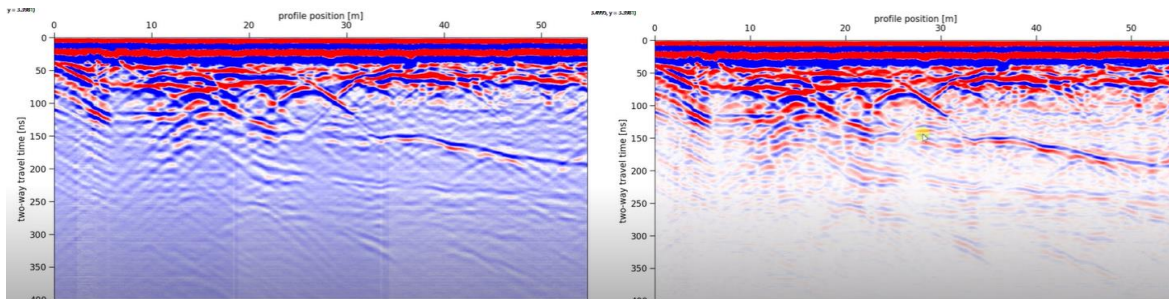


- set zero time: Establece el tiempo 0, o de inicio de grabación de señal de las ondas.

- align traces: Alinea las trazas automáticamente de acuerdo con sus amplitudes moviéndolas hacia arriba o hacia abajo.
- truncate Y: Esta herramienta me permite eliminar datos en el eje Y hasta determinado rango, es usada cuando se adquieren datos que no son significativos para la interpretación.
- cut profile: Elimina datos en el eje X, definiendo un punto inicial y uno final.
- dewow: Filtro de corte bajo de pasa banda alta.

Figura 51.

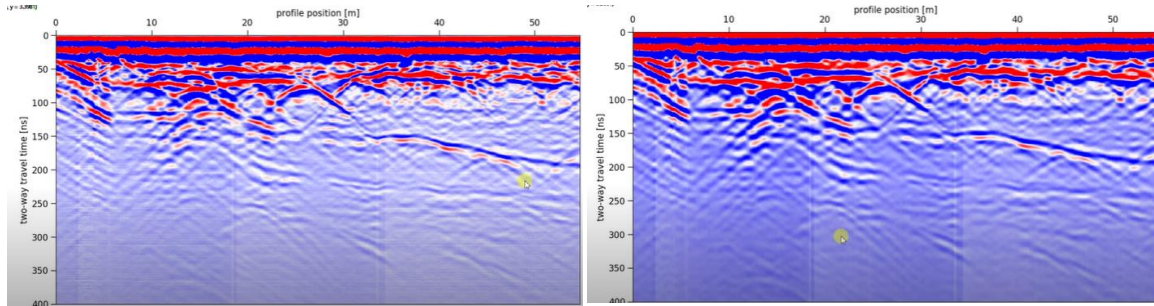
Perfil GPR antes y después de aplicar el filtro Dewow.



- rem mean tr: En una ventana de análisis, remueve el promedio de trazo largo de la señal, siendo útil para identificar y remover la onda de aire de la onda de tierra.
- smooth: Filtro de corte alto y pasa banda baja, aplicado en una ventana de análisis. Visualmente atenúa la intensidad de los reflectores, y modifica el valor de las trazas cambiándolas por un valor promedio de estas dentro de la ventana elegida por el usuario. Figura 52.

Figura 52.

Perfil GPR antes y después de aplicar el filtro Smooth.

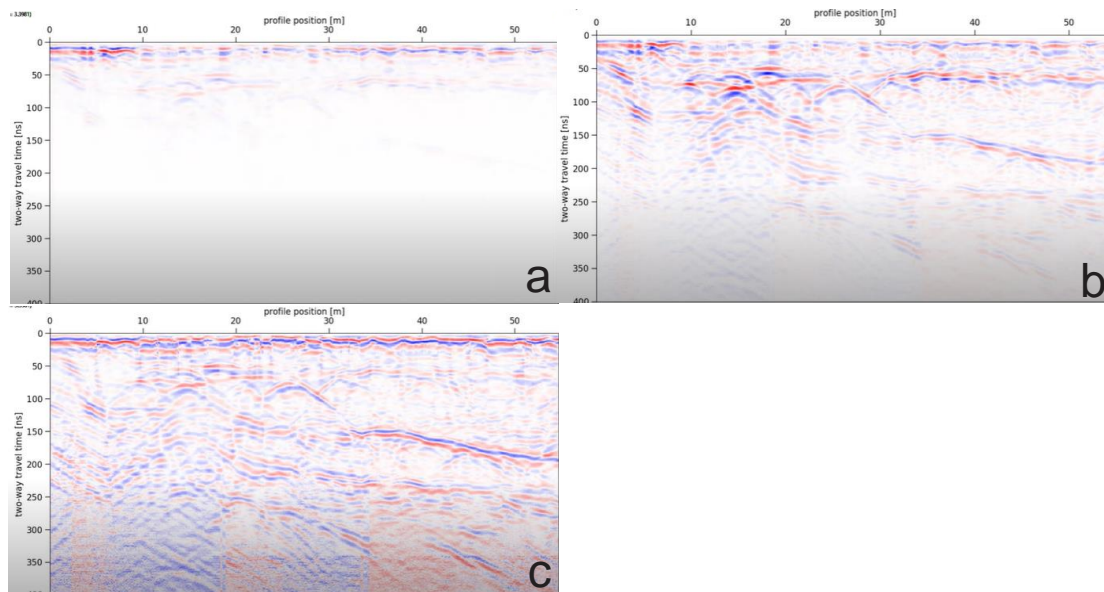


- profile smoothing: Filtro que suaviza y atenúa los cambios horizontales a lo largo del perfil, basándose en una ventana de análisis elegida por el intérprete.
- tpow: Herramienta de ganancia, que aumenta la amplitud de los reflectores usando como potencia un valor determinado por el usuario.
- agc: Herramienta de ganancia que nos ofrece el GPRPy, en la que la ganancia es determinada por el programa dándole mayor intensidad a los reflectores favoreciendo al contraste en la imagen.

Figura 53.

Figura 53.

Perfil GPR aplicando la ganancia AGC.

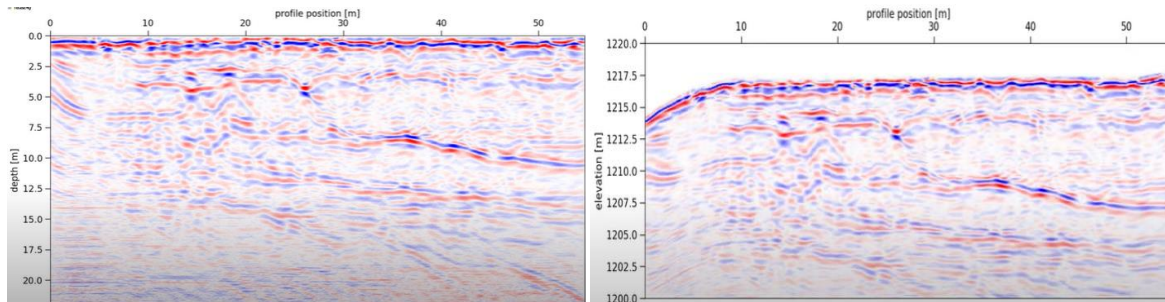


- show hyperb: Esta herramienta permite la determinación de una velocidad del medio, para hacer la conversión tiempo – profundidad a partir del análisis de una hipérbola
- set vel: Esta herramienta permite convertir los valores del eje Y, del dominio del tiempo a profundidad, usando una velocidad de las ondas en el medio definida por el usuario.
- antenna sep correct: Esta herramienta corrige la distorsión de los tiempos de llegada cerca de la superficie debido a la separación de la antena del transmisor y del receptor.
- fk migration: Esta herramienta permite hacer la migración, y eliminar hipérbolas que degeneran la calidad de los datos, si el dato de velocidad es el correcto.
- topo correct: corrección topográfica, se realiza con un archivo de texto donde se tiene la altura de la línea de adquisición discretizada en valores de distancia, se realiza si la topografía tiene variaciones drásticas. Figura 54.



Figura 54.

Perfil GPR aplicando la corrección de topografía.



- start pick – stop pick: Esta herramienta permite marcar un reflector en la imagen para recopilar información de la ubicación de este.
- save data: Guarda los datos procesados, incluido su historial, nombres de ruta absolutos de los archivos usados (de datos y topografía) en un archivo .gpr.
- print figure: Guarda la figura visible en pdf con la resolución elegida.
- export to VTK: Exporta la figura procesada a un formato VTK, que puede ser leído por Paraview o programas 3D similares.
- write script: Escribe una secuencia de comandos de Python para reproducir el estado actual del procesamiento, si los datos actuales son de un archivo .gpr, el script de Python contendrá todos los pasos que se remontan a los datos sin procesar. El script no contendrá configuraciones de visualización como configuraciones de rango x, a menos que se haya usado el comando "imprimir figura".

## **2 Fundamentos del GPR (Radar de Penetración Terrestre)**

El GPR es un método geofísico electromagnético (EM) para crear imágenes y mapear suelos subsuperficiales en alta resolución. El funcionamiento de este método consiste en la emisión de un pulso electromagnético de alta frecuencia (10-1000 MHz), que va a ser transmitido al suelo desde una antena instalada en el equipo, cuyas reflexiones serán recibidas por una antena receptora y grabadas por el equipo. La técnica permite hacer un sondeo desde unos pocos centímetros hasta unos cuantos metros dependiendo de la permitividad dieléctrica, siendo esta una constante que representa la capacidad de polarización de un material sometido a un campo eléctrico externo. Tomando en cuenta la conductividad de los materiales que se encuentren subyaciendo, por ejemplo, en materiales con una baja conductividad ( $>1$  S/m) como gravas, arena y agua dulce, la señal penetrará a grandes profundidades, a diferencia de materiales conductores como arcillas, limos y suelos de agua salina o poros contaminados, en los que el rango de sondeo es mucho menor (Davis & Annan, 1989).

Las adquisiciones de GPR y los datos resultantes dependen de factores como lo son la frecuencia de la onda que se emite, el arreglo de las antenas emisora y receptora, y las propiedades eléctricas del subsuelo. En la actualidad el método GPR es utilizado en variedad de aplicaciones tanto científicas como técnicas, en estudios arqueológicas, hidrogeológicas, de detección de minas explosivas enterradas, y geotecnia entre otros.

## 2.1 Fundamento físico del GPR

El método GPR utiliza ondas electromagnéticas (EM) que se transmiten a velocidades que dependen de la conductividad, permitividad dieléctrica y permeabilidad magnética de los materiales que conforman el subsuelo. Las frecuencias específicas que se usan en este método son susceptibles especialmente a la permitividad dieléctrica relativa (propiedad intrínseca de los materiales), por lo que el contraste de esta propiedad genera cambios de velocidad que determinan la generación y propagación de ondas electromagnéticas reflejadas. Estas ondas reflejadas, son captadas por la antena receptora, generando una imagen que permite el estudio del subsuelo.

### 2.1.1 Velocidad de onda

Las ondas electromagnéticas se propagan a diferentes velocidades, al pasar por distintos materiales, esto depende de las propiedades físicas de dicho material. Cuando un medio es homogéneo, se podría describir la velocidad como:

$$V = \sqrt{\frac{2}{\mu\epsilon} \left[ \left( 1 + \left( \frac{\sigma}{\omega\epsilon} \right)^2 \right)^{1/2} + 1 \right]^{-1/2}}$$

Al hacer una comparación entre la velocidad de una onda electromagnética en un medio determinado y en el vacío, es decir cuando los valores de permeabilidad magnética, permitividad dieléctrica y conductividad eléctrica; toman valores aproximados a cero (0), vemos que estas ondas son mucho más rápidas.

Partiendo del hecho de que generalmente las ondas emitidas por los equipos de GPR, se catalogan como de alta frecuencia, es posible asumir que  $\sigma \ll \omega\epsilon\sigma \ll \omega\epsilon$ , conocido también como régimen de olas. De esta forma podemos simplificar la ecuación de velocidad de la siguiente forma:

$$V = \frac{1}{\sqrt{\mu\epsilon}} = \frac{c}{\sqrt{\mu_r\epsilon_r}}, \text{ siendo } \mu_r \text{ y } \epsilon_r \text{ la permeabilidad y permitividad relativa}$$

Asumiendo que el material por el cual se propagan las ondas no es magnético, podemos decir que  $\mu_r$ , por lo cual, la ecuación de la velocidad se podría definir como:

$$V = \frac{1}{\sqrt{\epsilon_r}}$$

De acuerdo con la siguiente tabla de velocidades para distintos medios podemos asumir que cuando un medio se encuentra inmerso en agua, la velocidad de propagación disminuye notablemente en comparación del vacío, siendo el medio salubre el que presenta mayor disminución, esto debido a su alta conductividad eléctrica. Es decir, aquellas rocas, que no presenten incidencia de agua en sus poros o en su composición, ejercerán un efecto más rápido en la propagación de las ondas electromagnéticas dentro de las mismas.

Tabla 3.

*Propiedades electromagnéticas de diferentes materiales.*

Material	Permitividad relativa	Conductividad (mS / m)	Velocidad media (m / ns)
Aire	1	0	0,3
Agua dulce	80	0,5	0.033
Agua de mar	80	3000	0,01
Hielo	3-4	0,01	0,16
Arena seca	3-5	0,01	0,15
Arena saturada	20-30	0,1-1	0,06
Caliza	4-8	0.5-2	0,12
Lutitas	5-15	1-100	0,09
Limos	5-30	1-100	0,07
Arcillas	5-40	2-1000	0,06
Granito	4-6	0.01-1	0,13
Anhidritas	3-4	0.01-1	0,13

### 2.1.2 Reflexión y transmisión de ondas de radio

Las relaciones de la amplitud de reflexión se pueden cuantificar en un coeficiente de reflexión que se puede definir como la relación de las permitividades dieléctricas de los medios que están en contacto. Entre mayor sea la amplitud reflejada, mayor energía se va a reflejar. Como lo vemos evidenciado en la siguiente formula:

$$R = \frac{\text{Amplitud Reflejada}}{\text{Amplitud Incidida}} = \frac{\sqrt{\epsilon_1} - \sqrt{\epsilon_2}}{\sqrt{\epsilon_1} + \sqrt{\epsilon_2}}$$

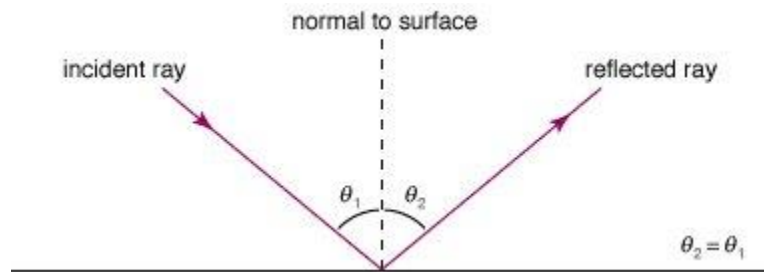
### 2.1.3 Refracción de ondas de radio

Cuando una onda que se propaga en un medio cambia de interfaz, es decir de un medio a otro con propiedades electromagnéticas diferentes; esta cambia de dirección. Una porción de ella presenta un fenómeno reflexivo, mientras que otra continua su paso de transmisión a través del segundo medio. El ángulo con el cual la onda se refleja y refracta, depende del ángulo inicial con el que la onda incide en la interfaz (Kearey, P. et al, 2002). Tomando la ley de Snell, podemos dar explicación a este fenómeno de la siguiente forma:

$$\frac{\sin\theta_1}{V_1} = \frac{\sin\theta_2}{V_2}$$

Figura 55.

Ley de Snell, tomada de [www.britannica.com/science/light/Reflection-and-refraction](http://www.britannica.com/science/light/Reflection-and-refraction).



Cuando las ondas electromagnéticas atraviesan materiales resistivos y no magnéticos, se puede definir la velocidad con la que se propagan de la siguiente forma:

$$V = \frac{c}{\sqrt{\epsilon_r}}$$

Tomando esto en cuenta, la ley de Snell se puede ajustar así:

$$\sqrt{\varepsilon_1} \sin\theta_1 = \sqrt{\varepsilon_2} \sin\theta_2$$

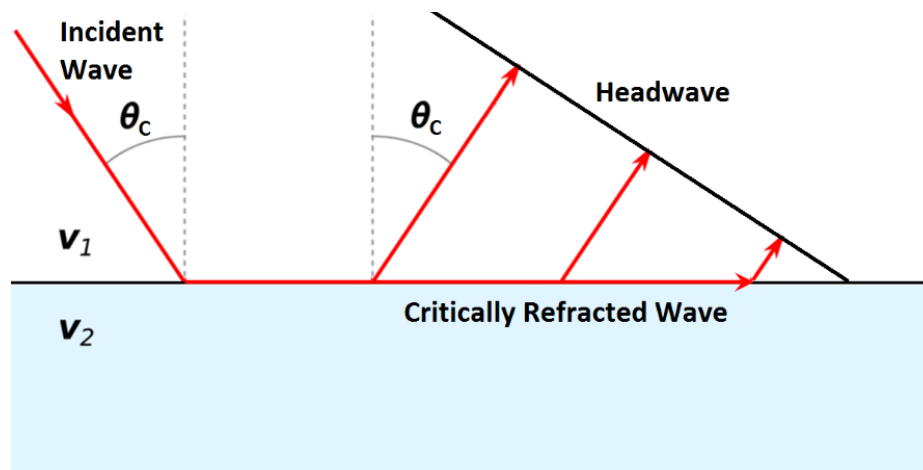
### 2.1.4 Refracción crítica

Cuando el ángulo de incidencia de la onda electromagnética  $\theta_1$  es de un valor determinado, y la velocidad del medio 2 es mayor a la del medio 1 ( $V_1 < V_2$ ), se puede presentar un ángulo crítico de propagación, por lo que la onda refractada se propaga a lo largo de la interfaz a la velocidad del medio 2. (Daniels, J., 2000). Este ángulo crítico se puede definir como:

$$\sin\theta_c = \frac{V_1}{V_2}$$

Figura 56.

Ángulo crítico, tomado de [gpg.geosci.xyz/content/GPR/GPR\\_fundamental\\_principles.html](http://gpg.geosci.xyz/content/GPR/GPR_fundamental_principles.html).



#### 2.1.4.1 Conductividad eléctrica ( $\sigma$ )

“La conductividad eléctrica está representada por la letra griega  $\sigma$ , y se define como la inversa de la resistividad ( $\rho$ ). Esto significa que una alta resistividad es igual a una baja conductividad”. (Webster, 2003). Esta conductividad eléctrica se evidencia en los diferentes materiales cuando se aplica un campo eléctrico. La conductividad eléctrica puede ser iónica, como en los fluidos, o electrónica, electrones moviéndose de un átomo a otro, como en los metales. Los iones disueltos en un líquido son cargas que se mueven conduciendo electricidad, es por esto por lo que el agua pura tiene una baja conductividad eléctrica. El agua salada en cambio tiene una alta conductividad eléctrica dado que a medida que la concentración de iones aumenta (Hayes & Drummond, 1995). El valor de conductividad es propio del material, a modo de ejemplo en la siguiente tabla se muestran diferentes materiales y sus valores de conductividad y resistividad.

*Tabla 4.*

*Propiedades eléctricas (Glover, 2015).*

<b>Roca</b>	<b>Conductividad (S m<sup>-1</sup>)</b>	<b>Resistividad (<math>\Omega</math> m)</b>
<b>Granito</b>	0.001 - 0.00001	1000 - 100000
<b>Gabro</b>	0.001 - 0.00001	1000 - 100000
<b>Basalto</b>	0.1 - 0.0001	10 - 100000
<b>Andesita</b>	0.001 - 0.00001	1000 - 100000
<b>Arena y grava</b>	0.05 - 0.002	20 - 5000
<b>Arenisca</b>	0.1 - 0.001	50 - 1000
<b>Caliza</b>	0.002 - 0.00001	5000 - 100000
<b>Dolomita</b>	0.001 - 0.0002	1000 - 50000
<b>Conglomerado</b>	0.01 - 0.0001	100 - 10000
<b>Carbón</b>	0.1 - 0.001	10 - 1000
<b>Shale</b>	0.001 - 1	50 - 1000
<b>Arcillas</b>	0.05 - 0.01	20 - 100
<b>Grafito</b>	100 - 0.01	0.01 - 100
<b>Sulfuros Masivos</b>	1000 - 1	0.001 - 1
<b>Agua Salada</b>	10 - 1	0.1 - 1
<b>Agua Fresca</b>	0.1 - 0.01	1 - 100
<b>Permafrost</b>	0.02 - 0.00001	500 - 100000
<b>Hielo Marino</b>	0.05 - 0.001	20 - 1000



La conductividad también se podría definir como el valor inverso de la resistividad dado por:

$$\sigma = \frac{J}{E},$$

Donde J es la densidad de corriente que fluye por un material en unidades de A / m<sup>2</sup> (amperios / metros al cuadrado), mientras que E representa el campo eléctrico en V / m (voltios / metro). Dado esto, la conductividad se puede representar por Siemens por metro (S / m) (Glover, 2015).

**2.1.4.2 Permeabilidad magnética ( $\mu$ ).** El paso de la densidad de flujo magnético a través de un material no solo depende del campo magnético, sino de la propiedad conocida como permeabilidad, esta se mide en Henrys / m siendo el Henry equivalente a un voltio por amperio. Existen dos tipos de permeabilidades para cualquier material magnético: permeabilidad absoluta y permeabilidad relativa.

**2.1.4.2.1 Permeabilidad absoluta.** Esta representa la magnetización de un material cuando es sometido a un campo magnético, se denota por la letra griega  $\mu$ , expresada de la siguiente forma:

$$\mu = \frac{B}{H}$$

Siendo B la densidad de flujo del campo magnético y H la intensidad del campo magnético.

**2.1.4.2.2 Permeabilidad magnética en el vacío.** Se define como la permeabilidad magnética dada en un vacío, la cual se produce cuando una corriente eléctrica o una carga en movimiento produce un campo magnético en dicho espacio. Se denota con la letra  $\mu_0$ .

Experimentando se encontró que, si tanto B, como H permanecen constantes en el vacío, el valor de la permitividad magnética en dicho vacío  $\mu_0$  sería  $4 \pi \times 10^{-7} \text{ H / m}$ .

**2.1.4.2.3 Permeabilidad relativa.** Se define como la relación entre la permeabilidad magnética absoluta y la permeabilidad magnética en el vacío cuando ambas están sometidas a un mismo campo magnético.

$$\mu_r = \frac{\mu}{\mu_0}$$

La permeabilidad magnética no debe ser considerada como una propiedad intrínseca de los materiales, pero es determinante para el análisis de la interacción de dicho material con el campo magnético al cual está sujeto.

Los materiales se clasifican magnéticamente dependiendo de cómo responden sus dominios magnéticos a la magnetización. Por ejemplo, los ferromagnéticos, tienden a atraer las líneas de campo en su interior, estos tienen un valor muy superior a 1. La siguiente es una tabla en la que se evidencian los valores de permeabilidad magnética dados para algunos minerales:

*Tabla 5.*

*Valores de permeabilidad magnética ( $\mu$ ) para algunos minerales (Zhdanov M. , 2018).*

Mineral	Permeabilidad magnética
Magnetita	2.5 - 16
Siderita	1.0035
Hematita	1.0004 – 1.001
Ankerita	1.00025
Pirita	1.000045

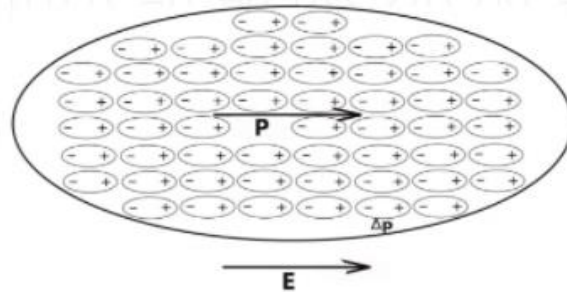
<b>Calcopirita</b>	1.00035
<b>Malaquita</b>	1.00035
<b>Casiterita</b>	1.00088
<b>Pirolusita</b>	1.0013
<b>Arsenopirita</b>	1.0024
<b>Cromita</b>	1.0024
<b>Pirrotita</b>	2.5
<b>Ilmenita</b>	1.36
<b>Augita</b>	1.0015
<b>Hornblenda</b>	1.00015
<b>Magnesita</b>	1.000250 – 1.000500
<b>Calcita</b>	0.999988 – 0.999994
<b>Silvita</b>	0.999986 – 0.999990
<b>Anhidrita</b>	0.999986
<b>Grafito</b>	0.999990
<b>Cuarzo</b>	0.999985 – 0.999988

### 2.1.5 Polarización eléctrica

La polarización eléctrica es la expresión de densidad de los momentos eléctricos dipolares permanentes o inducidos en un material dieléctrico, esto se ve ejemplificado en la figura 57. Al acercar un cuerpo cargado eléctricamente a un dieléctrico, hará que las cargas de dicho cuerpo se organicen, conociendo esto como polarización (Universidad de Vigo, 2012). La extensión con la que separan las cargas eléctricas se puede definir como momento de dipolo eléctrico y la suma de estos momentos se define como densidad de polarización  $P$  (C/m<sup>2</sup>) Coulomb por metro cuadrado.

Figura 57.

Polarización eléctrica (Fano, 2019).



### 2.1.6 Permitividad dieléctrica ( $\epsilon$ )

La permitividad dieléctrica, es una propiedad física de diagnóstico que caracteriza el grado de polarización eléctrica que experimenta un material cuando es influenciado por un campo eléctrico externo. Siendo también definida como la relación entre un campo eléctrico  $E$  dentro de un material y el desplazamiento de corriente eléctrica  $D$ :

$$D = \epsilon_0 E + P$$

Donde  $\epsilon_0$  es la permitividad del vacío  $8.85 \times 10^{-12} \text{ F / m}$ , esta define la relación entre el desplazamiento eléctrico  $D$  y el campo eléctrico  $E$  si el material no es susceptible a la polarización.

## 2.2 Propiedades dieléctricas de minerales formadores de rocas

Tanto los átomos, moléculas o cargas experimentan procesos de desplazamiento dieléctrico y polarización. La contribución del universo electrónico a la polarización resulta fácil de evidenciar porque esta puede ser calculada mediante índices de refracción óptica como el método de línea de transmisión y reflexión (Yaw, 2012). De acuerdo con lo anterior existe una relación entre dichos índices de refracción ( $n$ ) y las constantes ( $k_e$ ) de permitividad dieléctrica en los materiales, y la

permitividad dieléctrica en el vacío, siendo  $K_e = \xi/\xi_0$ . Según el trabajo realizado por (Zhdanov M. , 2018), es posible reconocer algunos valores de permitividad dieléctrica en ciertos minerales.

Tabla 6.

Permitividades dieléctricas de distintos materiales (Zhdanov M. , 2018).

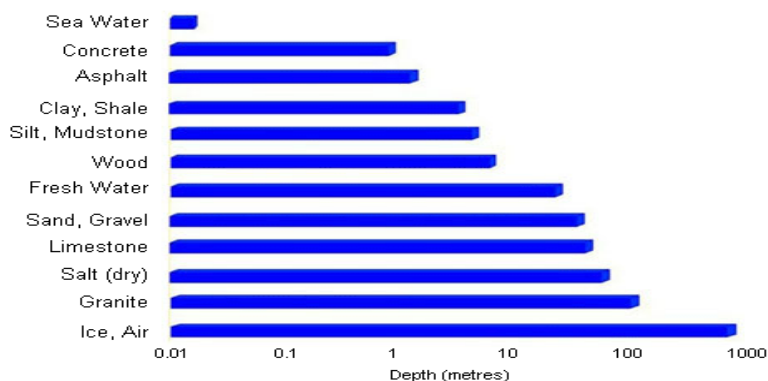
Mineral		Constante Dieléctrica pF/m
<b>Galena</b>	PbS	158
<b>Esfalerita</b>	ZnS	69.7
<b>Corindón</b>	Al <sub>2</sub> O <sub>3</sub>	97.2 a 117
<b>Hematita</b>	Fe <sub>2</sub> O <sub>3</sub>	207 a 212
<b>Rutilo</b>	TiO <sub>2</sub>	221
<b>Halita</b>	NaCl	50.4 a 54.8
<b>Fluorita</b>	CaF <sub>2</sub>	55.4 a 60.0
<b>Silvanita</b>	KCl	38.8 a 54.8
<b>Aragonito</b>	CaCO <sub>3</sub>	57.1 a 86.0
<b>Calcita</b>	CaCO <sub>3</sub>	69.0 a 75,2
<b>Dolomita</b>	CaMg(CO <sub>3</sub> ) <sub>2</sub>	60.1 a 70.7
<b>Apatito</b>	Ca <sub>5</sub> (F,Cl)(PO <sub>4</sub> ) <sub>3</sub>	65.5 a 92.8
<b>Anglesita</b>	PbSO <sub>4</sub>	644 a 4400
<b>Anhidrita</b>	CaSO <sub>4</sub>	50.5 a 55.7
<b>Barita</b>	BaSO <sub>4</sub>	69.5 a 109
<b>Celestita</b>	SrSO <sub>4</sub>	67.2
<b>Yeso</b>	CaSO <sub>4</sub> .2H <sub>2</sub> O	47.7 a 106
<b>Analcima</b>	NaAlSi <sub>4</sub> H <sub>2</sub> O	52.0
<b>Augita</b>	Ca(Mg,Fe,Al)(Al,Si) <sub>2</sub> O <sub>6</sub>	61.0 a 76.0
<b>Berilo</b>	Be <sub>3</sub> Al <sub>2</sub> Si <sub>6</sub> O <sub>18</sub>	50.2 a 58.3
<b>Biotita</b>	K(Mg,Fe) <sub>3</sub> AlSi <sub>3</sub> O <sub>10</sub> (OH) <sub>2</sub>	54.8 a 70.7
<b>Epidota</b>	Ca(Al,Fe) <sub>3</sub> (SiO <sub>4</sub> ) <sub>3</sub> OH	67.3 a 136
<b>Leucita</b>	KAlSi <sub>2</sub> O <sub>6</sub>	63.0
<b>Moscovita</b>	KAl <sub>3</sub> Si <sub>3</sub> O <sub>10</sub> (OH) <sub>2</sub>	54.8 a 70.7
<b>Albita</b>		48.2 a 49.1
<b>Oligoclasa</b>		53.3 a 53.6
<b>Andesina</b>		54.8 a 57.2
<b>Labradorita</b>		57.6 a 58.5
<b>Anortita</b>		62.3 a 64.0
<b>Cuarzo</b>	SiO <sub>2</sub>	36.4 a 37.8
<b>Sericita</b>		173 a 224
<b>Topacio</b>	Al <sub>2</sub> SiO <sub>4</sub> (F,OH) <sub>2</sub>	55.7 a 59.2
<b>Zircón</b>	ZrSiO <sub>4</sub>	a 106

### 2.3 Adquisición GPR

En las adquisiciones de GPR se utiliza una antena emisora y una receptora, de señales desde 12.5 mHz hasta 1000 mHz. La frecuencia específica emitida por el equipo es fundamental para determinar la profundidad máxima de penetración de la señal. Skin depth que se define como la distancia de propagación a la que la amplitud de una onda electromagnética se reduce en un factor de  $1/e$  ; es decir, reducido al 37% de su amplitud original. La profundidad de penetración ha sido determinada para materiales específicos de manera teórica. Figura 58. En otros casos debe hacerse modelamiento numérico, usando las ecuaciones de propagación de ondas electromagnéticas, y las propiedades de los materiales esperados de estudio. (Baker, Jordan, & Pardy, 2007)

Figura 58.

*Alcance de adquisición de las frecuencias típicas usadas en el GPR dependiendo el tipo de material. (P., Annan, s.f.).*



En general, a menores frecuencias se obtendrá una mayor profundidad de alcance, pero una menor resolución en las imágenes adquiridas, por lo que se hace necesario determinar previamente a hacer la adquisición, la frecuencia ideal para el tipo de material y objetivos a estudiar. De ahí la variabilidad de aplicaciones para este método geofísico. Algunas frecuencias y usos comunes aparecen en la figura 59.

Figura 59.

Aplicaciones GPR y frecuencias adecuadas (P., Annan, s.f.).

	12.5 MHz	25 MHz	50 MHz	100 MHz	200 MHz	250 MHz	500 MHz	1000 MHz
Geología profunda, Glaciología	✓	✓	✓					
Geología			✓	✓	✓			
Servicios públicos, geotécnica				✓	✓	✓		
Arqueología						✓	✓	
Medicina forense, nieve y hielo						✓	✓	
Minería, Canteras								✓
Hormigón, Carreteras, Puentes								✓

### 2.3.1 Tipos de arreglos en sondeos GPR

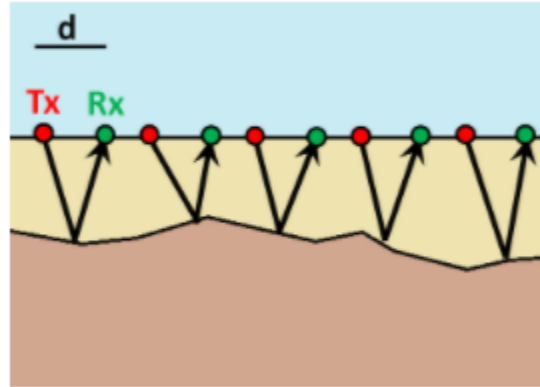
#### 2.3.1.1 Arreglo de compensación común

El esquema de adquisición de datos más utilizado en el GPR es el de compensación común. En este arreglo, la distancia entre la antena transmisora y la antena receptora es fija. Los datos se recopilan cada vez que el par transmisor-receptor se mueve a una nueva posición. En algunos casos, el transmisor y el receptor se colocan en un desplazamiento cero; también conocido como fuente y receptor coincidentes (zero offset).

Los levantamientos de zero offset son eficaces para localizar las profundidades de interfaces aproximadamente horizontales. Además, estos son muy efectivos para localizar tuberías, túneles y objetos enterrados compactos; ya que generan firmas hiperbólicas en los datos del radargrama y muy eficaces debido a que su adquisición es muy rápida, en tiempo inmediato se observa la imagen del perfil. Figura 60.

Figura 60.

Arreglo zero offset, recuperada de <https://gpg.geosci.xyz/content/GPR/index.html>.

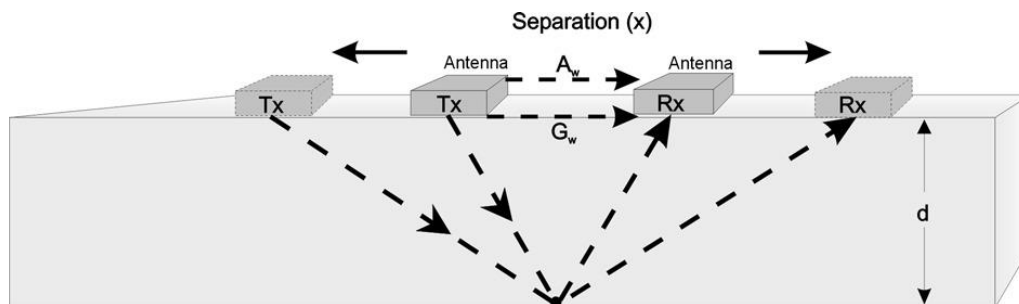


### 2.3.1.2 Arreglo de punto medio común

El sondeo punto medio común (CMP) es un ejemplo equivalente al arreglo usado en sismica de reflexión (Yilmaz, 2001). En esto (Baker, Jordan, & Pardy, 2007)s sondeos la velocidad de la señal del radar es estimada en función de la profundidad en el suelo, relación que se mide por medio de la variación de la distancia entre el transmisor y receptor para cada lectura, manteniendo la distancia al punto medio entre el transmisor y el receptor iguales. Figura 61.

Figura 61.

Arreglo CMP, recuperada de <https://gpg.geosci.xyz/content/GPR/index.html>.





En los sondeos con este arreglo CMP, las antenas se mueven controladamente, de tal forma que los tiempos de arribo de la señal para cada caso, permitan hacer una estimación matemática de la velocidad.

## **2.4 Procesamiento GPR**

Al realizar una toma de datos de GPR se obtienen imágenes de los perfiles, la calidad de estas imágenes en ocasiones no es la más adecuada para realizar una interpretación geológica optima debido las diferentes causas de interferencia a la hora de la adquisición (ruido). Por lo que se hace necesario realizar el procesamiento de estas aplicando las diferentes correcciones y ganancias, con la premisa de hacerlo de una forma lógica para no llegar a alterar los datos y recaer en errores en la interpretación.

### **2.4.1 Corrección de tiempo zero**

Esta corrección se debe realizar debido a las múltiples variaciones del entorno como son la temperatura, inestabilidad electrónica, dependiendo el arreglo la longitud del cable o el alcance de profundidad al que pueden llegar las señales emitidas por la antena; pueden generar saltos en el tiempo de llegada del primer frente de onda (Cassidy & Jol, 2009).

El  $T_0$ = tiempo cero, es el momento en el que empieza la grabación de señales por parte del equipo. En el GPR este tiempo se marca luego de que el  $R_0$ = primera señal recibida que viaja desde el transmisor, es detectable en los registros GPR. La definición del tiempo tiene un efecto en la posición de la interfaz de tierra en la sección, por lo tanto, las trazas deben ajustarse a una posición de tiempo cero común antes de poder aplicar los métodos de procesamiento (Cassidy & Jol, 2009).

### **2.4.2 Análisis de velocidad**

Este análisis es importante debido a que con el valor de la velocidad asignado es que se puede hacer la conversión de la profundidad de nanosegundos a metros. Existen dos formas de determinar esta velocidad: la primera de ellas es con una adquisición con el arreglo CMP, debido a que, para una topografía plana, los tiempos de llegada de una reflexión en el punto medio constante, dependen de la separación entre las antenas. En una gráfica de distancia entre las antenas y el punto medio vs. Tiempo, la pendiente de la relación lineal es  $1/v$ , donde  $V$  es la velocidad del aire. Las señales reflejadas por las interfaces horizontales, en el dominio del tiempo, tienen una forma hiperbólica definida por

$$t = \sqrt{x^2 + 4d^2}/V_{rms}$$

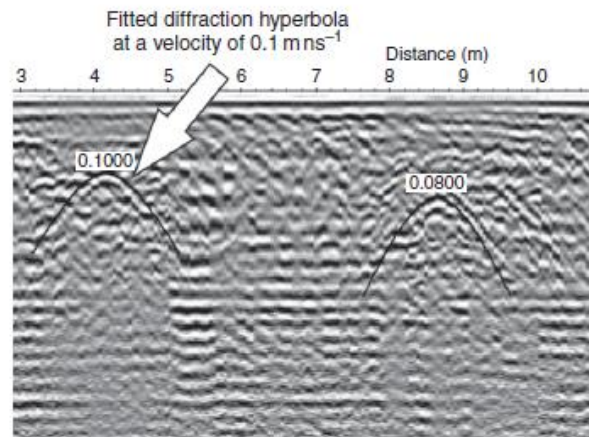
Donde:

x= separación de las antenas	$V_{rms}$ = velocidad promedio entre la
t= tiempo de viaje en las dos direcciones	superficie y la interfaz reflectante
d= profundidad del reflector	

La segunda forma de determinar una velocidad de propagación de ondas electromagnéticas en el medio rocoso es con el emparejamiento hiperbólico, entre la forma ideal de una función hiperbólica definida y cualquier hipérbola de reflexión o difracción que ocurra en las secciones, así pues, los programas de procesamiento permiten al usuario variar el radio, tamaño y velocidad para que coincidan con una hipérbola en cualquier lugar de la sección. Figura 62.

Figura 62.

Ejemplo de cómo determinar la velocidad a partir de la función hiperbólica (Cassidy & Jol, 2009).



Tanto el análisis CMP como las técnicas de emparejamiento hiperbólico producen valores de velocidad aproximados con valores de 10% o más de diferencia (Yilmaz, 2001). La conversión a profundidad de las secciones se hace con un solo valor en GPRPy, ingresado por el usuario.

### 2.4.3 Filtros

Los filtros usualmente son utilizados para mejorar la calidad de imagen del radargrama, eliminando de este la mayor parte de los tipos de ruido posible. Existen 3 tipos de filtros principales: los filtros de pasa baja, pasa alta y pasa banda.

#### 2.4.3.1 Filtro pasa baja

Este tipo de filtros se encargan de atenuar las frecuencias altas que normalmente son generadas por interferencias electromagnéticas entre dispositivos que usan este tipo de ondas como los teléfonos móviles, y la antena del radar (Benedetto, A et al, 2017). A su vez mantienen intactas las

bajas frecuencias en el radargrama. Este filtro es equivalente al filtro de suavizado (smoothing filter) (Szymczyk & Szymczyk, 2013).

#### **2.4.3.2      *Filtro pasa alta***

Este filtro remueve las señales de baja frecuencia en cada traza, ruido usualmente relacionado con las ondas terrestres y otras fuentes como vehículos, cercas eléctricas, árboles o edificio cercanos (Benedetto, A et al, 2017). Zonas en el radargrama con tonalidades grisáceas generalmente están asociadas a frecuencias bajas por lo que este tipo de filtros hace una mejora y las suprime (Szymczyk & Szymczyk, 2013).

#### **2.4.3.3      *Filtro pasa banda***

Este tipo de filtros se encarga de depurar las frecuencias que son muy bajas y a su vez muy altas manteniendo un rango medio de frecuencias (Szymczyk & Szymczyk, 2013). Estos filtros son de gran ayuda cuando se han aplicado ganancias de amplitud a la información, usualmente en terrenos muy irregulares o con suelos que provoquen pérdidas (E. C. Utsi, 2017).

#### **2.4.3.4      *Filtro Dewow***

Procesamiento básico y requerido de los datos de GPR. Se define como el filtrado de frecuencias bajas, llamado “Dewowing”. Estos componentes de baja frecuencia son causados principalmente por una saturación de la señal de las ondas directas u ondas de tierra – aire (Maruddani & Sandi, 2019). Su mal manejo podría generar una distorsión en el espectro de toda la señal causado por la descomposición de los datos de baja frecuencia que afectaría la eficacia de los filtros posteriores. Debido a esto los equipos de GPR actuales generan un filtrado de Dewow

de forma automática con estándares de filtro óptimos (Cassidy & Jol, 2009), (Maruddani & Sandi, 2019).

#### **2.4.4 Herramientas de Ganancia**

La señal emitida se atenúa por las pérdidas de dispersión geométrica al propagarse a través del subsuelo, por lo que se hace necesario aplicar una ganancia a los datos para incrementar las amplitudes de las señales que se reflejan a mayor profundidad (Izargain, 2004). Aunque estas ganancias mejoran de visualización de las secciones de GPR, también alteran la estructura de los datos adquiridos aumentando las amplitudes de los datos y de los ruidos. Por lo tanto, es importante que se comprendan los efectos de las funciones de ganancia antes de aplicarlas y que los datos se traten con cuidado al interpretarlos una vez estas se han aplicado (Davis & Annan, 1989).

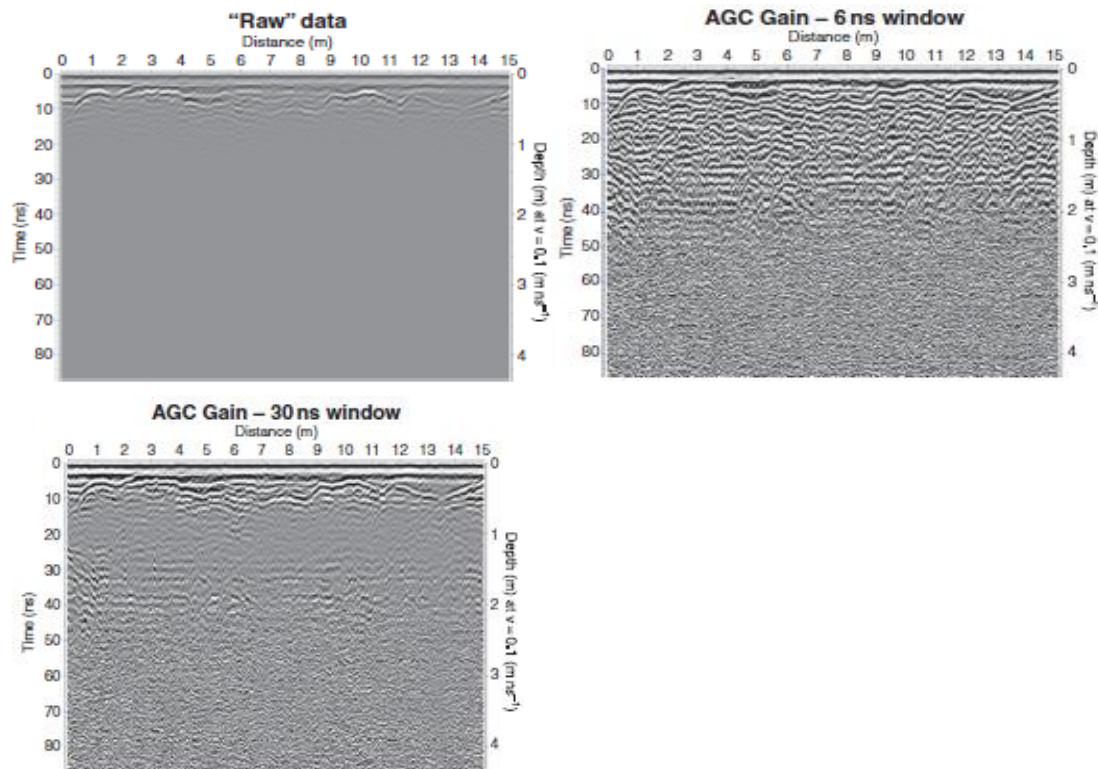
Existen diferentes tipos de ganancia, por ejemplo, ganancia constante, ganancia exponencial, AGC, SEC, tpow, cada uno con características diferentes, pero operando de la misma forma un factor multiplicador a regiones de la traza. (Cassidy & Jol, 2009).

Para casos como los que muestran continuidad del horizonte estratigráfico, es importante mostrar toda la información independientemente de la fidelidad de la amplitud, por esto, una

ganancia adecuada es AGC. Esta pretende restablecer un valor cuadrático medio a lo largo de las trazas, aumentando así las señales más débiles. Figura 63.

Figura 63.

Ganancia AGC para un perfil GPR (Cassidy & Jol, 2009).



## 2.4.5 Migración

Una de las ultimas herramientas utilizadas durante el procesamiento de los datos de GPR es la migración, siendo utilizada originalmente en la rama de la sismica, donde se considera de gran importancia. (Cassidy & Jol, 2009). La migración es un proceso de inversión que reacomoda los espacios de los elementos del radargrama, esto con el fin de ubicar las reflexiones y difracciones en su posición real, dando como resultado mucha más resolución (Izargain, 2004). También se puede definir como un conjunto de procesos matemáticos que establecen objetos lineales (ej:

tuberías) a partir de la corrección de hipérbolas. Al ser este un proceso matemático, se pensaría que se requieren conceptos avanzados de esta rama para hacer la migración de estos datos, pero no es así; esto debido a que todos softwares encargados de interpretar archivos de GPR poseen la migración dentro de su conjunto de herramientas de procesamiento (E. C. Utsi, 2017).

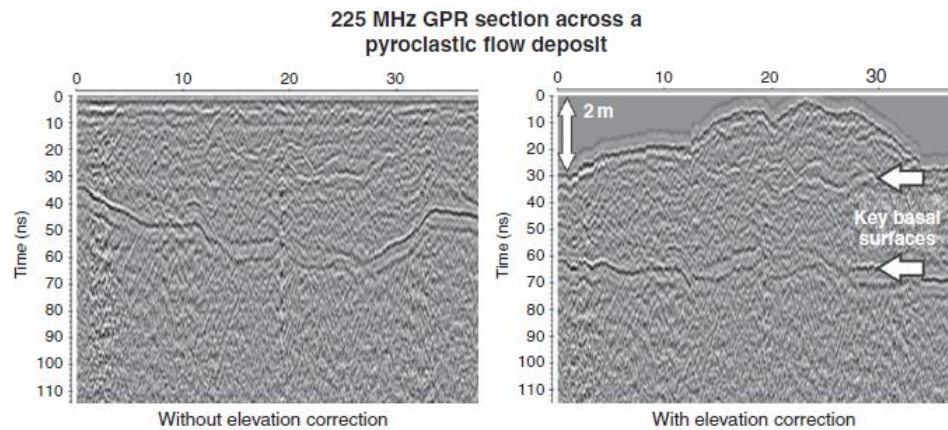
#### **2.4.6 Correcciones de elevación o topográficas**

Esta corrección es de gran ayuda para facilitar la interpretación y no hacerlo de forma errónea cuando el conjunto de datos se recopila sobre una superficie que no resulta plana ni nivelada. Con la corrección se posicionan los datos en su contexto espacial correcto. Estas denominadas correcciones estáticas de elevación se realizan normalmente con una simple corrección de velocidad constante que actúa en sentido vertical (Yilmaz, 2001). Esto es adecuado para una topografía relativamente suave, pero si las variaciones topográficas son muy abruptas, entonces se requieren rutinas de corrección topográfica más avanzadas (Lehmann & Green, 2000).

La corrección topográfica desplaza verticalmente cada traza para que el 0 de las trazas de la línea siga la superficie del terreno, proporcionando así un diagrama de elevaciones (Izargain, 2004). Las correcciones topográficas son importantes ya que ubican las reflexiones / difracciones de GPR en su contexto estratigráfico correcto, asegurando así que las interpretaciones sean realistas. Figura 64.

Figura 64.

Ejemplo de corrección por elevación topográfica (Cassidy & Jol, 2009).



### 3 Casos de Estudio Desarrollados en el Seminario

#### Set de datos:

Los datos con los que trabajamos durante el seminario de investigación fueron adquiridos en dos campañas de campo: 1.) en Mogotes, Santander. La cual inició el lunes 18 de marzo del 2019 y termino el 24 de marzo del 2019. Estas adquisiciones se realizaron por el Ing. Jheyston Omar Serrano Luna. Se adquirieron aproximadamente 50 líneas de GPR tanto en el abanico de Mogotes como en el batolito de Mogotes, con antenas de 100 y 400 MHz. El equipo utilizado para estas adquisiciones fue el GSSI Sir 3000. 2) en Bayunca, Cartagena en los alrededores del Volcán de Lodo Yarumal. Estas adquisiciones fueron realizadas por el Geólogo Álvaro Manrique del 7 al 18 de Mayo de 2019. Se adquirieron 10 líneas con la antena de 400 MHz. El equipo utilizado para estas adquisiciones fue el GSSI Sir 3000. Figura 65.



Figura 65.

Equipo de adquisición GPR utilizado, recuperada de [https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSmsr6\\_JeFyiYHcfH2XkDpyCFr352MWIupNw&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSmsr6_JeFyiYHcfH2XkDpyCFr352MWIupNw&usqp=CAU).



De estos set de datos adquiridos en parejas se procesaron e interpretaron dos líneas, una línea cada pareja. La selección de la línea trabajada tiene en cuenta distintos factores: el primero de ellos fue la ubicación que fuera geológicamente significativa; segundo que se pudiera hacer un amarre con la litología presente en la zona (presencia de afloramientos) o con datos descritos en

las bitácoras de campo, y tercero, que la calidad de la imagen del radargrama fuera óptima para procesar e interpretar de una forma más sencilla.

### **3.1 Caso de Estudio 1: Mogotes Santander**

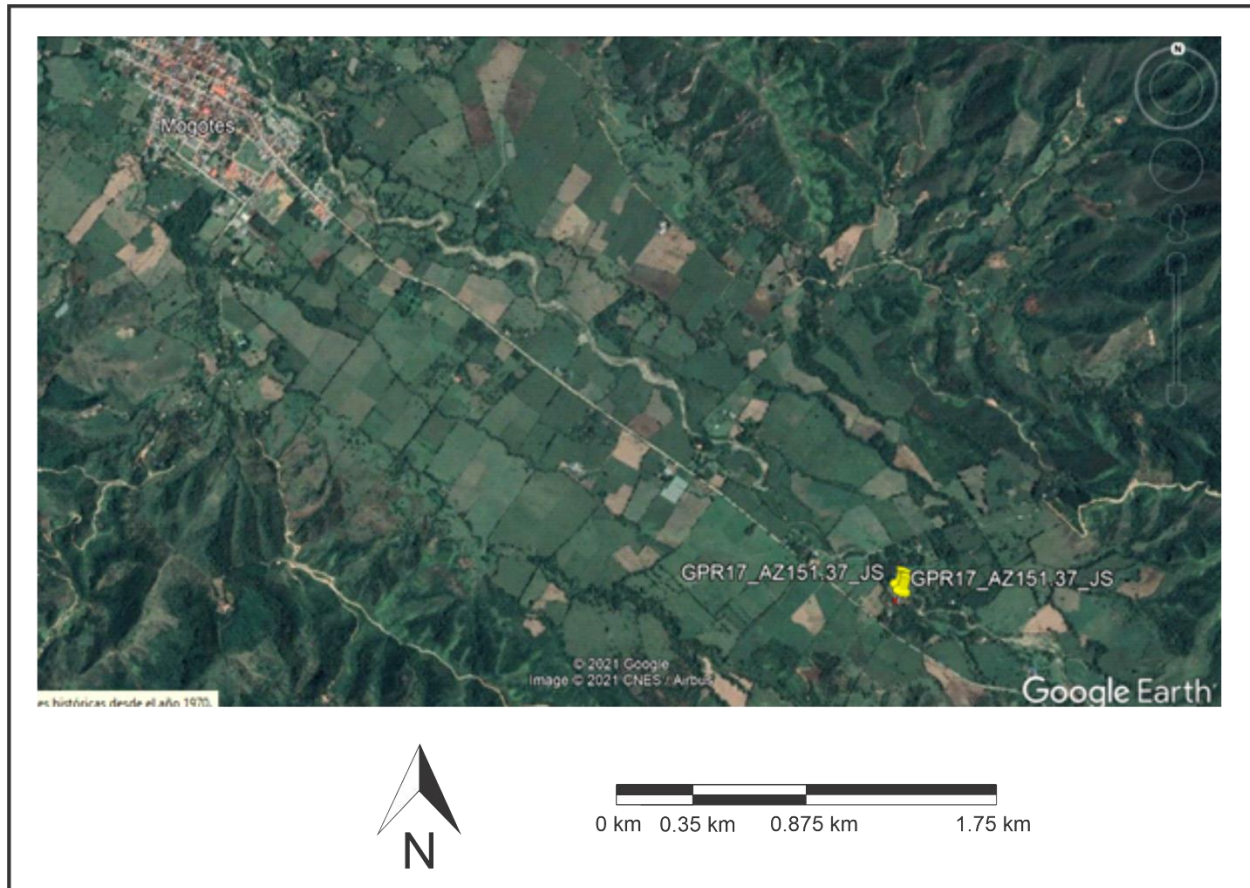
#### **3.1.1 Localización de la línea de estudio**

La línea seleccionada fue la Mogotes\_017.bzt. Esta línea fue adquirida con la antena de 400 MHz en las coordenadas (figura 66):

Punto inicial	Punto final
E: 1125731	E: 1125745
N: 1205898	N: 1205872

Figura 66.

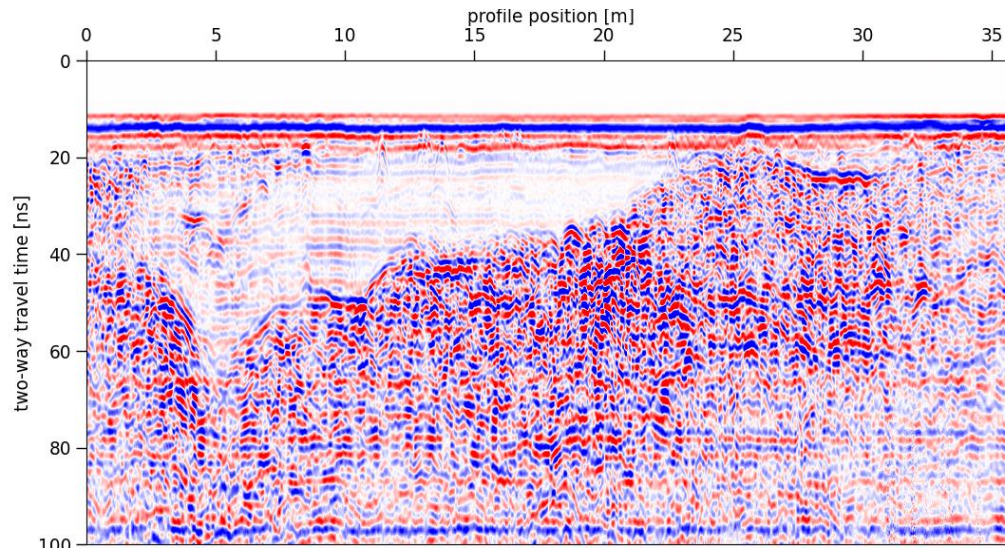
Ubicación de la línea número 1, Google Earth.



Esta línea fue adquirida en cercanías al río Mogoticos, la línea tiene un largo de 36m. En el informe de campo se documenta que la línea fue adquirida en cercanías al río, y se documenta una fotografía de afloramiento, figura 77.

Figura 67.

Línea número 1, imagen radargrama, GPRPy.



### 3.1.2 Resumen Geológico

Hacia la parte occidental de COR se encuentra el municipio de Mogotes, ubicado hacia el suroeste del Macizo de Santander. El batolito de Mogotes recibe su nombre debido a este municipio el cual limita al occidente con la falla de Bucaramanga (Ward D., 1973). Figura 68.

### 3.1.3 Geología local

Localmente en el área de interés (zona rural del municipio de Mogotes) afloran unidades ígneas y metamórficas, infrayaciendo secuencias de depósitos cuaternarios de tipo aluvial que serán nuestro objeto de estudio. Figura 68.

#### 3.1.3.1 Batolito de Mogotes (JTRcm)

El Batolito de Mogotes es resultante del magmatismo Triásico-Jurásico en el Macizo de Santander (Correa, 2016). Posee una extensión aproximada de 40 km de largo y 15 km de ancho

(Ward D., 1973). Se encuentra constituido principalmente por monzogranitos y en menor proporción por granodioritas; (Jerez & Rodriguez, 2018) basándose en las características mineralógicas definieron el Batolito de Mogotes principalmente como una cuarzomonzonita biotítica de color rosado naranja, gris rosácea o blanca rosácea, equigranular a subporfirítica y granito predominantemente equigranular de grano medio.

### **3.1.3.2 Formación Floresta metamorfoseada (Dfm)**

El termino floresta metamorfoseada fue postulado por (Ward D., 1973) para referirse a una secuencia de rocas de aproximadamente 1000 m de espesor, de edad Paleozóico Tardío (Devónico 390-380 Ma) según (Boinet, 1985). Según Vargas et al. (1981) esta formación se encuentra ubicada en la zona centro occidente del cuadrángulo I-13, compuesta por conglomerados, argilitas grises, metalimolitas calcáreas y mármol gris claro, siendo estos últimos la característica de reconocimiento litológico más importante de esta unidad en donde se puede evidenciar la presencia de restos de crinoideos (Pineda, 2017). Figura 68.

### **3.1.3.3 Depósitos cuaternarios**

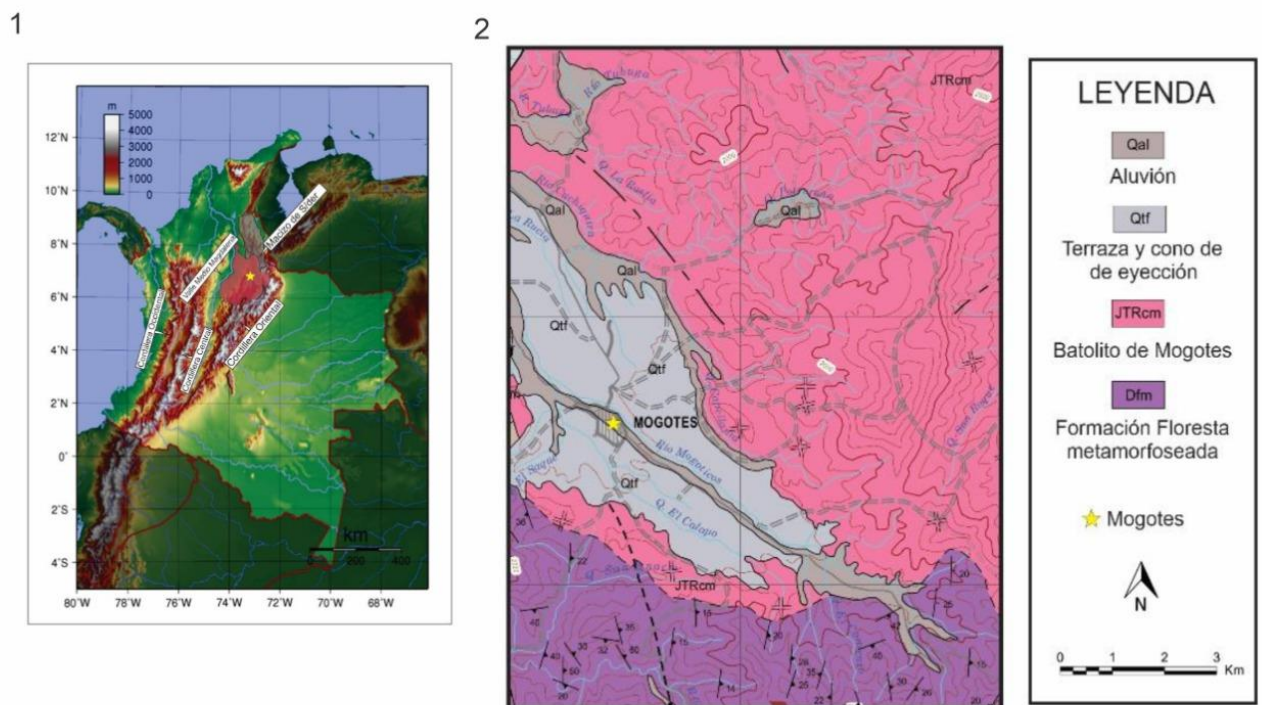
Los depósitos Cuaternarios que se encuentran dentro del departamento son muy variados en su origen, y se encuentran ampliamente dispuestos a lo largo de los ríos y quebradas principales (Royero & Clavijo, 2001). Los principales depósitos sedimentarios cuaternarios del Abanico de Mogotes forman un valle extenso infrayaciendo al municipio del mismo nombre, en donde su ápice se encuentra en la dirección ESE mientras que se va distanciando hacia el pie del abanico en dirección WNW (Mantilla & García, 2018).



Los depósitos aluviales presentes se dan principalmente en forma de franjas extendidas a lo largo de los bordes del río Mogoticos. Estos depósitos están formados básicamente por guijos y cantos de roca ígnea, metamórficas y sedimentarias, en una matriz arcillosa; en los alrededores del río Mogoticos predominan los cantos de material metamórfico e ígneo (Lucena, 2002).

Figura 68.

1) Mapa topográfico de Colombia ilustrado. 2) Mapa geológico de la plancha 136.



Geomorfológicamente, los depósitos cuaternarios están representados por planos elongados de morfología plana a suavemente ondulada y modelada sobre sedimentos aluviales. Su origen está relacionado con procesos de erosión y acumulación aluvial dentro de antiguas llanuras de inundación, por donde fluye una corriente. Esta unidad se caracteriza por presentar baja intensidad

de movimientos en masa, siendo las reptaciones el tipo más común, encontrándose en áreas de hasta 30 m x 30 m.

## **3.2 Caso de Estudio 2: Bayunca, Bolívar**

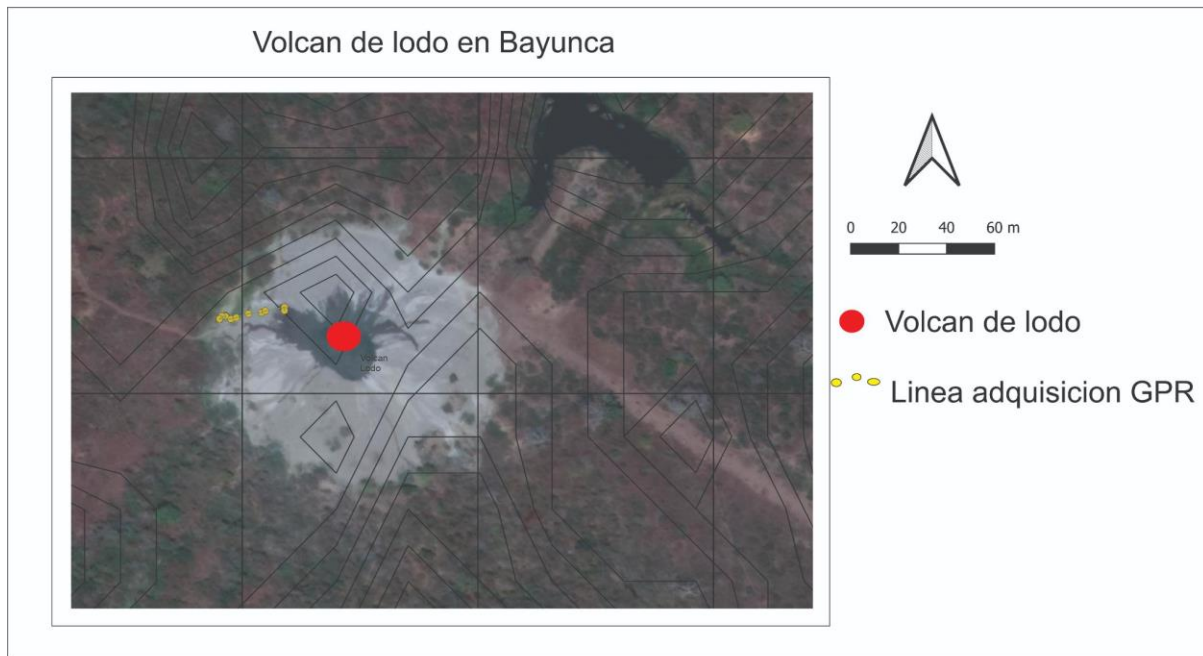
### **3.2.1 Localización de la línea de estudio**

La línea seleccionada fue la BAY\_002.bzt. Esta línea fue adquirida con la antena de 400 MHz en las coordenadas, figura 69:

Punto inicial	Punto final
E: 856320	E: 856331
N: 1660130	N: 1660135

Figura 69.

Ubicación de la línea número 2, Google Earth.



Esta línea fue adquirida en cercanías al volcán de lodo, la línea es de aproximadamente 27 metros.

#### 4.2.2 Resumen Geológico

La unidad estructural correspondiente al Cinturón de San Jacinto está ubicada en el flanco oriental del terreno Sinú – San Jacinto, limitada al occidente por el lineamiento del Sinú. La división de estos cinturones está marcada por la falla de Villanueva – El totumo, o lineamiento del Sinú. El cinturón del Sinú (CS) abarca desde la falla Urumita en el Golfo de Urabá, hasta Barranquilla. Al oriente limita con el lineamiento del Sinú y al oeste con el cinturón deformado del caribe meridional (Carvajal J. , 2011). El CS comprende rocas de edades entre el Oligoceno, hasta el Plioceno, correspondientes a 5000 m de pelagitas y hemipelagitas (aproximadamente);



además de lodolitas calcáreas y silíceas, chert y turbiditas muy finas de edad Mioceno superior - Plioceno, cubiertas por calizas arrecifales, arcillolitas, areniscas y conglomerados. Estos últimos pueden llegar a los 4000 m de espesor (Duque-Caro, 1979).

Por sus características estructurales, los cinturones de Sinú y San Jacinto se subdividen en bloques tectónicos, los bloques “Turbaco” y “Cartagena” (Reyes, Guzman, Barbosa, & Zapata, 2001). El bloque de Turbaco es el área de mayor importancia para este trabajo debido a que aquí se encuentra la zona de estudio. En esta región, afloran rocas de las formaciones Arjona, Bayunca, la Popa y Arroyo grande.

### **3.2.2 Resumen geológico**

El diapirismo de lodo es característico en las rocas del cinturón del Sinú, junto con algunos emplazamientos reportados (Carvajal J. , 2011).

Se llama diapiro de lodo, a la intrusión de una masa de lodo y fluidos de metros hasta kilómetros de tamaño, ascendiendo debido a contrastes de flotabilidad y densidad con las litologías circundantes pero que nunca aflora. Los diapiros de lodo han sido descritos en ambientes de márgenes pasivos, deltaicos como en el delta de Nigeria o Barei, donde el movimiento de los fluidos sobre-presionados en este caso es de tipo reactivo ya que la sobrecarga diferencial que se da en márgenes pasivos genera fallas y fracturas, que a su vez definen la depositación sintectónica (Quintero, 2012). El diapirismo de lodo en Colombia también se ha sido atribuido a la sobrepresurización, debida a esfuerzos tectónicos compresivos regionales o locales (Aristizábal, Ferrari, & Silva, 2009) y a la deposición de materiales con densidades diferentes que permitió el diapirismo sinsedimentario (Duque-Caro, 1979).

La ocurrencia del diapirismo de lodo en la región caribe colombiana es, en gran parte, el fenómeno responsable de la geomorfología de la región y particularmente de la configuración de la línea de costa. Otros autores han relacionado el modelo de formación de los diapiros de lodo en Colombia, con la formación del prisma de acreción (Carvajal & Mendivelso, 2017).

Sin embargo, se sugiere a su vez un modelo de colapso gravitacional que sugeriría un margen divergente (Rosello & Cossey, 2012) o de menor trascendencia en la formación del prisma de acreción (Duque-Caro, 1979). La expresión en superficie del fenómeno del diapirismo son los volcanes de lodo y en el área de Cartagena se presentan emplazados predominantemente en rocas de las formaciones Arjona y bayunca (Carvajal J. , 2011).

Un volcán de lodo es considerado como una geoforma de cerros con forma de domo, de morfología colinada y laderas convexas y generalmente cima plana; formados por la acumulación por flujo de lodo y por la deformación por empuje de los materiales confinados a profundidad (Carvajal J. , 2011). Los lodos aportados por estos volcanes se componen generalmente de minerales de arcilla y silicatos, escasos sulfuros y carbonatos y trazas de sulfatos y óxidos, y en gases predomina el metano con trazas de etano, propano, butano y CO<sub>2</sub> (Carvajal J. , 2011).

Están formados por gas que escapa hacia la superficie, cuando éste encuentra un conducto que generalmente es una falla o fractura, y este flujo de gas viene acompañado por material de poca cohesión, agua, salmueras y lodo de más baja densidad que las secuencias sedimentarias adyacentes (Quintero, 2012).

### **3.2.3 Geología local**

Las dos principales formaciones que afloran en la zona son:

#### **3.2.3.1 Formación bayunca (n1n2b)**

Descrita por (Germeraad, Hopping, & Muller, 1968), muestra morfología suave, ondulada a plana. Tiene un espesor aproximado de 500m, aunque este podría variar su espesor por repetición tectónica. Su deposición se dio en un ambiente de zona intermedia evidenciado por la presencia de canales y la abundante concentración de conchas, además de areniscas que rellenan canales abandonados (Ingeominas, 2001). Debido a la fauna del Mioceno encontrada, se da una edad entre el Mioceno y Plioceno. Son pocos los depósitos del cuaternario, como las Gravas de Rotinet, Formación Arroyo Grande y Formación La Popa, y algunos depósitos recientes de tipo llanura aluvial, llanura costera, coluvio-aluviales, de playa y de manglar (Ingeominas, 2001).

#### **3.2.3.2 Formación Arjona (e3n1a)**

Descrita por (Anderson, 1926) se define como una secuencia de sedimentos marinos del Eoceno y presenta una morfología ondulante. El espesor en distintas secciones varía desde los 2500 m hasta los 1600 m, sin embargo, la presencia de fallamientos puede aumentar su espesor. La formación Arjona se depositó en un ambiente de plataforma continental, mostrado por la presencia de sedimentos pelágicos y hemipelágicos con abundante concentración de material orgánico, de edad Miocena (Ingeominas, 2001).

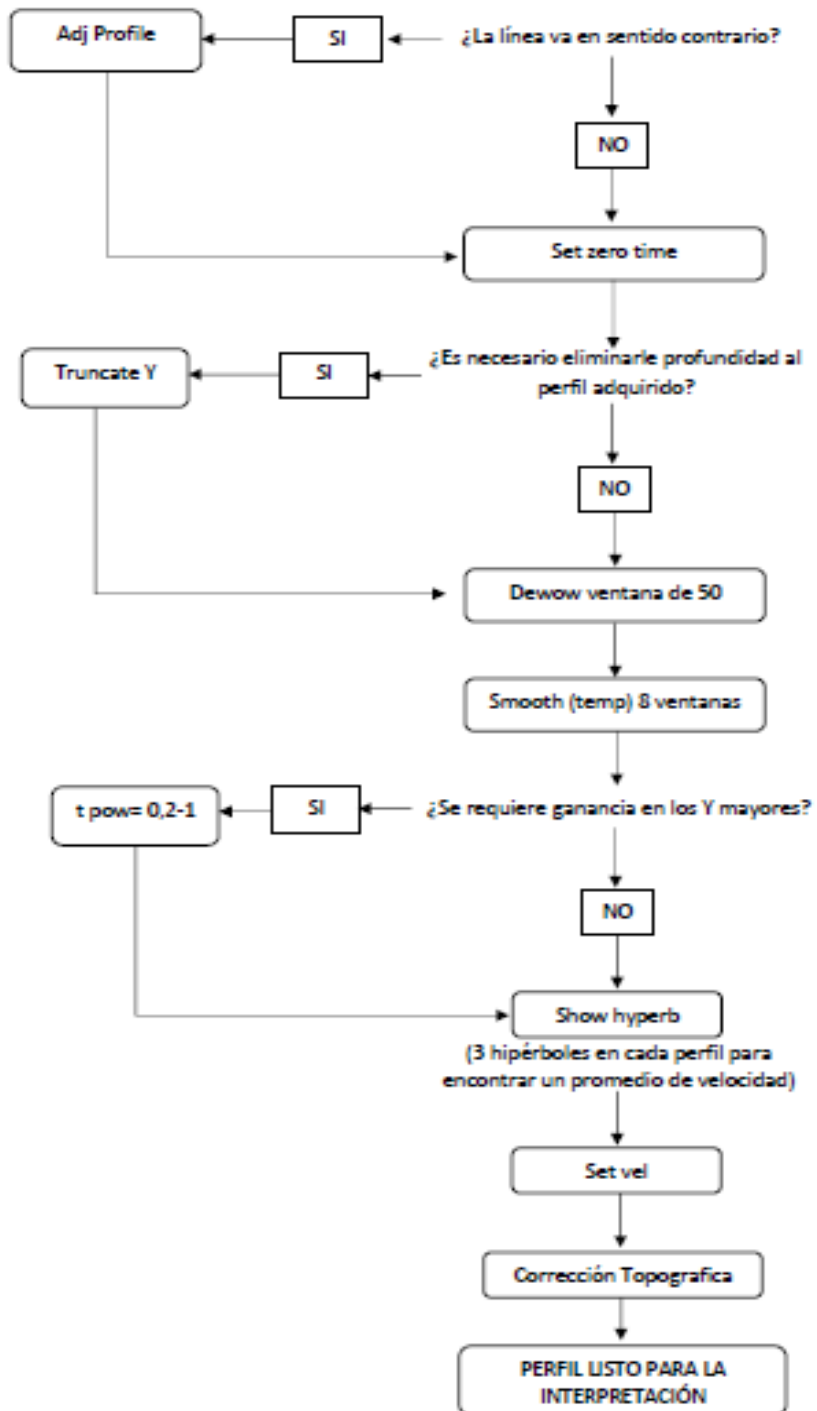
## **4 Procesamiento e Interpretación de Datos De GPR**

### **4.1 Caso de Estudio 1: Abanico aluvial de Mogotes, Santander**

En la figura 70 se observa el diagrama de flujo utilizado con las herramientas definidas como óptimas para el procesamiento de los datos crudos de GPR que se trabajaron. Este se diseñó con base a las herramientas ofrecidas por el programa GPRPy junto con la experimentación de procesamiento de la línea de estudio.

Figura 70.

Diagrama de flujo usado para el procesamiento de datos GPR.

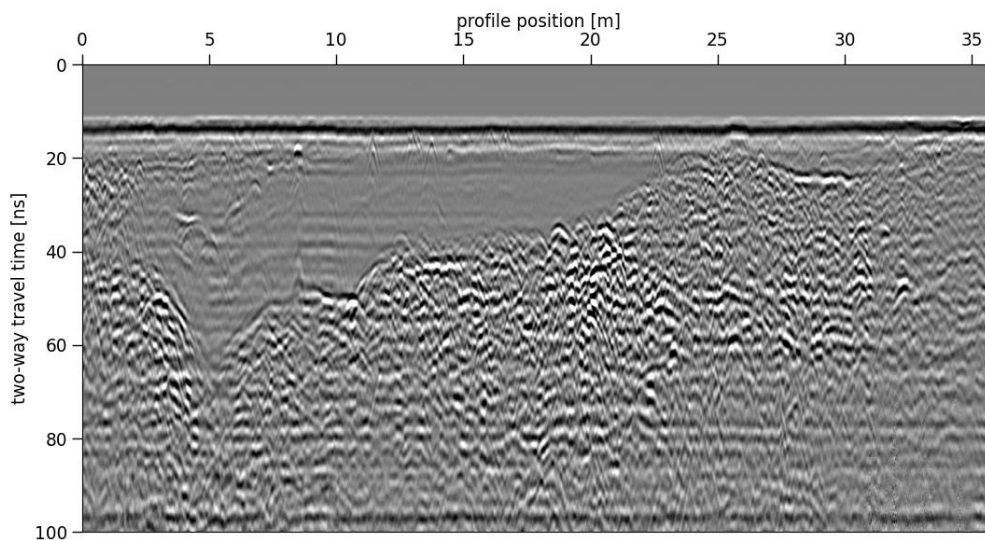


#### 4.1.1 Procesamiento

La línea de GPR inicial (Mogotes\_017.bzt) aparece en la figura 71. En ella se observa una profundidad en Two-Way-Time de 100 ns, un desplazamiento en la vertical del reflector inicial de 17 ns, y por lo menos 4 zonas con reflectores de diferentes amplitudes, e hipérbolas.

Figura 71.

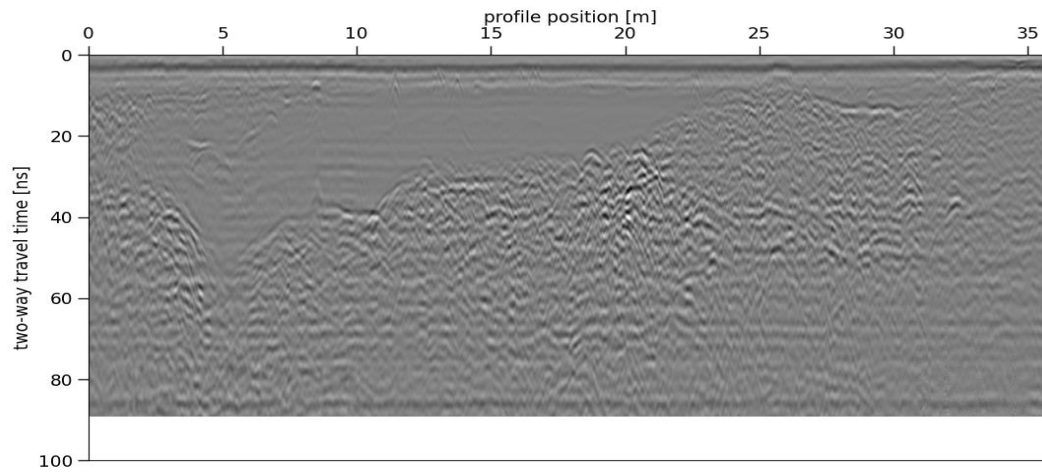
*Línea GPR sin procesamiento.*



Luego de confirmar la dirección de la línea ( en la bitácora se presenta la ubicación de los puntos inicial y final) se realiza la corrección de tiempo 0. Figura 72.

Figura 72.

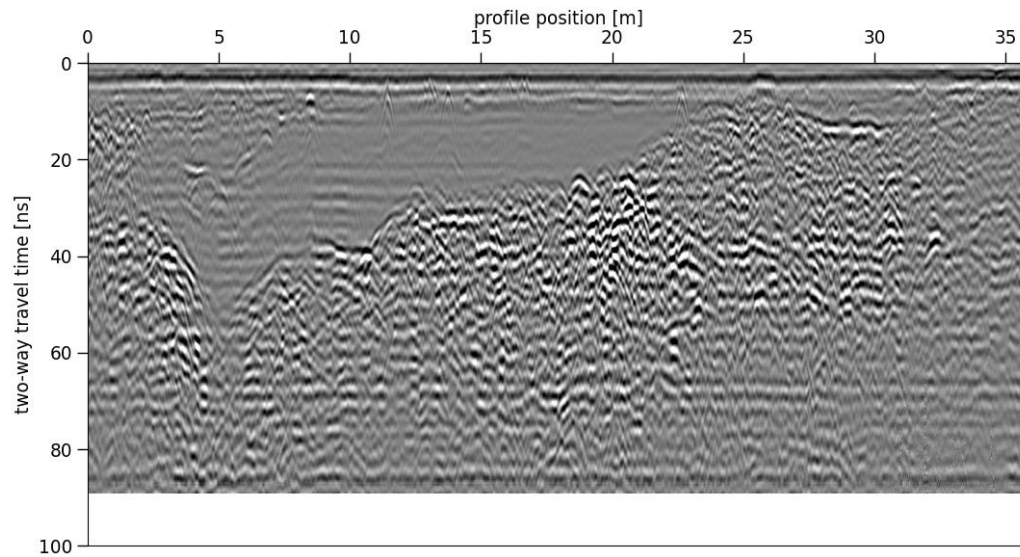
Línea GPR con la corrección de tiempo cero.



Posteriormente, se realiza el *Dewow*, usando una ventana de 25 trazas para eliminar la saturación de señal por llegada de ondas tempranas. Figura 73.

Figura 73.

Línea GPR con la aplicación del filtro Dewow.

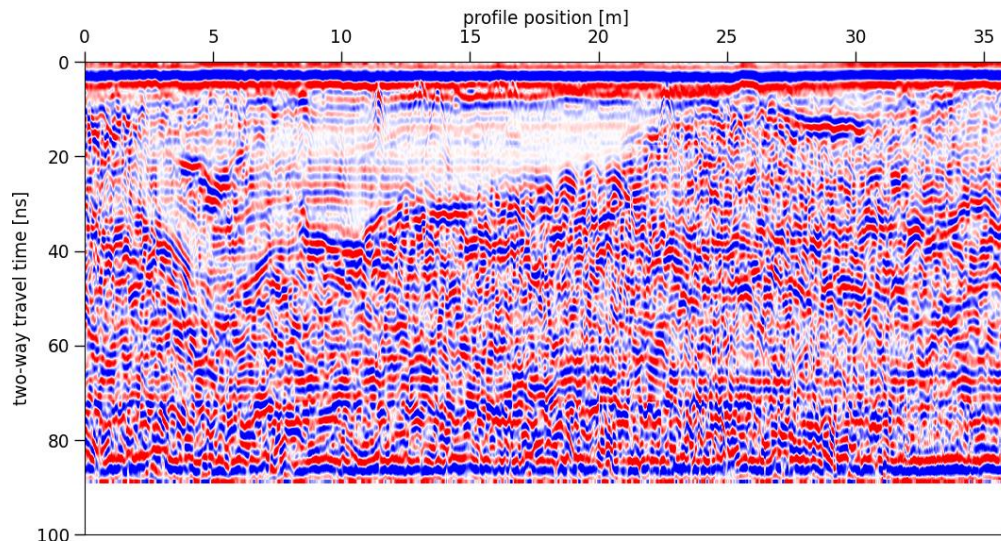


Adicionalmente se aplica el filtro de *Smooth* en una ventana de 7 trazas, y una ganancia AGC en una ventana de 180 trazas. La visualización de los resultados se realiza en una escala de colores azul-blanco-rojo (bwr). Figura 74.



Figura 74.

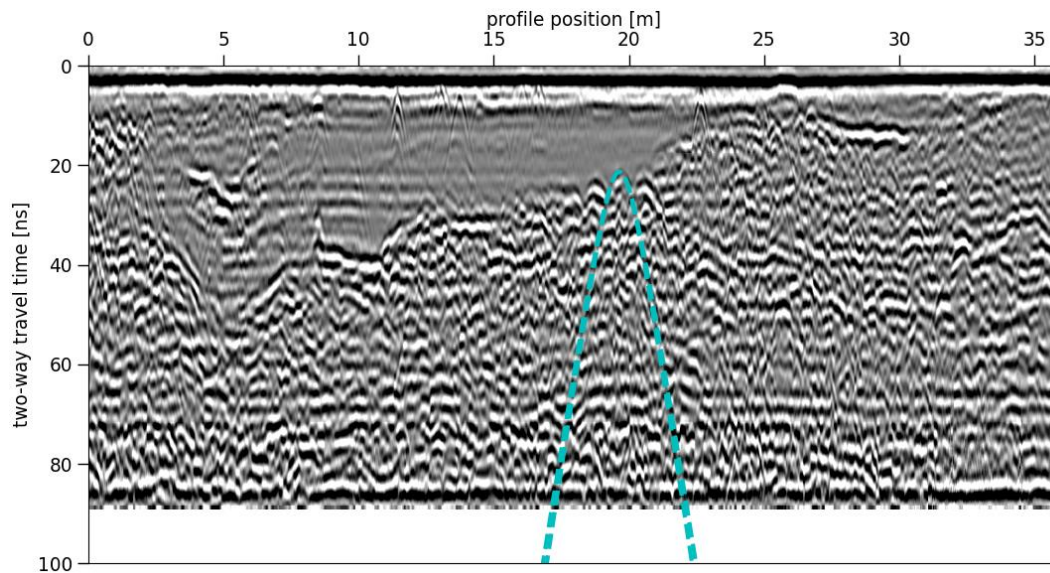
Línea GPR con la aplicación del filtro smooth y la herramienta AGC.



Luego, se determina la velocidad por medio del análisis de hipérbolas presentes en el radar-grama, hallando una velocidad de conversión de tiempo a profundidad de 0.058 m/ns. Figura 75.

Figura 75.

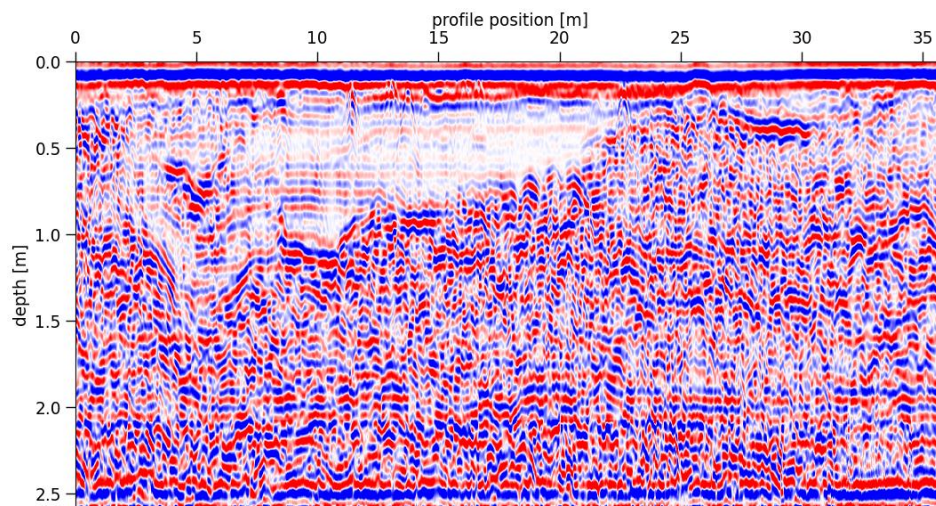
Línea GPR con la hipérbola seleccionada para determinar la velocidad.



Teniendo el valor de la velocidad de onda promedio se estima la profundidad que esta alrededor de 2.5 metros. Figura 76.

Figura 76.

*Línea GPR con el cambio de unidades de profundidad.*



Usando la comparación con la foto del afloramiento, se determina con la mejor calidad posible la ocurrencia de una discontinuidad que separa dos tendencias de reflectores.



## Litofacies

Figura 77.

Foto del afloramiento ilustrando las litofacies interpretadas.

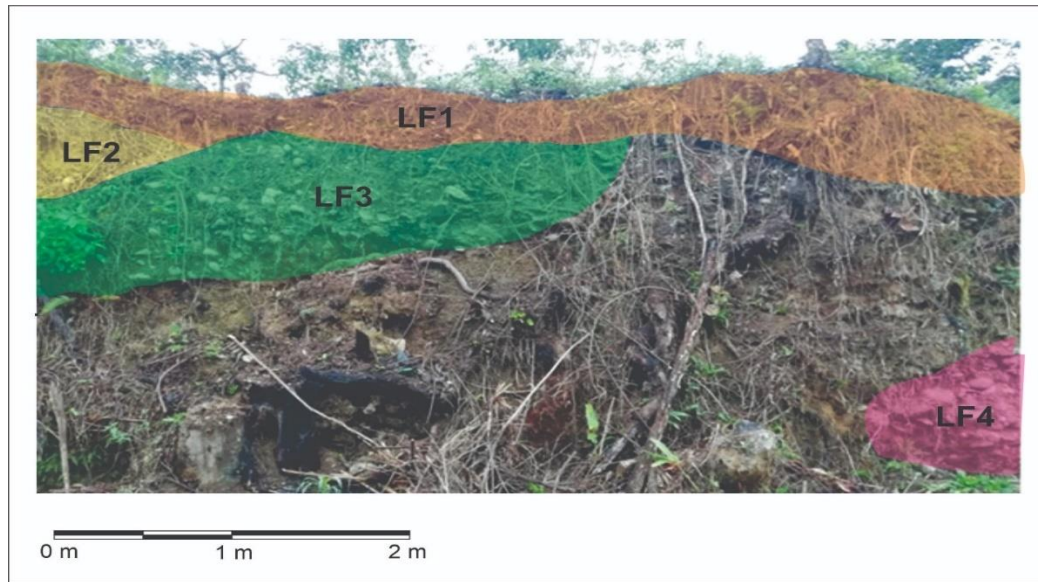


Figura 78.

Fotografía de afloramiento.



- LF1

Esta litofacie se compone de material de grano muy fino, cuenta con un espesor de aproximadamente 15 cm se encuentra a lo largo de todo el perfil adquirido con una probable alta presencia de material orgánico y sedimentos de granos finos.

- LF 2

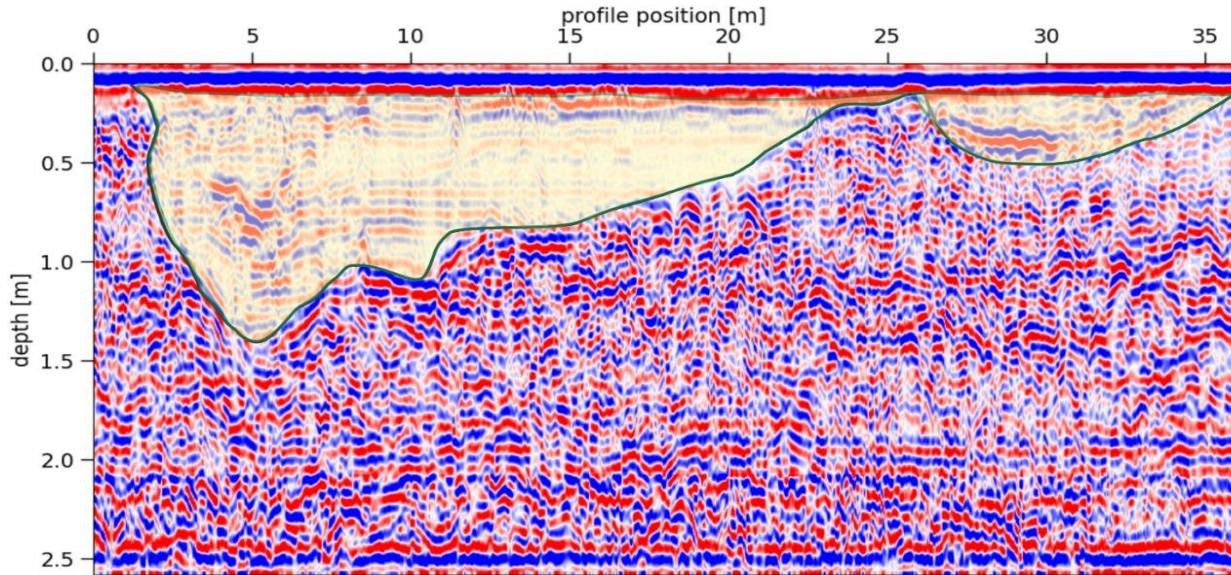
Litofacie compuesta de una material de finos de tonalidad rojiza posiblemente arenas.

- LF 3

Se definió esta litofacie como un conglomerado clastosoportado 40/60 de matriz/clastos respectivamente con aproximadamente un 60 % de cantos redondeados en su mayoría, embebidos en una matriz areno-arcillosa. El tamaño de los cantos varía desde 2 cm - hasta 20 cm.

- LF 4

Esta litofacie se compone de un conglomerado clasto soportado 20/80 matriz / clastos respectivamente, casi sin presencia de matriz, en su mayoría cantos imbricados, redondeados y solo aproximadamente el 10% de subangulares.

*Figura 79.**Línea GPR donde se evidencia la superficie de erosión.*

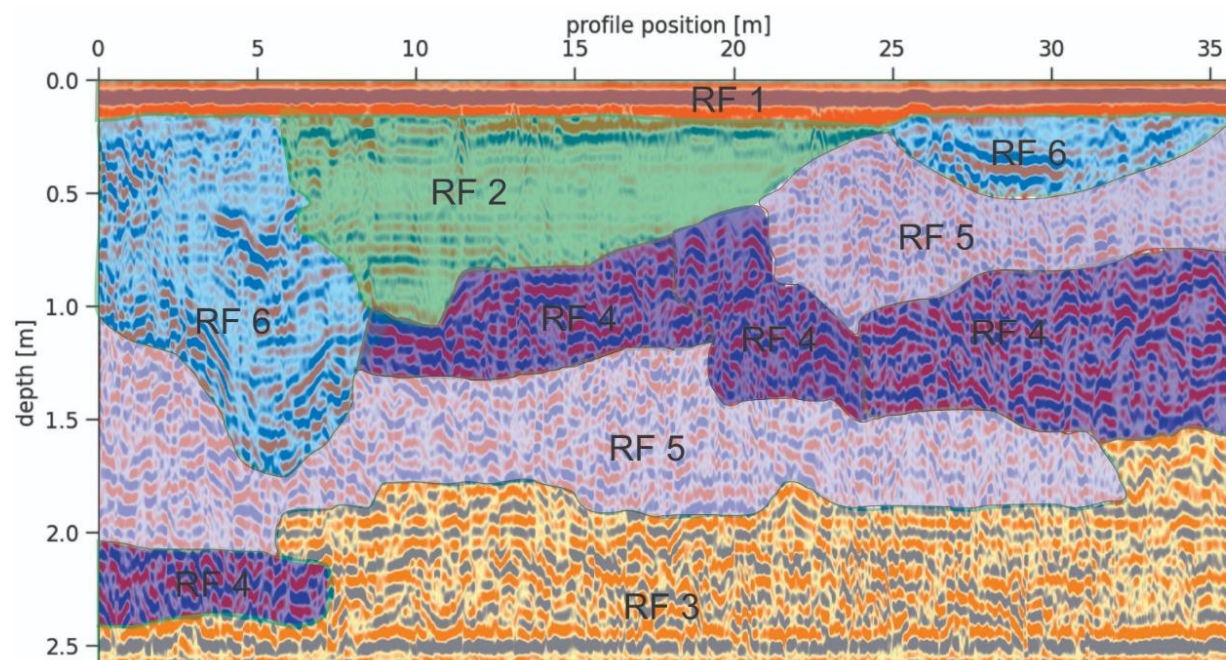
#### 4.1.2 Identificación y definición de Radarfacies

En el perfil GPR trabajado encontramos diferentes radar facies que definimos a partir de diferentes criterios como son la amplitud de los reflectores, el ángulo de inclinación, su continuidad y las superficies de erosión observadas. Figura 80.



Figura 80.

Radarfacies definidas por los intérpretes.



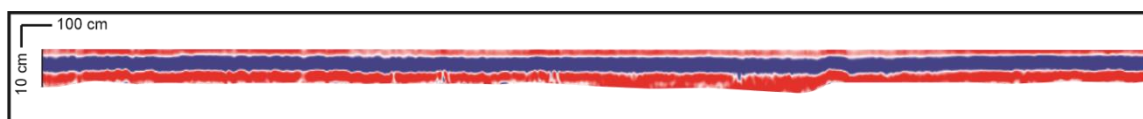
En total se describen siete (7) radar facies a continuación:

- Radar-facie 1 (RF 1)

Los reflectores son plano-paralelos, continuos y horizontales. Se encuentra a lo largo de todo el perfil y su espesor es de aproximadamente 0.2 metros.

Figura 81.

Radar facie 1.



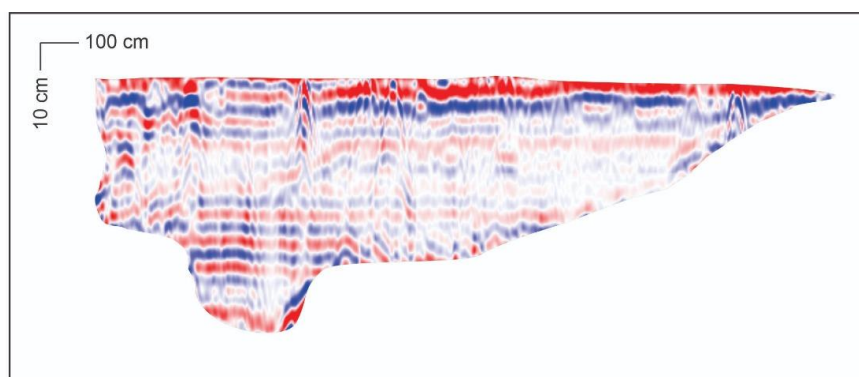
Potencial litología: LF1

### - Radar-facie 2 (RF 2)

En esta observamos la superficie de erosión marcada en la figura 79. Las reflexiones internas son paralelas a la base, inclinadas donde las amplitudes son bajas, y paralelas al tope. En esta radar-facie podemos evidenciar una amplitud media – baja.

*Figura 82.*

*Radar facie 2.*

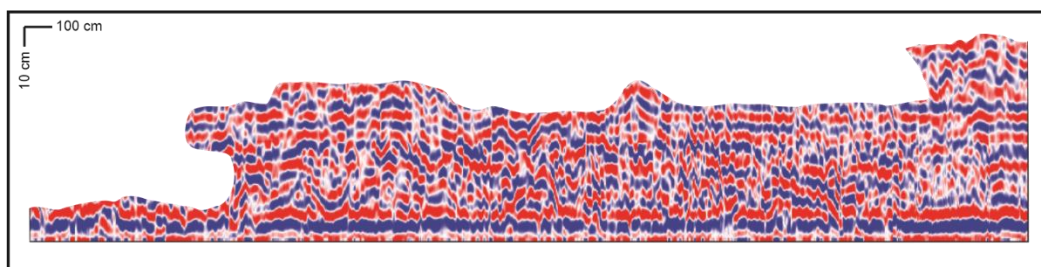


Potencial litología: LF2

- Radar-facie 3 (RF 3; Se define como una radar-facie de amplitudes medias a altas. En este caso, al tope y a la base se encuentran reflectores paralelos, mientras en la parte media los reflectores no son horizontales ni paralelos.

*Figura 83.*

*Radar facie 3.*



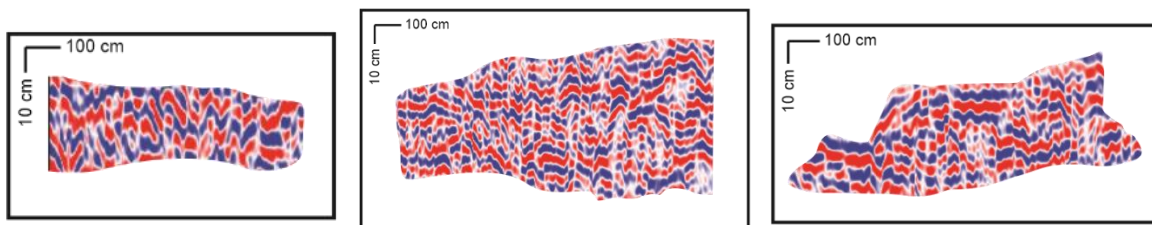
Potencial litología: LF3, LF4

- Radar-facie 4 (RF 4)

Se observan reflectores con tendencias de inclinación moderada de 5 a 9 grados y siendo estos paralelos, presentan continuidad moderada, ciertas zonas de esta radar-facie contienen amplitudes medias a altas formándose aspectos de ondículas.

*Figura 84.*

*Radar facie 4.*



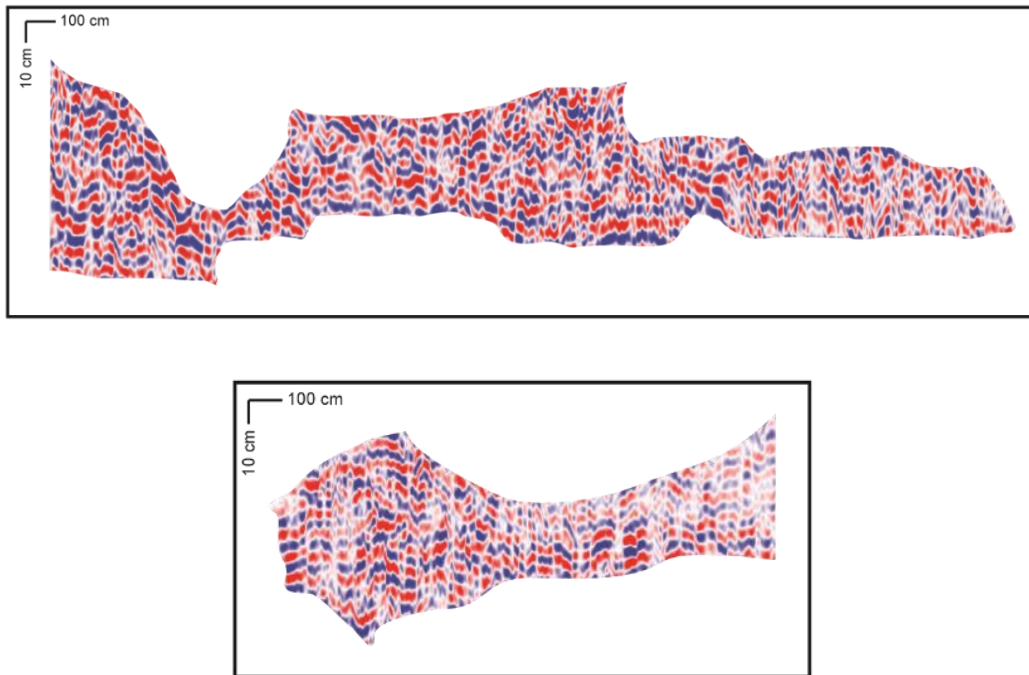
Potencial litología: LF3, LF4

- Radar-facie 5 (RF 5): Definimos esta radar-facie con una con una disposición sigmoidal compleja, en la que se encuentran amplitudes medias a bajas, en donde predominan las bajas. Estos reflectores son discontinuos y en ciertos lugares presentan cierto grado de inclinación.



Figura 85.

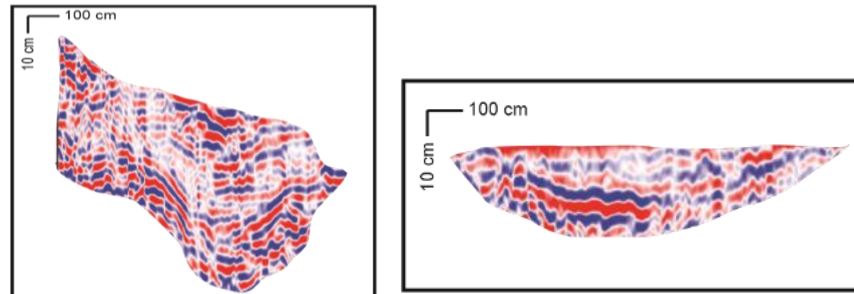
Radar facie 5.



Potencial litología: LF3, LF4

- Radar-facie 6 (RF 6)

Definida como un grupo de reflectores concordantes a la superficie de erosión causada por un canal de forma cóncava en la que presentan una continuidad con amplitudes medias y cierta inclinación en parte de estos.

*Figura 86.**Radar facie 6.*

Potencial litología: LF2

#### 4.1.3 Interpretación

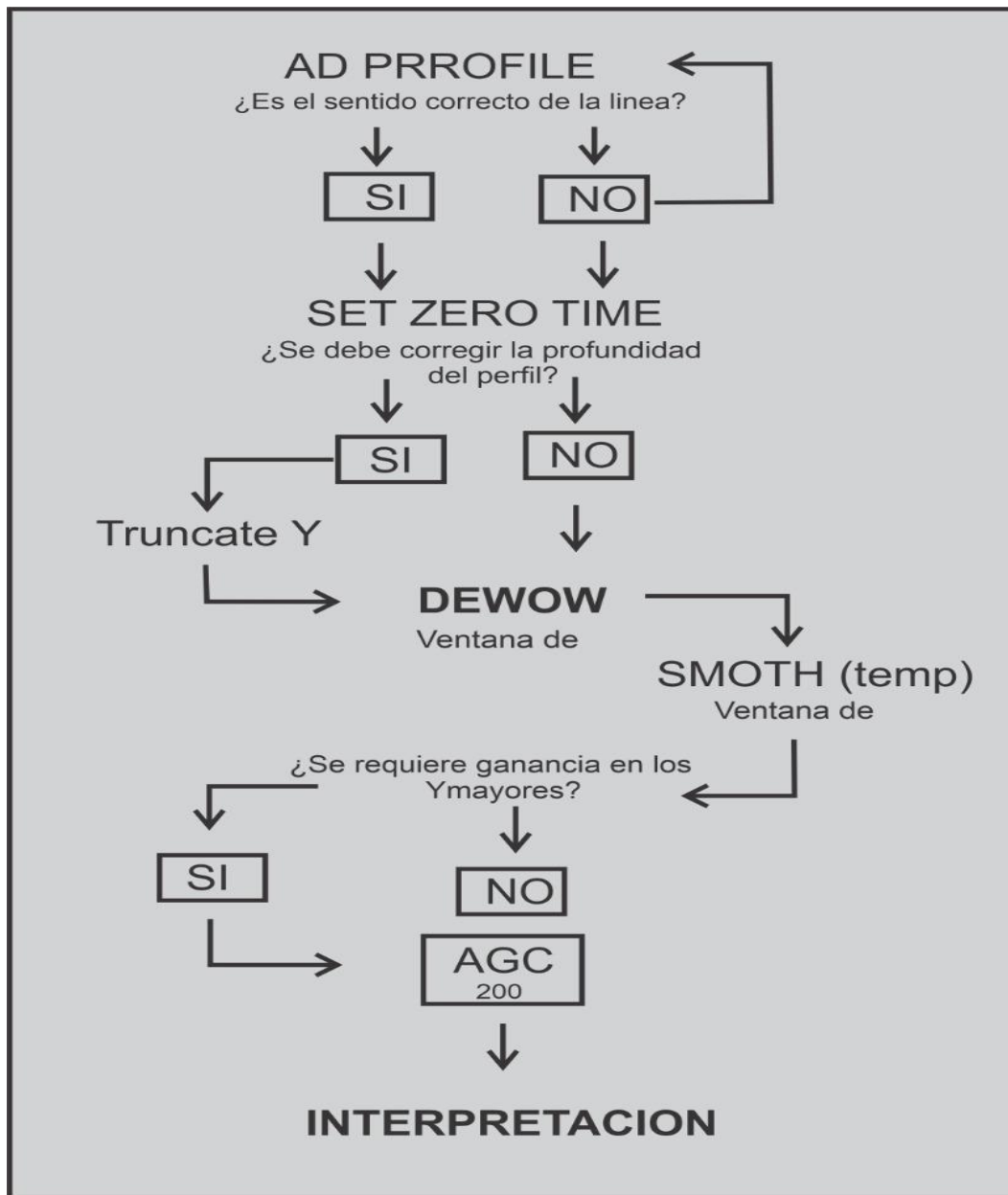
Teniendo en cuenta la poca diversidad litofacial presente en esta zona apical del abanico, en donde predominantemente se encuentran depósitos de conglomerados clastosoportados (LF3, LF4) con cantos altamente redondeados, lo que indica su larga duración y alta energía en el flujo que los transportó hasta el momento de depositarlos, ya que desplazó cantos de gran tamaño. Se observa una superficie de erosión evidenciada en una forma de canal, figura 79, que posteriormente se rellenó con sedimentos más finos (LF2) demostrando que se dio una depositación de baja energía, asemejan la superficie de erosión del canal.

#### 4.2 Caso de Estudio 2: Volcán de lodo en Bayunca, Cartagena

Al igual que con el caso de estudio 1, luego de hacer una revisión de la línea disponible, se propone un flujo de trabajo para realizar el procesamiento de la línea usando GPRPy. Figura 87.

Figura 87.

Diagrama de flujo para el procesamiento de los datos GPR.



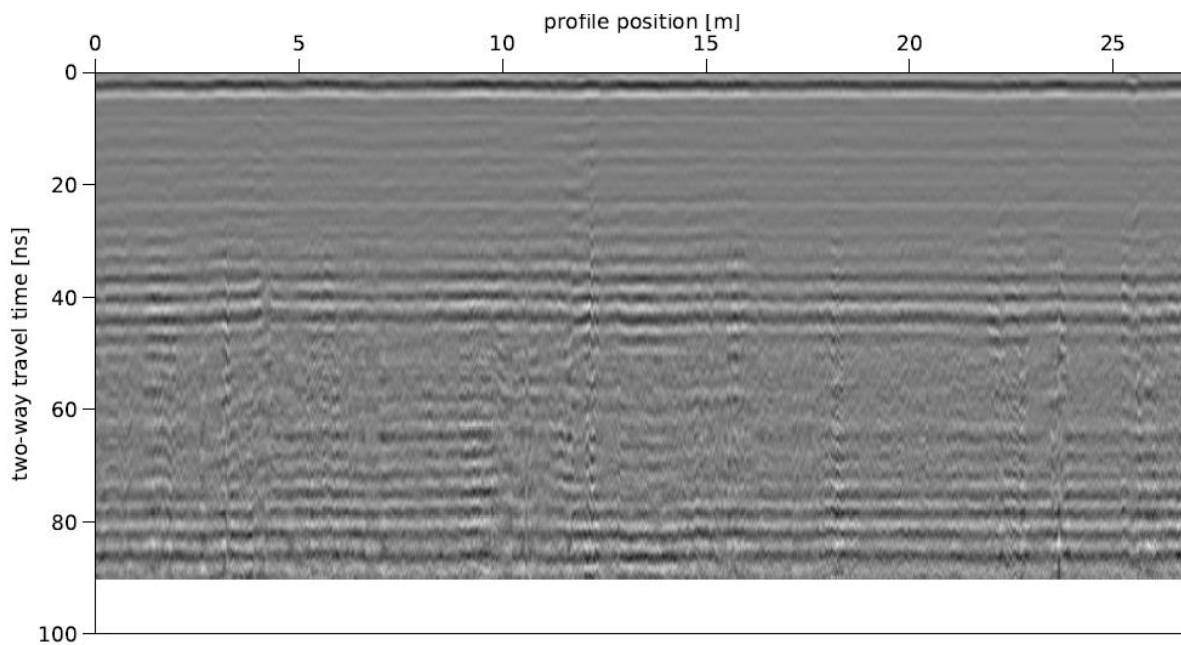
De 10 perfiles de GPR disponibles, se procede a demostrar el flujo de procesamiento e interpretación en la BAY\_002.bzt. El procesamiento de los datos consiste en una serie de pasos aplicados a cada uno de los perfiles de manera sistemática.

#### 4.2.1 Procesamiento

- Corrección de tiempo cero: La topografía de la zona en la longitud de la línea seleccionada es plana, por lo que la corrección sólo requiere ubicar el primer arribo, el tiempo de llegada de la onda aire/tierra. Figura 88.

*Figura 88.*

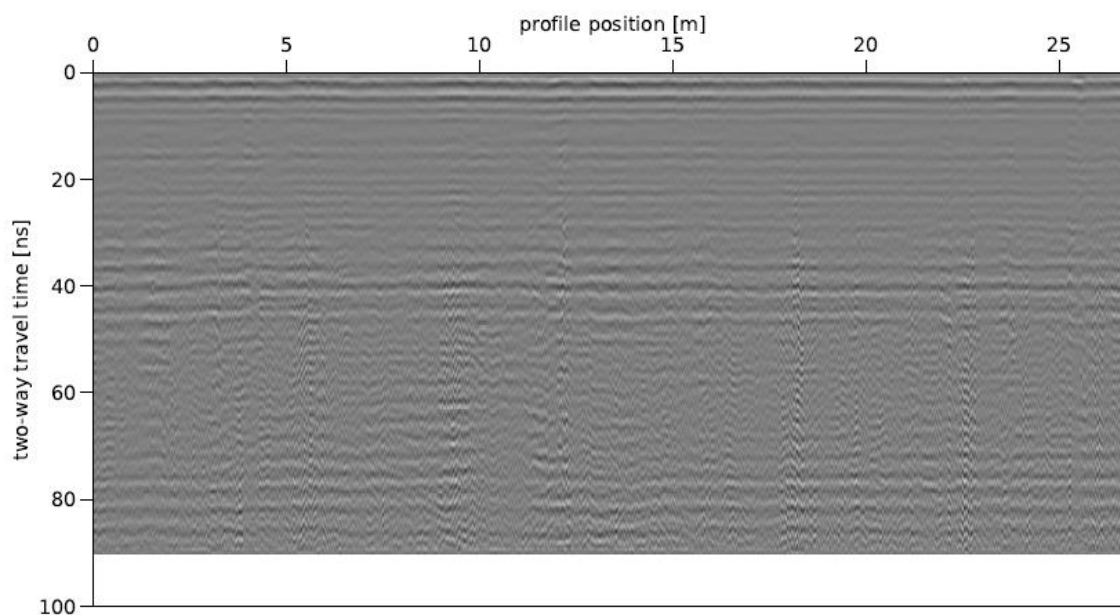
*Corrección tiempo cero.*



- Eliminación de la saturación de señal por llegadas tempranas, usando el Dewow, en una ventana con valor de 12. Figura 89.

Figura 89.

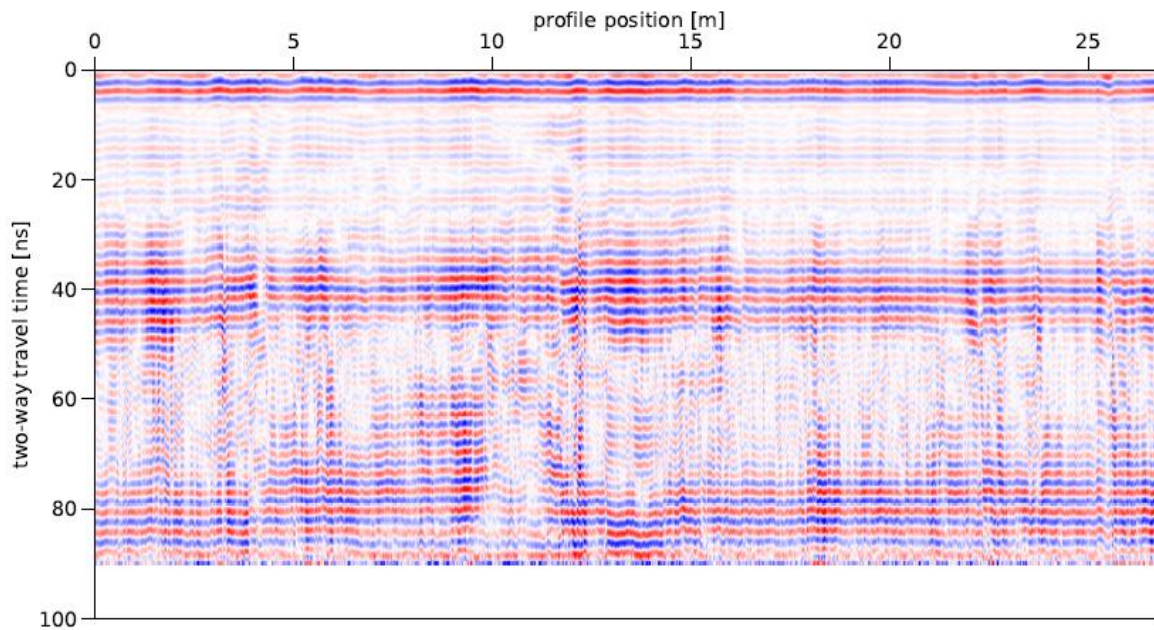
Línea de GPR con la aplicación del filtro Dewow.



- Eliminar y resaltar características de los perfiles mediante la aplicación de filtros como *Smooth* *temp* con ventana de 7 y se le aplica ganancia con filtro AGC con ventana de 200. Figura 90.

Figura 90.

Línea GPR con el filtro smooth.

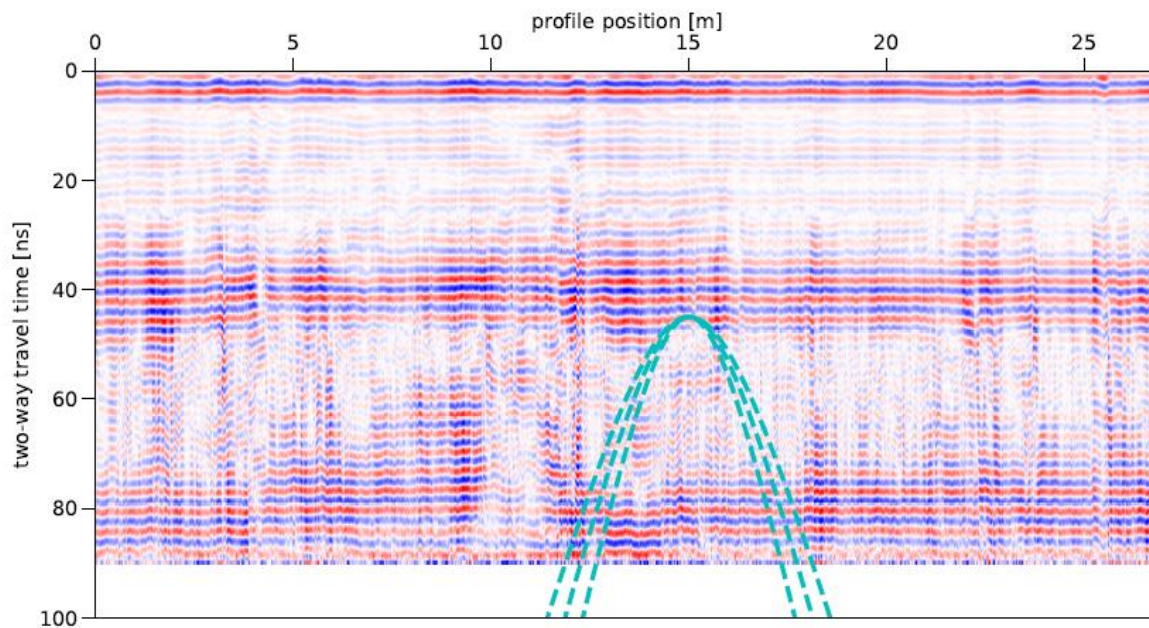


- Análisis de velocidad mediante el análisis de hipérbolas y la comparación de los reflectores obtenidos con los materiales observados en la sección velocidad de 0,08 ns /m. Figura 91.



Figura 91.

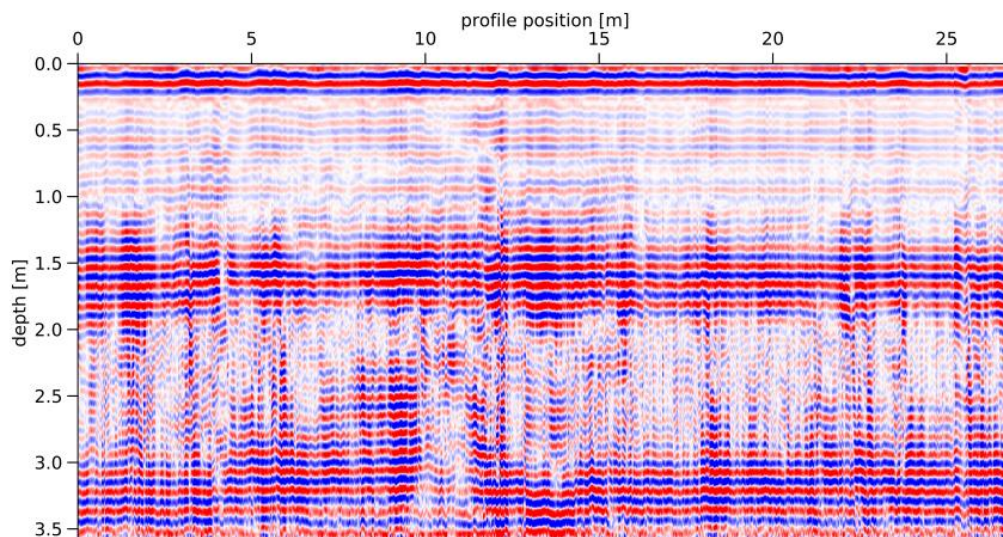
Análisis de velocidad mediante hipérbolas. (0.08 ns/m).



- Conversión a dominio de profundidad.

Figura 92.

Línea GPR con el cambio del dominio de profundidad.

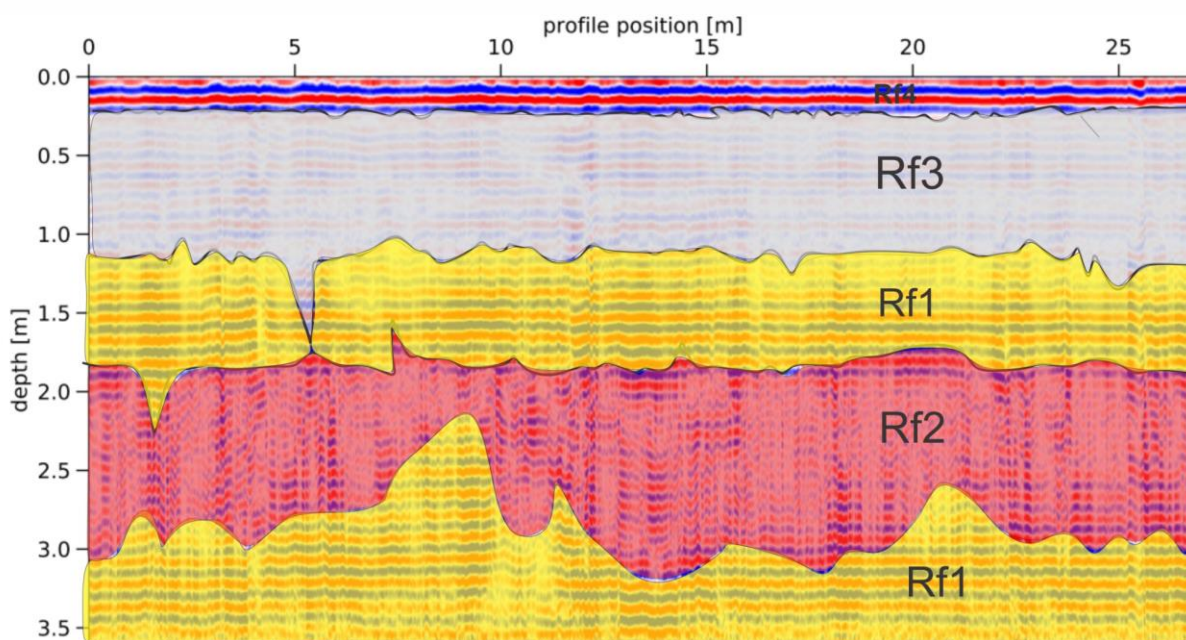


#### 4.2.2 Identificación y definición de radarfacies

Realizados los cálculos, cambios y la aplicación de filtros necesarios, se tiene como producto final el análisis de la siguiente figura guía de posteriores interpretaciones y conclusiones.

Figura 93.

*Perfil trabajado con las radar facies definidas por los intérpretes.*



En el perfil GPR Bayunca, se observan diferentes radar-facies que se definen, a partir de diferentes criterios como son la amplitud de los reflectores, el ángulo de inclinación, su continuidad y las superficies de erosión observadas. En total se definen cuatro (4) radar facies:

- Radar-facie 1 (RF 1)



Muestra reflectores de alta amplitud, horizontales a subpararelos, bien definidos, presentan un leve ángulo de inclinación y se encuentra una estructura de amplitud baja. Tiene un espesor aproximado de 0.8 metros en su parte inicial.

- Radar-facie 2 (RF 2)

Los reflectores son de amplitud baja, discontinuos, mayormente horizontales, con tendencia a ondularse, entre los 0 a 10 m en la horizontal los reflectores son inclinados . Posee un espesor aproximado de 1.5 metros.

- Radar-facie 3 (RF 3)

Esta radar facie presenta reflectores con amplitud baja, dispuestos de manera horizontal, un paquete se encuentra con una inclinación marcada hacia un extremo donde la tendencia de los reflectores es a levantarse, hacia la parte derecha está radar-facie presenta una amplitud media, siguiendo la horizontalidad. Presenta un espesor aproximado de 1 metro.

- Radar-Facie 4 (RF 4)

Se presenta de una manera plana-continua, completamente paralela y horizontal a lo largo del perfil. Es la radar-facie de menor espesor, aproximadamente 20 cm.

### **4.2.3 Interpretación**

En la zona de estudio, se presenta poca diversidad tanto litológica, como estructuralmente, debido a que se componen de lodos, los cuales han sido depositados por erupciones del volcán de lodo más cercano, las cuales, al estar alejadas en el tiempo, generan capas horizontales lodosas,

variando en espesor dependiendo tanto del tiempo de la erupción como la cantidad del material expulsado. A su vez cuando hay un tiempo prolongado entre cada evento del volcán, da condiciones propicias para que se produzca erosión de este, mostrando contactos entre capas horizontales, producto de la erosión, la cual tiende a aplanar las mismas; caso contrario a eventos rápidos y de corta duración entre cada evento eruptivo del volcán de lodo, generará contactos irregulares, no totalmente paralelos, como los es el contacto entre la RF-3 y la RF-4.

## **5 Conclusiones y Recomendaciones**

Las diferentes sesiones del seminario Alemán, desarrollado en la virtualidad, permitieron que los participantes conocieran los fundamentos de Python a través de la revisión de las nociones básicas de la programación e identificaran oportunidades de uso de estas herramientas para resolver problemas geológicos.

El GPR es un método geofísico de alta resolución y baja profundidad de estudio, que funciona con señales electromagnéticas de 100 MHz a 1000 MHz. En este rango de frecuencias la permitividad dieléctrica es la propiedad física determinante de la velocidad, y los coeficientes de reflexión y refracción de las ondas electromagnéticas. Las imágenes resultantes de las adquisiciones con zero offset son radargramas que muestran reflexiones, generadas en zonas de permitividad dieléctrica contrastante, en nuestro caso de estudio hasta 3 metros de profundidad.

La escala de estudio de este método puede llegar hasta los centímetros, por lo que geológicamente, corresponde al estudio de facies geológicas.

El procesamiento de las líneas de GPR, adquiridas en el modo de zero offset incluye el uso de filtros (dewow, smooth), la aplicación de correcciones topográficas y ganancias (ACG). La interpretación geológica de los resultados del procesamiento requiere la definición de radarfacies con sentido geológico para la escala de los radargramas (metros a centímetros). El escenario ideal de interpretación de datos de GPR contempla la descripción en detalle de las litofacies de un afloramiento donde existe radargrama.

### Referencias Bibliográficas

- Anderson, F. (1926). Original source of oil in Colombia. . *Am. Assoc. Petrol. Geol. Bull*, 10(4), 382- 404.
- Aristizábal, C., Ferrari, A., & Silva, C. (2009). Control neotectónico del diapirismo de lodo en la región de Cartagena, Colombia. *LA GEMAR*, 8, 42-50.
- Baker, G. S., Jordan, T. E., & Pardy, J. (2007). An introduction to ground penetrating radar (GPR). *Special Papers-Geological Society of America*, 1(432).
- Benedetto, A. T. (2017). An overview of ground-penetrating radar signal processing techniques for road inspections. *132*, 201-209.
- Boinet, T. B. (1985). *Age et repartition du magmatisme Premesozoïque des Andes de Colombie*. Comptes rendus hebdomadaires des séances de L'Académie des Sciences. Serie D: Sciences Naturales.
- Carvajal, J. (2011). Características del "volcanismo de lodo" del Caribe central colombiano. *Bogotá, Colombia: Servicio Geológico Colombiano*.
- Carvajal, J. M. (2010). Investigación del diapirismo de lodo y evolución costera del caribe colombiano. *Geomorfología Sector*, 1.
- Carvajal, J., & Mendivelso, D. (2017). Catálogo de volcanes de lodo del Caribe central colombiano., Bogotá, Colombia: Servicio Geológico Colombiano.

- Cassidy, N. J., & Jol, H. M. (2009). Ground penetrating radar: theory and applications. En N. J. Cassidy, & H. M. Jol, *Ground penetrating radar data processing, modelling and analysis*. Amsterdam: Elsevier.
- Chariguin, M. (1964). *Geología General*. Barcelona, España.
- Christofferson, E. (1984). Structure and magnetism of the Central Caribbean sea floor. *Research and Proposal, Department of Geology and Geography, Howard University, Washington*.
- Correa, e. a. (2016). *Catalogo de unidades litoestratigraficas de Colombia*. Medellin: SGC.
- Davis, J. L., & Annan, A. P. (1989). Ground-penetrating radar for high-resolution mapping of soil and rock stratigraphy 1. *Geophysical prospecting*, 37(5), 531-551.
- Duque-Caro, H. (1979). Major Structural Elements and Evolution of Northwestern Colombia: Small Basin Margins. *Colombia: Geological and Geophysical Investigations of Continental*.
- E. C. Utsi. (2017). *Ground Penetrating Radar: Theory and Practice*. Oxford, United Kingdom: Publishing.
- Fano, W. G. (2019). *The Electrical Properties of Soils with Their Applications to Agriculture and Engineering*. In Electromagnetic Fiel Radiation in Matter. IntechOpen.
- Franke, D., Hornung, J., & Hinderer, M. (2015). A combined study of radar facies, lithofacies and three-dimensional architecture of an alpine alluvial fan (Illgraben fan, Switzerland). *Sedimentology*, 62(1), 57-86.

Germeraad, J., Hopping, C., & Muller, J. (1968). Palynology of Tertiary sediments from tropical áreas,. *Rev. Palaebota. Palynol.*, 6, 3-4.

Glover, P. W. (2015). Geophysical Properties of the Near Surface Earth: Electrical Properties. *Treatise in geophysics.*, 89-137.

Gómez, A. C., Consuegra, N. P., & Ochoa, F. L. (2015). Levantamiento de la Cordillera Oriental de los Andes colombianos. *Hipótesis: Apuntes Científicos Uniandinos*, 19, 66-73.

González Duque, R. (2014). *Python para todos*. Creative Commons Reconocimiento, .

Hayes, D. C., & Drummond, F. E. (1995). Use of Fathometers and Electrical Conductivity Probes to Monitor Riverbed Scour at Bridge Piers: U.S. *Geological Survey Water-Resource Investigation Report*, 94-4164.

Ingeominas. (2001). Memoria Explicativa, Geología de las Planchas 23 y 29-30 Arjona. Colombia.

Izargain, J. J. (2004). *Técnicas de procesamiento de datos de georradar y su aplicación al estudio del régimen termodinámico de los glaciares fríos y politérmicos*. Universidad Politécnica de Madrid: Doctoral dissertation.

Jerez, D., & Rodriguez, K. (2018). *Caracterización mineralógica y génesis de las venas hidrotermales presentes en el Batolito de Mogotes (Vereda las Flores, Mogotes, Santander)*. Bucaramanga: Universidad Industrial de Santander.

Kopf, A. (2002). Significance of mud volcanism. *Rev. Geophys*, 40(2).

Lehmann, F., & Green, A. G. (2000). Topographic migration of georadar data: Implications for acquisition and processing. *Geophysics*, 65(3), 836-848.

- Lucena, S. (2002). *Esquema de ordenamiento territorial del municipio de Mogotes*. Mogotes.
- Mantilla, L., & García, C. (2018). *Geología y geocronología de las litologías aflorantes en el tramo Mogotes-San Joaquín (sector SW del Macizo de Santander)*. (Vol. 40). Bucaramanga: Boletín de Geología UIS.
- Maruddani, B., & Sandi, E. (2019). The Development of Ground Penetrating Radar (GPR) Data Processing. *International Journal of Machine Learning and Computing*, 9(6).
- O'Brien, G. (1968). Survey of diapirs and diapirics: Diapirism and diapirs: American Association of Petroleum geologist. *Memoir 8, p 1 – 9. Tulsa, USA*.
- P., Annan. (s.f.). *recuperado de*. Obtenido de <https://www.sensoft.ca/es/blog/what-is-gpr/>
- Pineda, W. (2017). *Caracterización mineralógica, petrográfica de las pizarras de la formación floresta metamorfoseada (FFM), aflorante entre los municipios de Mogotes y San Joaquín (Cordillera Oriental, Colombia)*. Bucaramanga: UIS.
- Plattner, A. M. (2020). GPRPy: Open-source ground-penetrating radar processing and visualization software. *The Leading Edge*, 39(5), 332-337.
- Quintero, J. (2012). Interpretación sísmica de volcanes de lodo en la zona occidental del abanico del delta del Río Magdalena, Caribe colombiano. *Medellín, Colombia: Universidad EAFIT*.
- Reyes, G., Guzman, G., Barbosa, G., & Zapata, G. (2001). Geología de las Planchas 23 Cartagena y 29 - 30 Arjona Escala 1:100.000. Memoria Explicativa. *Ingeominas*.

- Roberts, M. C., Niller, H. P., & Helmstetter, N. (2003). Sedimentary architecture and radar facies of a fan delta, Cypress Creek, West Vancouver, British Columbia. *Geological Society, London, Special Publications*, 211(1), 111-126.
- Rosello, E., & Cossey, S. (2012). What is the evidence for subduction in the Caribbean Margin of Colombia? In: Extended abstract in Memoirs of the XII Simposio Bolivariano de Cuencas Subandinas: Cartagena, Colombia. *Asociación Colombiana de Geólogos y Geofísicos del Petróleo*.
- Royero, J., & Clavijo, J. (2001). *Mapa geológico generalizado departamento de Santander. Memoria explicativa*. : INGEOMINAS.
- Sanchez Pinto, J. K. (2020). *Determinación del espesor del subsuelo inconsolidado en el abanico aluvial de Mogotes Santander utilizando sísmica de refracción. Tesis de Maestría en Geofísica, Universidad Industrial de Santander*. Bucaramanga.
- Szymczyk, M., & Szymczyk, P. (2013). Preprocessing of GPR data. *Image Processing & Communications*, 18(2-3), 83-90.
- Universidad de Vigo. (2012). *Polarización eléctrica*. Vigo: Física 2.
- Ward D., G. R. (1973). Geología de los cuadrángulos H-12 Bucaramanga y H-13 Pamplona, Departamento de Santander. (B. G. XXI, Ed.) *Ingeominas*, 1-132.
- Webster, J. (2003). Electrical measurement, signal processing, and displays. Palo Alto Research Center.



Wentworth, P., Elkner, J., Downey, A. B., & Meyer, C. (2020). *How to think like a computer scientist: Learning with Python 3*.

Yaw, K. C. (2012). Measurement of dielectric material properties. . *Application Note. Rohde & Schwarz*, 1-35.

Yilmaz, Ö. (2001). *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists.

Zhdanov, M. (2018). *Foundations of Geophysical Electromagnetic Theory and Methods* (Vol. 2). USA: Elsevier.

Zhdanov, M. (2018). *Foundations of Geophysical Electromagnetic Theory and Methods* (Segunda ed.). Elsevier.