



## TESINA DE LICENCIATURA

**Título:** Desarrollo de una interfaz de visualización de datos de radar

**Autores:** Primost, Esteban Joaquín

**Director:** Luengo, Miguel

**Codirector:** --

**Asesor profesional:** --

**Carrera:** Licenciatura en Sistemas

### Resumen

*El radar como dispositivo, es una herramienta de trabajo muy importante, que ha sido explotada a lo largo del tiempo en diferentes ramas como la industria militar, la aeronavegación, la astronomía y la meteorología, entre tantas otras. La agricultura es una de las principales áreas de explotación económica de Argentina, fuertemente relacionada a las condiciones meteorológicas actuales como así también a los estudios realizados sobre los datos arrojados por los sistemas de radar, que permiten tomar decisiones a corto y largo plazo, en especial en situaciones climáticas desfavorables para el desarrollo agropecuario.*

*La red de radares meteorológicos del Instituto Nacional de Tecnología Agropecuaria (INTA), cuenta con la posibilidad de presentar los resultados de las actividades diarias de escaneo y procesamiento a través de aplicaciones y formatos propietarios. Como alternativa se propuso desarrollar una herramienta capaz de obtener diferentes tipos de productos meteorológicos, como reflectividad y velocidad, combinados con múltiples tipos de presentaciones visuales en donde el desarrollo como el prototipo propuesto para esta tesina, utilicen software libre.*

*Para lograr este objetivo se investigó acerca de la utilización e implementación actual de los sistemas de radar de INTA, utilizando información provista por el Instituto de Clima y Agua de dicha institución.*

### Palabras Claves

RADAR, INTA, METEOROLOGÍA, AGRICULTURA, CLIMA, VISUALIZACIÓN DE DATOS, PROCESAMIENTO, ESCANEO, REFLECTIVIDAD, VELOCIDAD, DOPPLER.

### Conclusiones

*La utilización de software propietario motiva a desarrollar alternativas que, en principio, cumplan con la misma funcionalidad pero además puedan extender las capacidades de este software. Éste fue el caso en esta tesina.*

*Se investigó el funcionamiento de los radares, en especial en la mecánica de escaneo, procesamiento y presentación de datos, por parte del software actual.*

*Se logró desarrollar una herramienta capaz de: interactuar con datos de escaneo de radar, procesar datos en bruto, con formato propietario, aplicar las correcciones necesarias y presentar los productos de escaneo en diferentes formatos y visualización.*

### Trabajos Realizados

*Investigación acerca del funcionamiento básico de los radares meteorológicos utilizados actualmente en INTA.*

*Documentación de los resultados obtenidos en la investigación anterior.*

*Desarrollo de un prototipo para la lectura, procesamiento y presentación de los datos de radar, "Meteorax", como resultado del trabajo realizado en esta tesina.*

### Trabajos Futuros

*Extender el prototipo construido, agregando más formatos/productos de exportación.*

*Permitir el análisis a partir de otros tipos de archivos.*

*Vistas en 3D.*

*Sitio online para el procesamiento de archivos.*

*Conexión directa con la estación de radar*



Universidad Nacional de La Plata

Facultad de Informática

Tesina de Licenciatura

Desarrollo de una interfaz de  
visualización de datos de radar

**Primost, Esteban Joaquín**

Director: Luengo, Miguel

Asesores profesionales: Lic. Banchemo, Santiago y  
Lic. Mezher, Romina Nahir

Septiembre 2019

## Agradecimientos

A mis viejos, **Mabel** y **Daniel**, que me enseñaron a caminar.

A mis hermanas, **Iva**, **Moni** y **Jeza**, que me marcaron el camino.

A mi novia, **Lu**, que caminó conmigo.

A mis **amigos**, que me alentaron en todo el recorrido.

A mi director, **Miguel**, que confió en la propuesta y trabajó conmigo para construir esta tesina.

A la **Universidad Nacional de La Plata** y a la **Facultad de Informática** por brindarme el soporte para lograr este objetivo.

# Índice

<b>1. Introducción</b>	<b>5</b>
<b>2. Radar</b>	<b>7</b>
2.1. Historia . . . . .	7
2.2. Funcionamiento . . . . .	11
2.2.1. Ciclo de escaneo . . . . .	12
2.2.2. Reflectividad y dispersión . . . . .	13
2.2.3. Bandas de frecuencias . . . . .	14
2.2.4. Efecto Doppler . . . . .	15
2.2.5. Polarización . . . . .	17
2.3. Componentes . . . . .	18
2.3.1. Transmisor . . . . .	19
2.3.2. Modulador . . . . .	20
2.3.3. Receptor . . . . .	20
2.3.4. Antena y reflector . . . . .	20
2.3.5. Duplexador . . . . .	21
2.3.6. Guía de onda . . . . .	22
2.3.7. Computadora . . . . .	22
2.3.8. Display . . . . .	23
2.4. Radares utilizados en INTA . . . . .	27
2.4.1. Software . . . . .	29
<b>3. Obtención de los datos</b>	<b>32</b>
3.1. Escaneo . . . . .	33
3.1.1. Movimientos de la antena . . . . .	33
3.1.2. Unidades y medidas . . . . .	34
3.1.3. Geometrías de escaneo . . . . .	35
3.1.4. Configuración actual . . . . .	37
3.2. Almacenamiento y acceso . . . . .	38

<b>4. Procesamiento</b>	<b>40</b>
4.1. Archivos de registro	40
4.1.1. Registro Rainbow	40
4.1.2. Registro <b>BITE</b>	41
4.1.3. Registro <b>RainAdmin</b>	41
4.1.4. Registro <b>RainRCS (Rainbow® Control Center)</b>	41
4.2. Archivos de resultados	41
4.2.1. Sección XML	42
4.2.2. Sección de datos en binario	42
4.2.3. Archivos de datos de producto	44
4.2.4. Archivos de datos en bruto	51
4.3. Tipos de archivos utilizados	59
4.4. Lectura	60
4.4.1. Utilización de librerías	60
<b>5. Visualización: aplicación Meteorax</b>	<b>63</b>
5.1. Herramientas utilizadas	63
5.1.1. Lenguaje: Python	63
5.1.2. Py-ART	64
5.1.3. Wradlib	69
5.1.4. GTK+	71
5.2. Variables procesadas	73
5.3. Formatos de exportación	77
5.3.1. GeoTIFF	77
5.3.2. PNG	79
5.3.3. ASCII	81
5.3.4. Productos de radar	82
5.4. Aplicación	83
5.4.1. Ejecución	83
5.4.2. Inicio: selección de archivo	83

<i>ÍNDICE</i>	4
5.4.3. Información de escaneo . . . . .	83
5.4.4. Productos: PPI . . . . .	86
5.4.5. Productos: RHI . . . . .	88
5.4.6. Productos: GeoTIFF . . . . .	90
5.4.7. Productos: ASCII . . . . .	91
<b>6. Conclusión</b>	<b>95</b>
<b>7. Trabajos relacionados</b>	<b>97</b>
<b>8. Trabajos a futuro</b>	<b>98</b>
<b>9. Referencias</b>	<b>100</b>

## 1. Introducción

La tesina se ha organizado de la siguiente manera:

**Capítulo 1:** Introducción. Motivación y contexto de esta tesina.

**Capítulo 2:** Radar, definiciones, historia, componentes y funcionamiento. Radares utilizados en INTA.

**Capítulo 3:** Esquema de flujo de los datos. Proceso inicial de obtención. Características de los escaneos. Configuraciones. Almacenamiento y acceso a los datos generados.

**Capítulo 4:** Etapa de procesamiento de los datos. Archivos de registro. Archivos de resultados. Tipos de datos de entrada y análisis de formatos. Proceso de lectura de los datos. Introducción a las librerías que intervienen.

**Capítulo 5:** Etapa de visualización y presentación de los datos. Herramientas utilizadas en el desarrollo. Variables, definición. Formatos de exportación de los datos. Aplicación como resultado de desarrollo: **Meteorax**, archivos, productos de salida, detalles y ejemplos de utilización.

**Capítulo 6:** Conclusión acerca del desarrollo de esta tesina. Dificultades, aprendizaje y experiencia personal.

**Capítulo 7:** Mención de colaboradores y referencias a los trabajos relacionados con la obtención y/o la presentación de datos de radar.

**Capítulo 8:** Posibles trabajos a futuro a partir de la presente tesina.

**Capítulo 9:** Referencias a los recursos consultados para elaborar esta tesina.

La palabra “**radar**” proviene del acrónimo **Radio Detection And Ranging**, en inglés, que significa: *Detección y medición por radio*. Es un dispositivo capaz de detectar, dimensionar y localizar objetos a su alrededor, incluso a varios kilómetros de distancia. Claro que no son sus únicas capacidades ya que además, por ejemplo, puede medir la velocidad de los objetivos que detecta, entre otras características. Su funcionamiento se basa en la utilización de ondas de radio y de varias propiedades físicas, como por ejemplo la reflectividad, las propiedades de las ondas y el efecto Doppler, entre otras.

El radar, como herramienta de detección y medición, ha sabido aplicarse en diversas áreas, como la **industria militar**, para la detección de naves/aviones enemigos y para el descubrimiento de campos minados, entre otros usos. Para el **transporte aéreo** el radar (en torres de control y sobre los mismos aviones) cumple varias funciones como la de sistema de alerta ante la cercanía de objetos o la de brindar indicaciones precisas de posición y condiciones climáticas y la de instrumento de ayuda para aterrizajes con niebla u otras condiciones climáticas adversas. Otros usos del radar incluyen: **astronomía**, **oceanografía**, **topografía** y la más importante para esta tesina: **meteorología**. En particular nos interesa su desarrollo en el sector agropecuario.

En Argentina, el **INTA** (Instituto Nacional de Tecnología Agropecuaria)

cuenta con tres radares meteorológicos, ubicados en Buenos Aires, Entre Ríos y La Pampa, los cuales constituyen una red de radares que cubre una superficie aproximada de  $46000 \text{ km}^2$ , mayormente en el área central de nuestro país. El proveedor de dichos radares es la empresa alemana **Selex ES GmbH**, que además de los instrumentos físicos, provee el software necesario para controlar y acceder a la información generada por los mismos, denominado **Rainbow® 5**. Este software es propietario junto con el formato de archivos que genera en donde se encuentra la información para el usuario. Es decir, el punto de acceso de los datos generados (y procesados) es a través de las aplicaciones propietarias de Rainbow® 5.

La motivación de esta tesina comienza con la característica anteriormente mencionada: **software y formato propietarios**, y es la de **desarrollar una herramienta abierta, interactiva, gráfica, de procesamiento y visualización de datos de radar**, accesible a cualquier tipo de usuario (no solo un meteorólogo) con interés en la meteorología. La necesidad de esta herramienta nace en el Instituto de Clima y Agua de INTA [1], más específicamente en su sede de Castelar, Buenos Aires.

Las tareas principales para lograr este objetivo, son *investigar* de qué manera procesar la información que generan los radares luego de cada escaneo y *desarrollar* una herramienta necesaria para procesar los datos de entrada y presentar los resultados.



## 2. Radar

Un radar es un sistema electrónico con el cual podemos **detectar, localizar y dimensionar objetivos**. Su nombre se deriva del acrónimo en inglés: *Radio Detection And Ranging*.

Su funcionamiento se basa en la utilización de ondas electromagnéticas que son enviadas al exterior y que impactan con un objetivo (una gota de lluvia, una partícula de granizo, un pájaro, un avión, etc.) produciendo cierta energía de respuesta. Parte de esta energía, es capturada de nuevo por el radar y posteriormente procesada junto con otras variables que deben ser tenidas en cuenta para determinar características del objetivo como tamaño, velocidad, ubicación y dirección.

Aviación, aeronáutica, control de tráfico, servicios militares y (el más importante para esta tesina) **meteorología**, son algunos de los sectores donde el radar cumple una función principal, ya sea detectando una precipitación cercana a un cultivo, localizando un avión para controlar su ruta, alertando a los ciudadanos de una posible situación de granizo en los próximos minutos o descubriendo una nave enemiga acercándose a una flota militar en operaciones marítimas.

Para los alcances de esta tesina se analizarán en detalle las características y el funcionamiento de un tipo muy común de radar: **el radar meteorológico**, cuyo objetivo es, justamente, brindar información acerca del estado actual de los elementos del clima.

### 2.1. Historia

A continuación se enumeran los sucesos históricos más importantes y los descubrimientos logrados, ligados a los conceptos de radiotransmisión, ondas electromagnéticas, localización y detección de blancos, entre otros, para concluir en una versión funcional de un radar:

- 1865 • El físico escocés James Clerk Maxwell presentó su “**Teoría dinámica del campo electromagnético**” [2].
- 1886 • Heinrich Rudolf Hertz, físico alemán, consigue demostrar experimentalmente la existencia de ondas electromagnéticas, concepto introducido previamente en las ecuaciones de Maxwell [3].
- 1897 • Por primera vez se logra una transmisión de ondas electromagnéticas a larga distancia. Dicho acontecimiento se le atribuyó al ingeniero e inventor Guglielmo Marconi, que luego de varios intentos y tras no haber recibido apoyo del gobierno italiano, se muda a Inglaterra para posteriormente lograr transmitir, un 13 de mayo de 1897 la primer señal inalámbrica sobre el agua, a través del canal de Bristol desde la isla Flat Holm a Lavernock [4][5].

- 1900 • Basado en sus investigaciones, Nikola Tesla, físico, ingeniero e inventor, sugiere que el reflejo de las ondas electromagnéticas puede usarse no sólo para detectar, sino también para medir objetos distantes.
- 1904 • El 30 de abril de ese año, Christian Hülsmeyer presenta la patente del “Telemobiloscopio” [6] (Del Griego: *τηλε* (tele) = “lejos”, del Latín: *mobilis* = “movible” y del Griego: *σκοπειν* (skopein) = “para ver, examinar”) en Düsseldorf, Alemania. Este sistema emisor-receptor, basado en el uso de ondas eléctricas, fue pensado originalmente para evitar colisiones entre barcos, pero su funcionalidad se extendía para la detección otros objetos metálicos, como trenes.
- 1921 • Albert Wallace Hull, físico e ingeniero estadounidense, consolida el primer “magnetron” funcional, gracias a las pruebas realizadas en el mismo año por el físico sueco Heinrich Greinacher. Este dispositivo, un tubo de vacío de alta potencia, funciona como oscilador de microondas auto-excitado. Los electrones cruzados y los campos magnéticos son utilizados en el magnetron para producir la salida de alta potencia requerida en los equipos de radar.
- 1922 • En septiembre de este año, los ingenieros Albert Hoyt Taylor y Leo Clifford Young, se encontraban realizando experimentos para el laboratorio de investigación de las fuerzas navales de los Estados Unidos, a orillas del río Potomac, en la ciudad de Anacostia, Washington. Al realizar uno de sus experimentos, que consistía en transmitir señales de radio desde la costa de Anacostia hacia Haines Point atravesando así el río en su amplitud, se dieron cuenta de que estas señales se veían interferidas por los barcos de madera que navegaban por el canal. El experimento demostró la primera detección no ambigua de señales bio estáticas. Posteriormente, en 1934, comprobaron el mismo síntoma de interferencia entre las señales de radio pero esta vez con aeronaves. A raíz de esto, proponen el uso de pulsos de energía para la detección de objetivos.
- 1930 • Lawrence A. Hyland, ingeniero eléctrico, basándose en los experimentos de Taylor y Young, consigue detectar por primera vez una aeronave. Lo importante de este acontecimiento fue que además logró determinar la posición aproximada del objetivo. Estos descubrimientos lo motivaron a presentar una patente [7] para su invento: “System for detecting objects by radio”.
- 1931 • William A. S. Butement junto a P. E. Pollard, como miembros de la SEE (Signals Experimental Establishment) en Woolwich Common, Londres, Inglaterra proponen un método para usar ondas de radio para la localización de barcos [8].

En sus experimentos utilizaron el oscilador Barkhausen-Kurz, el primero que pudo producir ondas de radio en frecuencias por encima de los 300 MHz, conocidas también por sus siglas en inglés como **UHF** (**U**ltra **H**igh **F**requency). La longitud de onda que lograron transmitir fue de 50 cm. Los resultados fueron positivos: lograron detectar las señales de radio reflejadas de las hojas de metal a una distancia de 91 metros aproximadamente. Butement y Pollard documentaron su experiencia acerca del “sistema de radar” en su propuesta: “Coastal Defence Apparatus”.

- 1933 • Basado en sus trabajos previos con experimentos relacionados al uso de la tecnología **SONAR** (acrónimo de **S**ound **N**avigation **A**nd **R**anging, en español: “navegación por sonido”), Rudolf Kühnhold, físico experimental alemán, recurre a la idea de utilizar ondas electromagnéticas de corta longitud para determinar en qué rango un “echo” (respuesta) puede ser obtenida desde de un objeto metálico. Debido al fracaso de sus pruebas con longitudes de onda de 13.5 cm, en donde la recepción era buena pero la reflectividad no, Kühnhold busca ayuda en la empresa Telefunken, sin conseguir interés por sus experimentos. Así es como decide apoyar a dos radio operadores, Paul-Gunther Erbsloh y Hans-Karl von Willisen, a constituir la empresa GEMA (Gesellschaft für Elektroakustische und Mechanische Apparate, traducido a español: Sociedad para la electroacústica y aparatos mecánicos) cuyo primer objetivo fue obtener resultados de los experimentos iniciados tiempo atrás por Kühnhold. En GEMA se trabajaba con longitudes de onda de 48 cm (630 MHz) y en los ensayos realizados sobre el viejo buque Hessen, se encontraron con que el impulso de transmisión debía ser cerrado para permitir al receptor obtener la información de la respuesta. A raíz de estos descubrimientos se empezaron a producir los radares Freya a partir de 1938.
- 1935 • En febrero de este año, el ingeniero y físico escocés Robert Watson-Watt, redacta un borrador, con la ayuda de su asistente Arnold Wilkins, en el cual proponía los principios del uso del radar [9]. La propuesta, titulada “The Detection of Aircraft by Radio Methods” (En español: “Detección de aeronaves utilizando métodos de radio”), fue elevada al comité para el estudio científico de defensa aérea y el 26 de febrero de 1935 se llevó a cabo un ensayo utilizando transmisores con antenas de la BBC de onda corta, en la ciudad de Daventry, Inglaterra, que impactaron sobre los aviones de bomberos de Heyford.

Producto del éxito de estos ensayos, Watson-Watt se convierte en Superintendente de la BRS (Bawdsey Research Station) y como primer trabajo en su nueva conducción, se encargó del diseño de una cadena de estaciones de radar a lo largo de la costa Este y Sur de Inglaterra antes del comienzo de la guerra en 1939. Esta cadena fue de vital importancia para la obtención de información que ayudó a la RAF (Royal Air Force, en español: Real Fuerza Aérea) a triunfar en la Batalla de Inglaterra.

- 1937 • Los hermanos Russell Harrison Varian y Sigurd Fergus Varian, de la Universidad de Stanford, desarrollan el prototipo del tubo de **Klystron**, un amplificador de alta frecuencia (En las frecuencias superiores a las UHF) y generador de microondas, ayudados por las investigaciones del físico y profesor William Webster Hansen sobre los resonadores de cavidad. Este invento, que fue publicado en enero de 1939 [10], tuvo una gran importancia para los sistemas de radar en la guerra así como también en la posteridad en el campo de la televisión, la comunicación satelital y los aceleradores de partículas actuales.
- 1940 • Henry Albert Howard Boot y John Turton Randall, de la universidad de Birmingham, desarrollan un magnetrón multi-cavidad, con 8 cavidades. El primer magnetrón multi-cavidad fue patentado por Hollman en 1938, pero sólo contaba con 4 cavidades [11] y la potencia generada no era suficiente. Así es como Howard y Randall construyen un magnetrón, que contaba con enfriamiento líquido y con 8 cavidades resonantes y concéntricas con el objetivo de incrementar la potencia de las radiofrecuencias. Además de ser muy pequeño, ventaja para el acoplamiento a los sistemas de radar de la época, como los que portaban los aviones de guerra B-17, el dispositivo generaba pulsos a 3 GHz con una potencia de salida de 15 KW.

#### Cuadro 1: Historia del radar

El desarrollo del magnetrón multi-cavidad por parte de Boot y Randall en 1940, contribuyó al éxito en la utilización del radar, ya que hizo posible contar con radares moderadamente ligeros que operaran con microondas en frecuencias altas (UHF) y con sistemas de longitud de onda más largos. Como consecuencia, el uso de señales de alta frecuencia hizo posible reducir el tamaño de la antena considerablemente, lo cual permitió al radar determinar la dirección de los objetivos con mucha más precisión.

Para el fin de la Segunda Guerra Mundial, el radar se había desarrollado y utilizado exitosamente especialmente por las fuerzas aliadas. Después de la guerra, gran parte del equipamiento militar excedente estuvo disponible para el uso civil. Gracias a esta disponibilidad, los interesados en el estudio del clima con radares fueron algunos de los primeros en adquirir estos radares excedentes y utilizarlos para investigación.

Desde la Segunda Guerra Mundial, el radar ha sido objeto del desarrollo evolutivo. Se han hecho mejoras en todos los componentes del radar, como los transmisores, las antenas, los receptores, los monitores y los procesadores. Las invenciones logradas en otros campos (como el transistor) pronto encontraron la forma de acoplarse a los sistemas de radar. Nuevos tipos de sistemas de antenas como las antenas en fase fueron desarrollados por los ejércitos aunque también empezaron a encontrarse en usos civiles. Y por supuesto, la aplicación de computadoras en los radares ha desplazado muchos de los nuevos desarrollos en el campo de los radares del hardware al software.

Uno de los grandes avances en la tecnología de radar, al menos en lo que respecta a su utilización en el clima, fue el desarrollo de técnicas Doppler. Esto se verá en detalle en la subsección 2.2.4.

Otro desarrollo importante en la utilización de los radares, desde el fin de la guerra, ha sido la utilización de información de polarización. El concepto de polarización aplicada en radares se desarrolla en la subsección 2.2.5.

## 2.2. Funcionamiento

El funcionamiento de un radar se basa en la utilización de ondas electromagnéticas que son producidas en su interior y, previa amplificación, se envían a la atmósfera esperando que las mismas “choquen” con un objeto. El proceso continúa al generarse cierta cantidad de energía producto de esta colisión que es captada nuevamente por el radar y procesada de manera tal que se obtenga información acerca del objeto localizado, como el tamaño, la velocidad, la densidad y la forma, entre otras.

A continuación se detallan las etapas involucradas en la operación normal del radar. Cabe destacar que este proceso involucra la operación de los diferentes componentes del radar (Ver Figura 2 en la sección 2.3).

En primer lugar, la **computadora**, encargada de sincronizar la operación de todos los componentes que conforman el sistema de radar, recibe una orden de escaneo, ya sea introducida por el operador del radar o bien auto disparada, en este caso, previamente cronometrada y programada, que le permite al radar escanear continuamente cada cierto lapso de tiempo. En este punto del flujo de operación y antes de comenzar efectivamente con el escaneo es donde se deben configurar los parámetros para el escaneo del radar.

La tarea siguiente conlleva la participación del **modulador** que interactúa con el transmisor alternando su encendido y apagado y proveyéndole la energía que obtiene de su propio almacén, la potencia y la duración de pulso a transmitir, todo esto en los intervalos necesarios para el escaneo.

Luego de recibido el conjunto de elementos previamente mencionados por parte del modulador y necesarios para continuar, el **transmisor** genera una señal de radio de alta frecuencia para, posteriormente, ser enviada al exterior. En radares meteorológicos, la frecuencia utilizada para la señal de microondas transmitida ronda entre 2 y 8 GHz y la longitud de onda se encuentra entre los

3 y los 10 cm, equivalentes a las bandas S, C y X (Ver subsección 2.2.3) [3].

En este momento, el **duplexador**, que actúa como un puente/interruptor entre el proceso de transmitir y recibir, se encuentra en el modo de transmisión. Una vez generada la señal de microondas de alta frecuencia, la misma llega a la **antena** desde el transmisor a través de la **guía de onda**, que actúa como comunicadora entre los dos anteriores y que tiene características particulares que permiten transportar las señales electromagnéticas con muy baja pérdida. Posteriormente, la antena envía la señal al **reflector** con el objetivo de propagar la energía electromagnética hacia el exterior, direccionándola de acuerdo a las características del reflector y/o a las especificaciones de escaneo configuradas.

Esta señal que es enviada al exterior en forma de pulsos de energía, viaja a la velocidad de la luz (aproximadamente  $3 \times 10^8$  m/s). Parte de esa energía ahora impactará sobre un objetivo y gracias al principio de reflectividad, otra parte de la energía volverá hacia el radar (Ver capítulo 2.2.2). Cabe aclarar que el objeto impactado puede ser cualquier objeto en esa dirección, como una gota de lluvia, una partícula de granizo, un ave, etc.

Aquí es donde el radar vuelve a tomar protagonismo, pero en sentido inverso, es decir, parte de la energía devuelta, producida por la incidencia de la señal electromagnética que impacta sobre un objetivo, es captada por la **antena** y direccionada hacia el **receptor**, nuevamente a través de la **guía de onda** y el **duplexador** (esta vez en modo de recepción).

El receptor, como se verá más adelante, limpia, clasifica y amplifica la señal obtenida para luego enviarla de manera entendible a la **computadora**, que se encarga de realizar los cálculos necesarios que permitan obtener información útil del escaneo. Parte de los cálculos realizados, se basan en los datos almacenados previamente en el momento de la transmisión, como el tiempo transcurrido desde la última transmisión, la energía enviada, el ángulo de la antena en ese instante, etc.

Finalmente, la información obtenida, es presentada al operador del radar a través del **display**, o los indicadores, que pueden ser de diferentes tipos dependiendo del radar en cuestión o de la información a mostrar.

### 2.2.1. Ciclo de escaneo

El conjunto de operaciones anteriormente mencionado, que transcurren de forma sincronizada se repite varias veces en un ciclo de escaneo. En un radar meteorológico, gracias a las características físicas y mecánicas de la antena, suele utilizarse un barrido rotativo de  $360^\circ$  y en varios ángulos de elevación, que dura aproximadamente 5 minutos en completarse. A continuación se presenta el ciclo de barrido completo, asumiendo como punto de origen el Norte o los  $0^\circ$  y la dirección de rotación en sentido horario [12]:

1. La antena se posiciona en dirección horizontal a los  $0^\circ$  y con un ángulo de elevación de  $0.5^\circ$ .
2. Emite un pulso de energía electromagnética y luego espera o “escucha”

por las señales de respuesta relativas a la dispersión de los objetivos impactados.

3. La antena gira en sentido horario y horizontalmente un incremento, relativo al ancho del haz de la antena.
4. Se repiten los pasos 2 y 3 hasta lograr un giro horizontal completo ( $360^\circ$ ).
5. Se incrementa el ángulo de elevación. Este incremento dependerá también del ancho del haz de la antena. Ejemplo: si el ancho del haz de la antena es de  $0.95^\circ$  y el ángulo actual de elevación es de  $0.5^\circ$ , la siguiente elevación de la antena será de  $1.45^\circ$ .
6. Se repiten los pasos 2 a 5 hasta alcanzar el número de elevaciones requeridas.

Para un escaneo de precipitaciones, se realizan alrededor de 10 a 20 elevaciones, dependiendo del radar y de la cobertura por parte del haz de la antena. Mientras que para un escaneo en modo aire claro, se requieren aproximadamente 5 elevaciones.

### 2.2.2. Reflectividad y dispersión

Antes de entrar en los detalles de los componentes anteriormente mencionados, se explicarán brevemente dos conceptos fuertemente relacionados a la utilización del radar como “aparato dimensionador de objetos”: **reflectividad** y **dispersión**.

#### Reflectividad

Desde el punto de vista meteorológico, la **reflectividad** se puede definir como la capacidad de una partícula de devolver la energía recibida de regreso hacia el radar. Cuando un pulso de energía emitido por la antena del radar se topa con un objetivo, una pequeña parte de esa energía “rebota” de vuelta hacia la antena del radar. Su valor depende de características del objeto (como su forma) y de la relación entre la longitud de onda utilizada y la dimensión del objeto [12] [13].

#### Dispersión

La **dispersión** es el proceso por el cual las ondas electromagnéticas son desviadas al interceptar un objeto. En este caso, la energía que influye sobre el objeto no retorna con la misma intensidad hacia el radar, sino que se divide en “hilos” de energía más pequeños que se dispersan, propiamente dicho, en diferentes direcciones. Si la señal impacta sobre varias partículas, las señales que retornan hacia el radar se combinan en una sola de mayor intensidad. Esto significa que varias gotas de lluvia grandes producirán una respuesta más fuerte que unas cuantas gotas pequeñas.

### 2.2.3. Bandas de frecuencias

Como se mencionó anteriormente, en los radares meteorológicos se utilizan las bandas de frecuencias S, C y X, cuyos rangos de frecuencias utilizados son 2 a 4 GHz, 4 a 8 GHz y 9 a 12 GHz, respectivamente. Adicionalmente, otras bandas de frecuencias son utilizadas en radares y pueden apreciarse en la Tabla 2 [14]:

Banda	Rango de frecuencia	Usos
HF	3-30 MHz	Radares de vigilancia OTH ( <b>O</b> ver <b>T</b> he <b>H</b> orizon, en español: <i>Sobre el horizonte</i> )
VHF	30-300 MHz	Vigilancia en distancias muy largas
UHF	300-1000 MHz	Vigilancia en distancias muy largas
L	1-2 GHz	Vigilancia de larga distancia Control de tráfico en rutas
S	2-4 GHz	Vigilancia en distancias moderadas Control de tráfico en terminales Control del clima en largas distancias
C	4-8 GHz	Seguimiento de larga distancia Control del clima en el aire
X	8-12 GHz	Seguimiento de corta distancia Direccionamiento de misiles Intercepciones aéreas
K <sub>u</sub>	12-18 GHz	Cartografía de alta resolución Altimetría satelital
K	18-27 GHz	Absorción por vapor de agua (poco uso)
K <sub>a</sub>	27-40 GHz	Cartografía de muy alta resolución Vigilancia en aeropuertos
Milímetro	40-100+ GHz	Usos experimentales

Cuadro 2: Bandas de frecuencias



Existen diferentes regímenes que relacionan la longitud de onda  $\mathbf{A}$ , utilizada por la banda de frecuencia elegida y el tamaño de las partículas detectadas  $\mathbf{D}$ , especialmente útiles para el caso de detección de partículas de lluvia o granizo [14]:

- Uno de ellos es el de **Rayleigh**, el cual expone que cuando el tamaño de una gota  $\mathbf{D}$  es menor que la longitud de onda  $\mathbf{A}$ , existe una relación lineal para la dispersión obtenida.
- El siguiente es el régimen de **Mie**, que se aplica cuando  $\mathbf{D} / \mathbf{A}$  es aproximadamente igual a 1, es decir la longitud de onda es casi igual en tamaño al de las partículas detectadas.
- El tercero es el régimen de la zona óptica, basado en la suposición de que  $\mathbf{D} / \mathbf{A}$  es grande, o que el tamaño de las partículas es mayor a la longitud de onda utilizada.

En las ecuaciones de los radares meteorológicos se utiliza el régimen de Rayleigh, aunque esto pueda provocar errores si en el escaneo aparecen objetos de un tamaño mayor que las gotas de lluvia, como aves o alguna clase de insecto. Estos errores derivan en dificultad a la hora de realizar los cálculos de dispersión o reflectividad por parte del sistema de radar [12].

Dado que el diámetro de las partículas de interés para los radares meteorológicos va de pocos milímetros a centímetros, se utilizan bandas de frecuencias con longitudes de onda chica (de pocos milímetros a pocos centímetros), como lo son las bandas **S**, **C** y **X**.

#### 2.2.4. Efecto Doppler

Se ha mencionado que hay cálculos que se realizan comparando la señal de salida del radar con la señal de respuesta recibida. Un ejemplo de cálculo importante en la tecnología de radares es la velocidad de los objetivos detectados (o *velocidad Doppler*, para estos tipos de radares), para lo cual se utiliza un principio, denominado **Doppler** en base a su descubridor Christian Andreas Doppler.

El principio se basa en la alteración de frecuencias que se produce entre dos fuentes de emisión de ondas al desplazarse al menos uno de ellos respecto del otro. Aplicado en un radar, la longitud de onda de la señal de respuesta que se recibe en el radar difiere de la enviada, si la distancia entre el radar y el objetivo se incrementa o decremента.

A continuación se presentan las alteraciones producidas en base a la movilidad de las fuentes de emisión, adaptado al ejemplo de un radar Doppler. En la Figura 1 puede verse en la parte izquierda, para los tres casos, el radar (origen de las ondas) y la señal emitida en cada caso y en la parte derecha, la señal recibida en base a la variación de la distancia entre los dos puntos [15].

1. **Sin cambios:** Si la distancia entre el radar y el objetivo permanece constante, la frecuencia de la señal recibida por el radar permanecerá igual en el transcurso del tiempo. Esto significa, que el objetivo no se ha trasladado a otra posición, al menos durante el envío y la recepción de las señales electromagnéticas.
2. **Más distancia:** Cuando la distancia entre el radar y el objetivo aumenta, la frecuencia de la señal recibida disminuirá. Cada vez que el radar ha completado una onda, el objetivo se ha movido lo suficiente como para que las ondas **se reciban con menos frecuencia**. Este es el caso en el que, por ejemplo, el objetivo que devuelve la señal se está alejando del radar. Se debe tener en cuenta que el alejamiento del objeto no significa que su dirección sea en línea recta desde el radar, sino que simplemente “se aleja” del mismo.
3. **Menos distancia:** Al disminuir la distancia, la frecuencia de la señal recibida irá incrementándose en relación al tiempo. Dado a que el objetivo se está acercando, las ondas **llegan con mayor frecuencia**.

Hay que tener en cuenta que existen tiempos que se utilizan en el cálculo de la velocidad Doppler relativos a los instantes en los cuales se emiten las ondas electromagnéticas y en los que se reciben. Los mismos son utilizados para establecer con precisión los valores resultantes de aplicar los cálculos basados en el efecto Doppler.

Debemos comprender la importancia del descubrimiento realizado por C.A. Doppler y su aplicación en diversas áreas como medicina, seguridad, aviación, audio, meteorología, entre otras. Sin duda alguna el enfoque de esta tesina se concentra en la aplicación del efecto Doppler en radares meteorológicos.

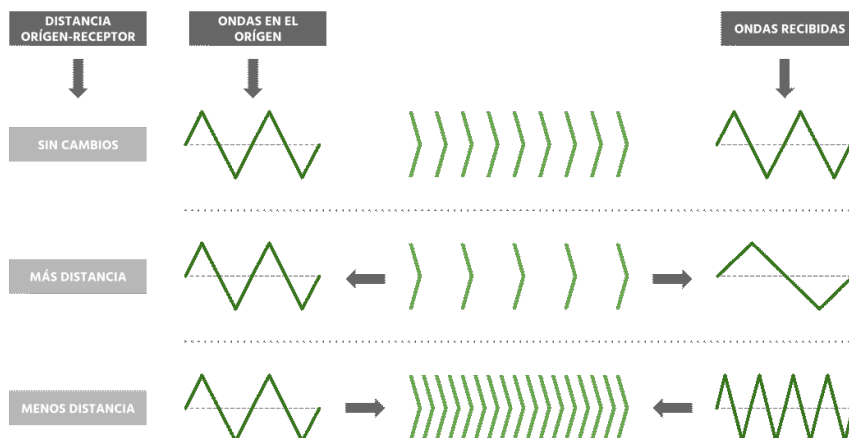


Figura 1: Efecto Doppler

### 2.2.5. Polarización

Una onda electromagnética se compone de un componente eléctrico y un componente magnético, también nombrados como **campo eléctrico** y **campo magnético** de la onda. Estos campos oscilan en dirección perpendicular a la dirección en la que se propaga la onda. A su vez, el campo magnético oscila de forma perpendicular al campo eléctrico. La **polarización** es una propiedad que relaciona la forma de oscilar entre los campos que componen una onda. Para el caso de esta tesina, interesa estudiar la polarización aplicada a ondas de tipo electromagnéticas, aunque es posible hablar de polarización también en otros tipos de ondas como las *mecánicas transversales*.

Como el campo magnético es perpendicular al campo eléctrico, suele mencionarse la polarización de una onda como la polarización de su campo eléctrico (puede derivarse fácilmente la polarización del otro campo).

La polarización de una onda puede clasificarse según la relación existente entre las amplitudes de ambas componentes (eléctrica y magnética) a lo largo del tiempo. Éstas pueden ser:

- **Lineal:** Tanto el campo eléctrico como el magnético se producen *en fase*. Con igual amplitud, ambos alcanzan sus máximos y mínimos al mismo tiempo. Este caso se produce inclusive con un desfase de  $180^\circ$  entre los campos. Se subclasifican en **polarización lineal horizontal**, si el campo eléctrico oscila de forma horizontal en relación a la superficie terrestre y en **polarización lineal vertical** caso análogo.
- **Circular:** Misma amplitud de onda en ambos campos pero a razón del tiempo, están desfasados  $90^\circ$ . A su vez, se subclasifican en: *Circular en sentido horario* y *Circular en sentido antihorario* dependiendo del tipo de desfase (campo eléctrico  $-90^\circ/+90^\circ$  en relación al magnético).
- **Elíptica:** Este es el caso que cubre las excepciones de los dos anteriores, es decir, ambos campos tienen amplitudes diferentes y a su vez, el ángulo de desfase es distinto de  $0^\circ$  y  $180^\circ$ .

El nombre que recibe cada clasificación anterior, se debe a la forma que toma el dibujo que describe la variación del vector del campo eléctrico a lo largo del tiempo (visto desde el punto receptor de la onda directamente mirando al punto emisor de la onda).

Es posible clasificar a su vez los tipos de polarizaciones dependiendo de la orientación del vector del campo eléctrico de la onda:

- **No polarizado:** Es el ejemplo de una onda de luz, en donde la orientación de la oscilación se da en múltiples direcciones.
- **Parcialmente polarizado:** En donde existen ciertas (limitadas) orientaciones preferenciales en las cuales oscila el campo eléctrico.
- **Linealmente polarizado:** Coincide con el primer caso de la clasificación anterior, en donde las oscilaciones ocurren en un solo plano.

En la actualidad existen radares de **polarización simple** capaces de transmitir y recibir ondas electromagnéticas en una sola orientación y radares de **polarización doble** que suman un plano más para transmitir/recibir ondas electromagnéticas. Incluso existen radares que pueden transmitir ondas polarizadas circularmente.

Debido a la naturaleza de la energía reflejada por un objetivo del radar, la polarización de las ondas emitidas por el este objetivo puede ser diferente en relación a la polarización emitida por el radar. A los efectos de una mejor extracción de datos, un radar de polarización simple no podrá obtener el mismo nivel de información (forma, tamaño, etc.) de una gota de precipitación, por ejemplo, que si usara un sistema de radar de doble polarización en el cual estos datos pueden ser obtenidos de forma directa.

Aún conociendo la mayor capacidad que posee un radar con las característica de utilizar la polarización doble para manejar las ondas electromagnéticas, es útil en ciertos casos, la utilización de ondas de simple polarización. Esta aproximación en la utilización de un radar suele ser útil, por ejemplo, para obtener cierto tipo de información (suficiente con ondas de polarización simple) aislando datos, o ruido obtenido por la utilización de más de un vector de polarización.

### 2.3. Componentes

Luego de haber introducido los conceptos básicos en los cuales se basan los radares, en la sección anterior, se analizarán los componentes de un radar meteorológico. La siguiente figura muestra un esquema conceptual de dichos componentes [13].

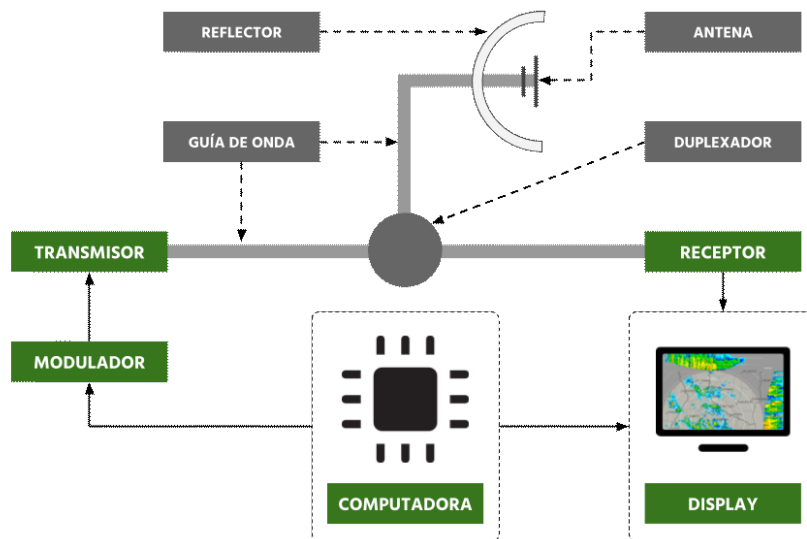


Figura 2: Esquema de radar

### 2.3.1. Transmisor

El **transmisor** es el encargado de generar la señal de radio de alta frecuencia que será enviada a la atmósfera a través de la **antena**. Esta señal será transmitida a través de pulsos cortos de energía, cuya potencia dependerá de aspectos como usos del radar, la escala del sistema de radar entre otras y por lo tanto puede variar desde cientos de watts (W) a unos cuantos miles de watts (kW) [16].

A continuación se presentan algunas de las características con las que los transmisores deberían contar [17]:

- Capacidad de generar la señal de radiofrecuencia de potencia media y de potencia más alta.
- Contar con un adecuado ancho de banda de radiofrecuencia.
- Alta estabilidad de radiofrecuencia para cumplir con los requisitos de procesamiento de señales.
- Facilidad para modularlo acorde a las necesidades de requerimiento de diseño de la forma de la onda.
- Eficiencia, confiabilidad y facilidad de mantenimiento.

Existen diferentes tipos de transmisores, los cuales están altamente relacionados con el tipo de material/componentes utilizados para su fabricación [13] [17]:

- **Klystron**. Tubos de vacío de microondas de alto poder. Suelen tener mayor tamaño que los transmisores de magnetrón pero mayor poder de amplificación, con lo cual resulta más fácil controlar la forma de la onda en los pulsos transmitidos.
- **Magnetron**. Para radares viejos, los cuales utilizaban solamente datos de intensidad, no era necesario obtener información de fase y en consecuencia el uso de magnetrones como transmisores era aceptable. Tienen un bajo costo, debido, entre otros aspectos, a su fácil construcción, aunque por esta misma facilidad presenta una desventaja: generalmente, solo puede trabajar en una frecuencia fija de manera constructiva.
- **De estado sólido**. El transistor es un dispositivo de estado sólido que amplifica la señal controlando el flujo de portadores de carga a través de sus materiales semiconductores. Generalmente tienen mucho menor potencia de transmisión que la mayoría de los transmisores de radar. Sin embargo, combinando varios transmisores de estado sólido en una formación y controlando la sincronización de cada elemento apropiadamente, es posible obtener buenas potencias de salida.

### 2.3.2. Modulador

El **modulador** es el componente encargado de controlar al transmisor. Esta tarea es llevada a cabo alternando el encendido y apagado del transmisor para generar la forma de onda adecuada para el pulso que se transmitirá. Es decir el modulador indica al transmisor, cuando transmitir y por cuánto tiempo. Esta actividad se ejecuta cuando un pulso temporizado acciona al modulador, que entrega la energía indicada al transmisor a través de un pulso corto de alto voltaje y alta potencia.

La energía mencionada anteriormente es obtenida desde un almacén que acumula en el lapso que transcurre entre dos pulsos. De esta manera el almacén contendrá la suficiente energía para repetir el ciclo de transmisión.

### 2.3.3. Receptor

La función del **receptor** es la de detectar las señales de respuesta provenientes de la captura de la antena. Cabe aclarar que no todas las señales que llegan son útiles y es aquí en donde el receptor logra su primer objetivo: filtrar el flujo de señales, desechando aquellas que no interesan.

Posteriormente al filtrado, las señales son limpiadas de ruido, interferencia o desorden. Esta característica está ligada al tipo de respuesta “dispersa” que se recibe al impactar la señal emitida por el radar en el objetivo, como se ha mencionado con anterioridad.

Una vez limpiada la señal, sigue el proceso de amplificación. Los receptores utilizados en radares, son capaces de aceptar señales de respuesta débiles, dependiendo del MDS (del inglés, Mínima Señal Detectable), e incrementar su amplitud en un factor de 20 o 30 millones. El MDS para un radar corriente tiene un valor de 10-13 Watts (-100 dBm).

El diseño del receptor dependerá de factores como la forma de la onda a detectar, la naturaleza del ruido entrante, la interferencia y las respuestas “fantasmas” que vendrán junto con las respuestas útiles/relevantes.

El ruido puede llegar al receptor a través de las terminales de la antena junto con las señales deseadas o puede ser generado dentro del mismo receptor. Dadas las frecuencias de microondas utilizadas generalmente en un radar, el ruido externo que entra por la antena es bastante bajo, por lo que la sensibilidad del receptor es configurada, más bien, por el ruido interno que se produce dentro del mismo.

### 2.3.4. Antena y reflector

La **antena** es el instrumento por el cual el radar irradia de forma eficiente los campos de ondas electromagnéticas generadas por el **transmisor** y recibe eficientemente las respuestas de las dispersiones. La forma y la dirección del haz que contiene el campo electromagnético irradiado está determinado por la

distribución en la apertura de la antena. Otros parámetros comprendidos en la regulación de una antena, además de la apertura son el ancho de haz, la relación delante-atrás y la ganancia entre otros.

Los radares meteorológicos generalmente tienen una antena y un **reflector** (ver Figura 2). La antena real es el elemento radiante que transmite la señal de radar hacia la atmósfera en dirección al reflector que luego refleja y direcciona la señal en sentido opuesto al radar.

Existen diferentes tipos de antenas, o conjuntos de antenas y reflectores, utilizados en los sistemas de radar. A continuación se detallan algunos de los mismos:

- **Parabólicas.** O de plato, por su forma, es una de las utilizadas con más frecuencia en los radares con antena incluida. Una antena de plato consiste de un reflector circular parabólico y una fuente puntual de alimentación, situada en el punto focal de este reflector. El plato circular parabólico está cubierto por una capa de metal en su interior, con el objetivo de reflejar la energía del radar, proveniente de la fuente. Cuanto más grande sea el reflector, mejor direccionada será la señal y más estrecho el haz de la antena [17].
- **Antenas en fase.** Consiste en un conjunto de elementos radiantes que se disponen generalmente con un cierto orden. Si se controla la onda electromagnética proporcionada a cada antena individualmente, la misma se denomina antena en fase. Éstas tienen la capacidad de cambiar la dirección del haz de forma eléctrica haciendo posible escanear el haz del radar con una antena estacionaria [16].

### 2.3.5. Duplexador

En general, la transmisión y recepción de las señales desde y hacia el radar, respectivamente, son efectuadas por una misma antena. El control del camino a tomar para un pulso transmitido y para una señal recibida, es llevado a cabo por el **duplexador**, también denominado Intercambiador de transmisión/recepción o en inglés T/R switch. Para transmisiones de potencia menor se utiliza, generalmente, un diodo semiconductor como duplexador, mientras que para intercambios de potencia mayores el intercambiador suele estar compuesto por un circulador de guía de onda (duplexador de anillo híbrido) [16].

En resumen, las funciones básicas de un duplexador son [18]

- Conectar el transmisor a la antena, desconectando el receptor durante la fase de transmisión.
- Conectar el receptor a la antena, desconectando el transmisor, durante la fase de recepción.
- Proveer el aislamiento necesario entre transmisor y receptor.

A continuación, se presentan las propiedades principales del duplexador [19]:

- Baja pérdida entre transmisor y antena, en fase de transmisión (es aceptable menos de 1 dB).
- Buen aislamiento al intercambiar transmisión a recepción, en la fase de transmisión (como mucho 80 dB ).
- Baja pérdida entre la antena y el receptor, en fase de recepción (es aceptable menos de 1 dB).
- Intercambio rápido entre el estado de transmisión y recepción, ya sea causado por la señal de transmisión o por una señal de comando.

### 2.3.6. Guía de onda

La **guía de onda**, es un conector, fabricado como un tubo metálico hueco y generalmente rectangular, que cumple la función de *conducir* la energía electromagnética de un punto a otro. En el caso de un radar meteorológico, los puntos a unir son el transmisor/receptor y la antena. De este modo, los componentes involucrados en esta unión pueden estar a cierta distancia, con fines de disponer por ejemplo a la antena en una torre, para evitar interferencias en el escaneo, y al transmisor/receptor sobre la base de la torre, al resguardo.

Una de las ventajas que se obtienen de la utilización de la guía de onda es la posibilidad de transmitir en ráfagas de alta potencia y la baja pérdida de energía a la hora de propagar las ondas electromagnéticas. Esto último, en gran medida, se debe a la utilización de superficies metálicas para su elaboración, como puede ser el cobre, plata u oro inclusive. De hecho la guía de onda está completamente blindada, lo que produce excelente aislamiento entre señales adyacentes [20].

Como desventaja, podemos mencionar su elevado costo. Los volúmenes de fabricación son bajos y el costo de los insumos como el cobre y la plata es relativamente alto [21].

Toda guía de onda tiene una frecuencia de corte asociada a cada tipo de onda electromagnética que se propague. La misma se deriva de sus dimensiones físicas y determina un nivel de filtro en el cual la energía que se encuentre por encima de esta frecuencia, pasará a través de la guía de onda, mientras que el resto por debajo de este límite se verá atenuado y convertido en un valor ilegible en una corta distancia relativa. El modo dominante de transmisión para una guía de onda particular es el modo que posee la menor frecuencia de corte. La frecuencia de corte depende de la forma y el tamaño del corte transversal de la guía de onda: cuanto más larga sea la guía de onda, menor será la frecuencia de corte.

### 2.3.7. Computadora

En viejos radares la **computadora**, se denominaba también **reloj maestro** y su función principal es la de controlar y coordinar las acciones llevadas a cabo



por el resto de los componentes del radar.

Así mismo, dentro de las funcionalidades de este componente, se incluye la de servir como interfaz de control y configuración para los usuarios del radar.

Además de controlar la operatoria del radar, las computadoras (también denominadas, *procesadores de señales*) controlan el procesamiento de la información recibida en los escaneos. Estos procesadores de señales pueden obtener la señal de entrada, efectuar chequeos de calidad en los datos, promediarlos, convertirlos en nuevos productos, almacenarlos para futuras reproducciones, transmitirlos a ubicaciones remotas y mostrarlos para ser consumidos por los usuarios. Hoy en día, la computadora acoplada al radar, es probablemente más importante que el hardware. De hecho, el hardware que comprende un radar casi no ha cambiado mucho en los últimos años a diferencia del software utilizado con el radar.

Gracias a la computadora, se pueden controlar una gran variedad de parámetros de radar, entre ellos los no menos importante, **los de escaneo**. Para denotar la importancia de una buena interfaz de configuración combinada con las acciones llevadas a cabo, un usuario de radar puede ajustar opciones de escaneo como:

- Rango de escaneo para mostrar.
- El ángulo de elevación de la antena.
- Los ángulos azimuts a escanear.
- El número de pulsos transmitidos por segundo.
- Duración (largo) de los pulsos emitidos.

### 2.3.8. Display

La señal de respuesta de un radar, posterior al proceso de escaneo, debe ser presentada al operador del radar de manera entendible y adecuada. Para esto se utiliza un monitor o **display**.

Puede verse al display como una funcionalidad incluida en el componente anteriormente mencionado: **la computadora**, aunque a los efectos de la descripción que se hará en esta sección, no se incluirá al **display** como una parte de la computadora, sino que se lo tratará como un componente más en todo el sistema de radar que se comunica con el resto.

En la mayoría de ocasiones, un display se compone de una pantalla de dos dimensiones, que muestra la localización de los objetivos con respecto a algún punto de referencia. También puede presentarse como una simple luz para indicar el estado de alguna condición o un medidor para indicar el valor de algún parámetro como el ángulo a donde apunta la antena o la distancia que existe a un objetivo encontrado. Otra versión de display que suele verse en antiguos radares es un visor numérico para mostrar valores clásicos como distancia, velocidad, intensidad de señal o el estado de alguna funcionalidad del radar.

El dispositivo que compone al display del radar en la mayoría de los casos es un tubo de rayos catódicos (CRT en inglés). Podemos distinguir dos tipos de displays que utilizan la tecnología CRT:

- **CRT de deyección modulada:** un objetivo es indicado por la deflexión del haz de electrones.
- **CRT de intensidad modulada:** un objetivo es indicado intensificando el haz de electrones y presentando un punto luminoso en la pantalla del CRT.

Existen distintos formatos para displays que utilizan CRT, que se derivan de explotar las distintas dimensiones ( $X, Y$ ) y características de esta clase de displays como el brillo y la paleta de colores disponible en cada caso. Algunos de los más relevantes son:

- **A-Scope:** Uno de los modos más simples, en el que se permite visualizar la distancia a un objetivo (eje  $X$ ) y la intensidad de la señal en forma de deflexión vertical (eje  $Y$ ), como se aprecia en la Figura 3. Utilizado generalmente en instrumentación o en sistemas de radares de recolección de datos, en donde la antena no está normalmente escaneando. Si la antena escanea, este formato es difícil de interpretar. Este formato de visualización se aprecia en los displays anteriormente clasificados como CRT de deyección modulada [14].

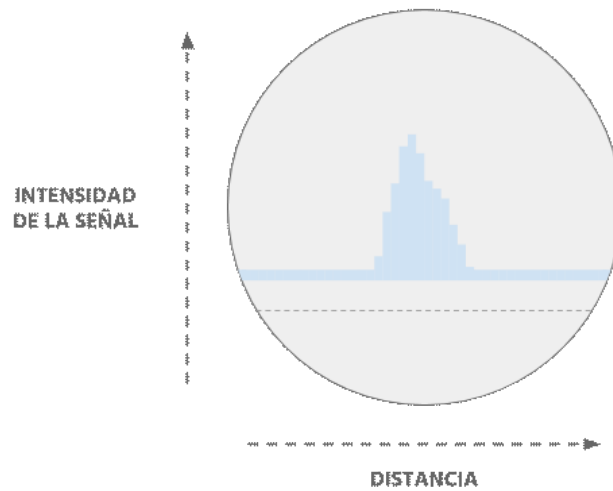


Figura 3: Formato A-Scope

- **PPI:** Este formato de display, cuyo acrónimo es derivado del inglés *Plan Position Indicator*, es utilizado en radares que realizan escaneo en azimut. De los displays de dos dimensiones es el más utilizado y lo que puede

apreciarse como resultado es un mapa de coordenadas polares en donde figuran las señales de respuesta producidas por la reflexión de los objetivos (Ver Figura 4). El mapa se centra en la antena del radar, mostrada en el centro de la pantalla, usualmente con el norte en la parte central superior de la pantalla y una línea radial que rota acompañando el movimiento de escaneo circular de la antena. Esta línea al ejecutar su barrido limpia los puntos marcados en la rotación anterior y marca los nuevos puntos, producto del escaneo más reciente [14].

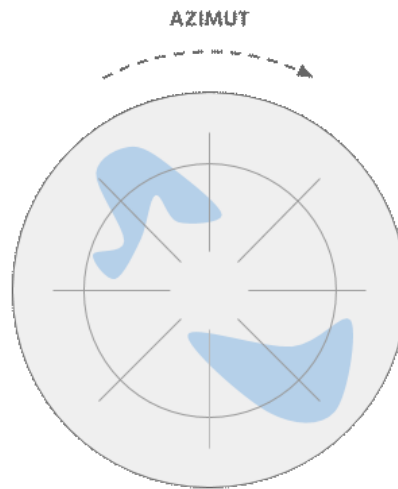


Figura 4: Formato PPI

- **Offset-PPI:** Tomando como base los displays con formato PPI, existen ocasiones en las cuales es preferible desplazar el centro del mapa, que representa la ubicación de la antena, hacia un punto que no sea el centro de la pantalla. Este es el motivo de su denominación: el desplazamiento (*offset* en inglés). Es útil para explotar determinadas áreas específicas alrededor del radar (ver Figura 5) [14].
- **B-Scope:** Este formato, similar al PPI, está basado en coordenadas rectangulares  $x$  e  $y$ , las cuales representan el ángulo de azimut y la distancia, respectivamente. Las señales también aparecen como puntos brillantes a medida que una línea vertical recorre el eje  $X$ , barriendo la pantalla hacia la derecha y acompañando el movimiento de la antena al escanear [14].
- **RHI:** Su nombre proviene del inglés: Indicador de distancia-altura (*Range-Height indicator*) y es utilizado generalmente con el fin de obtener y presentar datos relevantes a la altura de un objetivo. Se aprecia un plano de dos dimensiones, como puede verse en la Figura 7, en el cual el eje vertical representa la altura del objetivo, mientras que el eje horizontal denota la distancia al mismo. También suele denominarse **RHI-Scope** [3].

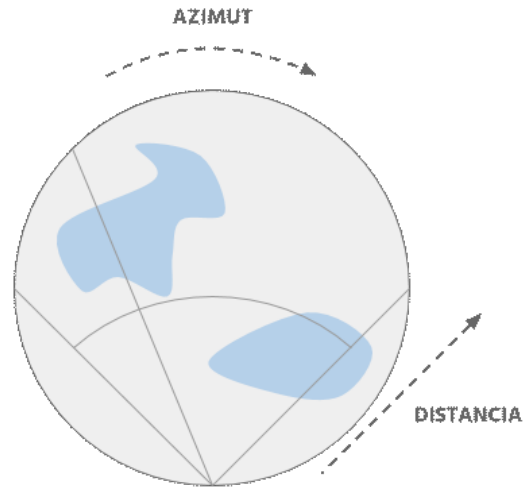


Figura 5: Formato Offset-PPI

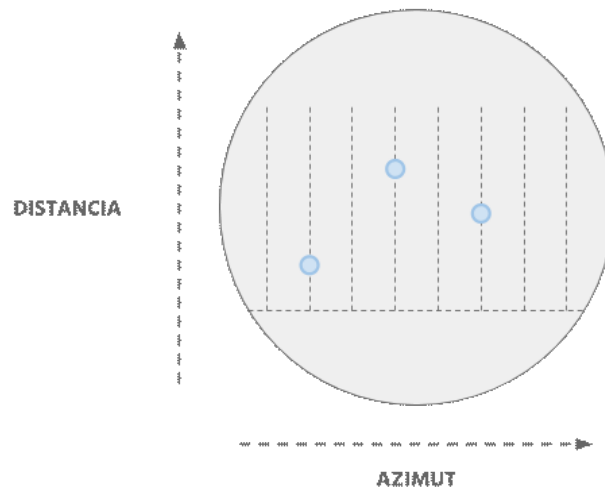


Figura 6: Formato B-Scope

Los anteriores formatos sólo conforman una parte de la lista de formatos conocidos de displays [14]:

- A, B, C, D, E, F, G, H, I, J, K, L, M, N, O y R Scope.
- PPI, Offset-PPI, Delayed-PPI, Sector-PPI y Expanded-PPI.
- Situation display.

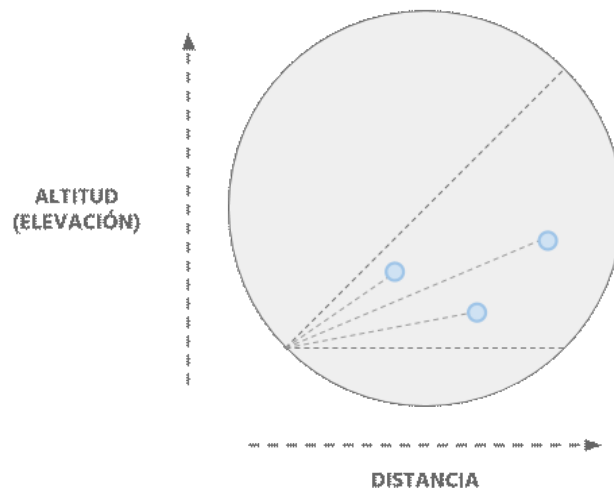


Figura 7: Formato RHI

- Range-Height Indicator ó RHI-Scope. (Indicador distancia-altura).

Generalmente el display se conecta directamente a la salida de video del receptor y en este esquema la información visualizada puede no ser de gran utilidad para el operador, dada la velocidad con la que se obtienen los datos. Así pues, la información proveniente del receptor, pasa por un procesador automático de datos con el fin de condensar y sintetizar la información que se presentará al operador del radar [3].

#### 2.4. Radares utilizados en INTA

Para el desarrollo de esta tesina, se han utilizado archivos generados por tres de los radares meteorológicos que utiliza **INTA**, fabricados por la empresa **Selex ES – Gematronik**, que conforman una red que alcanza una cobertura de escaneo de 46 millones de hectáreas en la región Pampeana de Argentina. Los mismos se encuentran en las siguientes ubicaciones (ver el mapa de la Figura 8) y poseen las siguientes características [22] [1]:

- **Paraná, Entre Ríos**
  - Modelo: Meteor 600C.
  - Doble polarización.
  - Puesto en funcionamiento en el año 2008.
- **Anguil, La Pampa**
  - Modelo: Meteor 600C.

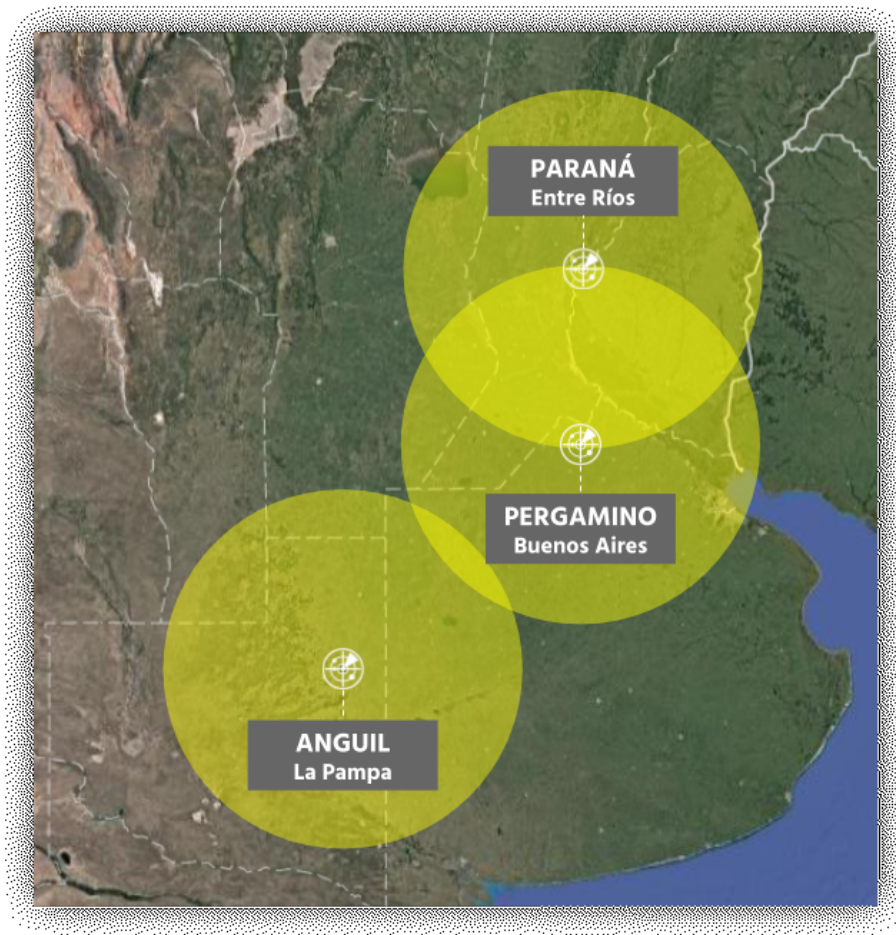


Figura 8: Cobertura de los radares de INTA en la Argentina

- Doble polarización.
- Puesto en funcionamiento en el año 2008.
- **Pergamino, Buenos Aires**
  - Modelo: Meteor 360C.
  - Simple polarización.
  - Puesto en funcionamiento en el año 2004.

Además los tres comparten las siguientes características:

- Banda de frecuencia: **C (4 a 8 GHz)** (Ver subsección 2.2.3).
- Longitud de onda: **5.4 cm**.
- Alcance de cobertura: **120 km, 240 km y 480 km**, dependiendo del modo de trabajo.

- Software de usuario: Rainbow®.

#### 2.4.1. Software

##### Características

El radar analizado en esta tesina utiliza el software Rainbow® en su versión 5.22.0. El mismo ofrece características importantes como:

- Control del radar de forma local y remota.
- Soporte para redes multi radar.
- Arquitectura cliente-servidor.
- Múltiples plataformas: Windows, Linux. Con posibilidad de configuraciones mezcladas.
- Capa de comunicación TCP/IP heterogénea, sin utilización de protocolos propietarios.
- Interface de fácil navegación con MDI (del inglés **M**últiple **I**nterface **N**avigation).
- Amplio rango de productos meteorológicos avanzados.
- Productos hidrológicos incluyendo integración con pluviómetro/disdrómetro y productos de polarización dual.
- Sistema de monitoreo y registro amplio y completo.
- Soporte de visualización de 24 *bits* de color verdadero y 256 niveles para todos los productos.

##### Tareas principales

El software Rainbow® tiene la capacidad de cumplir las siguientes tareas:

- Administración de sensores y supervisión de radar.
- Procesamiento de datos meteorológicos, incluyendo pre y post procesamiento.
- Visualización de datos y post-análisis en diferentes niveles.
- Distribución y control de datos e interfaces a otros sensores.

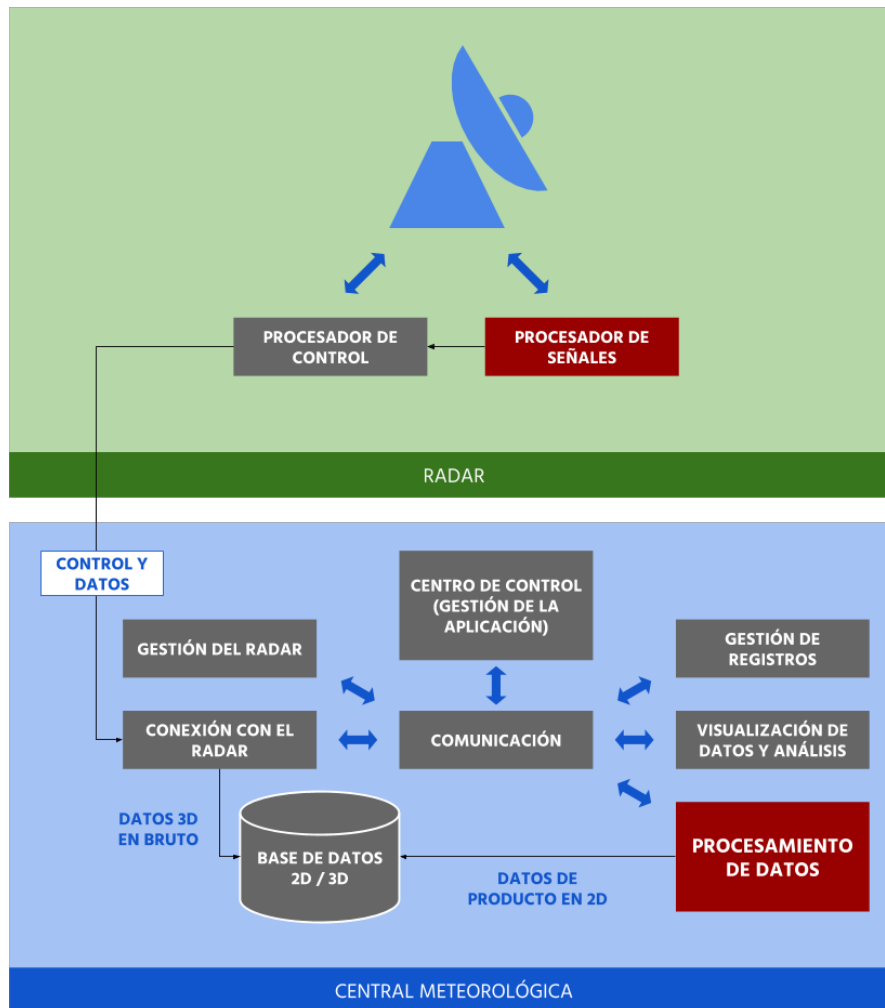


Figura 9: Arquitectura y distribución de software Rainbow®

### Flujo de procesamiento

La Figura 9 representa la arquitectura de software para el sistema Rainbow®. Es sabido que los sistemas de radar generan una gran salida de datos que debe ser presentada a los usuarios de forma amigable. Esto requiere de una serie de tareas que intervienen en dicha información en diferentes niveles:

- Obtención de la información, incluyendo la administración del proceso que requiere control de radar, ajuste de parámetros de adquisición, recepción de la información y control de calidad (tanto para operación con datos y sensores).
- Procesamiento de datos.



- Distribución de datos.
- Visualización de datos y análisis interactivo.

La parte de **procesamiento de datos** es una de las actividades principales en el software y puede dividirse en dos etapas:

### 1. Procesamiento de señal

Es la operación realizada en el radar y en tiempo real. No forma parte de la red meteorológica directamente pero es la base de su alimentación. Esta tarea la lleva a cabo el procesador de señales del radar, que aplica toda la fuerza de procesamiento necesaria para producir los principales momentos como reflectividad ( $Z$ ), velocidad radial ( $V$ ), ancho de espectro ( $W$ ), etc.

### 2. Procesamiento de datos

Segunda etapa que utiliza todo el poder de computación necesario para la evaluación tridimensional, como interpolación en azimuth y elevación, es decir un conjunto de datos tridimensionales en bruto de reflectividad, velocidad radial, ancho de espectro, etc. Además convierte la información tridimensional en productos meteorológicos bidimensionales necesarios para mostrar una imagen escalar o cartesiana.

El **procesador de señal** realiza el pre-procesamiento inicial de los datos del volumen escaneado. El **procesador de control del radar (RCP)** representa la conexión inteligente entre el radar y la red meteorológica. Este se comunica con el procesador de señal y almacena temporalmente la información en bruto, previamente procesada. Luego la transferencia de la información hacia las estaciones de trabajo, que también utilizan el software Rainbow®, es llevada a cabo sobre el protocolo TCP/IP.

### 3. Obtención de los datos

Previamente se ha examinado el funcionamiento y en especial el proceso por el cual un radar meteorológico, de forma genérica, es capaz de detectar objetos en la atmósfera, medir su tamaño, calcular la velocidad de su movimiento y obtener datos relevantes, entre otras capacidades. Éstas son características que podemos encontrar en la mayoría de los radares que se encuentran en funcionamiento en la actualidad.

Habiendo comprendido las características generales de los radares meteorológicos, es necesario organizar el **flujo de los datos** desde la etapa de **obtención** hasta la de presentación a los usuarios interesados o también denominada **visualización**, pasando por el **procesamiento** de la información.

El enfoque anterior se puede ver en el siguiente esquema:

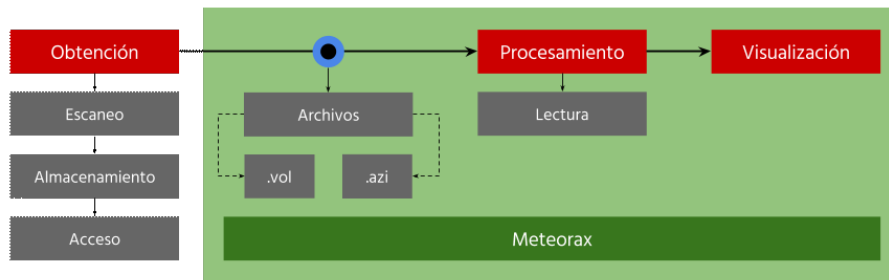


Figura 10: Flujo de los datos

En este capítulo se abordará en particular la primer etapa: la obtención de los datos, que comprende las tareas de **escaneo**, **almacenamiento** y **acceso** a los datos.

Este esquema, que intenta ilustrar cómo los datos se manipulan y sufren transformaciones entre las diferentes etapas, es el utilizado para el desarrollo de esta tesina y tiene como fuente principal de alimentación tres de los radares de la red instalada en INTA, los cuales están ubicados en:

- **Anguil**, La Pampa
- **Paraná**, Entre Ríos
- **Pergamino**, Buenos Aires

Sus principales características fueron presentadas en el capítulo 2.

De ahora en adelante se mencionará al *radar* haciendo referencia a las características y configuraciones, en particular, de estos tres ejemplares que se encuentran en funcionamiento en los puntos de escaneo previamente mencionados.

Es importante destacar que a partir de las dos últimas etapas se hace presente la intervención de la aplicación **Meteorax** construida como prototipo de esta tesina. Su análisis se extiende en el capítulo 5.

### 3.1. Escaneo

La primer tarea que se realiza en el radar meteorológico es el escaneo, en donde básicamente se emiten pulsos de energía a la atmósfera que al chocar con las partículas, generan más energía que es devuelta e interceptada por el radar y desde allí se realizan los cálculos necesarios para determinar variables como reflectividad, velocidad, tamaño de las partículas, distancia al radar, etc. Los detalles más específicos acerca de este proceso fueron vistos en el capítulo 2.

Como se ha visto, los radares poseen una antena con capacidad de movimiento en múltiples direcciones. Esto permite abarcar la mayor cantidad del espectro en cuanto a la captura de datos a su alrededor. Esta capacidad de movimiento en diferentes direcciones facilita el escaneo de la atmósfera en diferentes geometrías:

- Escaneo en **volumen**
- Escaneo en **azimut**
- Escaneo en **elevación**

Antes de entrar en el análisis de cada tipo de escaneo, es necesario aclarar algunos conceptos que serán de utilidad para comprender en detalle la información brindada.

#### 3.1.1. Movimientos de la antena

Como se ha mencionado anteriormente, el radar emite continuamente señales electromagnéticas que al ser devueltas al radar, se procesan para obtener información acerca de los objetivos interceptados por las mismas en su recorrido. Estas señales pueden ser disparadas en cualquier dirección, que se identifica por un **grado de azimut** y un **grado de elevación**, alrededor del radar (debemos tener en cuenta las limitaciones físicas existentes en el movimiento de la antena):

- **Horizontalmente:** La antena puede moverse por los  $360^\circ$ , lo equivalente a un giro completo, si se observa al radar desde la parte superior (Figura 11).
- **Verticalmente:** La antena se mueve desde los  $-2^\circ$  a los  $182^\circ$  de elevación. Suponer la vista frontal del radar, sobre la superficie (Figura 12).

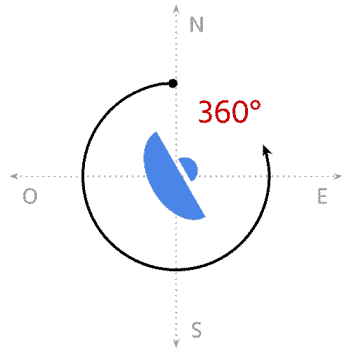


Figura 11: Movimiento horizontal de la antena (vista superior)

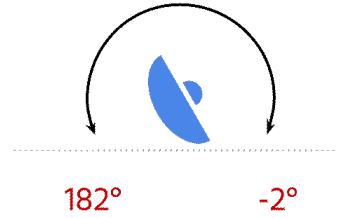


Figura 12: Movimiento vertical de la antena (vista frontal)

### 3.1.2. Unidades y medidas

En lo que respecta a la representación de los datos a nivel tridimensional podemos destacar lo siguiente:

- Un **rayo** (*ray* en inglés) representa un vector de información obtenido por cada señal emitida al exterior, en una determinada dirección. Suele representarse un rayo por grado entero en el plano horizontal (máximo: 360).
- La información contenida en el vector que representa al rayo, se divide en unidades denominadas **bins**, o lo que es lo mismo para las posteriores definiciones: **volumen de muestreo**. Las dimensiones están dadas por la relación existente entre el ancho de haz de la antena, la distancia del volumen de muestreo desde la antena y la longitud, configurada para cada escaneo, en relación a la dirección del vector del rayo.
- El conjunto de datos obtenidos al realizar un giro horizontal de 360° suele denominarse **corte horizontal** o **slice**. Según lo expresado con anterioridad y a modo de ejemplo, un corte horizontal se compone de:
  - 480 **volúmenes de muestreo** por cada rayo.
  - 0,5 metros de longitud por cada volumen de muestreo, con respecto al vector con dirección a la antena.

Esta configuración (tomada como ejemplo) arrojaría un corte horizontal de 240 km de radio con centro en la antena del radar.

### 3.1.3. Geometrías de escaneo

#### Escaneo en volumen

En esta geometría de escaneo, la antena del radar realiza rotaciones horizontales, cubriendo los  $360^\circ$  del espacio físico circundante al radar, incrementando la elevación actual en ciertos grados de incremento, como se aprecia en la figura Figura 13. En este caso, el concepto de “volumen” viene dado por la cobertura que se hace sobre el área en tres dimensiones que rodea al radar (Figura 13).

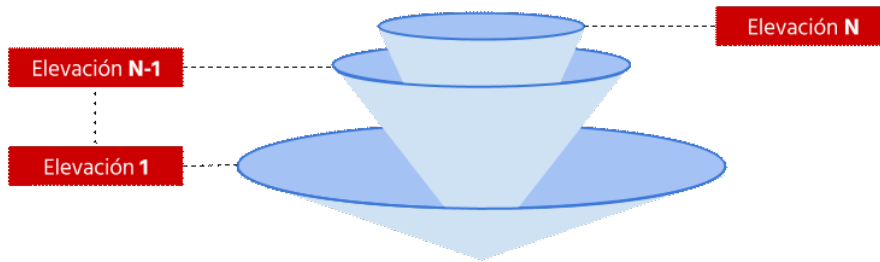


Figura 13: Elevaciones incrementales sobre un escaneo en volumen

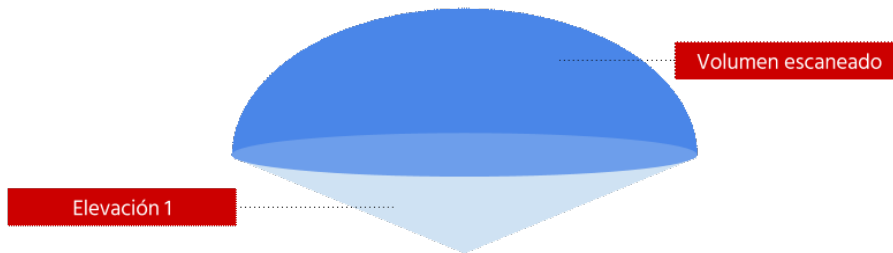


Figura 14: Volumen de escaneo cubierto por el radar

Como resultado del escaneo obtenemos un volumen de información  $V$  compuesto por  $N_c$  cortes horizontales iguales a la cantidad de elevaciones elegidas en la configuración. Cada corte se compone de  $N_r$  rayos que a su vez contienen  $N_v$  volúmenes de información. En este sentido se obtiene un volumen total que comprende:

$$V = N_c N_r N_v$$

unidades de información disponibles en el volumen escaneado.

La configuración de este escaneo permite establecer, entre otros parámetros:

- Número de elevaciones a escanear, entre 2 y 20.
- Grado de la elevación más baja, entre  $-2^\circ$  y  $89.9^\circ$ .
- Grado de la elevación más alta, entre  $-2^\circ$  y  $89.9^\circ$ .
- Distancia de barrido (o de cobertura), entre 25 km y 500 km.

### Escaneo en azimut

Este escaneo se realiza de manera similar al escaneo de volumen, con la salvedad de que se opera sobre un solo corte horizontal en una elevación previamente establecida. De esta forma se obtiene un conjunto de datos  $V$  equivalentes a un corte horizontal compuesto de  $N_r$  rayos radiales, que finalmente contienen  $N_v$  volúmenes de información:

$$V = N_r N_v$$

Existe la posibilidad de realizar un escaneo sobre una parte del horizonte del radar, denominado **escaneo en sector** (o *sector scan* en inglés), es decir, para una elevación dada se escanea desde un cierto azimut mínimo a otro máximo, dentro de los  $360^\circ$  que conforman un corte horizontal. Para este caso, el radar admite los siguientes parámetros de configuración:

- Distancia de barrido (o de cobertura), entre 25 km y 500 km.
- Grados de la elevación a escanear, entre  $-2^\circ$  y  $89.9^\circ$ .

En el caso de necesitarse un escaneo en sector, deberán indicarse:

- Grado de azimut de inicio, entre  $0^\circ$  y  $360^\circ$ .
- Grado de azimut de fin, entre  $0^\circ$  y  $360^\circ$ .

En la Figura 15 podemos ver lo que conforma un corte horizontal de un escaneo en azimut realizado.

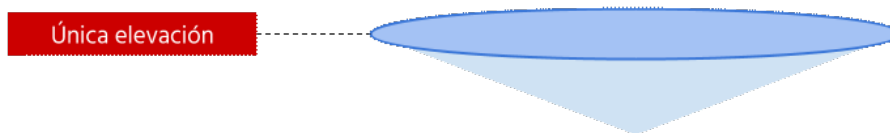


Figura 15: Escaneo en azimut. Un solo corte horizontal/elevación cubierta

### Escaneo en elevación

A diferencia de los escaneos anteriores que se basan en su mayoría en los movimientos de rotación horizontal de la antena, en un escaneo en elevación, la antena gira verticalmente, manteniendo un grado de azimut fijo. En términos informales podemos ver que se realiza un “corte vertical” de la situación sobre el radar, como se puede apreciar en la Figura 16:

Si se utiliza esta estrategia, pueden configurarse los siguientes parámetros:

- Distancia de barrido (o de cobertura), entre 25 km y 500 km.

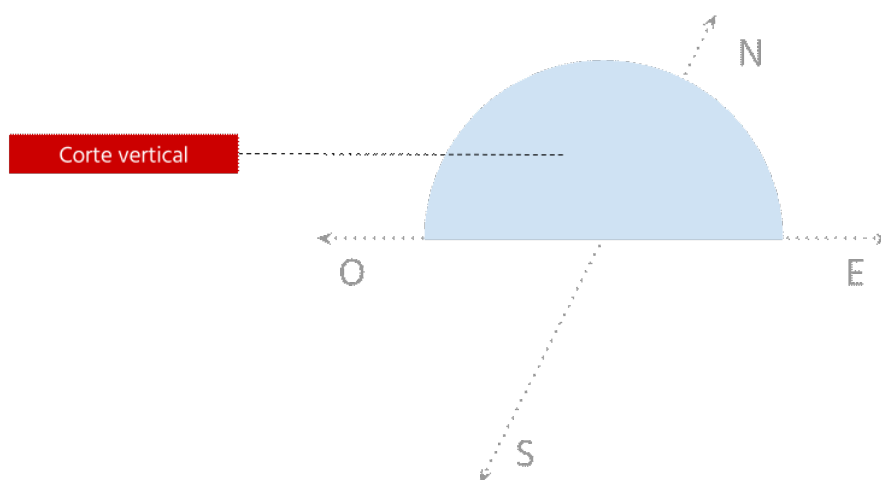


Figura 16: Escaneo en elevación. Un solo corte vertical (en la figura, de Este a Oeste)

- Azimut del cual se realiza el corte vertical, entre  $0^\circ$  y  $360^\circ$ .
- Grado de elevación inicial, entre  $-2^\circ$  y  $182^\circ$ .
- Grado de elevación final, entre  $-2^\circ$  y  $182^\circ$ .

#### 3.1.4. Configuración actual

Actualmente, los radares de la red de INTA, están configurados para operar bajo tareas de escaneo programadas, que producen los datos necesarios para ser consumidos y analizados por los especialistas de la organización. Ésta recolección de datos se comporta de la siguiente manera:

- Se utiliza el **escaneo en volumen**.
- Cada escaneo se repite en intervalos de **10 minutos**.
- El grado de elevación para el **primer corte horizontal** es de  **$0.5^\circ$** .
- El **último corte** se sitúa a los  **$15.1^\circ$**  de elevación.
- En total el escaneo se compone de **12 cortes horizontales**.
- Los radios de cobertura utilizados para estos escaneos son de **120 km, 240 km o 480 km**.

Más allá de que la red de radares cumpla con su tarea de escaneo diaria, también suelen utilizarse otros tipos de escaneos como el **escaneo en azimut**, el cual bajo ciertos criterios focaliza su atención en una elevación determinada. Una gran utilidad de este modo de barrido, por ejemplo, es la obtención de

datos en la primera elevación, ya que permite apreciar lo que ocurre en alturas cercanas a la superficie.

### 3.2. Almacenamiento y acceso

Como resultado de las tareas de escaneo programadas, el sistema de control del radar, elabora archivos en los que almacena los datos en diferentes formatos, dependiendo del tipo de escaneo: en **volumen**, en **azimut** o en **elevación** y de los productos que fueron analizados. Estos formatos de almacenamiento de datos están restringidos al criterio de la empresa fabricante de los radares y se detallan en el capítulo 4.

Posterior a la elaboración de los distintos archivos, se procede al almacenamiento. El mismo se da lugar en el espacio de disco del servidor de la estación de radar. La denominación **servidor** se aplica con la finalidad de denotar el esquema de comunicación que suele utilizarse para finalmente obtener los archivos resultantes. Es decir, las estaciones de radares suelen ubicarse en un punto tal que facilite, por ejemplo, la extensión de la red de radares ya instalada, o bien con intenciones de cubrir un área específica de interés por parte de la entidad que haga uso de los mismos.

Assumiendo esta condición, los consumidores de los resultados de los escaneos realizados, que interpretan estos archivos, no necesariamente se encuentran en una ubicación cercana al almacenamiento del radar, como para extraerlos personalmente en algún dispositivo de almacenamiento portátil.

Es por esto que esta clase de radares cuenta con un **servidor FTP** al cual los **clientes** se conectan para obtener los archivos en sus estaciones de trabajo, tal vez ubicadas a cientos de kilómetros de distancia de la ubicación del radar. El software que controla actualmente los radares incluye, dentro del conjunto de módulos que ofrece, una aplicación cliente para facilitar la conexión al servidor utilizando el protocolo FTP como base. Este servicio que forma parte del software de control de los radares, destinado a la transferencia de datos vía FTP, se denomina RainFTP. Igualmente es posible conectarse al servidor del radar utilizando otro tipo de cliente FTP convencional para obtener entre otros contenidos, los volúmenes de información escaneados.

En la figura Figura 17 puede verse este esquema de comunicación de forma gráfica.



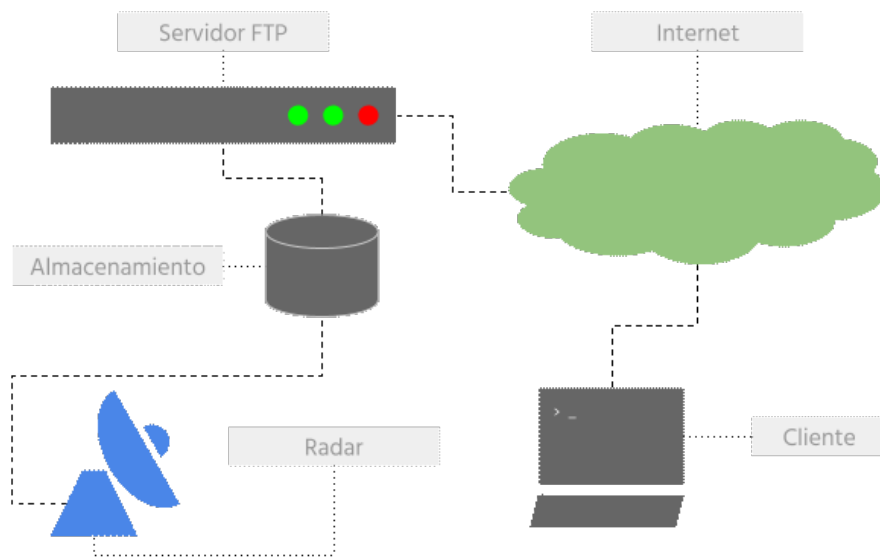


Figura 17: Esquema de almacenamiento y acceso a los datos mediante un cliente y servidor FTP

## 4. Procesamiento

Continuando con el proceso que cubre la manipulación de los datos de radar, en el capítulo 3 se ha analizado cómo se obtienen los datos mediante el **escaneo** y posteriormente cómo los mismos son **almacenados** y **accedidos**. Lo que sigue es analizar el formato de los archivos que contienen los datos efectivos de los escaneos de los radares de INTA, que serán la entrada o *input* de la siguiente etapa: el **procesamiento**, como se ilustró en la Figura 10. Es aquí donde interviene la aplicación Meteorax mencionada con anterioridad, producto del trabajo de investigación de esta tesina.

La clase de radares utilizados para el desarrollo de esta tesina, genera diferentes tipos de archivos. Una clase de archivos son los clasificados como **de registro**, que muestran información de eventos ocurridos relacionados al funcionamiento del radar. Por otro lado se encuentran los archivos **de resultados**, que contienen información acerca de los escaneos propios del radar, así como los valores obtenidos de las variables analizadas.

### 4.1. Archivos de registro

#### 4.1.1. Registro Rainbow

Son escritos en dos formatos diferentes:

- Registro **Rainbow XML**

Archivo con el formato **XML** <sup>1</sup> que contiene una secuencia de etiquetas `<log />` conteniendo el mensaje producido por el evento y las variables requeridas por dicho mensaje. El nombre de archivo que tiene se define bajo el siguiente formato: **aaaammdd.log**, en donde **aaaa**, **mm** y **dd** corresponden al año, mes y día, respectivamente, en el que se generó el registro.

- Registro **Rainbow ASCII**

Cumple la misma función que el formato previo, salvo que este se encuentra en formato **ASCII** <sup>2</sup>. Cada línea de este registro contiene un mensaje completo y los campos de este mensaje se dividen por un punto y coma. Se generan sólo si se configura este formato de salida, es decir, son opcionales. El nombre del archivo, siguiendo el formato anterior posee una extensión **txt: aaaammdd.log.txt**.

---

<sup>1</sup>en inglés eXtensible Markup Language, lenguaje de enmarcado basado en un árbol de etiquetas

<sup>2</sup>en inglés American Standard Code for Information Interchange, codificación de caracteres utilizada para intercambio de información basada en el alfabeto latino

### 4.1.2. Registro BITE

El formato del contenido de este tipo de registro es **csv** y se compone de un archivo con los sucesos que alteraron algún estado BITE<sup>3</sup> del Lenguaje de Control de Radar, (**RCL, Radar Control Language**), desarrollado por Gematronik para el control del radar. Su nombre de archivo lleva un formato establecido por: **<radar ID>\_BITEaammdd.csv**, en donde **aa**, **mm** y **dd** identifican al año, mes y día, respectivamente, en el que se generó el registro y **<radar ID>** corresponde al identificador utilizado en la configuración del radar.

### 4.1.3. Registro RainAdmin

Archivo con formato de nombre: **RainADMIN\_aaaammdd.log** que se crea diariamente con el objetivo de almacenar los mensajes arrojados por la aplicación RainAdmin. Esta aplicación comprueba ciertos directorios cada 60 minutos, por ejemplo. Luego de este análisis los resultados son escritos en este archivo.

### 4.1.4. Registro RainRCS (Rainbow® Control Center)

La aplicación RainRCS genera dos archivos:

- **RainRCS\_aaaammdd.log**: contiene el historial de comandos y el registro de mensajes de los eventos de los procesos (comienzos / reinicios / paradas / terminaciones).
- **RainRcsStatus.txt**: correspondiente al estado actual de la aplicación RainRCS, este estado incluye información acerca del modo standby y de una serie de procesos bajo la supervisión de RainRCS. Este archivo se actualiza cada 15 segundos.

## 4.2. Archivos de resultados

Como se ha mencionado con anterioridad, estos archivos representan la forma básica de volcado de datos por parte de los radares meteorológicos de la red de radares de INTA. Con esta estrategia, los datos permanecen almacenados en el servidor de cada radar para que, en etapas posteriores, sean accedidos mediante un medio como **internet**, utilizando además el protocolo **FTP**.

Una vez obtenidos los archivos, será momento de procesarlos para extraer los datos relevantes, y para este procesamiento, primero es necesario comprender y analizar el formato con el cual los desarrolladores del software controlador del radar decidieron escribir los resultados.

---

<sup>3</sup>**Built-In Test Equipment**, componente en sistemas generalmente de aviación con el fin de contribuir al proceso de testeo, mantenimiento y corrección de errores de un sistema

Esta clase de archivos se almacena en el formato **XML**, de las siglas en inglés **eXtensible Markup Language** y se compone de dos secciones bien definidas:

- Sección **XML**
- Sección de **datos en binario**

#### 4.2.1. Sección XML

En primer lugar se detallan algunas características básicas del formato XML:

- Es un meta-lenguaje de formato abierto para la representación de **información estructurada**.
- Mediante el mismo, se pueden definir lenguajes de enmarcado (Ejemplo: **HTML**).
- Desarrollado por la **World Wide Web Consortium**.
- Actualmente cuenta con dos estándares.
- Se basa en una estructura de **árbol de etiquetas**.
- Formalmente se utiliza la extensión **.xml** para los archivos con este formato.

Un ejemplo de un archivo XML podría ser el siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<elemento>
  <hijo nombre="Primer hijo">1</hijo>
</elemento>
```

Sección de código 1: Ejemplo de archivo XML

Continuando con el formato que utiliza este tipo de archivos, esta sección contiene datos en formato XML capaces de ser procesados por un lector/procesador XML convencional. La organización de las etiquetas dentro de esta sección depende del tipo de archivo con el que se esté trabajando, **archivos de datos en bruto** o **archivos de productos**.

#### 4.2.2. Sección de datos en binario

Esta sección se caracteriza por contener datos que no son manipulables por un procesador XML corriente. El motivo de esta diferencia radica en el tipo de contenido: **datos en binario**. El problema con este tipo de datos y el formato XML, es que existen caracteres que pueden romper el procesamiento del archivo, encontrando por ejemplo, etiquetas sin cerrar.

Para determinar la finalización de los datos en formato puro XML que no rompen el procesamiento se inserta una línea, en forma de comentario, inmediatamente **después de la sección XML** conteniendo lo siguiente:

```
<!-- END XML -->
```

#### Sección de código 2: Comentario de finalización de datos

Cabe aclarar que sólo los archivos que contengan datos en binario a continuación de la sección en XML contendrán la anterior etiqueta, que no es más que un comentario útil para los usuarios de este tipo de archivo.

La parte binaria está compuesta por una o más secciones, encapsuladas como etiquetas **<BLOB />**. Si bien se utiliza la etiqueta **<BLOB />** para los datos en binario, esto no significa que pueda ser leída por un procesador de XML ya que por ejemplo, accidentalmente, esta etiqueta puede contener a su vez otras etiquetas que mal-formen el árbol XML y hagan imposible trabajar con la información. El valor para esta etiqueta contiene los datos como **BLOB** (Binary Large Objects), es decir, los datos en formato binario que pueden ser de gran tamaño y que se encuentran delimitados en este caso por el byte con valor 0x0A.

A continuación se presenta un ejemplo de una etiqueta **<BLOB />**, en la que podemos observar:

- La etiqueta junto con sus parámetros resaltadas con azul.
- El valor de la etiqueta en verde.
- Los caracteres delimitadores (0x0A) en rojo.

```
00 3c 42 4c 4f 42 20 62 6c 6f 62 69 64 3d 22 30 → <BLOB blobid="0
15 22 20 73 69 7a 65 3d 22 31 30 22 20 63 6f 6d → " size="10" com
30 70 72 65 73 73 69 6f 6e 3d 22 4e 6f 6e 65 22 → pression="None"
45 20 3e 0a 31 32 33 34 35 36 37 38 39 30 0a 3c → >.1234567890.<
60 2f 42 4c 4f 42 3e → /BLOB>
```

La etiqueta **<BLOB />** contiene los siguientes atributos:

- **blobid**: Nombre que identifica esta sección **<BLOB />**. Todos las secciones de datos en binario de un archivo son referenciadas siempre en la sección XML situada al principio del mismo, mediante el atributo **blobid** situado en el elemento que referencia al **<BLOB />**.
- **size**: Tamaño del **<BLOB />** en bytes (sin contar los bytes referidos al carácter de control para nueva línea).
- **compression**: Nombre del algoritmo utilizado para comprimir los datos en BLOB. Se utiliza el valor "none" para indicar que los datos no están comprimidos.

### 4.2.3. Archivos de datos de producto

Los archivos enmarcados en esta clasificación proveen información de escaneos con un nivel de procesamiento adicional, que permite la visualización de los datos de una manera más directa que los archivos de datos en bruto, los cuales requieren de transformaciones adicionales para ser mostrados de forma correcta al usuario.

La desventaja de esta aproximación que provee el software controlador de los radares de INTA, es que las imágenes producidas a partir de esta clase de archivos, quedará sujeta a la interpretación y a las limitaciones gráficas del propio software que corra en el radar.

Dadas estas circunstancias, estos archivos no serán la fuente de entrada de la aplicación **Meteorax**, aunque se proveerá un análisis de los mismos, en la siguiente sección, que ayudará a comprender la organización de los datos producidos por los radares.

#### Formato de nombre

El formato de nombre para los archivos de datos de producto es el siguiente:

```
AAAAMDDhhmssNN<tipo-dato>.<tipo-producto>
```

Sección de código 3: Formato de nombre de un archivo de datos de producto

En donde cada sección corresponde con la siguiente información:

- Fecha de escritura del archivo.
  - **AAAA**: Año en 4 dígitos.
  - **MM**: Mes en 2 dígitos.
  - **DD**: Día en 2 dígitos.
  - **hh**: Hora en 2 dígitos.
  - **mm**: Minutos en 2 dígitos.
  - **ss**: Segundos en 2 dígitos.
- **NN**: Contador para distinguir archivos con la misma fecha.
- **<tipo-dato>**: Tipo de dato del producto (dBZ, V, W, ZDR, etc.).
- **<tipo-producto>**: Tipo de producto (ppi, cappi, max, etc.).

#### Ejemplo

2	0	1	6	0	2	0	7	0	8	0	9	1	3	0	0	d	B	Z	.	p	p	i
1	2	3	4	5	6	7	8	9	10													

El anterior nombre corresponde a un archivo de un **único escaneo** para esta fecha, ya que el contador de escaneos (7 y 8) es igual a 0, realizado el **07/02/2016** (1, 2 y 3) a las **08:09:13** (4, 5 y 6) del producto **ppi** (10) que contiene información acerca del tipo de dato **reflectividad** o **dBZ** (9).

### Formato de contenido

La estructura del árbol de etiquetas para este tipo de archivo es la siguiente:

```
<product
  datatype= ""
  datetime= ""
  name= ""
  owner= ""
  type= ""
  version= "" >

  <history />
  <alerts />
  <data />
</product>
```

Sección de código 4: Archivo de datos de producto

En la cual se corresponde:

- **datatype:** (Atributo) Tipo de dato del producto, vacío si hay más de uno.
- **datetime:** (Atributo) Fecha de la última modificación o creación.
- **name:** (Atributo) Nombre del producto.
- **owner:** (Atributo) Nombre de usuario.
- **type:** (Atributo) Tipo de producto.
- **version:** (Atributo) Número de versión del archivo.
- **history:** (Etiqueta) Historial de los datos obtenidos y los productos generados.
- **alerts:** (Etiqueta) Información de alertas.
- **data:** (Etiqueta) Información necesaria para mostrar el producto.

#### Etiqueta `<history />`

La estructura de esta sub-etiqueta del elemento `<product />` será diferente para productos simples y productos compuestos. Las diferencias pueden verse a continuación.

```

<history
  pdfname= " "
  ppdfname= " "
  sdfname= " " >

  <rawdatafiles>
    <file />
  </rawdatafiles>

  <preprocessedfiles>
    <file />
  </preprocessedfiles>

  <scheduler datetime= " " >
    <task
      type= " "
      name= " " >

      <nodeid />
    </task>
  </scheduler>
</history>

```

Sección de código 5: Etiqueta `history` (archivo de datos de producto simple)

Para la anterior estructura se tiene:

- **pdfname:** (Atributo) Nombre del archivo de la definición de producto.
- **ppdfname:** (Atributo) Nombre del archivo de la definición del pre-proceso (opcional).
- **sdfname:** (Atributo) Nombre del archivo de la definición de escaneo.
- **rawdatafiles:** (Etiqueta) Lista de archivos de datos en bruto utilizados.
  - **file:** (Etiqueta) Nombre de archivo y ruta de acceso.
- **preprocessedfiles:** (Etiqueta) Lista de archivos de datos pre-procesados utilizados.
  - **file:** (Etiqueta) Nombre de archivo y ruta de acceso.
- **scheduler:** (Etiqueta) Contiene la información de planificación.
  - **datetime:** (Atributo) Fecha y hora del planificador.
  - **task:** (Etiqueta) Provee información de una tarea.
    - **type:** (Atributo) Tipo de tarea.
    - **name:** (Atributo) Nombre de la tarea.



- **name:** (Etiqueta) Contiene información acerca del nodo de la tarea.

```

<history
  pdfname= ""
  ppdfname= ""
  sdfname= "" >

  <pdffiles>
    <file />
  </pdffiles>

  <radarinfo
    alt= ""
    lon= ""
    lat= ""
    id= "" >

    <name />
    <wavelen />
    <beamwidth />
  </radarinfo>
</history>

```

Sección de código 6: Etiqueta `history` (archivo de datos de producto compuestos)

Para la anterior estructura se corresponde:

- **pdfname:** (Atributo) Nombre del archivo de la definición de producto.
- **ppdfname:** (Atributo) Nombre del archivo de la definición del pre-proceso (opcional).
- **sdfname:** (Atributo) Nombre del archivo de la definición de escaneo.
- **pdffiles:** (Etiqueta) Lista de archivos de productos de entrada del radar.
  - **file:** (Etiqueta) Nombre de archivo y ruta de acceso.
- **radarinfo:** (Etiqueta) Ubicación de los radares individualmente.
  - **alt:** (Atributo) Altura de la antena en metros sobre el nivel del mar.
  - **lon:** (Atributo) Longitud geográfica de la ubicación del radar.
  - **lat:** (Atributo) Latitud geográfica de la ubicación del radar.
  - **id:** (Atributo) Identificador del radar (Usualmente se utilizan 3 letras).
  - **name:** (Etiqueta) Nombre de la ubicación del radar.

- **wavelen:** (Etiqueta) Longitud de onda del transmisor del radar, en metros.
- **beamwidth:** (Etiqueta) Ancho del haz de la antena del radar, en grados.

Etiqueta `<alerts />`

```
<alerts
  level="" >

  <table2d>
    <table1d>
      <pdtype />
      <datatype />
      <peak_x />
      <cell
        lon=""
        lat=""
      />
    </table1d>
  </table2d>
</alerts>
```

Sección de código 7: Etiqueta `alerts`

Por último, el anterior árbol de etiquetas con sus atributos, corresponde a:

- **level:** (Atributo) Nivel de alerta: “none”, “low”, “medium”, “high”.
- **table2d:** (Etiqueta) Tabla de 2 dimensiones. Contiene un conjunto de elementos `<table1d />`
  - **table1d:** (Etiqueta) Tabla con los datos del alerta.
    - **pdtype:** (Etiqueta) Tipo de producto (ppi, capped, max, etc.).
    - **datatype:** (Etiqueta) Tipo de dato del producto (dBZ, V, W, ZDR, etc.).
    - **peak\_x:** (Etiqueta) Valor tope dentro del área de alerta.
    - **cell:** (Etiqueta) Celdas referenciadas por `<lat />` (latitud) y `<lon />` (longitud).
      - ◇ **lon:** (Atributo) Valores de longitud de los puntos característicos que conforman el polígono de alerta (en grados).
      - ◇ **lat:** (Atributo) Valores de latitud de los puntos característicos que conforman el polígono de alerta (en grados).

Etiqueta `<data />`

```

<data
  time= " "
  date= " " >

  <datainfo />
  <viewparams />
  <radarinfo />

  <radarpicture
    placeid= " "
    type= " "
    min= " "
    name= " "
    max= " " >

    <datamap
      blobid= " "
      rows= " "
      columns= " "
      depth= " " />

    <flagmap
      blobid= " "
      rows= " "
      columns= " "
      depth= " " >

      <flag />
    </flagmap>

    <projection />
  </radarpicture>

  <table2d>
    <table1d />
  </table2d>
</data>

```

Sección de código 8: Etiqueta `data`

La anterior estructura contiene:

- **time:** (Atributo) Hora más importante para este producto (Ej.: hora de escaneo).
- **date:** (Atributo) Fecha más importante para este producto (Ej.: fecha de

escaneo).

- **datainfo**: (Etiqueta) Información adicional del producto. No son ajustes presentes en `<viewparams />`
- **viewparams**: (Etiqueta) Parámetros para mostrar en la leyenda de la imagen del producto.
- **radarinfo**: (Etiqueta) Información acerca de la ubicación del radar. Ídem a la etiqueta `<radarinfo />` explicada con anterioridad.
- **radarpicture**: (Etiqueta) Información acerca de la imagen.
  - **placeid**: (Atributo) Tipo de imagen.
  - **type**: (Atributo) Tipo de dato.
  - **min**: (Atributo) Menor valor mostrado.
  - **max**: (Atributo) Mayor valor mostrado.
  - **name**: (Atributo) Nombre de la fuente de la información.
  - **datamap**: (Etiqueta) Arreglo de datos de la imagen
    - **blobid**: (Atributo) Identifica una sección en la parte binaria.
    - **rows**: (Atributo) Número de filas de información que componen la imagen (Altura de la imagen).
    - **columns**: (Atributo) Número de columnas de información que componen la imagen (Ancho de la imagen).
    - **depth**: (Atributo) Número de bits utilizados para describir un punto de los datos.
  - **flagmap**: (Etiqueta) Conjunto de flags<sup>4</sup>
    - **blobid**: (Atributo) Identifica una sección en la parte binaria (mapa de píxeles)
    - **rows**: (Atributo) Número de filas de información que componen el arreglo de flags (Altura de la imagen).
    - **columns**: (Atributo) Número de columnas de información que componen el arreglo de flags (Ancho de la imagen).
    - **depth**: (Atributo) Número de bits utilizados para describir un píxel.
    - **flag**: (Etiqueta) Descripción e información acerca de un flag.
  - **projection**: (Etiqueta) Contiene varios parámetros de la proyección, como latitud y longitud de las cuatro esquinas de la imagen, dimensión, tipo de proyección, etc.
- **table2d**: (Etiqueta) Tabla de 2 dimensiones con la información de los datos escalares resultantes. Contiene un conjunto de elementos hijos, denominados `<table1d />` :
  - **table1d**: (Etiqueta) Tabla con los datos escalares resultantes.

---

<sup>4</sup>También se puede nombrar como **notificación** o **bandera**, por la naturaleza de su funcionalidad.

#### 4.2.4. Archivos de datos en bruto

Esta clase de archivos, a diferencia de los archivos de datos de producto, provee la información *cruda* del escaneo. Es decir, los datos no ha sufrido modificación alguna más que la necesaria para transformar las señales físicas en datos que serán volcados en estos archivos.

Sabiendo esto, es importante destacar la importancia de estos datos *en bruto* que serán el punto de entrada de la aplicación **Meteorax**.

Cada archivo de datos en bruto contiene la información sobre un escaneo realizado, dependiendo del tipo de escaneo en particular. Se genera un archivo distinto por cada tipo de producto analizado, es decir, existirán para una misma fecha uno o más archivos resultantes del mismo escaneo.

Existen tres tipos de archivos de datos en bruto y los mismos dependen de la estrategia de escaneo que se realice:

- Escaneo en **volumen**
- Escaneo en **azimut**
- Escaneo en **elevación**

Estos tipos de escaneos se han analizado en la sección 3.1.

Como se verá en la siguiente sección, de la elección de uno de estos tres métodos para obtener los datos escaneados dependerá la extensión de los archivos a generar: **.azi**, **.ele** o **.vol**.

#### Formato de nombre

Para representar los archivos en el almacenamiento se utiliza el siguiente formato de nombre:

```
AAAAAMDDhhmmsNN<tipo-dato>.<tipo-escaneo>
```

Sección de código 9: Formato de nombre de un archivo de datos en bruto

- Fecha de escritura del archivo.
  - **AAAA**: Año en 4 dígitos.
  - **MM**: Mes en 2 dígitos.
  - **DD**: Día en 2 dígitos.
  - **hh**: Hora en 2 dígitos.
  - **mm**: Minutos en 2 dígitos.
  - **ss**: Segundos en 2 dígitos.

- **NN**: Contador para distinguir archivos con la misma fecha.
- **<tipo-dato>**: Tipo de dato del producto (dBZ, V, W, ZDR, etc.).
- **<tipo-escaneo>**: Tipo de escaneo (azi, vol o ele).

### Ejemplo

```
2 0 1 6   0 2   0 5   1 0   0 6   1 1   0 0   d B Z . v o l
  1       2       3       4       5       6       7       8       9       10
```

El anterior nombre corresponde a un archivo de un **único escaneo**, ya que el contador de escaneos (7 y 8) es igual a 0, realizado el **05/02/2016** (1, 2 y 3) a las **10:06:11** (4, 5 y 6) del producto **Reflectividad** o **dBZ** (9) y mediante un tipo de escaneo **en volumen**, dada su extensión *.vol*.

### Formato de contenido

En este tipo de archivo, el árbol de etiquetas fue diseñado de la siguiente manera:

```
<volume
  version=""
  datetime=""
  type=""
  owner="" >

  <scan />
  <radarinfo />
</volume>
```

Sección de código 10: Archivo de datos en bruto

Que se corresponde con la siguiente especificación:

- **version**: (Atributo) Número de versión de este archivo.
- **datetime**: (Atributo) Fecha de la última modificación o creación.
- **type**: (Atributo) Tipo de escaneo.
- **owner**: (Atributo) Nombre de usuario.
- **type**: (Atributo) Tipo de producto.
- **scan**: (Etiqueta) Contiene los parámetros de escaneo.
- **radarinfo**: (Etiqueta) Ubicación del radar.

**Etiqueta `<scan />`**

```

<scan
  name= " "
  time= " "
  date= " " >

  <unitid />
  <advancedchanged />
  <detailedchanged />
  <scantime />
  <pargroup />
  <slice />
</scan>

```

Sección de código 11: Etiqueta `scan`

Para la anterior estructura se tiene:

- **name:** (Atributo) Nombre del archivo de la definición del escaneo.
- **time:** (Atributo) Hora de recolección de los datos (Se encuentra en el nombre del archivo).
- **date:** (Atributo) Fecha de recolección de los datos (Se encuentra en el nombre del archivo).
- **unitid:** (Etiqueta) Sistema de unidades utilizado.
- **advancedchanged:** (Etiqueta) Indicador de uso del complemento “opciones avanzadas” para generar el archivo de definición de escaneo.
- **detailedchanged:** (Etiqueta) Indicador de uso del complemento “opciones detalladas” para generar el archivo de definición de escaneo.
- **scantime:** (Etiqueta) Tiempo estimado de escaneo, en segundos.
- **pargroup:** (Etiqueta) Contiene todos los parámetros generales de escaneo, configurados además en el archivo de definición de escaneo.
- **slice:** (Etiqueta) Parámetros específicos de escaneo para un corte horizontal. En un escaneo en volumen, habrá múltiples instancias de esta etiqueta (una por cada corte horizontal que se realice).

**Etiqueta `<radarinfo />`**

Ésta etiqueta se ha visto con anterioridad para los archivos de datos de producto. Para los archivos de datos en bruto, posee la misma estructura:

```
<radarinfo
  alt= ""
  lon= ""
  lat= ""
  id= "" >

  <name />
  <wavelen />
  <beamwidth />
</radarinfo>
```

Sección de código 12: Etiqueta `radarinfo`

Para la anterior estructura se tiene:

- **alt:** (Atributo) Altura de la antena en metros sobre el nivel del mar.
- **lon:** (Atributo) Longitud geográfica de la ubicación del radar.
- **lat:** (Atributo) Latitud geográfica de la ubicación del radar.
- **id:** (Atributo) Identificador del radar (Usualmente se utilizan 3 letras).
- **name:** (Etiqueta) Nombre de la ubicación del radar.
- **wavelen:** (Etiqueta) Longitud de onda del transmisor del radar, en metros.
- **beamwidth:** (Etiqueta) Ancho del haz de la antena del radar, en grados.

Etiqueta `<pargroup />` ( `<scan />` )

Dentro de esta etiqueta se encuentran los parámetros de configuración generales de escaneo. A continuación se presentan aquellos que resultan más relevantes para la visualización de la información por parte de la aplicación **Meteorax**:



```

<pargroup
  refid="" >

  <stoprange />
  <numele />
  <firstele />
  <lastele />
  <posele />
  <startazi />
  <stopazi />
  <posazi />
  <datatypez />
  <datatypeuz />
  <datatypev />
  <datatypew />
  <datatypezdr />
  <datatypephidp />
  <datatypekdp />
  <datatypeukdp />
  <datatyperhohv />
  <datatypeldr />
  <rangestep />
  <rangesamp />
  <start_range />
  <antspeed />
  <anglestep />

  <!-- Existen más parámetros -->
</pargroup>

```

Sección de código 13: Etiqueta `pargroup`

- **stoprange:** (Etiqueta) Distancia de alcance del escaneo, para la cual adquirir datos (en metros).
- **numele:** (Etiqueta) Número de elevaciones realizadas<sup>5</sup>.
- **firstele:** (Etiqueta) Grado de la primera elevación<sup>5</sup>.
- **lastele:** (Etiqueta) Grado de la última elevación<sup>5</sup>.
- **posele:** (Etiqueta) Grado de elevación<sup>6</sup>.
- **startazi:** (Etiqueta) Grado de inicio de azimut<sup>6</sup>.
- **stopazi:** (Etiqueta) Grado de fin de azimut<sup>6</sup>.

---

<sup>5</sup>Sólo para escaneo en volumen

<sup>6</sup>Sólo para escaneo en azimut

- **datatypez**: (Etiqueta) Indicador de tipo de dato **Z**.
- **datatypeuz**: (Etiqueta) Indicador de tipo de dato **uZ**.
- **datatypev**: (Etiqueta) Indicador de tipo de dato **V**.
- **datatypew**: (Etiqueta) Indicador de tipo de dato **W**.
- **datatypezdr**: (Etiqueta) Indicador de tipo de dato **ZDR**.
- **datatypephidp**: (Etiqueta) Indicador de tipo de dato **PhiDP**.
- **datatypekdp**: (Etiqueta) Indicador de tipo de dato **KDP**.
- **datatypeukdp**: (Etiqueta) Indicador de tipo de dato **uKDP**.
- **datatyperhoHV**: (Etiqueta) Indicador de tipo de dato **RhoHV**.
- **datatypeldr**: (Etiqueta) Indicador de tipo de dato **LDR**.
- **rangestep**: (Etiqueta) Resolución; tamaño de las unidades de muestreo en kilómetros.
- **start\_range**: (Etiqueta) Distancia de inicio de escaneo en kilómetros.
- **antspeed**: (Etiqueta) Velocidad de rotación de la antena en grados por segundo.
- **anglestep**: (Etiqueta) Salto entre ángulos, en grados.

**Etiqueta** `<slice />` ( `<scan />` )

Esta etiqueta representa al conjunto de parámetros configurados para un corte horizontal específico. La misma se verá más de una vez para el caso de los archivos que representan escaneos en volumen. A continuación, se presentan aquellos parámetros más relevantes:

```

<slice
  refid="" >
  <dynz min="" max="" />
  <dynv min="" max="" />
  <dynw min="" max="" />
  <dynzdr min="" max="" />
  <dynldr min="" max="" />
  <dynkdp min="" max="" />

  <posangle />
  <startangle />
  <stopangle />
  <stoprange />
  <start_range />
  <rangestep />
  <pw_index />
  <antspeed />
  <anglestep />

  <slicedata
    time=""
    date="" >

    <rayinfo
      refid=""
      blobid=""
      rays=""
      depth="" />

    <rawdata
      blobid=""
      rays=""
      type=""
      bins=""
      min=""
      max=""
      depth="" />
  </slicedata>

  <!-- Existen más parámetros -->
</slice>

```

Sección de código 14: Etiqueta `slice`

- **refid:** (Atributo) Contador de corte horizontal. Para el escaneo en volumen (.vol) identifica el número de elevación. Para archivos .azi y .ele será un valor fijo ("0").

- **dynz:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de reflectividad (Z) en dBZ.
- **dynv:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de velocidad radial (V) en m/s.
- **dynw:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de ancho espectral (W) en m/s.
- **dynzdr:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de reflectividad diferencial (ZDR) en dB.
- **dynldr:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de radio de de-polarización linear (LDR) en dB.
- **dynkdp:** (Etiqueta) Valor mínimo y máximo para el rango dinámico de los datos de fase diferencial específica (KDP) en grados/km.
- **posangle:** (Etiqueta) Ángulo de elevación en el caso de escaneo en volumen o en azimut. Para escaneo en elevación representa el ángulo de azimut.
- **startangle:** (Etiqueta) Ángulo de inicio para el caso de escaneo en volumen o en azimut. Para escaneo en elevación representa el ángulo de inicio de elevación.
- **stopangle:** (Etiqueta) Ángulo de fin para el caso de escaneo en volumen o en azimut. Para escaneo en elevación representa el ángulo de fin de elevación.
- **stoprange:** (Etiqueta) Distancia de alcance del escaneo, para la cual adquirir datos (en metros).
- **start\_range:** (Etiqueta) Distancia de inicio de escaneo en kilómetros (por defecto: “0”).
- **rangestep:** (Etiqueta) Resolución; tamaño de las unidades de muestreo en kilómetros.
- **pw\_index:** (Etiqueta) Ancho de pulso:
  - Corto: “0”.
  - Medio: “1”.
  - Largo: “2”.
- **antspeed:** (Etiqueta) Velocidad de rotación de la antena en grados por segundo.
- **anglestep:** (Etiqueta) Salto entre ángulos, en grados.
- **slicedata:** (Etiqueta) Contiene la información de los datos obtenidos del escaneo **para este corte horizontal**.
  - **time:** (Atributo) Hora de recolección de los datos (al finalizar el corte horizontal).

- **date:** (Atributo) Fecha de recolección de los datos.
- **rayinfo:** (Etiqueta) Parámetros y datos para los rayos de este corte horizontal.
  - **refid:** (Atributo) Valor fijado a “startangle”.
  - **blobid:** (Atributo) Identifica (referencia) a una sección de datos en binario que contiene la información en bruto de los rayos de este corte horizontal.
  - **rays:** (Atributo) Número de rayos almacenados en la sección en binario.
  - **depth:** (Atributo) Número de bits para un valor en el `<BLOB />`.
- **rawdata:** (Etiqueta) Información relativa a la sección de datos en bruto.
  - **blobid:** (Atributo) Identifica (referencia) a una sección de datos en binario que contiene la información en bruto.
  - **rays:** (Atributo) Número de rayos almacenados en la sección en binario.
  - **type:** (Atributo) Tipo de dato en bruto (dBZ, dBuZ, V, W, ZDR, etc.).
  - **bins:** (Atributo) Número de unidades de escaneo almacenadas por cada rayo. Cada unidad es representada por un valor con resolución **depth**.
  - **min:** (Atributo) Rango dinámico mínimo del tipo de dato.
  - **max:** (Atributo) Rango dinámico máximo del tipo de dato.
  - **depth:** (Atributo) Número de bits para un valor en el `<BLOB />`.

### 4.3. Tipos de archivos utilizados

Ahora que se ha analizado el formato de los archivos portadores de los datos que provienen del escaneo de los radares, se continuará con la siguiente etapa, de acuerdo al esquema de etapas (Figura 10): el **procesamiento**. Como se expone en dicho esquema, la actividad llevada a cabo por el procesamiento, toma como entrada sólo un subconjunto de archivos producidos por los radares:

- Archivos de **datos en bruto y escaneo en volumen**, los cuales poseen la extensión **.vol**.
- Archivos de **datos en bruto y escaneo en azimut**, con extensión **.azi**.

En primer lugar, la elección de datos en bruto se debe a que los datos contenidos en este tipo de archivo, si bien requieren una decodificación adicional, permiten trabajar con los datos con un alto grado de libertad, ya que representan datos *puros* provenientes de las señales captadas por el radar. De este modo es posible aplicar cualquier tipo de procesamiento posterior sobre los datos originales, como transformaciones, correcciones, transposiciones, etc. con el fin de lograr la correcta visualización de los mismos.

La elección de los archivos de escaneo en volumen y azimut se debe en parte a la forma en que los datos se representan en estos tipos de escaneo y su relación con las visualizaciones finales requeridas. Por ejemplo, un gráfico muy utilizado es aquel que muestra las condiciones meteorológicas en la primera elevación, por lo cual, tanto los archivos .azi y .vol son los indicados para proveer esta información.

Otro motivo por el cual estos dos tipos de archivos/escaneos resultan de gran utilidad es la cobertura que realizan sobre el área circundante al radar, como se pudo ver en la sección 3.1. Tanto el escaneo en azimut como en volumen son capaces de cubrir un radio de hasta 480 km, con lo cual, se puede obtener suficiente información acerca de los sucesos meteorológicos que rodean la zona del radar.

#### 4.4. Lectura

El primer paso del procesamiento es la lectura de los datos contenidos en los archivos con extensión .azi y .vol. Se ha mencionado con anterioridad que el formato en el que estos archivos fueron diseñados, es **XML**, pero con la salvedad de que además de organizar los datos en un árbol de elementos o etiquetas, aptos para el lenguaje, la empresa fabricante del radar y del software para controlarlo, ha incluido datos binarios como parte de estos archivos. Los datos binarios a los que se hace referencia son, ni más ni menos que los datos en bruto del resultado de los escaneos. Conforman una parte fundamental para el posterior análisis de la información y vale la pena tomar especial consideración a la hora de manipularlos.

Si bien estos datos en binario fueron ubicados dentro de la etiqueta denominada `<BLOB />`, el problema con datos en binario y el formato XML es que ningún procesador de XML podrá leer los datos de manera correcta sin producir un error, ya que es muy probable que algún conjunto de bytes contenidos en el BLOB, rompa el procesamiento del mismo.

Esta fue la primer dificultad encontrada a la hora de encarar un prototipo de aplicación, que involucra la lectura de formatos no estándares, o más bien, formatos personalizados por los desarrolladores de software, que extienden de otros conocidos, como en este caso: XML. Es por esto que para poder realizar de forma correcta la lectura de los archivos, se utilizaron dos librerías muy populares en el mundo del procesamiento de datos de radar, como lo son: **Py-ART** y **Wradlib**. Éstas últimas y otras librerías son analizadas en el capítulo siguiente.

##### 4.4.1. Utilización de librerías

Dentro de la aplicación Meteorax el proceso de “lectura” de los datos se refiere a la utilización de las librerías anteriormente mencionadas para obtener otra representación digital de los datos contenidos en los archivos de entrada. Estos datos son mantenidos en memoria mientras se trabajan y se encapsulan bajo un

modelo de datos personalizado por los desarrolladores de Py-ART denominado “Radar”.

Py-ART intenta cubrir varias funcionalidades relativas al procesamiento de datos de radar, parseo de archivos de entrada y visualización de productos, entre otras. Es por esto y por la vasta diversidad de formatos conocidos de modelado de datos de radar, que se requieren dependencias adicionales a la hora de operar con determinados tipos de archivos y bajo ciertas demandas. Un ejemplo de este caso se da, justamente, con los archivos .vol y .azi del software Rainbow. En este caso, Py-ART requiere el uso de la librería Wradlib, la cual se encarga de leer los datos de este tipo de archivos, entre otras características que ofrece.

Dado a que la utilización que hace Py-ART sobre Wradlib sólo cubre la lectura de ciertos datos de los archivos Rainbow, se ha hecho uso de la segunda librería de forma independiente, permitiendo, de esta manera tener el conjunto de datos completos disponibles para el usuario de la aplicación.

Continuando con la lectura de datos, de forma más genérica, esta tarea puede dividirse en los siguientes pasos:

### 1 Validación de dependencias

Relacionado a lo explicado en los párrafos anteriores, acerca de la dependencia que tiene Py-ART sobre Wradlib, la primera valida que la segunda librería se encuentre importada, para poder realizar correctamente la lectura completa de los datos contenidos en los archivos de volumen.

### 2 Validación de entrada

No se permitirá el ingreso de archivos que no correspondan a los establecidos por el software Rainbow. Esta validación se realiza teniendo en cuenta la extensión del archivo de entrada. En nuestro caso los dos tipos de archivos son, como ya se ha mencionado previamente, aquellos con extensión .vol y .azi, aunque Py-ART y en definitiva Wradlib permiten además la lectura de archivos .ele.

### 3 Chequeo de corte horizontales en el volumen

Se comprueba el número de cortes horizontales presentes en el escaneo, a través de la cantidad de elevaciones encontradas. Este número servirá para posteriores cálculos y procesamientos sobre el archivo. Se obtiene del valor de la etiqueta `<numele />`, accedido mediante el árbol de etiquetas del archivo XML:

```
<volume>
  <scan>
    <pargroup>
      <numele />
    </pargroup>
  </scan>
</volume>
```

Sección de código 15: Cortes horizontales

#### 4 Determinación del tipo de variable

En este paso, se extrae el tipo de variable analizada en el escaneo (Ej.: dBZ).

#### 5 Obtención de metadatos

Se leen los datos básicos del radar y del escaneo. Muchos de ellos requieren de cálculos adicionales, conversiones de medidas y/o relaciones con otros datos contenidos en el archivo. Alguno de los datos que podemos nombrar son:

- Ubicación (latitud, longitud) del radar.
- Altitud del radar.
- Fecha de escaneo y de escritura de resultados.
- Velocidad de la antena.

#### 6 Obtención de datos específicos de cada corte horizontal

Se itera sobre cada corte horizontal encontrado en el volumen para extraer información relevante:

- Cantidad de rayos por cada corte horizontal encontrado.
- Unidades de muestreo por cada corte horizontal y por cada rayo
- Ángulo de cada unidad de muestreo por rayo y por corte horizontal.

#### 7 Creación del objeto *Radar*

Con la recolección de todos los datos involucrados en las etapas anteriores, se construye un objeto de tipo Radar (del módulo `pyart.core`) con el cual se podrá interactuar para obtener los datos necesarios para la visualización.



## 5. Visualización: aplicación Meteorax

Esta etapa, la *visualización* (según el esquema presentado en la Figura 10), como su nombre lo indica, comprende las actividades que tienen como objetivo presentar los datos relevados en la etapa de **obtención**, y posteriormente **procesados** y volcados en archivos `.vol` y `.azi`, al usuario de la aplicación.

En este capítulo, se presentará lo que corresponde al *producto* de esta tesina, como resultado de un proceso de investigación en la obtención, el procesamiento y la visualización de datos de radar: **una aplicación que facilita la exportación de los datos en formatos comprendidos por un usuario final**. Al utilizar la expresión *facilita*, estamos incluyendo el trabajo realizado para lograr transformar las salidas de productos cerrados y por ende, limitados a la funcionalidad provista por los desarrolladores (como lo es justamente el software Rainbow®) a formatos comprensibles, adaptables y reutilizables.

En primer lugar se abordarán las herramientas utilizadas para la construcción de **Meteorax**. En este caso librerías de software capaces de procesar datos, modificarlos y exportarlos en formatos amigables y de uso común en el ambiente de la presentación de los datos de radar.

Estas librerías, ya fueron introducidas y mencionadas con anterioridad en la subsección 4.4.1, sin embargo, en este capítulo se las analizará con más detalle, ya que además de leer los datos que provienen de los archivos generados por el software de los radares, brindan facilidades para la obtención y visualización de los productos finales.

En segundo lugar, se mencionarán las variables dentro de los archivos que se utilizan como fuente de los datos, generados por el software Rainbow®, como por ejemplo la Reflectividad, una de las más utilizadas en meteorología.

Posteriormente se presentan los detalles y especificaciones de los formatos que la aplicación **Meteorax** puede exportar: imágenes, grillas, archivos de texto ASCII, etc. Se incluyen en esta sección, además, la definición de los llamados **productos de radar** que pueden generarse desde la aplicación, teniendo en cuenta ciertas limitaciones en la obtención de los datos.

Por último se brindará un recorrido por la aplicación, mostrando su utilización, sus características y facilidades. Se incluyen imágenes ilustrativas que ayudan a comprender el flujo de la aplicación desde la lectura de los archivos `.vol` y `.azi` hasta la obtención de lo que el usuario busca: datos en formatos conocidos.

### 5.1. Herramientas utilizadas

#### 5.1.1. Lenguaje: Python

Para la realización de la aplicación se ha elegido **Python**[23], un lenguaje de programación interpretado, orientado a objetos y de alto nivel. Además, se

pueden destacar otras características importantes, que facilitan el desarrollo de software como lo son:

- Variedad de estructuras de datos
- Tipado dinámico
- Sintaxis simple
- Multi-plataforma
- Facilidad de aprendizaje
- Performance
- Documentación amplia y completa
- Licencia de código abierto

Otra realidad que acompaña a las características del lenguaje anteriormente mencionadas es la compatibilidad con las librerías existentes para el desarrollo de herramientas de procesamiento y visualización de datos de radar, como las que se analizan a continuación.

### 5.1.2. Py-ART

Es una librería para el lenguaje Python, construida con el fin de analizar, procesar, corregir y visualizar datos de radar[24]. Es de código abierto y posee una licencia BSD. Es multiplataforma y tiene porciones de código escritas en Cython (un superconjunto de Python). Además provee una suite de tests de unidad, con el fin de asegurar la calidad del desarrollo.

Esta librería está basada en otras de uso científico:

- **SciPy**[25]

Suite de herramientas científicas para python, de código abierto. En sí SciPy además representa un conjunto de servicios disponibles a los interesados:

- **Stack de librerías**<sup>7</sup> de uso científico.
- **Comunidad de desarrolladores** que trabajan en estas librerías.
- **Conferencias** de computación científica. Ej.: SciPy, EuroSciPy y SciPy.in.
- **Librería SciPy**, que forma parte del stack de SciPy y que provee la funcionalidad para el desarrollo de aplicaciones con utilización de recursos científicos.

---

<sup>7</sup>Para el contexto informático, simboliza un conjunto de utilidades agrupadas por ciertas características comunes.

- **Numpy**[26]

Provee abstracción de datos de múltiples dimensiones. Permite contener y manejar volúmenes de datos de una forma sencilla, y en diversos formatos. Dado a su versatilidad, es posible integrarla con diversas bases de datos. Pertenece a la suite de librerías de SciPy y posee licencia BSD.

- **Matplotlib**[27]

Librería de ploteo de gráficos en dos dimensiones. También de código abierto, es una de las herramientas más utilizadas en materia de representación de datos en forma gráfica. La motivación que derivó en el desarrollo de esta librería, fue la de emular los comandos incluidos en MATLAB, una herramienta de software de uso matemático, pero desde Python.

Si bien en su mayoría está desarrollada sobre Python puro, su desarrollo en gran medida hace uso de la anterior librería mencionada: **Numpy**.

A continuación se listan algunos de los principios en los cuales se basa Matplotlib:

- Facilidad de comprensión y extensión de su código fuente.
- La forma de plotear datos debe ser sencilla y accesible.
- Múltiples formatos de exportación de los resultados obtenidos.
- Calidad de los gráficos resultantes.

- **NetCDF**[28] (Del inglés: **N**etwork **C**ommon **D**ata **F**orm) Comprende un conjunto de librerías y formatos de datos que permiten la manipulación de información científica. Su desarrollo es libre de plataforma y su implementación incluye varios lenguajes como **R**, **IDL**, **MATLAB**, **C++**, **Ruby**, **Perl**, **Java** y **Python**, entre otros. Para esta tesina, es de principal importancia su implementación en el último lenguaje mencionado: **netcdf4-python**.

Características principales del formato de dato NetCDF:

- **Auto-descriptivo**: Un archivo NetCDF provee información acerca del contenido que almacena.
- **Portable**: Un archivo NetCDF puede ser accedido por sistemas con diferente forma de almacenamiento de valores como enteros, caracteres y números flotantes.
- **Escalable**: NetCDF provee acceso eficiente a los datos, incluso a subconjuntos de datos de gran volumen.
- **Apto**: Es posible agregar información a un archivo NetCDF sin alterar su estructura y/o sin duplicar información.
- **Históricamente compatible**: Versiones antiguas y actuales de un archivo NetCDF son y serán compatibles con versiones futuras.
- **Compatible**: Es posible encontrar un escenario en donde un escritor y múltiples lectores estén accediendo a un archivo NetCDF sin problemas.

NetCDF es, además, una comunidad para el intercambio de información científica. Su formato es utilizado en diferentes entidades y organizaciones que hacen uso de su estándar, como **NASA** (del inglés: National Aeronautics and Space Administration), **NCAR** (Del inglés: National Center for Atmospheric Research) y el **NOAA** (del inglés National Oceanic and Atmospheric Administration).

Su desarrollo lo lleva a cabo la organización Unidata, parte de **UCAR** (del inglés: University Corporation for Atmospheric Research).

La organización de Py-ART, para una mejor utilización ha sido dividida en los siguientes submódulos:

- **Core** (pyart.core)
- **Entrada y salida** (pyart.io)
- **Entrada y salida auxiliar** (pyart.aux\_io)
- **Correcciones de radar** (pyart.correct)
- **Retribuciones de información de radar** (pyart.retrieve)
- **Gráficos** (pyart.graph)
- **Mapas** (pyart.map)
- **Filtros** (pyart.filters)
- **Utilidades** (pyart.util)
- **Conexión con otras herramientas** (pyart.bridge)
- **Testing** (pyart.testing)

## Core (pyart.core)

### Radar

El submódulo Core de Py-ART se compone de las clases: **Radar**, **Grid** y **HorizontalWindProfile**, más una serie de métodos para realizar transformaciones entre sistemas de coordenadas. Las dos primeras son las que se utilizarán como parte de esta investigación.

La clase **Radar**, que representa la información de un volumen de radar en memoria, almacena los datos propios del escaneo (variables, tiempos, etc.) como así también los datos del instrumento (altura, localización geográfica, etc.). En su mayoría, Py-ART utiliza la estructura de datos `dict` [29] para representar a estos datos.

Es importante mencionar que, además, utiliza un formato basado en estructuras para almacenar información de radar en el disco, siguiendo la convención CF/Radial[30]. En este formato, los valores del escaneo son almacenados **por**

**rayos** y por **volúmenes de muestreo**, que fueron detallados en la subsección 3.1.2 del capítulo 3. Estos bins, son indexados utilizando dos dimensiones: **tiempo** y **distancia al radar**, en ese orden. Es decir, un valor en concreto para una variable estará determinado por un tiempo  $t$ , que indica un rayo en particular y una distancia  $d$  al radar. Incluso, este formato, provee una manera de dividir un escaneo en volumen en **cortes horizontales**, almacenando los índices de los rayos de comienzo y fin para cada corte horizontal.

Podemos mencionar alguno de los campos más relevantes que conforman la clase Radar:

- **time:** ( dict ) Tiempo de cada rayo al centro del escaneo.
- **range:** ( dict ) Distancia al centro del escaneo de cada volumen de muestreo.
- **fields:** ( dict ) Variables analizadas.
- **metadata:** ( dict ) Información del instrumento.
- **scan\_type:** ( str ) Tipo de escaneo.
- **latitude:** ( dict ) Coordenada de latitud del instrumento.
- **longitude:** ( dict ) Coordenada de longitud del instrumento.
- **altitude:** ( dict ) Altura del radar sobre el nivel del mar.
- **sweep\_number:** ( dict ) Número de slices en el volumen, indexado desde 0.
- **fixed\_angle:** ( dict ) Ángulos de elevación para cada slice en un escaneo en volumen. Para un escaneo en azimut representa el grado (0-360) de escaneo.
- **sweep\_start\_ray\_index:** ( dict ) índice del primer rayo en cada slice, relativo al comienzo del volumen. Indexado desde 0.
- **sweep\_end\_ray\_index:** ( dict ) índice del último rayo en cada slice, relativo al comienzo del volumen. Indexado desde 0.
- **rays\_per\_sweep:** ( dict ) Número de rayos en cada slice.
- **ngates:** ( int ) Número de bins en un rayo.
- **nrays:** ( int ) Número de rayos en el volumen.
- **nsweeps:** ( int ) Número de slices en el volumen.

## Grid

La clase **Grid**, la cual almacena los datos de radar en el formato de grilla rectilínea, basándose en coordenadas cartesianas, es utilizada también para representar los datos de los escaneos de radar y para obtener resultados en otros formatos como en una grilla ASCII y GeoTIFF, que se analizarán con más detalle en la sección 5.3. Además provee algunas funciones relacionadas a conversiones de estructuras en coordenadas cartesianas, vectores y distancias, entre otras.

Como campos importantes de esta clase se encuentran, entre otros:

- **time:** ( `dict` ) Marca de tiempo de la grilla.
- **fields:** ( `dict` of `dict` ) Variables de la grilla u otras variables.
- **metadata:** ( `dict` ) Descripción de la grilla
- **origin\_longitude, origin\_latitude, origin\_altitude:** ( `dict` ) Coordenadas geográficas del punto de origen de la grilla.
- **x, y, z:** ( `dict` ) Distancia desde el origen para cada eje de coordenada cartesiano. Define el espacio en los tres ejes, que se repite por toda la grilla, convirtiéndola en una grilla rectilínea.
- **nx, ny, nz:** ( `int` ) Número de puntos de la grilla para una dimensión cartesiana dada.
- **projection:** ( `dict` o `str` ) Parámetros de proyección utilizados para transformar coordenadas cartesianas en geográficas.
- **point\_x, point\_y, point\_z:** ( `dict` ) Ubicaciones cartesianas de todos los puntos de la grilla desde el origen y para los 3 ejes.

## Entrada y salida auxiliar (`pyart.aux_io`)

En este módulo se encuentran las funciones para realizar la lectura de archivos de radar pero que provienen de formatos no estándares, como el utilizado por los archivos `.vol` y `.azi` generados por el software Rainbow, analizados en esta tesina.

Estas funciones de lectura especiales fueron agrupadas en un módulo diferente, ya que alguna de ellas requieren dependencias adicionales a diferencia de las que se encuentran bajo el módulo **Entrada y salida (`pyart.io`)**. En especial la función detallada a continuación, que provee la interfaz de lectura de archivos Rainbow, hace uso de la librería **Wradlib**, detallada más adelante en la subsubsección: `wradlib`.

La función `pyart.aux_io.read_rainbow_wrl` es la que permite obtener los datos de archivos Rainbow y acepta los siguientes parámetros:

- **filename:** ( `str` ) Nombre del archivo a leer.

- **field\_names:** (`dict`, opcional) Campos del archivo a leer (Ej.: *reflectivity*).
- **additional\_metadata:** (`dict` de `dict`, opcional) Información adicional a incluir en la lectura del archivo.
- **file\_field\_names:** (`bool`, opcional) Indica si se debe usar el formato MDV de nombres para los campos del archivo.
- **exclude\_fields:** (`list`, opcional): Lista de campos a ignorar en la lectura del archivo.

Esta función retornará un objeto de tipo Radar, previamente visto, con la información del archivo que actualmente está siendo procesado.

### Mapas (`pyart.map`)

### Gráficos (`pyart.graph`)

#### 5.1.3. Wradlib

La librería Wradlib[31] es otra herramienta desarrollada sobre el lenguaje Python con el fin de cubrir ciertas demandas (técnicas y científicas) relacionadas a la digitalización, procesamiento y visualización de los datos de radar provenientes de diversas clases de archivos. Como principales características se pueden destacar:

- Desarrollo de código abierto.
- Diseño modular.
- Construida sobre Python.
- Su API <sup>8</sup> es muy sencilla y la utilización de la misma no presenta ningún tipo de dificultad al usuario.
- Multi-plataforma.
- Proyecto construido en comunidad.
- Desarrollado bajo la licencia MIT, la cual permite que los mismos colaboradores de la librería puedan utilizar su trabajo para otros fines, incluso comerciales.
- Amplia variedad de formatos de entrada, incluyendo el formato Rainbow de Gematronik.

---

<sup>8</sup>Del inglés: **A**pplication **P**rogramming **I**nterface. Conjunto de funciones, clases, atributos, etc. que provee una librería

Como se mencionó en las anteriores características, Wradlib, fue diseñada para una utilización modular, permitiendo a los usuarios de la misma, combinar funcionalidades de la manera en que les resulte más conveniente, brindando la posibilidad de crear cadenas de procesamiento de datos de radar con los módulos que forman parte de la misma. Esta es una característica importante de Wradlib, que sus desarrolladores han remarcado: Wradlib no es una aplicación, es una librería.

Una sección de mucha utilidad presente en Wradlib, es la funcionalidad de alterar los datos, como por ejemplo transformar desde y hacia decibeles u obtener la profundidad de lluvia desde la intensidad de la misma en un período en particular. También incluye varias funciones dedicadas a la georreferenciación, como la conversión de datos en coordenadas polares a geográficas y viceversa o la proyección de coordenadas geográficas a sistemas de coordenadas cartesianas.

Finalmente, la corrección de errores que provienen de las aplicaciones que generan los archivos que la librería toma como entrada, es una característica valiosa a la hora de seleccionar el software encargado de procesar los datos de radar.

## Módulos

Wradlib se divide en los siguientes módulos:

- **Importación y exportación de datos** (wradlib.io).
- **Atenuación de ruido** (wradlib.clutter).
- **Corrección de atenuación** (wradlib.atten).
- **Análisis de datos verticales** (wradlib.vpr).
- **Transformación de unidades** (wradlib.trafo).
- **Conversiones** (wradlib.zr).
- **Transformación espacial y geo-referenciación** (wradlib.georef).
- **Interpolación espacial** (wradlib.ipol).
- **Indicadores de calidad de medición** (wradlib.qual).
- **Composiciones desde múltiples radares** (wradlib.comp).
- **Ajustes de medidor de lluvia** (wradlib.adjust).
- **Verificación** (wradlib.verify).
- **Mapeo y ploteo de gráficos** (wradlib.vis).
- **Utilidades** (wradlib.util).



Como puede verse, la librería **Wradlib** cubre una gran variedad de funcionalidades que completan, extienden y facilitan la extracción y el procesamiento de los datos de radar. Para el trabajo realizado en esta tesina sólo tomaremos lo brindado por el módulo **wradlib.io** el cual es el encargado de contener los métodos necesarios para leer los datos de los diferentes formatos de archivos que contienen datos de radar y que será brevemente detallado a continuación.

### Importación y exportación de datos (wradlib.io)

Se mencionó con anterioridad que la librería Pyart hace uso de Wradlib, en especial para la lectura de archivos. Es el módulo **wradlib.io** quien provee dicha funcionalidad. Mediante la utilización del mismo se leen los datos de los archivos de entrada y se crean objetos en memoria que utilizan la estructura de datos arreglo (*ndarray*) de Numpy, previamente mencionada. A su vez, existen metadatos que también son cargados en una estructura de datos `dict`, la cual permite que los datos puedan ser accedidos mediante claves.

El método que permite realizar esta tarea primordial es: **wradlib.io.read\_\_Rainbow**. El mismo toma como parámetros:

- **f:** (`str`) Nombre del archivo a leer.
- **loaddata:** (`bool`) Si su valor es `False` los datos en binario no serán leídos por el método.

Este método retornará una estructura de tipo `dict` conteniendo los datos leídos (según los parámetros indicados) y su estructura será un espejo de la organización de etiquetas del archivo de entrada. Se debe tener en cuenta que la librería anteriormente mencionada **PyArt** utiliza este método para realizar una representación propia de los datos y para posteriores utilidades, por ejemplo en la parte de la visualización de los datos. Es por esto que **Wradlib** también toma un papel importante en el desarrollo de esta tesina.

#### 5.1.4. GTK+

GTK+ es un conjunto de herramientas para el desarrollo de interfaces gráficas. Su nombre deriva de **The GIMP Toolkit** ya que, inicialmente, estas librerías que componen a GTK+ fueron pensadas para desarrollar la interfaz de la aplicación GIMP (Sigla que corresponde a **GNU Image Manipulation Program**), utilizada para la creación y edición de imágenes, aunque hoy día su utilización se expande a varias aplicaciones de uso cotidiano.

Su autor inicial fue Peter Mattis con la ayuda de Spencer Rimball y Josh Macdonald. Para el resto de sus funcionalidades como el *binding* con lenguajes de desarrollo, mejoras en los widgets principales, soporte para temas y la configuración inicial para los lanzamientos y publicaciones se sumaron: Marius Vollmer, Lars Hamann, Stefan Jeske, Carsten Haitzler y Shawn Amundson.

## Plataformas

Una ventaja principal es su característica de **multiplataforma**, tanto para su uso como herramienta de desarrollo e incluso para las interfaces que se generan. Es decir, se puede utilizar para construir, desde Linux, Windows o Mac OS (entre otras plataformas) y asimismo la interfaz desarrollada podrá utilizarse en las mencionadas plataformas. Ésta es a su vez una de las razones por las cuales se eligió para el desarrollo de la interfaz de Meteorax.

## Licencia

GTK+ es un software libre y pertenece al **Proyecto GNU**. Más allá de esto, las condiciones del licenciamiento para GTK+ (GNU LGPL 2.1) permiten que sea utilizada por cualquier desarrollador, incluso aquellos que construyan software privativo, sin ningún derecho de licencia o de autor.

## Soporte de lenguajes

GTK+ está desarrollado en C aunque, además, puede ser utilizado en una amplia variedad de lenguajes:

- Ada
- C#
- C++
- D
- Fortran
- FreeBASIC
- Go (GTK+ 2)
- Go (GTK+ 3)
- Guile
- Haskell
- Java
- Javascript
- Lua
- OCaml
- Pascal
- Perl

- Python
- PHP
- R
- Ruby
- Rust
- Vala

Para el desarrollo de Meteorax, se ha utilizado **PyGTK**[32] que es justamente un binding de GTK+ para el lenguaje Python, pero que comparte la misma API que la librería base. En conjunción se utilizó **Glade**, una herramienta visual para el diseño y la construcción de interfaces de usuario para GTK+ [33].

### Arquitectura

GTK+ fue construida basándose en cuatro librerías, también desarrolladas por el equipo de desarrollo de GTK+:

- **GLib**: Librería principal de GTK+. Manejo de estructuras en C, sistema de objetos, main loop, etc.
- **Pango**: Diseño y representación de texto con soporte para la internacionalización.
- **Cairo**: Gráficos en 2D con soporte para varios dispositivos.
- **ATK**: Interfaces que proveen accesibilidad.

## 5.2. Variables procesadas

Como resultado del proceso de obtención de los datos (visto en la capítulo 3), el software controlador de los radares de INTA generará 2 tipos de archivos: **de producto** y **de datos en bruto**. De estos dos tipos de archivo nos quedaremos con el primero como entrada para **Meteorax** (ver subsección 4.2.3).

En este tipo de archivo encontraremos diferentes tipos de variables que describen de qué se trata el resultado del escaneo, es decir, cómo interpretar los datos codificados para ser presentados. A continuación se describen aquellas variables que serán utilizadas en la aplicación:

- **Factor de reflectividad ( $dBZ$ ,  $[dB]$ )**: Como se ha mencionado con anterioridad, la reflectividad puede resumirse como la intensidad con la que la energía impactada en un objetivo es reflejada. Parte de esta energía es la que retorna al radar y con la cual es posible calcular el **factor de reflectividad  $Z$** .

El factor de reflectividad es proporcional a la suma de la sexta potencia del diámetro ( $D$ ) de todos los objetivos en el volumen muestreado. Dado a que el tamaño de las gotas de lluvia es medido generalmente en unidades de milímetros, y el volumen es expresado en unidades de metros cúbicos, el factor de reflectividad se compone de unidades de  $mm^6/m^3$  (Ver Ecuación 1).

$$Z = \sum_{i=1}^n D_i^6 \quad (1)$$

Los valores típicos de factores de reflectividad para nubes sin precipitación o de llovizna ligera irán desde  $10^{-5}$  a 10. Para lluvias muy intensas y granizo, el factor debería ser como máximo 107. Dado a que estos valores oscilan entre varios órdenes de magnitud y son difíciles de representar gráficamente con mucho detalle, los productos de radar usan, generalmente, una escala logarítmica en decibeles  $dBZ$  (decibeles de  $Z$ ) más fácil de entender (Ver Ecuación 2).

$$dBZ = 10 \log_{10} \frac{Z}{1 \text{ mm}^6 / 1 \text{ mm}^3} \quad (2)$$

Esta escala se comprende en un rango que va:

- de **-28** a **28** para el modo de aire claro y
- de **0** a **75** para el modo de precipitación.

En el modo de precipitación, valores bajos de  $dBZ$  (colores azul y verde) indican una precipitación leve, mientras que valores altos en el rango de los colores amarillo, naranja y azul, significan una precipitación más pesada. Valores debajo de los  $45 \text{ dBZ}$ , aproximadamente, significan una precipitación intensa y casi siempre se deben a tormentas eléctricas. Cualquier cosa por encima de los  $60 \text{ dBZ}$  generalmente significa que el volumen escaneado contiene algo de granizo.

Generalmente es la variable que primero se presenta en las visualizaciones de los escaneos de radares meteorológicos. Esto se debe a que por su naturaleza permite comprender un panorama general de forma sencilla y rápida a simple vista [12].

- **Factor de reflectividad (sin corrección) ( $dBuZ$ ,  $[dB]$ ):** Al hablar del anteriormente mencionado «factor de reflectividad ( $dBZ$ )», estamos haciendo referencia a valores que han sufrido una **corrección** en el momento en que se han procesado por los sistemas de radar.

Esta corrección se debe a la naturaleza de las condiciones dadas generalmente en el momento del escaneo, como por ejemplo obstáculos preestablecidos en el radio del radar que producirán falsos valores en los resultados.

En los sistemas Rainbow® se utilizan algoritmos específicos que atacan situaciones como:

- Corrección de atenuación.

- Corrección de perfil vertical.
- Corrección de banda de brillante.
- Bloqueo parcial de haz.
- Corrección por ocultamiento.

Entonces cuando se habla de **factor de reflectividad (dBZ)**, sabemos que los valores son valores corregidos, mientras que para el **factor de reflectividad sin corrección (dBuZ)** estamos hablando de los datos resultantes del escaneo que no han sufrido corrección en la reflectividad.

- **Velocidad:** ( $V$ ,  $[m/s]$ ): También denominada **Velocidad Radial**, en la subsección 2.2.4 se ha hablado de los principios del **efecto doppler** y que es de suma importancia para esta sección, ya que es gracias a este principio que se puede obtener la información brindada por esta variable. Es una representación de la información que se obtiene de la velocidad del viento junto con su dirección. En general, colores más cálidos (como el rojo) en los gráficos que muestran la velocidad, muestran movimientos que se *alejan* del radar, mientras que los colores más fríos, como el verde, indican movimientos que se *acercan* al radar. Los objetivos que estén estacionados (sin movimiento) o con movimiento perpendicular al haz del radar, no tienen velocidad radial y son coloreados en gris [12].

Como ejemplo, se puede ver en la Figura 18 [12], en donde se aprecia que el viento proviene del suroeste con dirección al noreste.

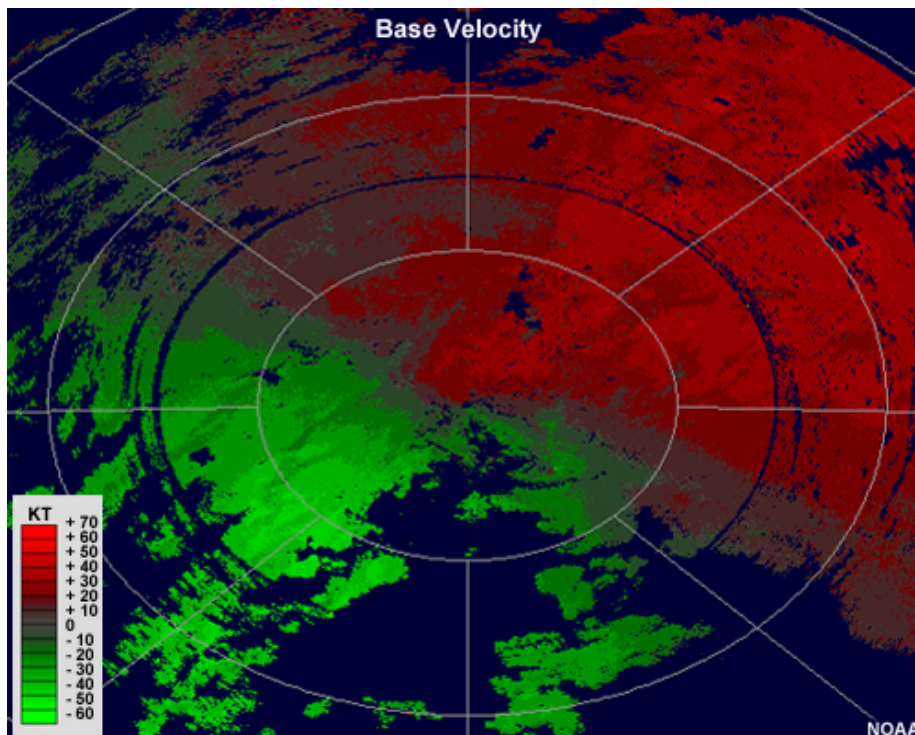


Figura 18: Ejemplo de visualización de la velocidad

- **Ancho espectral** ( $W$ ,  $[m/s]$ ): El ancho espectral es una medida de la dispersión de las velocidades dentro de un volumen de muestra de radar. Se puede definir como la desviación estándar de las medidas de la velocidad. Valores grandes de ancho espectral indican la presencia de fuertes turbulencias o cortes de viento potentes, los cuales producen un decremento en la capacidad del radar para definir de forma certera la velocidad radial ( $V$ ).
- **Reflectividad diferencial** ( $ZDR$ ,  $[dB]$ ): La reflectividad diferencial es la relación que existe entre la reflectividad observada al procesar las señales transmitidas y recibidas mediante la utilización de la polarización vertical y las señales procesadas utilizando polarización horizontal [34].

Como ejemplo de utilidad podemos describir la siguiente situación: las gotas de lluvia, por ejemplo, no son siempre esféricas cuando caen, en especial las gotas grandes. Éstas tienden a tomar una forma oblata (achatada en los polos), por lo tanto la reflectividad percibida será mayor si la onda utilizada estaba polarizada horizontalmente o si la reflectividad observada con polarización horizontal ( $Z_h$ ) es mayor a la observada en polarización vertical ( $Z_v$ ). Entonces, para discriminar esta situación y poder obtener un resultado más aproximado de la situación se puede utilizar la reflectividad diferencial.

Una fórmula comúnmente utilizada para la reflectividad diferencial es:

$$ZDR \cong 10 \log(P_h / P_v) [dB] \quad (3)$$

Donde:

- $P_h$  es la intensidad de la señal horizontalmente polarizada recibida de un pulso horizontalmente polarizado transmitido.
- $P_v$  es la intensidad de la señal verticalmente polarizada recibida de un pulso verticalmente polarizado transmitido.

Los valores de  $ZDR$  para observaciones de respuestas meteorológicas oscilan entre los  $-2 dB$  y  $6 dB$  y se tiene que:

- Valores de  $ZDR$  **por encima de cero** indican que los hidrometeoros en el volumen muestreado están orientados horizontalmente, lo que significa que su eje horizontal es más largo que su eje vertical ( $P_h > P_v$ ).
- Valores de  $ZDR$  **por debajo de cero** indican que los hidrometeoros en el volumen muestreado están orientados verticalmente, lo que significa que su eje vertical es más largo que su eje horizontal ( $P_h < P_v$ ).
- Valores de  $ZDR$  **cercanos a cero** indican que los hidrometeoros en el volumen tienen en promedio, una forma aproximada a la de una esfera ( $P_h \approx P_v$ ).

La reflectividad diferencial ( $ZDR$ ) es una buena herramienta para discriminar gotas de lluvia grandes de partículas de granizo, que caen de forma aleatoria y tienden a sostener una forma esférica.

- **Coefficiente de correlación polar ( $Rho_{HV}$ , [-]):** Es una medida de la calidad de un radar con polarización doble. El coeficiente de correlación polar entre las señales polarizadas de radar, horizontales ( $H$ ) y verticales ( $V$ ), es una medida de la variabilidad de las formas de los hidrometeoros en un volumen muestreado.

Resulta útil para diferentes aplicaciones, como la identificación de la capa de derretimiento, el ruido en las respuestas de la superficie, las mezclas indefinidas entre lluvia y granizo y la potencia obtención de la distribución de los tamaños de las gotas de lluvia, entre otras.

Una correlación perfecta entre los dos canales ( $H$  y  $V$ ) debería arrojar un valor de 1. En una lluvia uniforme y ligera, el  $Rho_{HV}$  se comporta aproximándose a 1 sin un límite, excepto por alguna medida intrínseca en las capacidades del radar.

- **Fase de propagación diferencial sin corrección ( $uPhi_{DP}$ , [grados]):** Una ventaja importante de los radares de polarización doble, es la medida adicional de la fase de propagación diferencial, la cual puede ser utilizada para *corregir* la atenuación acumulada debido a la interferencia producida por las gotas de lluvia a lo largo del camino de propagación. Al atravesar materiales más densos, las ondas sufren un cambio de fase entre las señales vertical y horizontal, y esto depende incluso de las características de los hidrometeoros.

La fase de propagación diferencial entre las polarizaciones vertical y horizontal, debido a las gotas de lluvia orientadas a lo largo de la ruta de propagación, resulta fundamental para la estimación de lluvias y en la corrección de los efectos de atenuación producidos.

En este caso se trata de la variable  $Phi_{DP}$  pero **sin corrección**, análogamente a lo descrito en la variable  $dBuZ$ .

### 5.3. Formatos de exportación

Toda la información resultante de procesar las anteriores imágenes deben ser *proyectadas* a un formato conocido y exportable. En el caso de **Meteorax**, la aplicación es capaz de generar tres tipos de formatos exportables, los cuales son analizados en las siguientes secciones:

- **GeoTIFF**
- **PNG**
- **ASCII**

#### 5.3.1. GeoTIFF

Antes de mencionar este formato es necesario comprender la base del mismo: el formato **TIFF**.

TIFF (del inglés **T**ag **I**mage **F**ile **F**ormat) es, como sus iniciales en inglés lo denotan, un *formato de archivo de imágenes con etiquetas* que almacena en su composición una imagen de mapa de bits y además un encabezado con **información descriptiva acerca de la imagen** en el mismo archivo y en forma de **etiquetas**.

Su primera versión fue desarrollada en el año 1987 por una empresa hoy perteneciente a **Adobe**, llamada **Aldus** y su última especificación se publicó en el año 1992, ésta fue la revisión 6.0.

Un archivo TIFF comienza con un encabezado de archivo de imagen de 8 bytes que apunta a un IFD (del inglés **I**mage **F**ile **D**irectory), que contiene información acerca de la imagen, así como los apuntadores a los datos reales de la imagen [35].

Algunas de sus ventajas son [35]:

- Es capaz de describir información de una imagen en varios espacios de colores, como bi-nivel, escala de grises, paleta de color y color full.
- Es portable y no está diseñado exclusivamente para un sistema operativo, sistema de archivos, compilador o procesador en particular.
- Está diseñado para ser extensible y evolucionar sencillamente a nuevas demandas.
- Permite la inclusión de una ilimitada cantidad de información privada o de propósito especial

El tamaño máximo de un archivo TIFF es de *4 GB* ya que se utilizan direccionamientos de *32 bits*.

El formato TIFF puede ser utilizado para almacenar y compartir imágenes satelitales, fotos aéreas, modelos de elevación, mapas o el resultado de varios tipos de análisis geográficos. Además es el único formato de dominio público capaz de soportar compresión, imágenes en mosaico y metadatos geográficos.

**GeoTIFF** implementa los metadatos geográficos formalmente, usando etiquetas y estructuras compatibles al formato TIFF.

Un archivo con formato GeoTIFF hace referencia a un archivo TIFF que contiene información geográfica o cartográfica embebida como etiquetas dentro del propio archivo. Ésta información puede ser utilizada para posicionar la imagen en la ubicación y geometría correcta en un display de información geográfica.

GeoTIFF es un formato de metadatos que provee información geográfica para asociar con la información de la imagen en sí.

Es un formato completamente abierto, de dominio público y no propietario. Fue producido por el Dr. Niles Ritter, en el NASA-JPL (Laboratorio de Propulsión a Chorro de la NASA, en español) y los cambios o adiciones al formato son propuestas a través de revisiones públicas utilizando email y recursos web.



No hay restricciones en la licencia, implementación, promulgación o cualquier uso del formato. El formato es totalmente abierto, y está disponible para todos [36].

### 5.3.2. PNG

La sigla **PNG** es el acrónimo de **P**ortable **N**etwork **G**raphics, en español: Gráficos de Red Portátiles. Es un formato de archivo para el almacenamiento, transmisión y visualización de imágenes, diseñado originalmente para reemplazar al formato GIF y TIFF.

Así como el formato GIF y TIFF, PNG es un formato *raster*, es decir, una representación de una imagen como un arreglo bidimensional de puntos (píxeles) coloreados. PNG no es explícitamente un formato vectorial, como por ejemplo uno que pueda almacenar formas (líneas, cuadrados, elipses, etc.) y que pueda ser escalado arbitrariamente sin pérdida de calidad [37].

PNG soporta compresión sin pérdidas, información de transparencia y una variedad de profundidades de color. Además soporta características como corrección gamma y un espacio de color estándar para la reproducción precisa de colores en diferentes sistemas. Como ventaja adicional y similar a lo que sucede con el formato TIFF, el formato PNG también soporta información textual embebida para el almacenamiento de información como título, nombre de autor y derechos de autor explícitos.

Existen campos fijos (bytes) en todo archivo PNG, denominados *bytes de firma* [37]:

Valor decimal	Significado ASCII
137	Un byte con su bit más significativo en 1
80	P
78	N
71	G
13	Carácter de retorno de carro (CR)
10	Carácter de nueva línea (LF)
26	CTRL-Z
10	Carácter de nueva línea (LF)

Cuadro 3: Bytes de firma del formato PNG

Luego de los 8 bytes de firma de PNG, viene la sección denominada IHDR (del inglés *Image Header*) que contiene todos los detalles acerca del tipo de la imagen:

- Alto y ancho
- Profundidad de los píxeles
- Métodos de compresión y filtrado
- Método de entrelazado
- Presencia de un canal alfa (transparencia)
- Indicador de imagen de *color verdadero*, en *escala de grises* o basada en una *paleta de colores*

En caso de tratarse de una imagen basada en paleta de colores, la sección a continuación será la denominada PLTE (del inglés *Palette*), es decir, la sección correspondiente a la información de la paleta. Contiene simplemente una secuencia de valores para los colores rojo, verde y azul, donde un valor de 0 es negro y 255 es la intensidad máxima.

A continuación sigue la sección IDAT (del inglés *Image Data*) que contiene toda la información comprimida de los píxeles que conforman la imagen. Ésta sección puede repetirse dependiendo de la imagen.

Y por último la sección IEND (del inglés *Image End*) que no contiene datos pero es utilizado para indicar que no hay más secciones en el archivo. En particular es de utilidad en las transmisiones de la imagen como un *stream*.

Los tipos de imagen PNG que podemos encontrar son:

- Basada en una paleta
- Basada en una paleta, con transparencia
- En escala de grises
- En escala de grises, con transparencia
- En escala de grises, con canal alfa
- RGB
- RGB con transparencia
- RGB con canal alfa

### 5.3.3. ASCII

Otra de los formatos de salida de la aplicación Meteorax es el formato **ASCII**. Existe una variante de formato binario privativa, **ARC/INFO GRID** y una no-propietaria, **ARC/INFO ASCII GRID**, que es la que se utiliza desde la aplicación Meteorax, para exportar la información en formato *grilla*. De ahora en adelante al hablar de *formato ASCII* nos referiremos a esta última variante.

También es un formato de imagen raster (matriz de puntos con información que conforman una imagen basada en la interpretación de esos puntos), aunque en este caso hablamos de una *grilla* conformada por *celdas* de valores numéricos [38].

Una grilla ASCII consiste de un encabezado que especifica el dominio geográfico y la resolución, seguido de los valores de cada celda de la grilla.

Para el encabezado se tienen los siguientes campos, cada uno en una línea de archivo:

- **ncols** (numérico): Número de columnas de la grilla.
- **nrows** (numérico): Número de filas de la grilla.
- **xllcorner** (numérico, decimal o entero): Coordenada X del origen, esquina inferior izquierda.
- **yllcorner** (numérico, decimal o entero): Coordenada Y del origen, esquina inferior izquierda.
- **cellsize** (numérico): Resolución de la grilla, utilizando la misma unidad que el sistema de coordenadas que indican la esquina inferior izquierda.
- **nodata\_value** (numérico, opcional): Representa el valor que aparecerá en una celda que no contenga información. ESRI asume un valor por defecto de -9999.

Notar que al permitir agregar información al archivo mediante campos (al igual que TIFF o PNG), en este caso los campos *xllcorner* y *yllcorner* es posible localizar los datos geográficamente para presentarlos sobre un mapa por ejemplo. Ésta es una de las características buscadas por las aplicaciones que presentan datos de radar, por su utilidad para georeferenciar imágenes.

A partir de la línea 7 del archivo comienzan las filas de la grilla, una por línea de archivo (esta cantidad debe corresponder al valor de *nrows*). Cada fila es una secuencia de valores separados por un espacio.

Generalmente la extensión para estos tipos de archivos es **.asc** pero pueden encontrarse otros tipos de extensiones como **.grd** en otras aplicaciones.

Un archivo ASCII es humanamente legible (Ver la Sección de código Sección de código 16) y puede abrirse con un editor de texto común, aunque su tamaño suele ser considerablemente grande, dependiendo de la cantidad de información que contenga o del área cubierta.

```

ncols 480
nrows 480
xllcorner -36.235589
yllcorner -60.001438
cellsize 100
nodata_value -9999
43 2 45 7 3 56 2 5 23 65 34 6 32 54 57 34 2 2 54 6 ...
35 45 65 34 2 6 78 4 2 6 89 3 2 7 45 23 5 8 4 1 62 ...

```

Sección de código 16: Ejemplo de grilla ASCII

#### 5.3.4. Productos de radar

Se puede definir a un “producto de radar” como un tipo de presentación de la información que se ha recolectado. Los mismos dependen de factores como el tipo de escaneo realizado, o la variable analizada.

En la aplicación Meteorax se exportan dos tipos de productos de radar, cuya definición se ha visto en la subsección 2.3.8:

- **PPI** (del inglés *Plan Position Indicator*). Utilizado en escaneo en azimut y en volumen.
- **RHI** (del inglés *Range Height Indicator*). Utilizado en escaneo en elevación.

Los dos productos anteriores, PPI y RHI, pertenecen a un conjunto de *productos estándares*, cuya lista puede ampliarse con los siguientes:

- **CAPPI** - *Constant Altitude PPI*. Soporta todos los tipos de datos escaneados.
- **MAX** - *Maximum Display*. Soporta todos los tipos de datos escaneados.
- **CMAX** - *Column Maximum*. Soporta todos los tipos de datos escaneados.
- **VCUT** - *Vertical Cut*. Soporta todos los tipos de datos escaneados.
- **MLVCUT** - *Multiple-Line Vertical*. Soporta todos los tipos de datos escaneados.
- **EHT** - *Echo Height*. Soporta datos de variables: *dBZ* y *dBuZ*.

Todos los anteriores productos se utilizan sólo en escaneos *en volumen*.

## 5.4. Aplicación

La aplicación desarrollada como resultado de la investigación realizada en esta tesina se denomina **Meteorax**<sup>9</sup>. Como se mencionó anteriormente, fue desarrollada en **Python** en su versión 3 en conjunción con las librerías de procesamiento de datos de radar: **PyArt** y **Wradlib** (entre otras) y se utilizó **GTK+** como herramienta de desarrollo de interfaz de usuario.

Meteorax fue pensada como una aplicación de escritorio para las plataformas Windows, Linux y OSX. El usuario deberá obtener la aplicación del repositorio, instalar las dependencias mencionadas y posteriormente la puede ejecutar por consola de comandos.

A continuación se presentan capturas de pantalla de diferentes instancias de la aplicación, junto con su descripción.

### 5.4.1. Ejecución

#### 5.4.2. Inicio: selección de archivo

En esta parte (Figura 19) se puede ver un widget de selección de archivo. En el mismo, como primer paso, se deberá clicar para que se abra un cuadro de diálogo (Figura 20) que permitirá seleccionar el archivo a procesar por Meteorax.

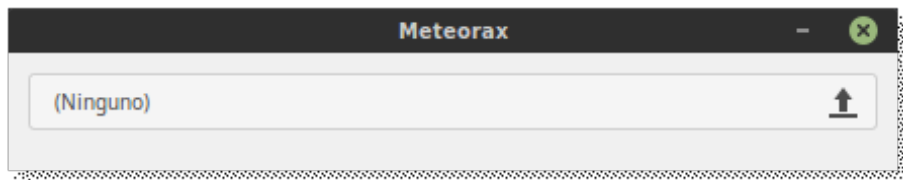


Figura 19: Meteorax - Inicio: selección de archivo

#### 5.4.3. Información de escaneo

Tanto en la Figura 21 como en la Figura 22 se aprecia la información básica del radar como del escaneo que se ha realizado, tal como:

- Radar
  - Radar que ha hecho el escaneo (Anguil, Paraná o Pergamino).
  - Longitud de onda.
  - Ancho del haz.
  - Ubicación: latitud y longitud.

---

<sup>9</sup>El nombre se deriva de una combinación entre *meteorología* y *rayos* (por las señales que envía el radar).

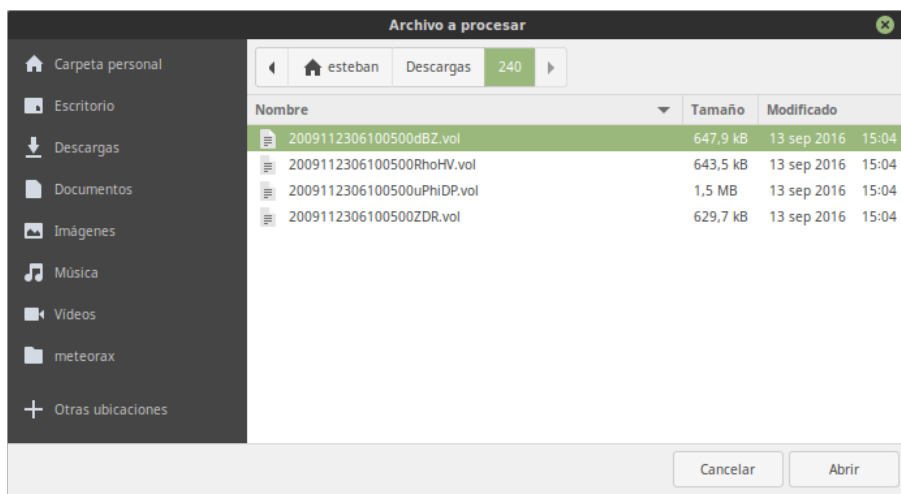


Figura 20: Meteorax - Cuadro de selección de archivo

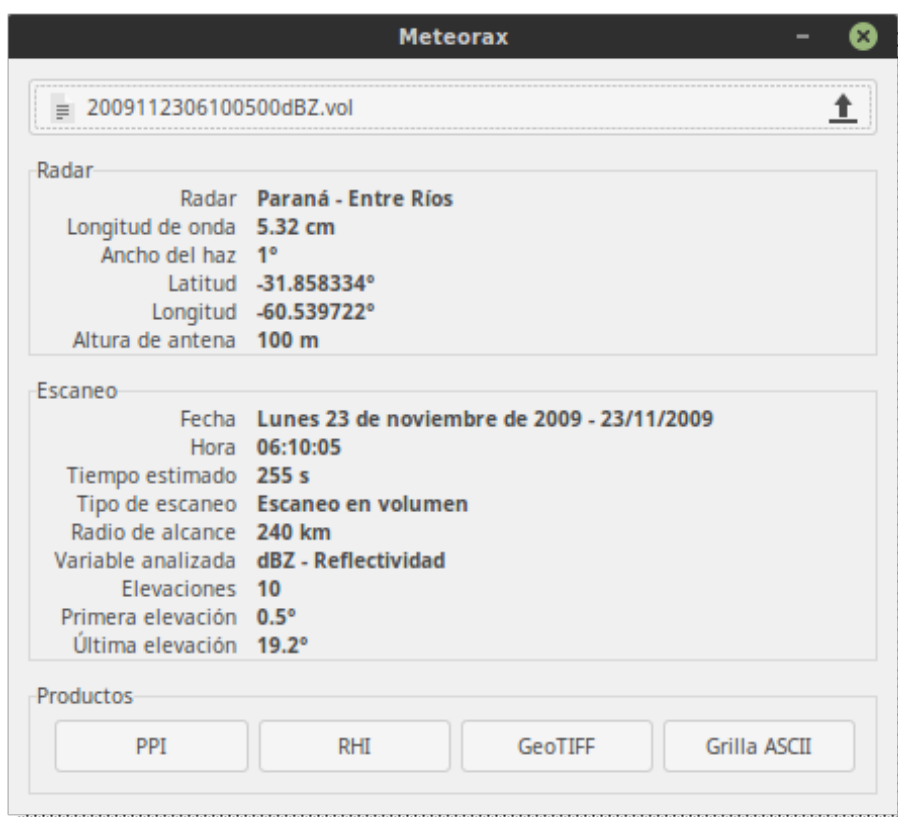


Figura 21: Meteorax - Información de escaneo, archivo .vol

- Altura de la antena.

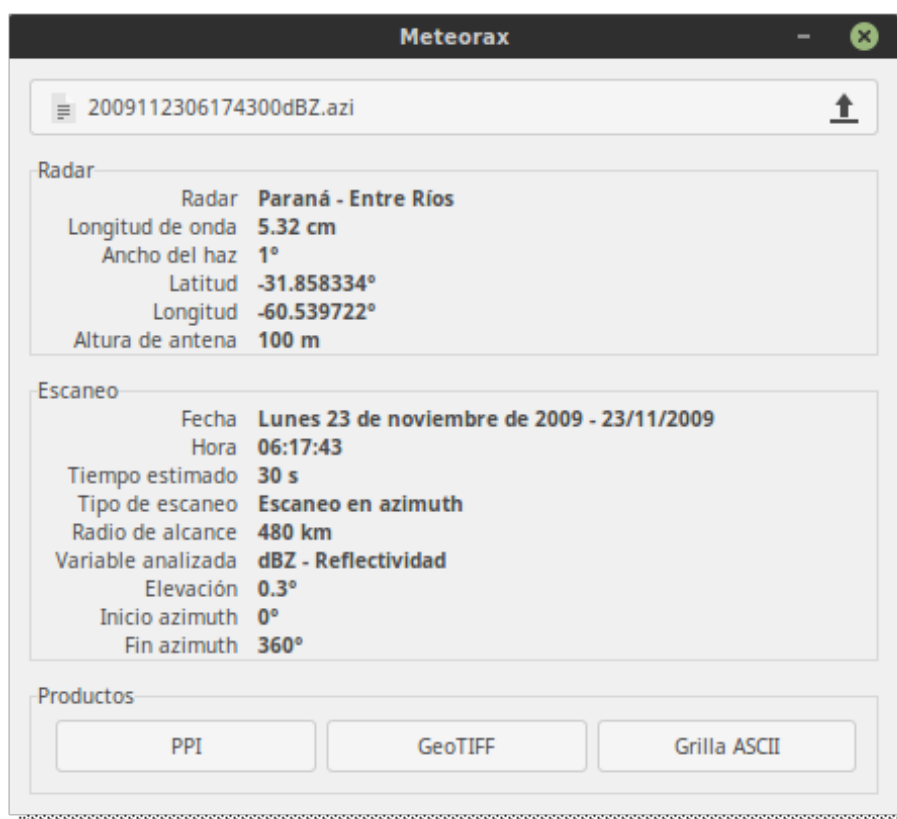


Figura 22: Meteorax - Información de escaneo, archivo **.azi**

#### ■ Escaneo

- Fecha y hora.
- Tiempo estimado.
- Tipo de escaneo (en volumen o en azimuth)
- Radio de alcance.
- Variable analizada.
- Para escaneo en volumen:
  - Cantidad de elevaciones.
  - Primera y última elevación.
- Para escaneo en azimuth:
  - Elevación analizada.
  - Primer y último azimuth.

Por último se puede ver una sección de **Productos** que corresponde a las acciones a llevar a cabo con el archivo cargado previamente. Esta sección contendrá diferentes acciones para los dos tipos de archivos permitidos (ver subsección 5.3.4):

Para ambos tipos de escaneos:

- **PPI**
- **GeoTIFF**
- **Grilla ASCII**

Para escaneo en volumen (archivos **.vol**), se agrega el tipo de producto **RHI**.

Lo anterior se debe a que el escaneo en azimut, captura información para una elevación única y fija, en los 360° a su alrededor, por ende, no contiene cortes verticales que permitan exportar datos para el producto RHI.

#### 5.4.4. Productos: PPI

Al presionar el botón **PPI** (ver Figura 21) se abrirá una nueva ventana con varios componentes para configurar la vista del producto PPI sobre el archivo de escaneo elegido.

Como se puede ver en la figura Figura 27, que corresponde a la misma ventana de *producto PPI*, se visualizan 4 elevaciones diferentes: primera, cuarta, séptima y última. Para poder cambiar la elevación actualmente mostrada, se utiliza el componente de la parte superior, un slider con las elevaciones y sus respectivos grados, definidos en el archivo cargado anteriormente.

En caso de tratarse de un archivo proveniente de un escaneo en azimut, sólo se tendrá una elevación, por lo que el slider de la parte superior desaparecerá (Figura 28).

En esta ventana es posible interactuar con el gráfico de variable presentado utilizando los botones de acción que se encuentran en la parte superior del mismo:

- Configurar los límites mostrados
- Exportar la imagen actual como PNG
- Hacer zoom (o alejar) en el área seleccionada
- Volver a la vista anterior o a la inicial

La barra de colores que se encuentra a la derecha de la ventana que muestra este producto, se denomina **mapa de color** la cual se utiliza como referencia para conocer los valores mapeados a colores en el gráfico. En este caso muestra en colores más cálidos los valores de reflectividad más elevados y en colores fríos los valores más bajos para esta variable.



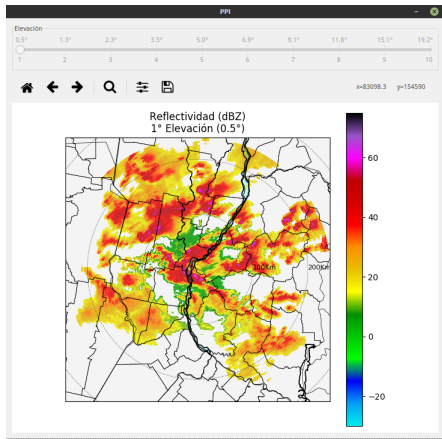


Figura 23: PPI, primer elevación

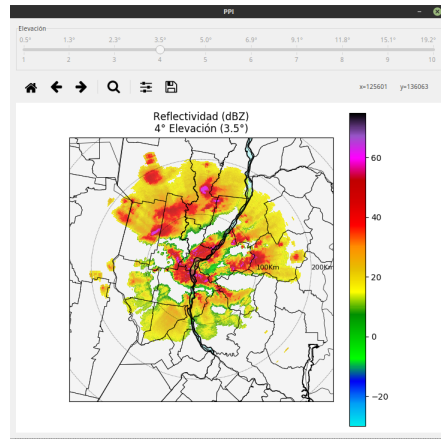


Figura 24: PPI, cuarta elevación

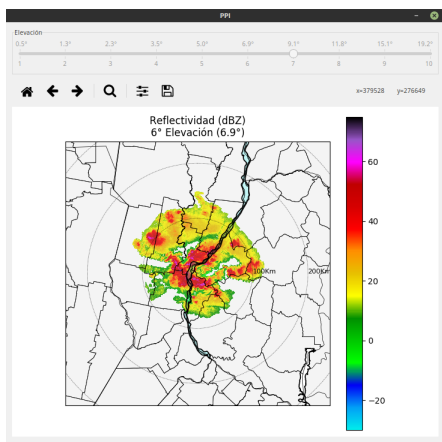


Figura 25: PPI, séptima elevación

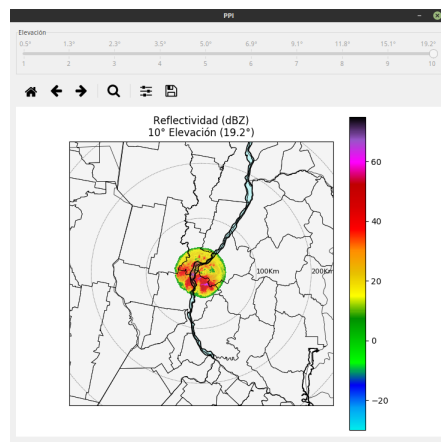


Figura 26: PPI, última elevación

Figura 27: Meteorax - Producto PPI

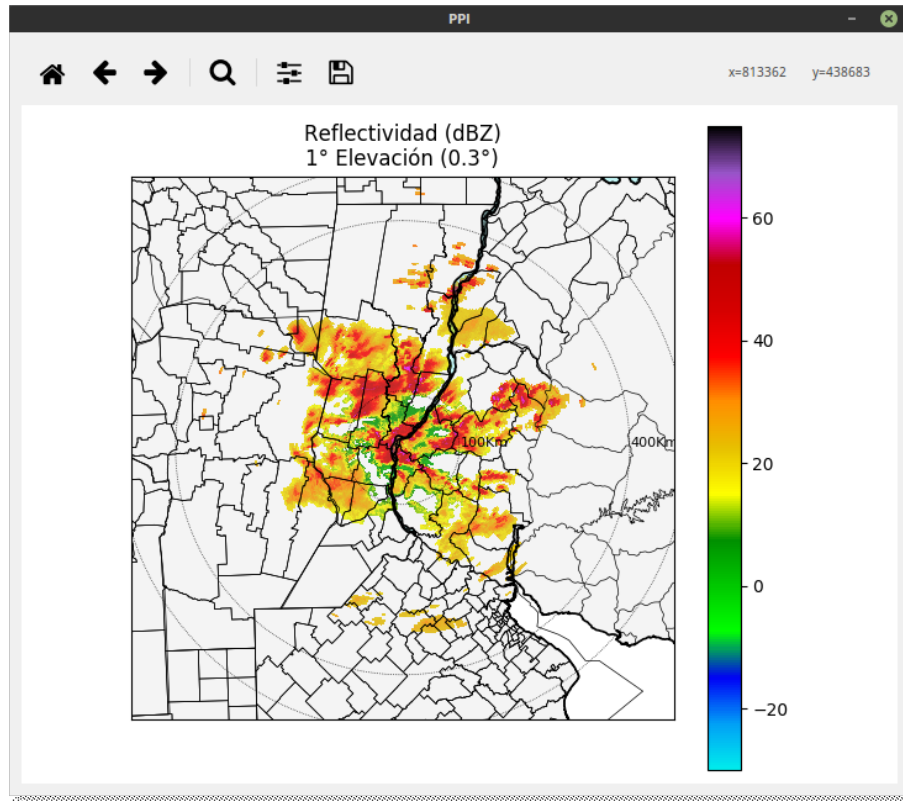


Figura 28: Meteorax - Producto PPI, archivo **.azi**

#### 5.4.5. Productos: RHI

Al igual que en la imagen que mostraba el producto PPI, aquí se ven 4 ventanas (Figura 33) correspondientes al mismo producto: **RHI**, pero en diferentes grados de azimut: 0, 90, 180 y 270 grados.

En el producto RHI se visualizan dos tipos de gráficos. El primero de ellos, en la parte superior corresponde propiamente al producto RHI y el que se encuentra debajo es un gráfico de referencia correspondiente al producto PPI en la primera elevación, centrado en la posición del radar, con una línea semi-transparente azul que indica geográficamente la dirección (el grado de azimut) que equivale al corte presentado en el gráfico superior. Este sub-gráfico PPI de referencia se actualizará a medida que cambiemos el grado de azimut en el componente de la parte superior de la ventana. Ambos gráficos incluyen la anteriormente mencionada barra de color.

Por último se incluyen los mismos controles que en el anterior producto, para poder manipular la vista actual en el gráfico del producto RHI actual.

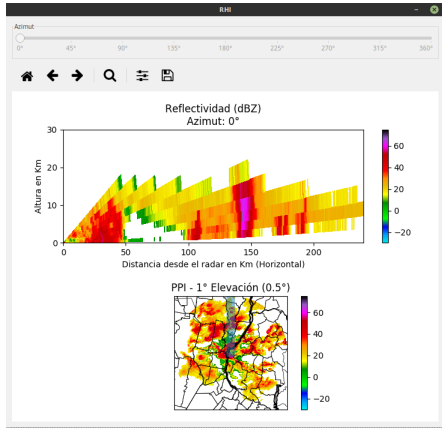


Figura 29: RHI, 0°

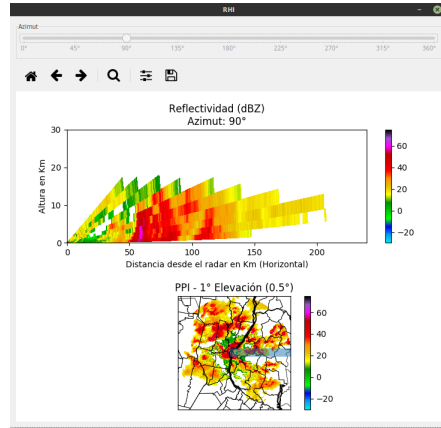


Figura 30: RHI, 90°

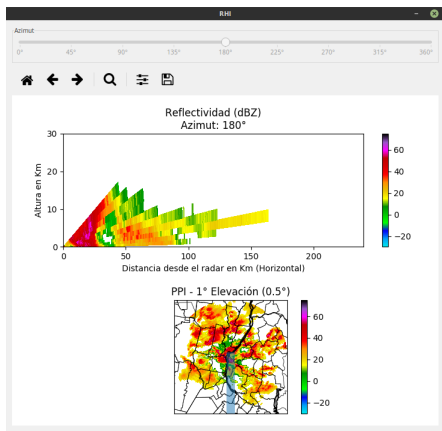


Figura 31: RHI, 180°

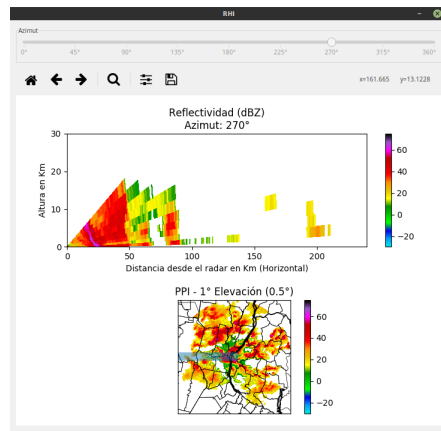


Figura 32: RHI, 270°

Figura 33: Meteorax - Producto RHI

#### 5.4.6. Productos: GeoTIFF

Al presionar el botón con la etiqueta «GeoTIFF» se abrirá una ventana cuyo contenido dependerá del tipo de escaneo realizado:

- Para escaneo **en volumen** (archivo **.vol**): Permitirá elegir las elevaciones a exportar, en forma de rango, dentro de las encontradas en el archivo de escaneo previamente seleccionado. En la figura de ejemplo (Figura 34), se exportarán las primeras 4 elevaciones del archivo.
- Para escaneo **en azimut** (archivo **.azi**): Informará que hay solamente una sola elevación y mostrará de cuál se trata (Figura 35).

A continuación se le consultará al usuario dónde guardar las imágenes TIFF generadas (Figura 36), equivalentes a las elevaciones comprendidas en el rango de la selección en el paso previo (una sola para archivos **.azi**).

Una vez seleccionado el destino de las imágenes, se procederá a generarlas internamente. Una vez finalizado el proceso se mostrará un mensaje de éxito (Figura 37), indicando la finalización de la escritura de las imágenes (Figura 38).

Recordar que para cada elevación seleccionada se generará un archivo correspondiente a los datos contenidos en la variable analizada, en formato **GeoTIFF**. El formato de nombre del archivo es el siguiente (todos los campos separados con un guión bajo):

- Iniciales de la ubicación del radar en minúsculas
- Variable analizada
- Fecha de escaneo
- Hora de escaneo
- Número de elevación

La extensión de los archivos será **.tif**.



Figura 34: Meteorax - GeoTIFF, selección de elevaciones, archivo **.vol**

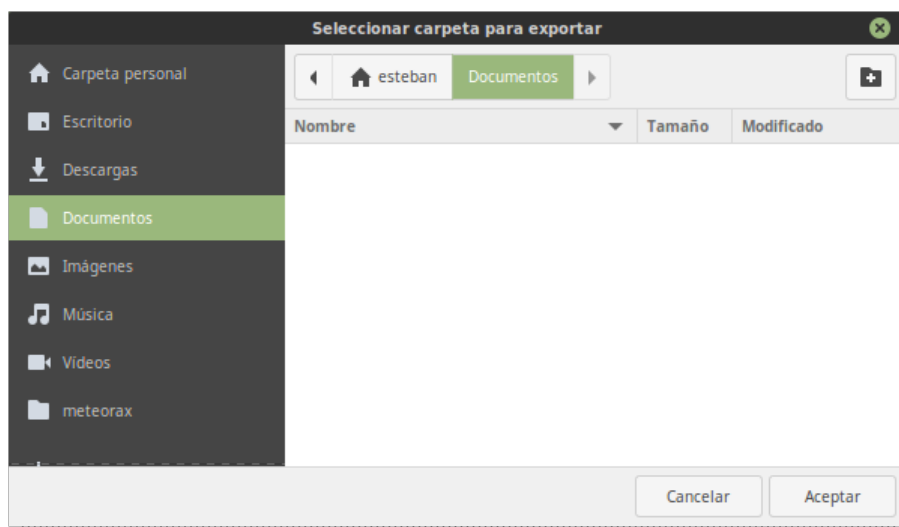
Figura 35: Meteorax - GeoTIFF, única elevación, archivo **.azi**

Figura 36: Meteorax - GeoTIFF, selección de carpeta de destino



Figura 37: Meteorax - GeoTIFF, mensaje de éxito

#### 5.4.7. Productos: ASCII

En cuanto a la exportación de los datos en formato de grilla ASCII, su comportamiento es similar al visto en la exportación de datos como GeoTIFF.

El usuario en primer lugar:

- Para escaneo **en volumen** (archivo **.vol**): Elegirá las elevaciones a exportar en este formato, seleccionando primer y última elevación en el rango

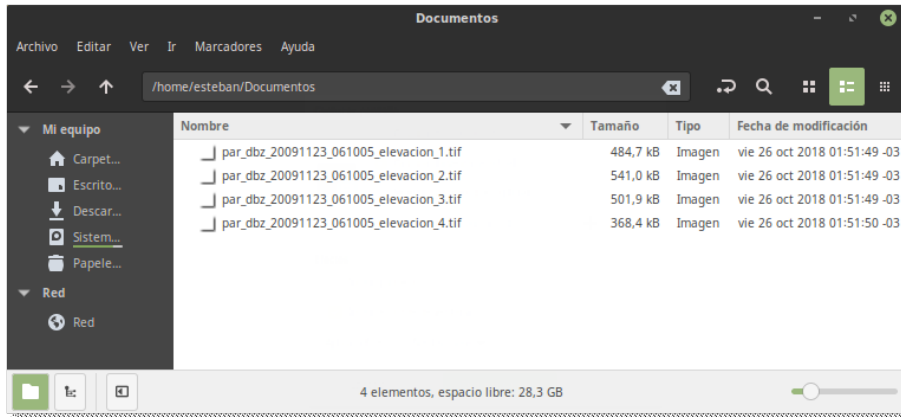


Figura 38: Meteorax - GeoTIFF, archivos exportados

de las elevaciones posibles (Figura 39). En la figura se puede ver que exportará las elevaciones 3 y 4.

- Para escaneo **en azimut** (archivo **.azi**): Será informado de que hay solamente una sola elevación y podrá ver de cuál se trata (Figura 40).

A continuación elige el destino de los archivos a generar por Meteorax (Figura 41). Paso siguiente, la aplicación procesará los datos para exportarlos en formato de grilla **ASCII** y escribirá los archivos resultantes en la carpeta seleccionada anteriormente (Figura 43), mostrando un mensaje de éxito (Figura 42).

El formato de nombre para este tipo de archivo es idéntico al visto en los archivos GeoTIFF, aunque, claro está, la extensión en este caso será **.asc**.

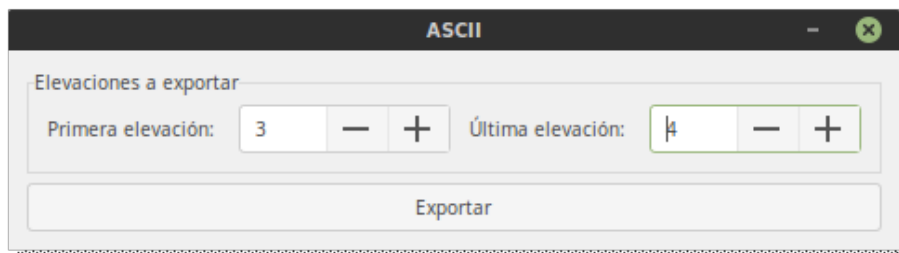


Figura 39: Meteorax - ASCII, selección de elevaciones

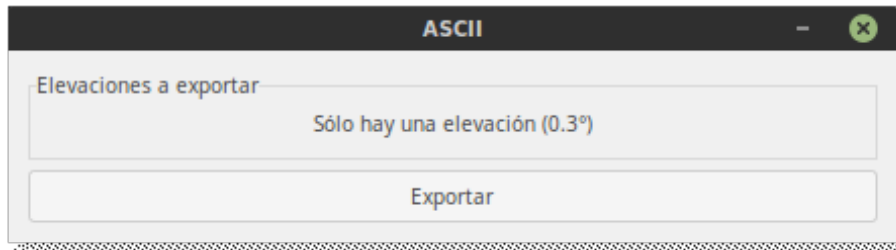
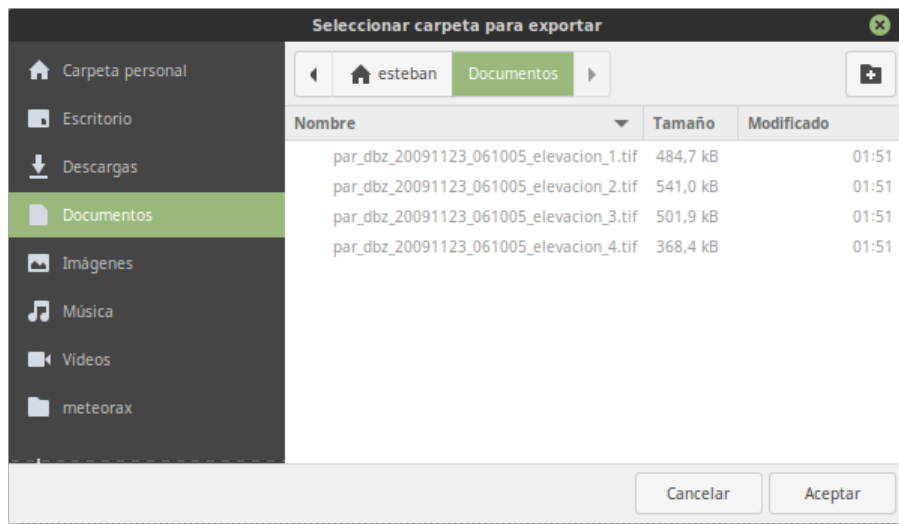
Figura 40: Meteorax - ASCII, única elevación, archivo **.azi**

Figura 41: Meteorax - ASCII, selección de carpeta de destino



Figura 42: Meteorax - ASCII, mensaje de éxito

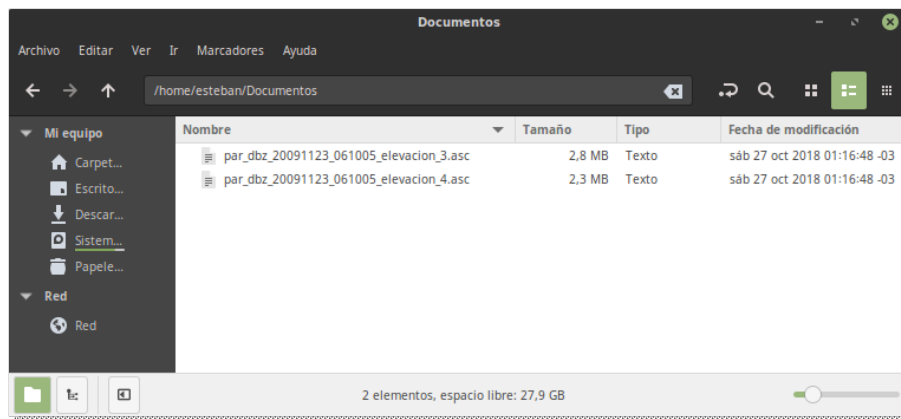


Figura 43: Meteorax - ASCII, archivos exportados



## 6. Conclusión

Este proyecto nace con la necesidad de contar con una herramienta gráfica capaz de permitirle a los miembros del Instituto de Clima y Agua de INTA Castelar, visualizar los resultados de los escaneos realizados por sus radares y poder exportarlos, en principio, en formatos básicos, conocidos en el área de la meteorología.

Pero la idea de esta tesina aparece a raíz de un problema de fondo: **el software y los formatos de datos utilizado en las estaciones de radar es propietario**, con lo que ello conlleva: costo en soporte y mantenimiento, limitaciones en las características ofrecidas e imposibilidad para extender funcionalidades propias del software (típico en esta clase de software), entre otras.

La ventaja con la que se cuenta es la de tener acceso a los archivos con los datos obtenidos de los escaneos, al menos para los usuarios en las estaciones de radar. Personalmente, el **acceso a los datos** lo es todo al inicio de un desafío de estas características: tenemos acceso a los datos salidos directamente como resultado de un proceso de escaneo realizado por un radar. Luego, el camino a transformar y presentar estos datos valiosos no debería ser una de las preocupaciones más importantes.

El primer paso evidente, era entender de qué se trataban los datos con los que se contaban y qué clase de información transportaban, cómo se procesarían y qué es lo que se buscaba obtener de los pasos anteriores. Entonces antes de comenzar nada, me tocó sumergirme en el mundo de la meteorología. Un mundo que me dejó cosas espectaculares y algunas otras me ayudaron a fortalecer conocimientos.

Desde el inicio de la investigación, hubo grandes encuentros con el conocimiento, como por ejemplo con la historia misma del radar, que incluye en su haber a grandes nombres de la ciencia, como **Marconi**, **Tesla** y **Hertz** (sí, el que le dió nombre a la unidad de frecuencia). Historia que data del radar como un instrumento incluso pensado y ampliamente favorable para fines bélicos, allá por la época de la Segunda Guerra Mundial. Luego, aprendiendo los fundamentos y las características de los componentes del radar conocí los “porqué” de la forma y la arquitectura de esos componentes.

Un punto fuerte de este proceso de aprendizaje fue entender el funcionamiento básico del radar como instrumento de detección de objetivos y medición. Fuerte, en forma positiva, debido a lo que me tocó conocer de las bases de un radar: las ondas electromagnéticas, el efecto Doppler, la dispersión y la reflectividad, entre muchas otras que recorrí. Ese aprendizaje me llevó a realizar cursos online de radares, como el dictado por la *University Corporation for Atmospheric Research* [12], el cual recomiendo ya que enseña y repasa lo básico de todo radar en un nivel muy comprensible. Parte de esa investigación me permite, hoy día, entender, con un cierto grado de profundidad, la “famosa” *imagen de radar* que vemos a diario en TV o internet. Esto es algo que debo agradecer: **aprendizaje continuo gracias a las posibilidades de acción de la informática.**

Toda esta investigación se fue plasmando en el presente documento, orde-

nando proceduralmente las secciones. Así luego llegó la hora de analizar los archivos de entrada. Ese proceso requirió de mucho análisis para comprender la forma de organización de los datos en los mismos. Una dificultad encontrada en el camino era la de lograr decodificar o procesar los datos binarios contenidos en los archivos XML (encuentro más cómodo para describir el proceso al verbo en inglés: “parse”). Ese desafío se pudo sortear relativamente fácil gracias a la utilización de librerías abiertas para el procesamiento de datos de radar, como lo son **PyART** [24] y **Wradlib** [31], que derivó en la utilización del lenguaje de programación **Python** [23] como base para construir lo que sería la aplicación prototipo de esta tesina. Al momento de comenzar con el desarrollo no tenía conocimientos del lenguaje o de su ecosistema, aunque no fue una limitación para continuar. Contrario a eso, sumo un lenguaje más a los *medianamente conocidos*. Existían inicialmente librerías en C++ que realizaban una parte del proceso de decodificación de los datos, pero eso derivaba en sumar y conectar mundos de lenguajes diferentes.

Por último en cuanto al desarrollo de la aplicación, fue una tarea agradable y entretenida aunque algunos procesos de exportación de los datos resultaban un poco más laboriosos de concretar debido en parte a mi conexión con el entendimiento de las estructuras de datos, las características ofrecidas por las librerías o por los ajustes que se debían realizar, en algunos casos en un modo *prueba y error*. Por suerte la ayuda de los colaboradores y de una investigación más profunda facilitaron esta tarea.

Ahora, analizando lo obtenido y más allá de las extensiones, mejoras y ajustes a realizar en la aplicación **Meteorax**, basados en su utilización y/o en las necesidades de los usuarios, quiero cerrar esta conclusión con algunos conceptos entrelazados, que me surgieron de todo este proceso:

- Acceso a datos originales, que dispara grandes posibilidades.
- Desarrollo de herramientas libres y multiplataformas.
- Posibilidad de extender funcionalidades.
- Acceso a la información por parte de cualquier tipo de usuario interesado en la meteorología.
- Reemplazo de productos cerrados.

## 7. Trabajos relacionados

- **Desarrollo y uso de herramientas libres para la explotación de datos de los radares meteorológicos del INTA** [39]

En este trabajo, integrantes del Instituto de Clima y Agua de INTA Castellar, presentan herramientas desarrolladas completamente utilizando software libre para el análisis de datos y la aplicación de un proceso de minería sobre los mismos. Se tomaron como fuente de entrada escaneos realizados en la estación de radar de Anguil, La Pampa. Los resultados fueron exitosos y el acoplamiento con el resto de las herramientas de software libre transcurrió sin problemas. Se logró clasificar nuevos eventos basados en los modelos obtenidos por el proceso de este trabajo.

Autores: Yanina Bellini Saibene, Martín Volpaccio, Santiago Bancharo y Romina Mezher.

- **Aplicación de línea de comandos para procesamiento de volúmenes: radar-cmd** [40]

Durante la elaboración de esta tesina, se desarrolló una herramienta de consola, desarrollada en Python que también utiliza Py-ART [24] como librería de procesamiento de datos de radar. Toma como entrada la misma clase de archivos que Meteorax y provee varias funcionalidades relacionadas a la exportación de imágenes de radar, como los formatos GeoTIFF, grilla ASCII e incluso composiciones en mosaico. Además, permite configurar el método de generación de imagen o la máscara para el procesamiento de las variables ingresadas.

Autor: Andres Giordano.

- **Wradlib** [31] y **Py-ART** [24]

Si bien estas dos librerías sirvieron como base de procesamiento para el desarrollo de la aplicación Meteorax, los usuarios con conocimiento en Python [23], pueden hacer uso de las mismas para procesar datos de diferentes tipos de archivos e incluso para diferentes tipos de radares. Es probable que se necesite un poco más de conocimiento en scripting, variables de radar y procesamiento de datos. En el ambiente del desarrollo de herramientas para radar son muy populares y cuentan con una gran comunidad de desarrolladores y colaboradores. Su documentación es buena y los ejemplos son comprensibles.

## 8. Trabajos a futuro

A continuación se presentan posibles extensiones, mejoras y propuestas a raíz de la investigación y el prototipo de aplicación desarrollada, **Meteorax**:

- **Agregar más formatos/productos de exportación**

Actualmente Meteorax exporta los productos **PPI** y **RHI**, vistos en la subsección 5.4.4 y en la subsección 5.4.5, respectivamente. Pero existen varios productos (Ver subsección 5.3.4) que podrían resultar de interés para los usuarios de la aplicación. Con respecto a esto, se tendría que evaluar los alcances de las librerías de procesamiento como lo es **Py-ART** y **Wradlib**, considerando las capacidades de las mismas para exportar dicho tipo de productos. Otra limitación es la compatibilidad del tipo de escaneo, y por ende la forma de organización de los datos, con el producto a agregar como exportación, ya que puede que dependiendo del tipo de escaneo (tipo de archivo de entrada) no sea posible producir los resultados buscados.

- **Permitir el análisis a partir de otros tipos de archivos**

Contar con el análisis de archivos tipo **.ele** y completar así la cobertura de los 3 tipos de archivos de escaneo (**.azi**, **.vol** y **.ele**), resultantes de los procesos de escaneo. La limitación actual es la configuración actual de las estaciones de radar que sólo generan cada cierto intervalo de tiempo archivos **.azi** y **.vol**, solamente. Cabe recordar que un escaneo en volumen contiene *varias* elevaciones, mientras que el escaneo de elevación sólo contiene una específica, que resulta en un archivo **.ele**.

- **Vistas en 3D**

Una funcionalidad muy interesante sería la de poder ver las estructuras de los eventos meteorológicos en **3 dimensiones**. Actualmente en Meteorax se exportan imágenes en un plano de dos dimensiones. En el producto RHI puede verse un corte *vertical* de la situación, mientras que en el producto PPI podemos observar un corte *horizontal*. ¿Qué sucedería si se logra combinar las dos vistas y sumar al eje *Z* en la exportación?. Esa pregunta podría dar inicio a esta nueva funcionalidad, que daría paso a una vista mucho más realista y detallada.

- **Sitio online para el procesamiento de archivos**

Ofrecer la misma funcionalidad que tiene actualmente Meteorax pero a través de un servicio online que permita subir un archivo, o seleccionarlo de una lista de archivos disponibles para ser procesados y posteriormente visualizar los productos seleccionados. Incluso darle la opción al usuario de descargar las imágenes obtenidas. La opción de contar con los archivos en la nube, dependerá de las posibilidades de acceso a los mismos. Los archivos actuales se obtienen por medio de una conexión FTP con la estación de radar y está limitada a los usuarios con permisos para realizar dicha descarga (Ver la sección 3.2).

- **Conexión directa con la estación de radar**

Relacionado al punto anterior, la obtención de los archivos está supeditada a permisos sobre el acceso al servidor que provee los archivos de radar, aunque sería de gran utilidad proveer al usuario acceso a los archivos generados en el último lapso de tiempo (o incluso buscarlos en un historial). De esta manera el paso de obtenerlos de forma manual, quedará reducida a una simple acción. No sería de gran dificultad implementar un cliente FTP en Python, para acceder a los escaneos recientes.

## 9. Referencias

- [1] *Sistemas de Información Clima y Agua* Instituto de Clima y Agua - INTA. <http://climayagua.inta.gob.ar/>.
- [2] Maxwell, J. C. VIII. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London* **155**, 459-512. <http://rstl.royalsocietypublishing.org/content/155/459.short> (1865).
- [3] Skolnik, M. *Introduction to radar systems* (McGraw-Hill, 1980).
- [4] Wales, A. C. -. N. M. *Marconi's Apparatus* <https://museum.wales/collections/online/object/adc9f773-0045-36fe-8e7a-a9b8355887ac/Marconis-Apparatus-photograph/>.
- [5] Raboy, M. *Marconi: the man who networked the world* (Oxford University Press, 2016).
- [6] Joachim, E. *98 Years of the radar principle: the inventor Christian Hülsmeyer* (2002).
- [7] H, T. A., C, Y. L. y A, H. L. *Granted Patent US 1981884 A* (United States). <https://patents.google.com/patent/US1981884A> (1934).
- [8] Swords, S. *Technical History of the Beginnings of Radar* (P. Peregrinus, 1986).
- [9] Austin, B. A. Precursors to Radar — The Watson-Watt Memorandum and the Daventry Experiment. *International Journal of Electrical Engineering Education* **36** (1999).
- [10] Varian, R. H. y Varian, S. F. A High Frequency Oscillator and Amplifier. *Journal of Applied Physics* (1939).
- [11] Hans, E. H. *Granted Patent US 2123728 A* (United States). <https://patents.google.com/patent/US2123728A>(1938).
- [12] For Atmospheric Research, U. C. *Fundamentos de radar meteorológico* [http://www.meted.ucar.edu/radar/basic\\_wxradar\\_es/index.htm](http://www.meted.ucar.edu/radar/basic_wxradar_es/index.htm). 2017.
- [13] Rinehart, R. E. *Radar for Meteorologists* (Rinehart, 2004).
- [14] Eaves, J. L. *Principles of Modern Radar* (Springer US, 1987).
- [15] NASA. *Basics – Solar System Exploration: NASA Science* <https://solarsystem.nasa.gov/basics/chapter6-4>.
- [16] Fukao, S. *Radar for meteorological and atmospheric observations* ISBN: 9784431543343 (Springer, 2013).
- [17] Wolff, C. *Fundamentos de Radar* <http://www.radartutorial.eu/>.
- [18] *Radar Duplexer basics* <http://www.rfwireless-world.com/ApplicationNotes/Radar-Duplexer.html>.
- [19] *Duplexers* <https://www.microwaves101.com/encyclopedias/duplexers>.

- [20] *Waveguide Primer* <https://www.microwaves101.com/encyclopedias/waveguide-primer>.
- [21] *Introduction to Waveguides* <http://www.antenna-theory.com/tutorial/waveguides/waveguide.php>.
- [22] *Guía de usuario Rainbow® 5* Inglés. Ver. 5.22.0. SELEX ES ().
- [23] Foundation, P. S. *Welcome to Python.org* <https://www.python.org>.
- [24] Helmus, J. J. y Collis, S. M. The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language. *Journal of Open Research Software* **4**. <https://doi.org/10.5334%2Fjors.119> (jul. de 2016).
- [25] Jones, E., Oliphant, T., Peterson, P. *y col. SciPy: Open source scientific tools for Python* <http://www.scipy.org/>.
- [26] Van der Walt, S., Colbert, S. C. y Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* **13**, 22-30. <http://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37> (2011).
- [27] Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **9**, 90-95. <http://aip.scitation.org/doi/abs/10.1109/MCSE.2007.55> (2007).
- [28] Rew, R., Davis, G., Emmerson, S. *y col. Unidata NetCDF*. <https://doi.org/10.5065/D6H70CW6> (1989).
- [29] *The Python Standard Library* 2017. <https://docs.python.org/3.7/library>.
- [30] *Cf/Radial RADAR/LIDAR Data Format* 2017. [https://www.ral.ucar.edu/projects/titan/docs/radial\\_formats/cfradial.html](https://www.ral.ucar.edu/projects/titan/docs/radial_formats/cfradial.html).
- [31] Heistermann, M., Jacobi, S. y Pfaff, T. An open source library for processing weather radar data (wradlib). *Hydrology and Earth System Sciences* **17**, 863-871. <https://doi.org/10.5194%2Fhess-17-863-2013> (feb. de 2013).
- [32] Documentation, P. *Overview — PyGObject* <https://pygobject.readthedocs.io/en/latest/>.
- [33] Project, T. G. *Glade - A User Interface Designer* <https://glade.gnome.org/>.
- [34] Society, A. M. *Glossary of Meteorology* <http://glossary.ametsoc.org>.
- [35] *TIFF 6.0 Specification* Adobe Systems Incorporated. <https://web.archive.org/web/20161015094213/http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>.
- [36] *GeoTIFF* OSGeo. <https://trac.osgeo.org/geotiff/>.
- [37] Roelofs, G. *PNG: The Definitive Guide* (ed. Koman, R.) ISBN: 1565925424 (O'Reilly & Associates, Inc., 1999).
- [38] Wiki, G. *ArcInfo ASCII Grid format - GeoTools - Codehaus* <https://web.archive.org/web/20150128024528/http://docs.codehaus.org/display/GEOTOOLS/ArcInfo+ASCII+Grid+format>.

- [39] Saibene, Y. B., Volpaccio, M., Banchemo, S. y Mezher, R. *Desarrollo y uso de herramientas libres para la explotación de datos de los radares meteorológicos del INTA en XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO)-VI Congreso Argentino de AgroInformática (CAI)(Buenos Aires, 2014)* (2014).
- [40] Giordano, A. *INTA-Radar/radar-cmd: Aplicación de línea de comandos para procesamiento de volúmenes y utilizando Py-ART* <https://github.com/INTA-Radar/radar-cmd>.