

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



**“Desarrollo de un Analizador de Protocolo
ASTERIX de Eurocontrol”**

PRESENTADO POR:
EDGAR MANUEL MORAZÁN BONILLA

PARA OPTAR AL TÍTULO DE:
INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, JULIO DE 2011

UNIVERSIDAD DE EL SALVADOR

RECTOR :

MSc. RUFINO ANTONIO QUEZADA SÁNCHEZ

SECRETARIO GENERAL :

LIC. DOUGLAS VLADIMIR ALFARO CHÁVEZ

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO :

ING. MARIO ROBERTO NIETO LOVO

SECRETARIO :

ING. OSCAR EDUARDO MARROQUÍN HERNÁNDEZ

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR :

ING. JOSÉ WILBER CALDERÓN URRUTIA

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

**“Desarrollo de un Analizador de Protocolo
ASTERIX de Eurocontrol”**

Presentado por :

EDGAR MANUEL MORAZÁN BONILLA

Trabajo de Graduación Aprobado por :

Docente Director :

ING. WERNER DAVID MELÉNDEZ VALLE

San Salvador, Julio de 2011

Trabajo de Graduación Aprobado por:

Docente Director :

ING. WERNER DAVID MELÉNDEZ VALLE

A Dios todopoderoso, que me ha permitido culminar con éxito el presente trabajo de graduación, sea la gloria, el honor y la majestad por los siglos de los siglos.

“Mira que te mando que seas valiente, no temas ni desmayes, porque yo, Jehová tu Dios, estaré contigo en cualquier lugar que tú fueres”.

Josué 1: 7 - 9.

Dios no me trajo hasta aquí para volver atrás...

Edgar Manuel Morazán Bonilla

AGRADECIMIENTOS

Agradezco por su apoyo incondicional y constante a Dios Todopoderoso que me ha brindado siempre las herramientas y medios para salir adelante. A mi Madre Vilma Gloria Bonilla, por su apoyo a lo largo del desarrollo del trabajo. A mi abuelita Velia Santos Bonilla, quien con un enorme sacrificio siempre ha estado en la disposición de dar todo, absolutamente todo por sus nietos y quien ha orado por mí; cuando ni siquiera yo mismo lo he hecho.

Agradezco a mi amigo que ha sido una persona que me ha ayudado en momentos críticos y de quien he aprendido, aprendo y seguiré aprendiendo siempre; Marvin Rafael Valencia Pérez. A mis maestros: Ing. Werner Meléndez, Dr. Carlos Martínez, Ing. Ricardo Ernesto Cortez, Ing. Wilber Calderón, Ing. Hugo Colato, Ing. Armando Martínez Calderón, entre muchos otros.

A las autoridades del gobierno de El Salvador, por permitirme realizar las pruebas de campo en su Centro de Control.

Finalmente agradezco a todas aquellas personas que de una u otra forma han influido en mi vida para ser lo que ahora soy y ayudarme para ser lo que el futuro ha de demandar de mí.

PREFACIO

La aplicación de los sistemas de radar comienza a finales de la década de 1930, cuando inicia su uso con fines militares, posteriormente se utilizaron en aplicaciones civiles, específicamente para el control de tráfico aéreo. En Centroamérica se instalaron los primeros radares a mediados de la década de 1990. Hace algunos años, cada fabricante de radares, implementaba tecnologías cerradas, de tal forma que el concepto de integración de sistemas era muy difícil de realizar; esto cambio con el surgimiento de nuevas normas de parte de los organismos reguladores de este tipo de servicios, de modo que se han desarrollado códigos “abiertos” de transmisión de datos para sistemas radar. Uno de los códigos abiertos más utilizados a nivel mundial es el ASTERIX de Eurocontrol, el cual facilita la integración de sistemas y el intercambio de datos radar entre diferentes regiones, lo que fortalece la seguridad del control de tráfico aéreo. El estudio de sistemas radar, forma parte de los temas abordados en la asignatura de Sistemas de Radionavegación Aérea de la Escuela de Ingeniería Eléctrica, de tal forma que el estudio del protocolo ASTERIX es muy importante en esta área. El presente trabajo de graduación ofrece un sistema didáctico de fácil utilización orientado a la enseñanza-aprendizaje, que proporciona información detallada sobre el contenido de las tramas ASTERIX y su relación con los conceptos básicos radar, para que un estudiante que no esté familiarizado con estos conceptos, lo pueda utilizar. El objetivo del presente trabajo de graduación es fortalecer el laboratorio de telecomunicaciones de la Escuela de Ingeniería Eléctrica, con el desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol.

Los alcances planteados al inicio de la tesis fueron: el desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol, que sea capaz de interconectarse con redes LAN de datos radar y extraer de ellas las tramas del protocolo en cuestión; posteriormente se analizarían los diferentes campos que la conforman y se presentarían los resultados en una interfaz grafica que facilite su comprensión e interpretación. Así mismo, deberá tener la capacidad de realizar análisis estadísticos básicos, entre ellos la probabilidad de detección, de falsos blancos y la probabilidad de detección de códigos, basados en las tramas capturadas. Deberá incluir la opción de actualizar datos con cada período de rotación de antena (información que se incluye dentro de las tramas a capturar) y desplegar los datos de forma grafica, en una pantalla que simule un presentador radar. Una vez desarrollado y aceptado el Analizador, se debían elaborar guías que permitieran su uso en prácticas de laboratorio en la Escuela de Ingeniería Eléctrica (EIE). Los alcances desarrollados al final de la tesis, fueron el desarrollo del Analizador de Protocolo ASTERIX de Eurocontrol, como lo plantean los alcances al inicio de la tesis, con la diferencia de que dicho software, no solo permite la visualización de todos los blancos, si no que tiene una opción para visualizar uno en específico, como también una opción para habilitar la visualización de pistas.

RESUMEN DEL TRABAJO

El presente trabajo de graduación se divide en cuatro partes principales, la primera contiene los conceptos básicos sobre radares, que describen el funcionamiento de un sistema radar y como los mismos han evolucionado hasta convertirse en una herramienta importante para la seguridad del transporte aéreo civil. Al final de esta primera parte, se abordan los temas de: concepto de integración de información o sistemas abiertos y los parámetros para establecer la calidad de un sistema radar.

La segunda parte del documento trata sobre el Protocolo Asterix de Eurocontrol, haciendo énfasis en su importancia, debido a que facilita la integración de sistemas y el intercambio de datos radar entre diferentes regiones, lo que fortalece la seguridad del control de tráfico aéreo. En esta parte se describe todo lo relacionado con el Protocolo Asterix de Eurocontrol, como se organizan sus datos, la aplicación a los radares que existen en Centroamérica (INDRA y ALENIA), el envío de datos radar, ejemplo de una trama Asterix y una breve descripción de analizadores comerciales de software libre (WRTADS y RBAT).

La tercera parte del documento describe de forma detallada la aplicación software que se desarrollo, a la cual se le llamo Analizador de Protocolo Asterix de Eurocontrol (ADPADEC), definiendo que es un sistema didáctico de fácil utilización orientado a la enseñanza-aprendizaje para estudiantes que comienzan a familiarizarse con conceptos básicos de radar y el protocolo Asterix de Eurocontrol. En esta parte es donde se mencionan las características generales de la aplicación, los criterios de diseño, descripción de cada uno de los algoritmos que integran el software ADPADEC, así como también la forma, de utilizar los tipos de análisis mediante la interface grafica de usuario de la aplicación.

En la cuarta parte se muestran las pruebas de campo realizadas utilizando el software, conectándose a un radar (ubicado sobre la carretera panorámica y administrado por autoridades del gobierno de El Salvador) y los resultados obtenidos del radar modelo IRS-20MP/L, fabricado por la empresa Indra de España. Es en esta parte donde se muestran resultados de: captura de datos, decodificación de datos, visualización de datos radar y estadísticas.

Los resultados obtenidos al final del presente trabajo de graduación, es un sistema didáctico de fácil utilización orientado a la enseñanza-aprendizaje, que proporciona información detallada sobre el contenido de las tramas ASTERIX y su relación con los conceptos básicos radar, para que un estudiante que está comenzando a familiarizarse con estos conceptos, lo pueda utilizar en prácticas de laboratorio, en la Escuela de Ingeniería Eléctrica.

TABLA DE CONTENIDOS

Capítulo	Página
1 CONCEPTOS DE RADAR.....	1
1.1 PRINCIPIOS BASICOS DE OPERACION DE UN RADAR.....	2
1.1.1 Determinación de Rango o Distancia del objeto	2
1.1.2 Determinación del Acimut o angulo	5
1.1.3 Altitud del blanco.....	6
1.1.4 Velocidad y dirección de desplazamiento del blanco	7
1.1.5 Imágenes del objeto	8
1.1.6 Volumen de lluvia o gases	9
1.1.7 Intercambio de información	11
1.1.8 Evolución de la altitud.....	12
1.2 CLASIFICACION DE RADARES.....	12
1.2.1 Clasificación de radares por tipo de transmisión	12
1.2.1.1 Radar de onda continua.....	12
1.2.1.2 Radar de onda pulsante.....	13
1.2.2 Clasificación de radares según el blanco	15
1.2.2.1 Radar Primario.....	15
1.2.2.2 Radar Secundario	15
1.2.3 Clasificación de radares por su aplicación.....	16
1.2.3.1 Radares Civiles	16
1.2.3.1.1 Radares Aeronauticos.....	16
1.2.3.1.2 Radares Meteorologicos.....	16
1.2.3.2 Radares Militares	17
1.2.4 Clasificación de radares por su cobertura y ubicación	17
1.2.4.1 Area terminal.....	17
1.2.4.2 Aproximación	17
1.2.4.3 Control de ruta.....	17
1.2.5 Clasificación de radares por el rango de frecuencias en el cual operan	17
1.3 DESARROLLO DE LA NAVEGACION AEREA.....	20
1.4 RADARES AERONAUTICOS.....	25
1.4.1 Historia de radares aeronauticos	25
1.4.2 Función de radares aeronauticos	28
1.5 CARACTERISTICAS TECNICAS DE RADARES AERONAUTICOS.....	28
1.5.1 Radar Primario de Vigilancia (PSR).....	28
1.5.2 Radar Secundario de Vigilancia (SSR).....	31
1.5.2.1 Transponder.....	33
1.5.2.2 Antena del SSR.....	35
1.5.2.3 Transmisor del SSR.....	35
1.5.2.4 Receptor del SSR.....	37
1.5.2.5 Coder del SSR.....	37
1.5.2.6 Decoder del SSR.....	37

1.5.2.6 Decoder del SSR.....	37
1.5.3 Radar Secundario de Vigilancia Modo S.....	40
1.6 ORGANISMOS NORMATIVOS DE SISTEMAS RADAR.....	42
1.7 CONCEPTO DE INTEGRACION DE INFORMACION O SISTEMAS ABIERTOS.....	43
1.8 PARAMETROS PARA ESTABLECER LA CALIDAD DE UN SISTEMA RADAR.....	43
1.9 ALGUNOS REQUISITOS DE OACI PARA SISTEMAS DE RADAR CIVILES.....	44
1.10 BIBLIOGRAFIA.....	45
2 PROTOCOLO ASTERIX DE EUROCONTROL.....	47
2.1 OBJETIVOS DEL PROTOCOLO ASTERIX DE EUROCONTROL.....	48
2.2 ORGANIZACION DE LOS DATOS ASTERIX.....	48
2.2.1 Data categories (Categorías de datos).....	50
2.2.2 Data Item (Item de Datos).....	50
2.2.3 Data Field (Campo de Dato).....	51
2.2.4 User Application Profile (UAP, Perfil de Aplicación de Usuario).....	51
2.3 ESTRUCTURA DEL MENSAJE.....	52
2.3.1 Data Block (Bloque de Dato).....	52
2.3.2 Registers (Registros).....	52
2.3.3 Organización Secuencial.....	55
2.3.4 Field Specification (FSPEC).....	55
2.3.5 Organización aleatoria.....	56
2.4 IDENTIFICACION DE LA FUENTE DE DATOS (SAC/SIC).....	56
2.5 DESCRIPCION DE LA CATEGORIA 001.....	58
2.6 DESCRIPCION DE ITEMS DE DATOS ESTANDAR DE LA CATEGORIA 001.....	61
2.6.1 ITEM I001/010 Identificación Fuente de Datos.....	61
2.6.1.1 Descripción de los bits del Item de Datos I001/010.....	62
2.6.2 ITEM I001/020 Descripción del informe de blanco.....	62
2.6.2.1 Descripción de los bits de la primera parte del Item de Datos I001/020.....	62
2.6.2.2 Descripción de los bits de la primera extensión del Item de Datos I001/020.....	63
2.6.3 ITEM I001/030 Condiciones de Alarma y Error.....	64
2.6.3.1 Descripción de los bits de la primera parte del Item de Datos I001/030.....	64
2.6.4 ITEM I001/040 Posición medida en Coordenadas Polares.....	65
2.6.4.1 Descripción de los bits del Item de Datos I001/040.....	66
2.6.5 ITEM I001/042 Posición calculada en Coordenadas Cartesianas.....	66
2.6.5.1 Descripción de los bits del Item de Datos I001/042.....	66
2.6.6 ITEM I001/050 Modo 2 en representación Octal.....	67
2.6.6.1 Descripción de los bits del Item de Datos I001/050.....	67
2.6.7 ITEM I001/060 Indicador de confianza del Modo 2.....	68
2.6.7.1 Descripción de los bits del Item de Datos I001/060.....	68
2.6.8 ITEM I001/070 Modo 3/A en representación Octal.....	68
2.6.8.1 Descripción de los bits del Item de Datos I001/070.....	69
2.6.9 ITEM I001/080 Indicador de confianza del Modo 3/A.....	69
2.6.9.1 Descripción de los bits del Item de Datos I001/080.....	70
2.6.10 ITEM I001/090 Código Modo C en representación Binaria.....	70
2.6.10.1 Descripción de los bits del Item de Datos I001/090.....	70

2.6.11 ITEM I001/100 Indicador de confianza del Modo C	71
2.6.11.1 Descripción de los bits del Item de Datos I001/100	71
2.6.12 ITEM I001/120 Velocidad Radial Doppler Medida	72
2.6.12.1 Descripción de los bits del Item de Datos I001/120	72
2.6.13 ITEM I001/130 Características del Plot radar	72
2.6.13.1 Descripción de los bits del Item de Datos I001/130	73
2.6.14 ITEM I001/131 Potencia recibida	73
2.6.14.1 Descripción de los bits del Item de Datos I001/131	73
2.6.15 ITEM I001/141 Tiempo Truncado del Día	74
2.6.15.1 Descripción de los bits del Item de Datos I001/141	74
2.6.16 ITEM I001/150 Presencia del Pulso X	74
2.6.16.1 Descripción de los bits del Item de Datos I001/150	75
2.6.17 ITEM I001/161 Numero de Pista/Plot	75
2.6.18 ITEM I001/170 Estado de la Pista	76
2.6.18.1 Descripción de los bits de la primera parte del Item de Datos I001/170	76
2.6.18.2 Descripción de los bits de la primera extensión del Item de Datos I001/170	77
2.6.19 ITEM I001/200 Velocidad de la pista calculada en coordenadas polares	77
2.6.19.1 Descripción de los bits del Item de Datos I001/200	78
2.6.20 ITEM I001/210 Calidad de la pista	78
2.6.20.1 Descripción de los bits del Item de Datos I001/210	78
2.7 DESCRIPCION DE LA CATEGORIA 002	79
2.8 DESCRIPCION DE ITEMS DE DATOS ESTANDAR DE LA CATEGORIA 002	80
2.8.1 ITEM I002/000 Tipo de Mensaje	80
2.8.1.1 Descripción de los bits del Item de Datos I002/000	80
2.8.2 ITEM I002/010 Identificación Fuente de Datos	80
2.8.2.1 Descripción de los bits del Item de Datos I002/010	81
2.8.3 I002/020 Número de Sector	81
2.8.3.1 Descripción de los bits del Item de Datos I002/020	81
2.8.4 ITEM I002/030 Tiempo de Día	81
2.8.4.1 Descripción de los bits del Item de Datos I002/030	82
2.8.5 ITEM I002/041 Período de rotación de antena	82
2.8.5.1 Descripción de los bits del Item de Datos I002/041	83
2.8.6 ITEM I002/050 Estado de configuración de la estación	83
2.8.6.1 Descripción de los bits del Item de Datos I002/050	83
2.8.7 ITEM I002/060 Modo de procesamiento de la estación	83
2.8.7.1 Descripción de los bits del Item de Datos I002/060	84
2.8.8 ITEM I002/070 Valor Contador de Plots	84
2.8.8.1 Descripción de los bits del Item de Datos I002/070	85
2.8.9 ITEM I002/080 Condiciones de Alarma/Error	85
2.8.9.1 Descripción de los bits del Item de Datos I002/080	86
2.8.10 ITEM I002/090 Error de colimación	86
2.8.10.1 Descripción de los bits del Item de Datos I002/090	86
2.8.11 ITEM I002/100 Ventana Dinámica – Tipo 1	88
2.8.11.1 Descripción de los bits del Item de Datos I002/100	88

2.9 APLICACION DE ASTERIX AL RADAR INDRA IRS 20MP/L.....	88
2.10 APLICACION DE ASTERIX A RADARES ALENIA.....	89
2.11 ENVIO DE DATOS RADAR.....	90
2.12 EJEMPLO DE UNA TRAMA ASTERIX.....	90
2.13 CAPTURA DE DATOS ASTERIX CON WIRESHARK.....	91
2.14 EVALUADORES COMERCIALES DE ASTERIX.....	92
2.14.1 Evaluador comercial WRTADS.....	92
2.14.1.1 Explicación de algunas opciones de WRTADS.....	92
2.14.1.2 Barra de herramientas de WRTADS.....	93
2.14.1.3 Leyenda de símbolo de blanco.....	95
2.14.1.4 Simbolos de seguimiento miscelaneo.....	95
2.14.1.5 Simbolos de busqueda.....	96
2.14.2 Evaluador comercial RBAT.....	97
2.14.2.1 Barra de herramientas de RBAT.....	97
2.14.2.2 Barra de herramientas de Real Time RBAT.....	99
2.14.2.3 Tipos de análisis de RBAT.....	100
2.14.2.4 Análisis de Vigilancia.....	101
2.14.2.5 Entradas de usuario del Análisis de Vigilancia.....	105
2.14.2.6 Salida de texto del Análisis de Vigilancia.....	107
2.14.2.7 Salida de plot del Análisis de Vigilancia.....	109
2.14.2.8 Archivo de extracción de salida del Análisis de Vigilancia.....	110
2.15 BIBLIOGRAFIA.....	111
3 DESCRIPCION DEL SISTEMA PROPUESTO.....	112
3.1 CARACTERISTICAS GENERALES DE LA APLICACION.....	112
3.2 CRITERIOS DE DISEÑO DE LA APLICACION.....	113
3.2.1 Algoritmo del software ADPADEC.....	113
3.2.2 Algoritmo de interface grafica de usuario.....	114
3.2.3 Algoritmo para la captura de datos.....	115
3.2.4 Algoritmo de extracción y decodificación.....	116
3.2.5 Algoritmo de procesamiento para análisis.....	121
3.2.6 Algoritmo para diferentes cálculos de probabilidad de detección.....	123
3.2.7 Algoritmo para el análisis de visualización radar.....	125
3.2.8 Algoritmo para el cálculo de las estadísticas.....	125
3.3 DESCRIPCION DE LA INTERFACE GRAFICA DE USUARIO.....	127
3.4 DESCRIPCION DE LOS ANALISIS DE LA APLICACION.....	128
3.4.1 Captura de datos.....	128
3.4.2 Extracción y decodificación.....	129
3.4.3 Procesamiento para análisis.....	131
3.4.4 Probabilidad de detección.....	132
3.4.4.1 Ventana de resultados de probabilidad de detección.....	132
3.4.4.2 Apartado de datos a ingresar de probabilidad de detección.....	133
3.4.4.3 Apartado de opción del área de gráficos y botones.....	134
3.4.4.4 Area de resultados.....	135
3.4.4.5 Area de gráficos.....	136

3.4.5 Visualización radar	138
3.4.5.1 Indicador de posición de plan	138
3.4.5.2 Opciones de visualización radar	139
3.4.5.3 Simbología de plots radar	143
3.4.5.4 Simbología de plots radar de baja confianza	144
3.4.5.5 Simbología de plots para códigos de emergencia	144
3.4.6 Estadísticas.....	145
3.5 VENTANA DE ERRORES	147
3.5.1 Error de nombre de archivo	147
3.5.2 Error de especificación de un código transponder.....	147
3.5.3 Error de tipo de dato	148
3.5.4 Error de intervalo de un dato	148
3.6 REQUERIMIENTOS DE HARDWARE DE LA APLICACION.....	148
3.7 REQUERIMIENTOS DE SOFTWARE DE LA APLICACION.....	148
3.8 INSTRUCCIONES DE EJECUCION DEL SOFTWARE ADPADEC.....	149
3.9 LIMITANTES DEL SOFTWARE ADPADEC	150
3.10 UTILIZACION DEL SOFTWARE ADPADEC.....	151
3.11 BIBLIOGRAFIA	152
4 APLICACION DEL SISTEMA PROPUESTO	153
4.1 RESULTADOS DE LA CAPTURA DE DATOS RADAR.....	153
4.2 RESULTADOS DE EXTRACCION Y DECODIFICACION DE DATOS RADAR.....	155
4.3 RESULTADOS DEL PROCESAMIENTO PARA ANALISIS DE DATOS.....	157
4.4 RESULTADOS DE CALCULOS DE PROBABILIDAD DE DETECCION.....	158
4.5 RESULTADOS DE VISUALIZACION RADAR.....	164
4.6 RESULTADOS DE ESTADISTICAS.....	165
5 CONCLUSIONES GENERALES.....	168
6 RECOMENDACIONES GENERALES.....	169
7 ANEXO A – GUIAS DE LABORATORIO.....	170
8 ANEXO B – CODIGOS FUENTES DEL SOFTWARE.....	204
9 ANEXO C – PASOS PARA INSTALAR SOFTWARE DE LA APLICACION	351

LISTA DE TABLAS

Tabla	Página
1.1 Algunos ejemplos de frecuencias utilizadas en aplicaciones radar	18
1.2 Bandas de frecuencia radar	19
1.3 Estadísticas del Aeropuerto de Londres Luton	24
1.4 Estadísticas del Aeropuerto Internacional El salvador del año 2010	24
1.5 Modos de interrogación	34
2.1 Organización de los Datos del Protocolo Asterix	49
2.2 Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM	57
2.3 Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM	57
2.4 Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM	58
2.5 Items de Datos estándar de categoría 001	59
2.6 UAP estándar para información de Plot	60
2.7 UAP estándar para información de Pista	61
2.8 Items de Datos estándar de categoría 002	79
2.9 Tabla de Aplicación de Usuario para Plots en Categoría 001	88
2.10 Tabla de Aplicación de Usuario para Marcación Norte en Categoría 002	89
2.11 Tabla de Aplicación de Usuario para Plots en Categoría 001 ALENIA	89
4.1 Resultados de las capturas realizadas con ADPADEC	155
4.2 Resultados de las capturas realizadas con ADPADEC	157
4.3 Resultados del análisis de probabilidad de detección de d12010.pcap	163
4.4 Resultados de las estadísticas de cada una de las capturas	166
4.5 Resultados de las estadísticas de cada una de las capturas	167

LISTA DE FIGURAS

Figura	Página
1.1 Diagrama básico para la medición de distancias en un sistema de radar	3
1.2 Medición de la Milla Náutica Radar.....	4
1.3 Determinación de acimut	5
1.4 Patrón de radiación de una antena en un gráfico de coordenadas polares	6
1.5 Radar de Vigilancia 3D, Indra LANZA	6
1.6 MSSR INDRA IRS-20MP/S	7
1.7 Radar Doppler utilizado por la policía.....	8
1.8 Radar Doppler utilizado en Meteorología	8
1.9 Imagen generada por un radar utilizado en un satélite	9
1.10 Imagen que genera un radar Meteorológico sobre un huracán	9
1.11 Trayectoria del haz y volumen escaneado por un Radar Meteorológico	10
1.12 Intensidad de lluvia registrada por Radar Meteorológico en El Salvador	11
1.13 Enlace de datos tierra-aire del Radar Secundario Modo S	11
1.14 Forma de onda del radar de onda continua	12
1.15 Diagrama de bloques de un radar de onda continua	13
1.16 Diagrama de bloques de un radar de onda pulsante	14
1.17 Características de propagación a diferentes frecuencias	19
1.18 Principio de funcionamiento de un radar primario básico	29
1.19 Patrón de radiación de una antena real de Radar Primario	30
1.20 PSR INDRA ASR12.....	30
1.21 Indicador de Posición de Plan de un Radar Primario	31
1.22 Periodo de los pulsos de un radar de onda pulsante	32
1.23 Transponder mecánico	33
1.24 Transponder digital	33
1.25 Patrones de radiación Suma (Σ) y Diferencia (Δ).....	35
1.26 Haz de control y de interrogación de la estación de tierra	36
1.27 Haces en el MSSR	36
1.28 Principio de funcionamiento de un Radar Secundario básico	37
1.29 Multiradar PSR + SSR INDRA.....	40
1.30 Principio de funcionamiento de un Radar Secundario Modo S básico	40
1.31 Compatibilidad entre los Modos A/C y el Modo S del SSR	41
1.32 Respuestas entrelazadas (garbling).....	43
2.1 Estructura del Protocolo Asterix.....	49
2.2 Estructura de un Bloque de Datos Asterix.....	52
2.3 Estructura de campos de datos de longitud fija	53
2.4 Estructura de campos de datos de longitud extendida	54
2.5 Estructura de campos de datos repetitivos.....	54
2.6 Estructura de campos de datos compuestos.....	54

2.7	Estructura de la especificación de campo (FSPEC)	55
2.8	Ejemplo de un octeto FSPEC	55
2.9	Ejemplo de un multi-octeto FSPEC.....	56
2.10	Bloque de Datos de categoría 001	58
2.11	Estructura de los dos octetos de longitud fija del Item de datos I001/010	62
2.12	Estructura de los dos octetos de longitud fija del Item de datos I001/020	62
2.13	Estructura del octeto de primera extensión del Item de datos I001/020.....	63
2.14	Estructura del octeto de primera parte del Item de datos I001/030	64
2.15	Estructura de los cuatro octetos del Item de datos I001/040	65
2.16	Estructura de los cuatro octetos del Item de datos I001/042	66
2.17	Estructura de los dos octetos del Item de datos I001/050.....	67
2.18	Estructura de los dos octetos del Item de datos I001/060.....	68
2.19	Estructura de los dos octetos del Item de datos I001/070.....	68
2.20	Estructura de los dos octetos del Item de datos I001/080.....	69
2.21	Estructura de los dos octetos del Item de datos I001/090.....	70
2.22	Estructura de los cuatro octetos del Item de datos I001/100	71
2.23	Estructura del octeto del Item de datos I001/120	72
2.24	Estructura del octeto de la primera parte del Item de datos I001/130	73
2.25	Estructura del octeto del Item de datos I001/131	73
2.26	Estructura de los dos octetos del Item de datos I001/141.....	74
2.27	Estructura del octeto del Item de datos I001/150	75
2.28	Estructura de los dos octetos del Item de datos I001/161.....	75
2.29	Estructura del octeto de la primera parte del Item de datos I001/170	76
2.30	Estructura del octeto de la primera extensión del Item de datos I001/170.....	77
2.31	Estructura de los cuatro octetos del Item de datos I001/200	78
2.32	Estructura del octeto del Item de datos I001/210	78
2.33	Bloque de Datos de categoría 002	79
2.34	Estructura del octeto del Item de datos I002/000	80
2.35	Estructura de los dos octetos del Item de datos I002/010.....	81
2.36	Estructura del octeto del Item de datos I002/020	81
2.37	Estructura de los tres octetos del Item de datos I002/030	82
2.38	Estructura de los dos octetos del Item de datos I002/041.....	82
2.39	Estructura del octeto del Item de datos I002/050	83
2.40	Estructura del octeto de la primera parte del Item de datos I002/060	84
2.41	Estructura de los octetos del Item de datos I002/070	85
2.42	Estructura del octeto del Item de datos I002/080	86
2.43	Estructura de los dos octetos del Item de datos I002/090.....	86
2.44	Estructura de los ocho octetos del Item de datos I002/100	87
2.45	Trama HDLC.....	90
2.46	Archivo Excel de la captura de paquetes con Wireshark	91
2.47	Caja de Dialogo del SETUP de WRTADS.....	93
2.48	Caja de Dialogo de Record de WRTADS	96
2.49	Ventana Principal de WRTADS.....	97
2.50	Ventana principal de RBAT	99
2.51	División en categorías de las estadísticas a nivel de sistema de RBAT	102

2.52	Grafica en 3D de la Probabilidad de Detección.....	104
2.53	Ventana de resultados del cálculo de la Probabilidad de Detección	104
2.54	Gráficos de barras de la Probabilidad de Detección vs altitud en pies.....	105
2.55	Caja de Dialogo de los parámetros de sitio de RBAT.....	105
2.56	Caja de Dialogo de los parámetros de usuario de RBAT	106
3.1	Flujograma en bloques del algoritmo del software ADPADEC.....	114
3.2	Flujograma en bloques del algoritmo de interface grafica de usuario.....	115
3.3	Flujograma en bloques del algoritmo para la captura de datos	116
3.4	Caso 1 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01.....	117
3.5	Caso 2 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01.....	118
3.6	Caso 3 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01.....	119
3.7	Flujograma en bloques del algoritmo de decodificación de plots Asterix categoría 01	120
3.8	Flujograma en bloques del algoritmo de decodificación de pistas Asterix categoría 01	122
3.9	Flujograma en bloques del algoritmo para la decodificación de mensaje de Norte.....	123
3.10	Flujograma en bloques del algoritmo utilizado para el análisis de visualización radar	125
3.11	Flujograma en bloques del algoritmo para calcular las estadísticas	126
3.12	Interface grafica de usuario del software ADPADEC.....	127
3.13	Ventana de captura de datos del software ADPADEC.....	129
3.14	Apartado de procesamiento de datos.....	129
3.15	Informe de la extracción y decodificación de datos en consola Linux.....	130
3.16	Ventana de resultados de la extracción y decodificación de datos.....	131
3.17	Apartado de procesamiento de datos.....	131
3.18	Informe del procesamiento para análisis en consola Linux	132
3.19	Ventana de resultados de probabilidad de detección.....	133
3.20	Apartado de datos a ingresar de la ventana de probabilidad de detección	134
3.21	Apartado de opción del área de gráficos y botones.....	135
3.22	Area de resultados de la ventana de probabilidad de detección	135
3.23	Area de resultados con información sobre seguimiento de un blanco.....	136
3.24	Grafico de barras de la probabilidad de detección de un blanco radar.....	137
3.25	Gráficos de barras de la probabilidad de detección vs rango	138
3.26	Gráficos de barras de la probabilidad de detección vs altitud.....	138
3.27	Area de Indicador de Posición de Plan.....	139
3.28	Apartado de opciones de visualización radar	140
3.29	PPI con mapa descriptivo	140
3.30	PPI con mapa de satélite.....	141
3.31	Ventana de ingreso de datos para visualización radar	142
3.32	Plots radar en PPI.....	143

3.33	Simbología utilizada en el PPI para un plot radar	143
3.34	Simbología utilizada en el PPI para un plot radar de baja confianza	144
3.35	Simbología para un plot radar con código especial de emergencia	145
3.36	Simbología para una pista radar en PPI.....	145
3.37	Ventana de estadísticas del software ADPADEC	147
3.38	Ventana de errores del software ADPADEC.....	148
3.39	Ejecutando el terminal de Linux Ubuntu 10.0.....	149
3.40	Instrucciones en consola Linux para ejecutar ADPADEC.....	149
3.41	Interface grafica de usuario del software ADPADEC.....	150
4.1	Fotografía de cómo es la captura de tramas Asterix con PC	154
4.2	Fotografía con mayor detalle de la captura de tramas Asterix	154
4.3	Informe de la extracción y decodificación de datos en consola Linux.....	156
4.4	Ventana de resultados de la extracción y decodificación de datos.....	156
4.5	Informe del procesamiento para análisis en consola Linux	158
4.6	Resultados de probabilidad de detección para el archivo d12010.txt.....	159
4.7	Códigos validos que se encuentran en el archivo d12010.txt.....	160
4.8	Resultados de probabilidad de detección para el código transponder 2346	160
4.9	Análisis de seguimiento de la aeronave con código transponder 2346	161
4.10	Resultados de Probabilidad de Detección vs Rango de d12010.txt.....	162
4.11	Resultados de Probabilidad de Detección vs Altitud de d12010.txt.....	162
4.12	Resultados de probabilidad de detección de falsos blancos de d12010.txt	163
4.13	Visualización radar de d12010.txt a 100 NM.....	164
4.14	Resultados obtenidos de la visualización radar de d12011.txt.....	165
4.15	Resultados obtenidos del análisis de estadísticas de d12010.txt	166

GLOSARIO

ACIMUT	Angulo medido con respecto al norte magnético, que indica la posición de un blanco, de forma similar a como lo hace una brújula
ADPADEC	Analizador de Protocolo Asterix de Eurocontrol
AIES	Aeropuerto Internacional El Salvador
ASTERIX	All Purpose Structure Eurocontrol Surveillance Information Exchange Exchange. Estructura Todo Propósito para el Intercambio de Información de Vigilancia de Eurocontrol
ATC	Control de Transito Aéreo
ATCRBS	Air Traffic Control Radar Beacon System. Sistema de aviso de radar para control de trafico aéreo. Nombre alternativo para SSR usado en los USA
ATM	Air Traffic Management. Sistema de gestión de tráfico aéreo
CAT	Categoría de Datos
EUROCONTROL	European Organization For The Safety Of Air Navigation Organización Europea para la Seguridad de Navegación Aerea
FAA	Federal Aviation Administration, USA. Administración de aviación federal de los Estados Unidos
FIR	Flight Information Region. Región de Información de vuelo
FRN	Field Reference Number. Número de Referencia de Campo
FRUIT	Respuestas falsas desincronizadas en el tiempo
FSPEC	Field Specification. Especificación de Campo
FX	Field extensión indicator. Indicador de extensión de campo
GARBLING	Solape de respuestas de SSR, correspondientes a la misma interrogación
GPS	Global Position System. Sistema de Posicionamiento global

HDLC	High-Level Data Link Control. Control de Enlace de Datos de Alto Nivel
LAN	Local Area Network. Red de área local
LEN	Length Indicator. Indicador de Longitud
LSB	Last Significant Bit. Último Bit Significativo
MSSR	Monopulse Secondary Surveillance Radar. Radar de Vigilancia Secundario Monopulso
OACI	Organización Internacional de la Aviación Civil
PISTA	Historial de Plots de una aeronave sobre un PPI
PLOT	Ecos radar sobre un PPI
PPI	Indicador de Posición de Plan
PSR	Primary Surveillance Radar. Radar Primario de Vigilancia
RANGO	Distancia que tiene el objeto a la antena radar
RBAT	Software que posee una gran variedad de análisis para los sistemas radar
REP	Field Repetition Indicator. Indicador de Repetición de Campo
SAC	System Area Code. Sistema de Código de Area
SIC	System Identification Code. Sistema de Código de Identificación
SP	Special Purpose Indicator. Indicador de Propósito Especial
SPI	Special Position identification. Identificación de Posición Especial
SSR	Secondary Surveillance Radar. Radar Secundario de Vigilancia
STFRDE	Surveillance Task Force for Radar Data Exchange de Eurocontrol
TCAS II	Sistema Anticolisión que utiliza interrogaciones SSR para proporcionar a la tripulación información que contiene un aviso de alerta de tránsito como una ayuda para el principio de “ver y eludir” versión II

TRANSPONDER	Equipo transmisor-respondedor que esta instalado en la aeronave, para emitir respuestas a interrogaciones hechas por un SSR. Equipo en el cual el piloto introduce un codigo octal, que fue asignado por el ATC, para identificación de la aeronave.
UAP	User Application Profile. Perfil de Aplicación de Usuario
UIT	Union Internacional de las Telecomunicaciones
UTC	Coordinated Universal Time. Hora Universal Coordinada
WRTADS	Software que se utiliza para la captura de datos radar en el formato Asterix, para luego hacer una evaluación grafica rápida sobre los blancos radar

CAPITULO I

CONCEPTOS DE RADAR

Introducción

El término Radar proviene del acrónimo **R**adio **D**etection and **R**anging (detección y medición de distancia por ondas de radio), es una tecnología desarrollada para detectar diferentes tipos de blancos como aeronaves y fenómenos climatológicos (tormentas, huracanes y lluvia). Las bases del radar se establecieron a finales del siglo XIX, El concepto fue inicialmente planteado por el físico alemán Heinrich Hertz en 1886 cuando a través de diversos experimentos demostró que las ondas electromagnéticas tienen las mismas propiedades que las ondas luminosas y por tanto se pueden reflejar; desde entonces, numerosos estudios se han desarrollado entre los cuales destacan:

1904 el ingeniero alemán Christian Hülsmeyer registró patentes alemanas y en el Reino Unido sobre un aparato de su invención que denominó el Telemoviloscopio un aparato que podía recibir el eco radioeléctrico reflejado, a pesar de que se le concedió las patentes la industria no mostro demasiado interés en él y el Telemoviloscopio fracaso como aplicación comercial.

1922 los Ingenieros eléctricos Albert H. Taylor y Leo C. Young del laboratorio de investigación de la Armada Americana (Estados Unidos) detectaron un barco de madera por primera vez, usando señales de radio.

1935 el mayor impulsor del radar el físico Robert Watson-Watt finalmente obtiene la patente del primer Sistema Radar en Inglaterra (patente británica GB593017).

1940 debido a la Segunda Guerra Mundial, se desarrollaron diferentes sistemas radar en los Estados Unidos, Rusia, Alemania, Francia y Japón.

Desde entonces, mucho se ha avanzado en este campo, aunque los conceptos básicos siguen siendo los mismos: la detección de un blanco de interés, por medio de ondas de radio. Después de la Segunda Guerra Mundial (1939 – 1945) y con el desarrollo de la aviación civil, los sistemas radar adquirieron una enorme importancia para el control de tránsito aéreo. En el presente capitulo se trataran los conceptos básicos que describen el funcionamiento de un sistema radar y como los mismos han evolucionado hasta convertirse en una herramienta importante para la seguridad del transporte aéreo civil. También se abordan los temas de: concepto de integración de información o sistemas abiertos y los parámetros para establecer la calidad de un sistema radar.

1.1 PRINCIPIOS BASICOS DE OPERACION DE UN RADAR

El radar es un sistema que usa ondas electromagnéticas, para detectar objetos. Para ello, el radar emite ondas de radio las cuales viajan por el espacio, chocan contra el objeto de interés y ese choque provoca que parte de la energía incidente sea reflejada de vuelta al punto de emisión, a dicho reflejo se le llama eco. A partir del análisis del eco, se puede extraer una serie de información para caracterizar el objeto o blanco de interés, por ejemplo:

- Rango o Distancia
- Acimut o Angulo

Dependiendo del tipo de radar, también se puede obtener información de:

- Altitud
- Velocidad y dirección de desplazamiento del blanco
- Imágenes del objeto
- Volumen de lluvia o gases

Incluso (para el caso de radares aeronáuticos Modo S) se puede obtener un canal de comunicación bidireccional que permite caracterizar de mejor manera al blanco.

1.1.1 Determinación de Rango o Distancia del objeto

El Rango, llamado así debido a que la palabra viene del ingles Range, que se traduce en español como Rango o Distancia, es la distancia que tiene el objeto a la antena radar.

La **Figura 1.1** nos muestra el diagrama básico para la medición de distancias en un sistema de radar; el transmisor emite ondas de radio que son representadas por líneas curvas de color rojo, así la dirección de estas ondas las indica la flecha de color rojo, que en este caso es la señal original, las ondas llegan y chocan con el objeto (que en la figura es de color azul) y provoca que parte de la energía incidente se refleje de vuelta al punto de emisión, estas ondas son representadas por líneas curvas no continuas de color verde, la flecha de color verde indica la dirección en la cual son reflejadas las ondas, que es el eco de la señal original el que llega al receptor. La función básica para medir la distancia del objeto a la antena radar se realiza por medio de la aplicación de la ecuación que relaciona velocidad, tiempo y distancia, es decir:

$$v = \frac{e}{t}$$

En la ecuación anterior, se conoce el valor de la variable v el cual es 300000 km/seg, que es la velocidad en el aire de las ondas electromagnéticas. Luego, si pudiéramos medir el tiempo que le toma a la señal, ir y venir del blanco, entonces podríamos conocer la distancia a la cual se encuentra.

Así para calcular la distancia, se despeja e y este resultado se divide entre dos, ya que el tiempo medido es, el tiempo en que la señal sale del transmisor, choca con el objeto y vuelve al punto de emisión, esta fórmula se muestra a continuación:

$$e = \frac{vt}{2}$$

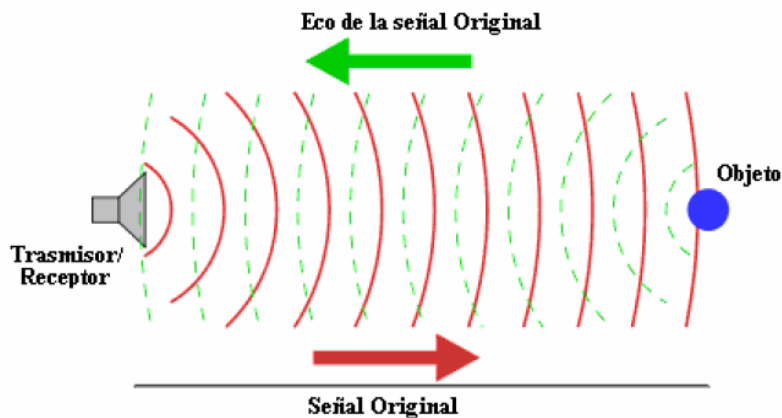


Figura 1.1: Diagrama básico para la medición de distancias en un sistema de radar
 Fuente: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/soriano_m_jc/apendiceJ.pdf

Por ejemplo si se quiere medir, la distancia a la cual está un objeto, conociendo el tiempo de retorno del eco el cual es de $t = 333.33 \mu s$, se calcula que la distancia a la cual está el objeto de la antena radar es de:

$$e = \frac{vt}{2} = \frac{300000 \times 333.33 \times 10^{-6}}{2} = 50 \text{ km}$$

Como se ha visto anteriormente para determinar el rango se usa la formula de la velocidad, así la fórmula para calcular el rango es la siguiente:

$$R = \frac{C_0 t}{2}$$

Donde:

C_0 = velocidad de la luz es la velocidad a la que las ondas electromagnéticas se propagan = 3×10^8 m/s

t = tiempo medido en segundos

R = rango o distancia medido en metros

En el campo aeronáutico surge el concepto de milla náutica radar, que es el tiempo que tardan las ondas de radio en recorrer (en un trayecto de ida y vuelta) una milla náutica, para esto se sabe que 1 milla náutica equivale a 1.853 km y usando la ecuación básica de la velocidad tenemos:

$$v = \frac{e}{t}$$

$$t = \frac{e}{v}$$

$$t = \frac{1.853 \times 10^3}{3 \times 10^8} = 6.177 \mu s$$

La **Figura 1.2** muestra como es el cálculo de la milla náutica radar, el tiempo que la señal de radio sale de la antena radar y llega a la aeronave se le llama t_1 , la señal que llega a la aeronave es la que indica la línea roja y la señal de eco que llega a la antena radar se indica por la línea verde. Así el tiempo que tarda la señal en recorrer una milla náutica y regresar a la antena del radar es dos veces t_1 con lo cual nos queda la siguiente ecuación:

$$t_{total} = 2t_1$$

De la ecuación de la velocidad sabemos que el tiempo que tarda en ir la señal de radio de la antena del radar a la aeronave es de $6.177 \mu s$, que es el valor que tiene t_1 por lo que el valor del tiempo total en que una señal de radio del radar va y regresa en forma de eco es:

$$t_{total} = 2t_1 = 2 \times 6.177 = 12.35 \mu s$$

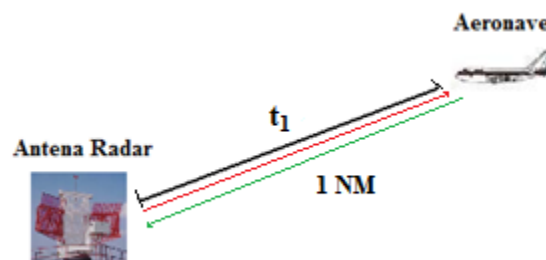


Figura 1.2: Medición de la Milla Náutica Radar

El grado de resolución en rango o distancia, depende del ancho del pulso transmitido, del tipo y tamaño de los blancos, de la eficiencia del receptor y de la pantalla.

1.1.2 Determinación del Acimut o Angulo

Los sistemas radar proporcionan información de la posición de un blanco con respecto al norte magnético, de forma similar a como lo hace una brújula. Dicha información se conoce como acimut o posición acimutal del blanco.

Esta medición se logra apreciar en la **Figura 1.3**. La referencia del norte puede obtenerse de diferentes maneras, antiguamente se utilizaban brújulas para esta medición, pero con el tiempo esto ha ido cambiando, en la actualidad la más utilizada es utilizando las redes satelitales de posicionamiento global o mejor conocido por sus siglas en inglés como GPS (Global Positioning System).

La posición acimutal del blanco debe darse de forma muy precisa, para lo cual se utilizan antenas con patrón de radiación muy directivo, esto significa que la antena concentra la energía radiada en un lóbulo principal, de ancho no mayor a 3° . Esta capacidad de concentración de energía de una antena está determinada por la ganancia de la misma.

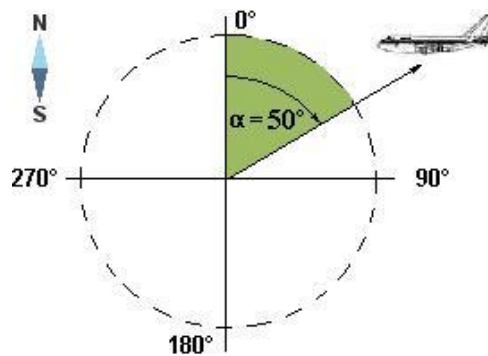


Figura 1.3: Determinación de acimut

La forma del patrón de radiación de una antena depende de la construcción de la misma. Las antenas de los sistemas radar están diseñadas para radiar energía en un solo lóbulo o haz unidireccional que puede ser movido simplemente moviendo la antena.

El ancho del haz de una antena (beam width) es solo la separación angular entre los dos puntos de media potencia (-3 dB) en el lóbulo principal del patrón de radiación del plano de la antena, por lo general tomado de uno de los planos "principales" como puede observarse en la **Figura 1.4**. Los lóbulos laterales son más pequeños que el lóbulo principal. Estos lóbulos laterales son por lo general la radiación en direcciones no deseadas que nunca pueden ser totalmente eliminados.

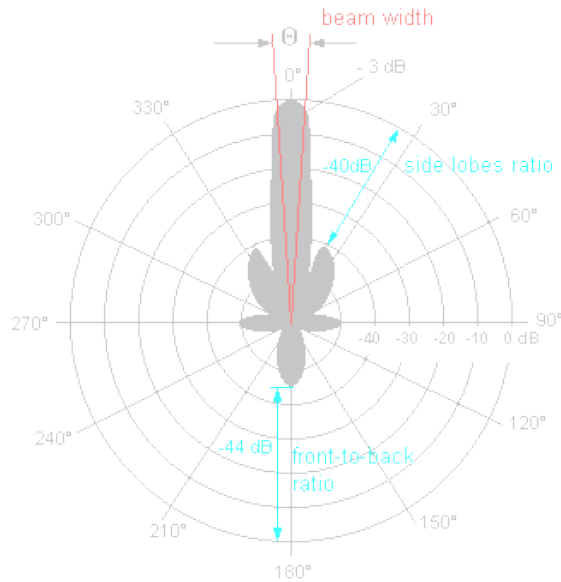


Figura 1.4: Patrón de radiación de una antena en un gráfico de coordenadas polares
Fuente: <http://www.radartutorial.eu/06.antennas/an10.de.html>

El acimut y el ángulo de elevación del radar con relación al blanco, pueden ser determinados midiendo la dirección en la que la antena apunta cuando se recibe el eco, la precisión de la medición de estos ángulos depende de la directividad de la antena. Para una frecuencia de transmisión dada (o longitud de onda definida), la directividad de la antena depende de sus propias dimensiones.

1.1.3 Altitud del blanco

Un radar que proporciona este dato, es el radar de vigilancia 3D, el cual tiene aplicaciones militares; Dicho radar es fabricado por la empresa INDRA Sistemas de España al cual le han llamado LANZA que se muestra en la **Figura 1.5**.



Figura 1.5: Radar de Vigilancia 3D, Indra LANZA

Otro radar que es capaz de proporcionar el dato de altitud obtenido de forma indirecta es el Radar Secundario. En este sistema, la aeronave envía al radar su dato de altitud y luego la incorpora con el resto de la información. La **Figura 1.6** muestra un Radar de Vigilancia Secundario Monopulso, por sus siglas en ingles MSSR, fabricado por la empresa INDRA Sistemas de España modelo IRS-20MP/S.



Figura 1.6: MSSR INDRA IRS-20MP/S

1.1.4 Velocidad y dirección de desplazamiento del blanco

Un radar que mide el dato de velocidad y dirección de desplazamiento del blanco es el denominado Radar Doppler, mediante el efecto Doppler, que es la diferencia aparente entre la frecuencia con que es emitido un sonido o una onda electromagnética por una fuente en movimiento y la frecuencia que percibe un observador, causada por la velocidad relativa entre la fuente y el observador. Este radar puede ser pulsante o de onda continua. Algunas aplicaciones en las cuales se utiliza este tipo de radar se observan en las **Figuras 1.7 y 1.8**. En La **Figura 1.9** se observa un radar Doppler utilizado por la policía para la medición de la velocidad de los automóviles. En la **Figura 1.10** se observa la imagen que proporciona un Radar Doppler utilizado en Meteorología.



Figura 1.7: Radar Doppler utilizado por la policía

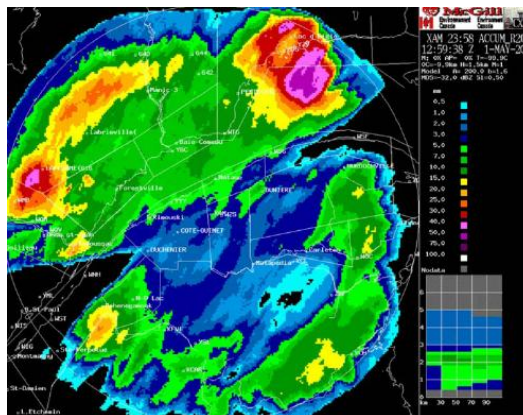


Figura 1.8: Radar Doppler utilizado en Meteorología

1.1.5 Imágenes del objeto

Utilizados por sistemas satelitales, es decir el satélite emite un haz de energía el cual sale del radar, choca contra la tierra y la porción de esta energía que es reflejada es la que se captura. Un ejemplo de las imágenes que este tipo de radar genera lo podemos observar en la **Figura 1.9**. También están los radares meteorológicos los cuales se utilizan para localizar lluvia, nieve, granizo, etc (precipitaciones, calcular sus trayectorias y estimar sus tipos). En la **Figura 1.10** se puede apreciar la imagen que genera un radar Meteorológico, que en este caso nos indica la presencia de un huracán.



Figura 1.9: Imagen generada por un radar utilizado en un satélite

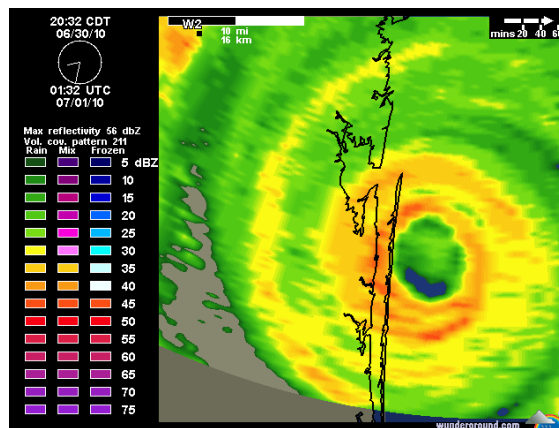


Figura 1.10: Imagen que genera un radar Meteorológico sobre un huracán

1.1.6 Volumen de lluvia o gases

Otro tipo de radares son los radares Meteorológicos, que son radares de tipo pulsante. Con un radar típico de pulso y su ancho de haz, el volumen escaneado varía grandemente, hasta 250 ó 300 km. Por ejemplo, el retorno de una distancia dada es el promedio de los ecos en un volumen del orden de 1 km³ de aire. El volumen de aire para un pulso, en un punto, en un tiempo puede calcularse aproximadamente por la fórmula:

$$v = hr^2\theta^2$$

Donde:

v = volumen encerrado por el pulso

h = ancho del pulso (medido en metros, calculados de la duración en segundos del pulso a la velocidad de la luz)

r = distancia desde el radar al pulso que está propagándose (medido en metros)

θ = ancho del rayo (en radianes)

La fórmula del volumen encerrado por el pulso asume que el haz es simétricamente circular, " r " es mucho mayor que " h " sobre " r " tomado al inicio o al final del pulso es prácticamente lo mismo, y la forma del volumen es un cono tronco de profundidad " h ". La **Figura 1.11** muestra la trayectoria del haz y volumen escaneado por un Radar Meteorológico

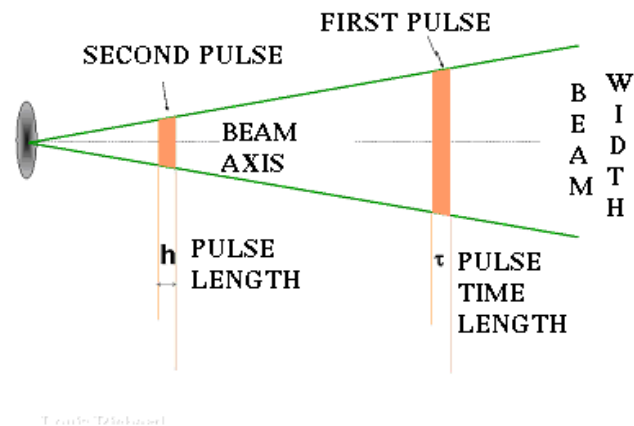


Figura 1.11: Trayectoria del haz y volumen escaneado por un Radar Meteorológico
Fuente: <http://es.wikipedia.org/wiki/Archivo:Radar-beam.png>

Los retornos que proporciona un Radar Meteorológico se representan usualmente en la pantalla del radar por un color o nivel. Los colores en una imagen de radar normalmente van del azul o verde para los retornos débiles, al rojo o magenta para los muy fuertes. Los números en un reporte oral se incrementan con la severidad de los retornos.

Cuando se describen los retornos del Radar Meteorológico, los pilotos, despachantes y controladores del tráfico aéreo se refieren a tres niveles de retorno:

- **Nivel 1:** corresponde al retorno verde, indicando usualmente ligera precipitación y poca o nada de turbulencia y con la posibilidad de visibilidad reducida.
- **Nivel 2:** corresponde al retorno amarillo, indica moderada precipitación, con la posibilidad de muy baja visibilidad, moderada turbulencia e incomfortabilidad para los pasajeros.
- **Nivel 3:** corresponde al retorno rojo, indica precipitación pesada, con la posibilidad de tormentas y severa turbulencia, con daños serios estructurales a la aeronave.

Las aeronaves deben tratar de evitar el nivel 2 cuando sea posible y obviar el nivel 3 a menos de que se tenga una aeronave especialmente diseñada. La **Figura 1.12** muestra la intensidad de lluvia registrada por un Radar Meteorológico en El Salvador.

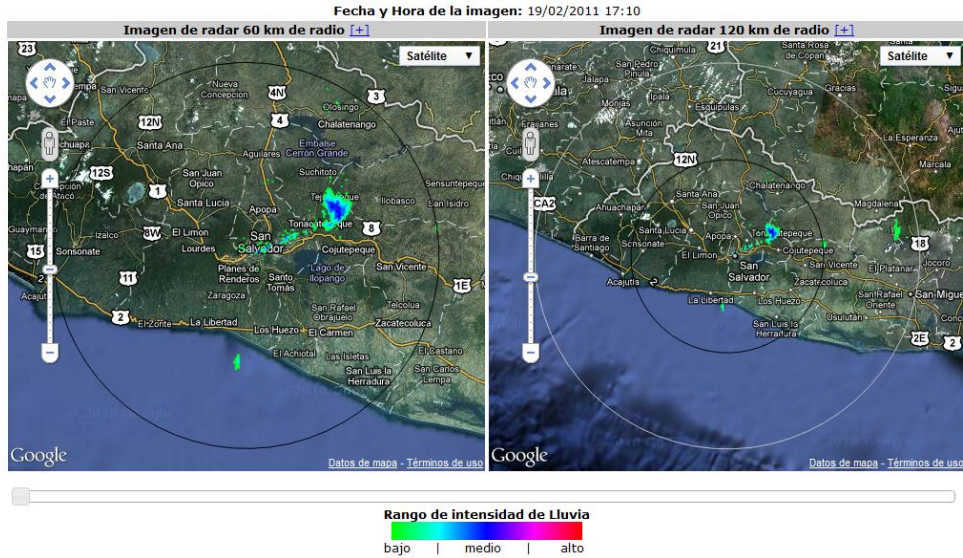


Figura 1.12 Intensidad de lluvia registrada por Radar Meteorológico en El Salvador
Fuente: <http://www.snet.gob.sv/googlemaps/radares/radar.php>

1.1.7 Intercambio de información

Entre este tipo de radares se encuentran los Radares Secundarios Modo S, que permiten el intercambio de información entre el piloto y el controlador de tráfico aéreo, ya que se establece un enlace de datos bidireccional tierra-aire. Estos servicios se pueden dividir en: Servicios ATC¹ y Servicios de asesoramiento de vuelo. La **Figura 1.13** muestra un esquema de cómo es este enlace de datos.

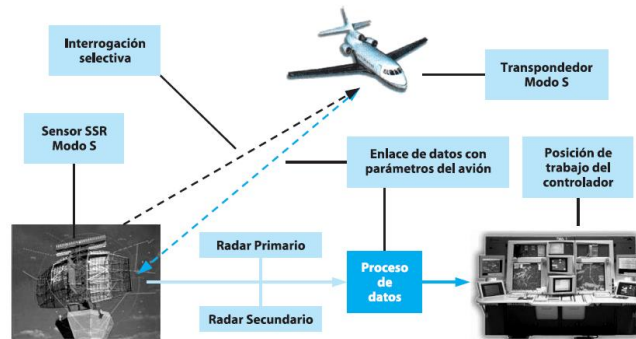


Figura 1.13: Enlace de datos tierra-aire del Radar Secundario Modo S
Fuente: <http://issuu.com/jontiveros/docs/del-radar-primario-al-modo-s--y-mas-alla>

¹ Abreviatura en inglés que significa Centro de Control de Tráfico Aéreo

1.1.8 Evolución de la altitud

Entre este tipo de radares se encuentran los Radares Secundarios Modo S, que para medir la evolución de la altitud de la aeronave, el altímetro es capaz de medir los cambios de altitud con mucha más exactitud, que el dato de altitud proporcionado al Radar Secundario. Debido a que se rebaja de 100 pies a 25 pies.

1.2 CLASIFICACION DE RADARES

1.2.1 Clasificación de radares por tipo de transmisión

1.2.1.1 Radar de onda continua

Es el que transmite ininterrumpidamente la señal de Radiofrecuencia (RF), utiliza transmisión continua en lugar de pulsante, La **Figura 1.14** muestra la forma de onda del radar de onda continua. En un radar de onda continua se transmite y recibe información de forma simultánea por lo que son necesarias dos antenas, para evitar interferencias, es necesario separar la ruta de transmisión de la ruta de recepción, esto muchas veces implica la necesidad de tener antenas separadas.

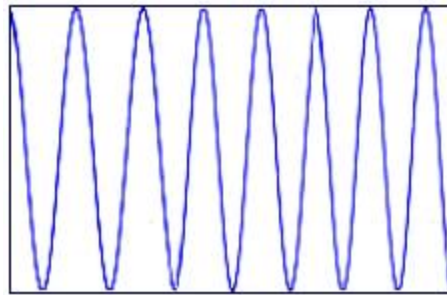


Figura 1.14: Forma de onda del radar de onda continua

Fuente: <http://personales.unican.es/perezvr/pdf/Introduccion%20al%20Radar.pdf>

Los radares de onda continua pueden trabajar con tecnología Doppler o de Modulación en Frecuencia (FM). El radar Doppler se utiliza para realizar medidas precisas de la velocidad de un objeto. Este tipo de radar transmite una onda continua de frecuencia fija, cuando esta señal encuentra un objeto en movimiento la frecuencia de la onda reflejada cambia con respecto a la transmitida que se toma de referencia, utilizando esta variación de frecuencia el radar determina la velocidad del objeto. Los radares FM también emiten señales continuas pero en este caso moduladas en frecuencia, a diferencia del radar Doppler, estas variaciones en frecuencia de la señal transmitida permiten no sólo conocer la velocidad del objetivo sino también su posición. En La **Figura 1.15** se observa el diagrama de bloques básico, de un radar de onda continua.

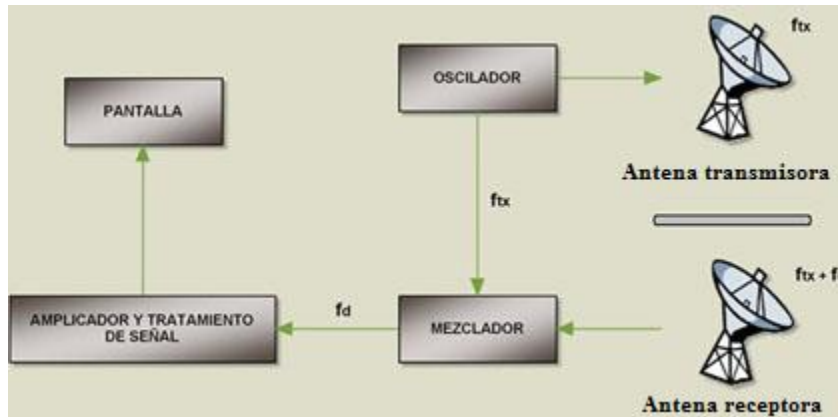


Figura 1.15: Diagrama de bloques de un radar de onda continua
Fuente: <http://www.tecnoradar.es/partesradar.html>

Un radar de onda continua posee los siguientes bloques o componentes:

Antena transmisora: Es la que transmite la onda continua (f_{tx}), que ha generado el oscilador la cual suele ser senoidal de frecuencia y amplitud constante. Esta antena suele ser de tipo Yagi.

Oscilador: Es el que genera la señal de salida hacia la antena, con amplitud constante y senoidal en frecuencia.

Mezclador: Forma la primera etapa de recepción, obtiene información sobre la amplitud y la variación de frecuencia del eco recibido con respecto a la señal enviada (f_d). La salida del mezclador tendrá una frecuencia muy inferior a la radiada con lo que su procesamiento será mucho más sencillo y de menor costo económico.

Antena receptora: Es la que recoge la señal reflejada en el objetivo ($f_{tx} + f_d$). Esta antena suele ser de tipo Yagi.

Amplificador y tratamiento de señal: En esta etapa el radar amplifica la señal y analiza su contenido. En un radar moderno esta operación se realiza utilizando una computadora.

Pantalla: Es el monitor donde se despliega la información recogida de las señales radar, para ser analizada y entendida.

1.2.1.2 Radar de onda pulsante

Es aquel donde se transmite periódicamente un pulso, que puede estar modulado o no. A diferencia de los anteriores, en estos la señal de radiofrecuencia se transmite de forma pulsante, esto permite utilizar la misma antena para transmitir y para recibir.

Los pulsos transmitidos, son pulsos de RF que se emiten por corto tiempo, por ejemplo 1 μ s, cuando se transmite un pulso, la recepción se inhibe, la recepción se habilita en el tiempo que separa dos pulsos de transmisión sucesivos ya que se espera un pulso de respuesta por parte del blanco.

En la **Figura 1.16** se muestra el diagrama de bloques de un radar de onda pulsante, donde se pueden apreciar cada uno de sus bloques o componentes.

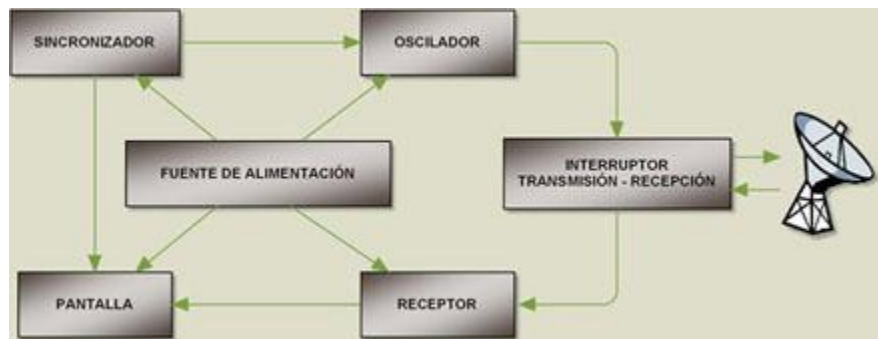


Figura 1.16: Diagrama de bloques de un radar de onda pulsante
Fuente: <http://www.tecnoradar.es/partesradar.html>

Los componentes básicos de un radar de pulsos son:

Antena: Es la que recibe o transmite los pulsos de RF. El tipo de esta antena puede ser Yagi o un arreglo de dipolos.

Sincronizador: Es el bloque que se encarga de generar la señal que marca la transmisión del pulso radar. Para inicializar su reloj interno el receptor utiliza también esta señal en espera del eco devuelto por el obstáculo.

Oscilador: Es el que produce la señal de radiofrecuencia que se transmite a través de la antena. Las frecuencias utilizadas van desde los MHz hasta los 30-40GHz. El oscilador también debe incluir un amplificador para obtener la potencia necesaria de 2-3 kW.

Receptor: Es el que está sintonizado a la misma frecuencia de transmisión y se encarga de amplificar los ecos devueltos y transformarlos en información capaz de ser representada en la pantalla radar.

Interruptor Transmisión-Recepción: Este bloque conecta la antena al receptor o al transmisor en función de la señal de sincronización. La mayoría de los radares de pulsos utilizan la misma antena para enviar y recibir la señal.

Fuente de alimentación: Proporciona la energía eléctrica que necesitan todos los componentes dentro del radar.

Pantalla: Es donde se presentan los resultados obtenidos.

1.2.2 Clasificación de radares según el blanco

1.2.2.1 Radar Primario

Es el que detecta al blanco mediante el reflejo de la onda electromagnética de radio que emite la antena del radar.

Los primeros radares fueron diseñados para que trabajaran de esta manera. Una ventaja de un Radar Primario, es que permite identificar a las aeronaves sin necesidad de que estas estén equipadas, con un equipo especial a bordo.

Una desventaja que posee un Radar Primario es que proporciona información limitada del blanco.

1.2.2.2 Radar Secundario

Es el que detecta al blanco mediante una respuesta que le envía la aeronave mediante un equipo especial que está instalado a bordo de la aeronave a una interrogación que ha hecho el radar.

El equipo en tierra está formado por una antena, que es un arreglo de dipolos, un transmisor, un receptor, un codificador y un decodificador. La señal que transmite el equipo en tierra se le conoce como interrogación.

El equipo a bordo de las aeronaves se le conoce como transmisor respondedor o transponder, cuando el transponder detecta o recibe la señal de interrogación, la procesa y si es válida le “contesta” o “responde” transmitiendo un código octal de cuatro números programables manualmente por los pilotos. Dicho código representa una identificación única de la aeronave y normalmente es asignado por el personal que administra el tránsito aéreo en la zona en la que se desplaza la aeronave.

Entre las ventajas de un Radar Secundario podemos mencionar:

- Permite la identificación de las aeronaves al asignarles un código octal de cuatro números.
- Las claves de cuatro números octales pueden utilizarse para indicar situaciones especiales, como fallo de radiocomunicaciones, interferencia ilícita y emergencia.

Entre las desventajas de un Radar Secundario tenemos:

- Al identificar a una aeronave con un código octal de cuatro números, no resulta práctico utilizarlo en aeropuertos donde hay más de 4096 aeronaves en un determinado momento.

- Las aeronaves deben de estar equipadas de un equipo especial, que permita su identificación.
- Se requiere un número determinado de respuestas para poder identificar con certeza a las aeronaves.
- Presenta el problema de que dos respuestas puedan estar solapadas (garbling), es decir que en la pantalla radar observaríamos como que dos aeronaves son una sola.

1.2.3 Clasificación de radares por su aplicación

1.2.3.1 Radares Civiles

1.2.3.1.1 Radares Aeronáuticos

Se llamara así a los radares empleados en aviación civil. Entre estos tenemos:

- **Radar Primario**
- **Radar Secundario**
- **Radar Secundario Modo S:** Es el radar que interroga al blanco de manera selectiva, pero que también necesita que el blanco este equipado con un equipo especial, el blanco responde, con una serie de datos (código transponder, altura de la aeronave, velocidad de la aeronave, etc).

Entre las ventajas de un radar Secundario Modo S tenemos:

- La Asignación de un código octal de 8 números octales, que permite una identificación única, para una determinada aeronave, ya que el número de códigos disponibles es un poco mayor a los 16 millones.
- Puede interrogar a las aeronaves de una manera selectiva, es decir que si existen muchas aeronaves en vuelo, el controlador de tránsito aéreo decide a que aeronave va a interrogar.

Las desventajas que posee un Radar Secundario Modo S son:

- Maneja mucha más información que la que se usa actualmente.
- Las aeronaves deben estar equipadas con un equipo especial (transponder Modo S), para poder ser identificadas por el Radar Secundario Modo S.

1.2.3.1.2 Radares Meteorológicos

El funcionamiento de un Radar Meteorológico es similar al de un faro, ya que la antena del radar gira continuamente para barrer, o iluminar, los blancos que se desea observar en un cierto volumen de la atmósfera. Después de un sofisticado procesamiento de la señal, el radar proporciona información valiosa sobre las características de estos ecos que se pueden relacionar con lluvia, de nieve o dirección de viento.

1.2.3.2 Radares Militares

Son por ejemplo radares de detección terrestre, radares de misiles autodirectivos, radares de artillería, radares de satélites para la observación de la Tierra.

1.2.4 Clasificación de radares por su cobertura y ubicación

1.2.4.1 Area terminal

Son los radares para áreas terminales con una cobertura de 5 millas náuticas, instalados sobre torres de control, con una rotación constante de antena de 40 y 60 revoluciones por minuto, se coloca en aeropuertos con mal tiempo o muchas pistas y son del tipo Primario.

1.2.4.2 Aproximación

Son radares con una cobertura entre 30 y 80 millas náuticas para el Radar Primario y 250 millas náuticas para el Radar Secundario instalados en cualquier punto del aeropuerto entre 0 y 10 millas náuticas, con una rotación constante de antena entre 12 y 15 revoluciones por minuto.

1.2.4.3 Control de ruta

Son los radares con una cobertura de 250 millas náuticas para el Radar Secundario y más de 80 millas náuticas para el Radar Primario, se recomienda trabajar solo con el Radar Secundario. Se colocan a mas de 30 millas náuticas y posee una rotación constante de antena entre 6 y 10 revoluciones por minuto.

1.2.5 Clasificación de radares por el rango de frecuencias en el cual operan

- **Banda L:** Rango de frecuencias de 1 a 2 Ghz, entre estos radares estan: radares civiles entre ellos radares de vigilancia aérea de largo alcance por encima de 250 NM, como el Radar Secundario de Vigilancia Monopulso, también se incluyen el Radar Secundario de Vigilancia convencional, radares Meteorológicos y el Radar de Vigilancia Primario.
- **Banda S:** Rango de frecuencias de 2 a 4 Ghz, entre estos se encuentran los radares civiles como el Radar de Vigilancia Primario (de 3.1 a 3.3 o de 2.7 a 2.9 Ghz) , el Radar Secundario de Vigilancia Monopulso y algunos radares Meteorológicos. Tambien estan el Radar de Energía Media, con una potencia de impulso de hasta 20 MW, radares Especiales de Vigilancia en Aeropuertos, con un alcance medio de 50 a 60 NM, que también se utiliza para uso militar.
- **Banda C:** Rango de frecuencias de 4 a 8 Ghz, está predeterminada para la mayoría de radares meteorológicos usados para localizar precipitaciones en zonas templadas como Europa. También se incluyen radares militares, entre estos sistemas móviles

de vigilancia, control de misiles y conjuntos de radares de vigilancia de corto o mediano alcance.

- **Banda X:** Rango de frecuencias de 8 a 12 GHz. Entre estos están aplicaciones militares como radares aerotransportados, radares civiles y militares para la navegación marítima. Se incluyen también radares de imágenes espaciales o aéreas usando como base el radar de apertura sintética, por sus siglas en inglés SAR (Synthetic Aperture Radar).
- **Banda K:** Rango de frecuencias de 12 a 24 GHz. Entre estos están radares de Movimiento en la Superficie, por sus siglas en inglés SMR (Surface Movement Radar) o Equipo de detección en la superficie del aeropuerto, por sus siglas en inglés ASDE (Airport Surface Detection Equipment).

La **Tabla 1.1** muestra algunos ejemplos de frecuencias utilizadas en aplicaciones radar.

HF	3 – 30 MHz		
VHF	30 MHz–300 MHz	↑	Exploración y búsqueda
UHF	300 MHz–1 GHz		
L-Band	1 GHz–2 GHz	↕	Búsqueda y seguimiento
S-Band	2 GHz–4 GHz		
C-Band	4 GHz–8 GHz	↑	Control de fuego y obtención de imágenes
X-Band	8 GHz–12 GHz		
Ku-Band	12 GHz–18 GHz	↑	Búsqueda de misiles
K-Band	18 GHz–27 GHz		
Ka-Band	27 GHz–40 GHz		
W-Band	40 GHz – 100+ GHz	↓	

Tabla 1.1: Algunos ejemplos de frecuencias utilizadas en aplicaciones radar
Fuente: <http://personales.unican.es/perezvr/pdf/Introduccion%20al%20Radar.pdf>

En la **Figura 1.17** se observa una imagen que muestra las características de propagación a diferentes frecuencias radar.

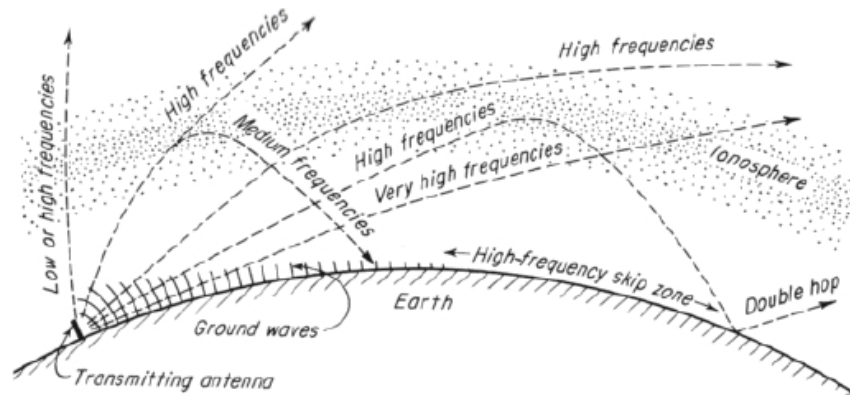


Figura 1.17: Características de propagación a diferentes frecuencias

Fuente: <http://agamenon.tsc.uah.es/Asignaturas/ittst/rdet/apuntes/Generalintroductionradar.pdf>

En la **Tabla 1.2** se muestran, las diferentes bandas de los sistemas radar, su margen de frecuencias y las bandas basadas en las asignaciones de la UIT.

Nomenclatura	Margen de Frecuencias	Bandas basadas en las asignaciones de la UIT
VHF	30 – 300 MHz	137 – 144 MHz 216 – 225 MHz
UHF	300 – 1000 MHz	420 – 450 MHz 890 – 940 MHz
Banda P	230 – 1000 MHz	
Banda L	1000 – 2000 MHz	1215 – 1400 MHz
Banda S	2000 – 4000 MHz	2300 – 2550 MHz 2700 – 3700 MHz
Banda C	4800 – 8000 MHz	5255 – 5925 MHz
Banda X	8000 – 12500 MHz	8500 – 1700 MHz
Banda Ku	12.5 -- 18 GHz	13.4 – 14.4 GHz 15.7 – 17.7 GHz
Banda K	18 – 26.5 GHz	23 – 24.25 GHz
Banda Ka	26.5 – 40 GHz	33.4 – 36 GHz
Milimétrica	> 40 GHz	

Tabla 1.2: Bandas de frecuencia radar

Fuente: <http://www.scribd.com/doc/48157376/Analisis-Teorico-Radar>

1.3 DESARROLLO DE LA NAVEGACION AEREA

Desde que en 1903 los hermanos Wright se convirtieron en los primeros en realizar un vuelo en un avión controlado, las personas empezaron a ocupar a los aviones como un medio de transporte para viajar de un lugar a otro. Los primeros aviones utilizados para este fin fueron los Hidroaviones. Al pasar el tiempo muchas más personas utilizaban los aviones para recorrer grandes distancias, con lo que se incremento su número en vuelo, pero a medida había muchos más aviones que podían transportar a muchas más personas (como un Airbus A380), empezaron a ocurrir accidentes, que hicieron que las personas ya no los utilizaran porque no eran un medio de transporte seguro. A raíz de esto se crearon organismos normalizadores con el fin de evitar pérdidas de vidas humanas, garantizando la seguridad a los pasajeros al utilizar aviones para viajar. El primero que apareció fue la Administración Federal de Aviación por sus siglas en ingles (FAA) en Estados Unidos, en el año de 1926 para administrar el tráfico aéreo civil, pero desde ese entonces la FAA se convierte en una entidad gubernamental responsable de la regulación de todos los aspectos de la aviación civil en los Estados Unidos. Desde los atentados terroristas del 11 de septiembre de 2001 la FAA ha estado trabajando conjuntamente con DHS (Departament of Homeland Security) para el control de los vuelos y la seguridad en el aire.

El 9 de diciembre de 1932 se crea la Unión Internacional de Telecomunicaciones (UIT), sustituyendo a la Unión Telegráfica Internacional (UTI) y la Unión Radiotelegráfica Internacional (URI) y su sede se encuentra en Ginebra (Suiza). La UIT se convirtió en la encargada de regular las telecomunicaciones a nivel internacional entre las distintas empresas operadoras y administraciones. La UIT establece los procesos para el uso de las frecuencias para los sistemas radioeléctricos aeronáuticos.

En noviembre de 1944, el gobierno de los Estados Unidos envió invitaciones a 55 estados para asistir a una conferencia sobre la aviación civil, que sería bautizada como la conferencia de Chicago. A esta conferencia solo llegaron 54 estados, que estuvieron reunidos desde el 1 de noviembre al 7 de diciembre. Al término de esta conferencia, se estableció un convenio sobre aviación civil, con el objetivo de asegurar la cooperación internacional en el asunto y llegar a un primer acuerdo sobre organización y seguridad en lo relacionado con el transporte y tráfico aéreo.

Lo que se había firmado en realidad había sido un convenio provisional en el que se involucraron 32 estados. Este convenio provisional promovió la creación de la PICAQ (Provisional International Civil Aviation Organization), de carácter asesor, con el objetivo de dar el primer paso hacia la colaboración internacional en lo relacionado con la aviación civil. Este convenio es conocido como “El Convenio de Chicago” y establecía, en su segunda parte (artículos 43 al 66) el establecimiento de una organización civil para mantener el Convenio actualizado y ejecutar las acciones previstas en él.

La PICAQ estuvo activa desde agosto de 1945 a abril de 1947, luego se convirtió en la actual ICAO (International Civil Aviation Organization) conocida también como la OACI (Organización de Aviación Civil Internacional) con exactamente las mismas funciones que su antecesora, cuya sede fue establecida en Montreal (Canadá), donde sigue actualmente.

Entre las dos finalidades principales de la OACI tenemos: la seguridad de la navegación aérea y la economía del transporte aéreo. Tanto la UIT como la OACI han establecido procedimientos operacionales detallados para las situaciones de emergencia, búsqueda y salvamento.

La OACI es un organismo técnico asociado a la ONU (Organización de las Naciones Unidas) cuyo objetivo es promocionar el desarrollo seguro y ordenado de la aviación civil en todo el mundo. Para llevar a cabo dicha tarea, establece normas internacionales y las regulaciones necesarias para salvaguardar la seguridad, la regularidad y eficiencia en el transporte aéreo. Los documentos de la OACI se organizan en Anexos al Convenio. Cada Anexo trata un tema diferente y todos ellos tienen una orientación técnica y normativa. A la fecha son 18 Anexos y se listan a continuación:

- *Anexo 1*: Licencias al Personal.
- *Anexo 2*: Reglamento del Aire.
- *Anexo 3*: Meteorología.
- *Anexo 4*: Cartas Aeronáuticas.
- *Anexo 5*: Unidades de Medida.
- *Anexo 6*: Operación de las Aeronaves.
 - Parte I: Transporte Aéreo Internacional Comercial - Aeroplanos
 - Parte II: Aviación General Internacional - Aeroplanos
 - Parte III: Operaciones Internacionales - Helicópteros
- *Anexo 7*: Nacionalidad de la Aeronave y Marcas de Registro.
- *Anexo 8*: Aeronavegabilidad de las Aeronaves.
- *Anexo 9*: Facilidades.
- *Anexo 10*: Telecomunicaciones Aeronáuticas.
 - Volumen I (Ayudas a la Radio Navegación)
 - Volumen II (Procedimientos de Comunicación)
 - Volumen III (Parte I: Sistemas de Comunicación de Datos Digitales - Parte II: Sistemas de Comunicación de Voz)
 - Volumen IV (Radares de Vigilancia y Sistemas para Evitar Colisiones)
 - Volumen V (Utilización del Espectro de Radiofrecuencias Aeronáuticas)
- *Anexo 11*: Servicios de Tránsito Aéreo.
- *Anexo 12*: Búsqueda y Rescate.
- *Anexo 13*: Investigación de Accidentes Aéreos.
- *Anexo 14*: Aeródromos.
 - Volumen I: Diseño de Aeródromos y sus Operaciones
 - Volumen II: Helipuertos
- *Anexo 15*: Servicios de Información Aeronáutica.
- *Anexo 16*: Protección ambiental.
 - Volumen I: Ruido de Aeronaves
 - Volumen II: Emisiones de Motores Aeronáuticos
- *Anexo 17*: Seguridad: Salvaguardando la Aviación Civil Internacional contra Actos de Interferencia Ilegítima.
- *Anexo 18*: Transporte Aéreo Seguro de Mercancías Peligrosas.

Es importante mencionar que actualmente hay una propuesta para incluir un Anexo adicional, el Anexo 19 que tratara sobre el SSP/SMS que se refiere en sus siglas en ingles al Safety State Program / Safety Management System.

La OACI también tiene una gran importancia de cara a la cooperación en todos los campos de la aviación civil entre los países miembros, ya que se compromete a ayudar a dichos países en lo relacionado a las instalaciones de la aviación civil o alcanzar las leyes establecidas por ella misma.

La OACI desarrolla importantes proyectos y programas que buscan estimular el desarrollo seguro, ordenado y eficiente de la aviación civil internacional.

Últimamente los esfuerzos de la OACI se han centrado en combatir dos problemas que no han sido resueltos y que son de una importancia esencial: en primer lugar, los daños producidos por los aeroplanos al medio ambiente por medio de ruidos y humos el segundo lugar y más importante, la lucha contra los secuestros y los atentados terroristas.

El servicio de control del tráfico aéreo, también conocido por sus siglas en inglés como ATC (Air Traffic Control) es el encargado del control de las operaciones en las regiones de información de vuelo, por sus siglas en ingles FIR (Flight Information Region) y cada país se hace responsable de este servicio en su área de responsabilidad. En muchos casos esta área de responsabilidad excede las aguas territoriales de un país. El servicio de control del tráfico aéreo se presta por los países firmantes del tratado de Chicago que dieron origen a la creación de la OACI en los términos especificados por las normas de esta organización internacional.

Con el establecimiento del ATC, en cada país firmante del tratado de Chicago, surgen los primeros controladores de tráfico aéreo, los cuales son personas que se encargan de controlar el tráfico aéreo y dar instrucciones a los pilotos por medio de un sistema de voz. El Radar Primario comenzó a tener importancia para la detección de aeronaves, debido a la necesidad que tenían los controladores de tráfico aéreo de poder tener un control sobre las aeronaves. La distancia desde la estación radar hasta la aeronave era estimada por el controlador de tráfico aéreo con la ayuda de un gráfico superpuesto en pantalla compuesto de círculos concéntricos. A medida la tecnología fue avanzando el Radar Primario calculaba el Rango y Acimut de las aeronaves, lo cual le ahorra trabajo a los controladores de tráfico aéreo. El anexo de la OACI que se refiere a los sistemas radar es el Anexo 10 de Telecomunicaciones Aeronáuticas Volumen IV (Radares de Vigilancia y Sistemas para Evitar Colisiones).

Con el fin de evitar pérdidas de vidas humanas, el organismo normalizador de gran importancia para Europa apareció el 13 de diciembre de 1960, ya que la República Federal Alemana, Francia, el Reino Unido, Bélgica, Holanda y Luxemburgo suscribieron el Convenio de EUROCONTROL (European Organisation for the Safety of Air Navigation) que es la Organización Europea para la Seguridad de la Navegación Aérea, que entró en vigencia en 1963. Esta agencia tenía como objetivo inicial resolver los enormes problemas que el excesivo límite de acceso a la información del espacio aéreo centroeuropeo genera en las rutas aéreas que lo atraviesan. Se buscaba entonces la unificación de los servicios de

Control y navegación aérea de Europa, más allá de las limitaciones que imponían los espacios aéreos nacionales. Esto implica la organización común del espacio aéreo, la explotación en común de los servicios ATC en el espacio aéreo superior, la planificación de las instalaciones, estandarización del equipamiento, la formación del personal y la percepción de las tarifas por ayudas a la navegación aérea. Durante un tiempo los pilares básicos de EUROCONTROL pasaron a ser la estandarización del equipamiento y la percepción de tarifas debido a la negativa de los ingleses, franceses y alemanes a que EUROCONTROL gestione su espacio aéreo en los años 70. A pesar de las diferencias, fue posible la creación del centro de Control de Maastricht, primer Centro de Control europeo y poco a poco la generalización y normalización de EUROCONTROL ha ido abarcando a todos los demás países de Europa.

A medida crecía el tráfico aéreo, surge la necesidad para los controladores de tráfico aéreo de disponer de un radar que les ayudara a la identificación de las aeronaves, ya que el Radar Primario no podía identificarlas, por lo que se desarrolló el sistema de Radar Secundario², así a cada controlador se le asigna un número específico de aeronaves en un determinado sector.

A medida fue creciendo el número de aeronaves en vuelo, el sistema de Radar Secundario comenzó a presentar problemas del tipo operativo, ya que en un determinado momento los 4,096 códigos asignados a las aeronaves, no alcanzaban para manejar la gran cantidad de aeronaves, por lo que los controladores de tráfico aéreo tenían que esperar a que una aeronave liberara un código transponder (se le llamo así a un equipo especial que tenían las aeronaves para poder ser identificadas mediante un código octal de cuatro números), para poder asignarlo a otra aeronave. Un ejemplo de esta situación la encontramos en uno de los aeropuertos más concurridos del mundo como es el Aeropuerto de Londres Luton (London Luton Airport), código IATA: LTN, código OACI: EGGW, en el cual la cantidad de aeronaves en un año excede los 49,152 códigos transponder que se dispondrían si cada mes se utilizaran los 4,096 códigos de los cuales dispone el Radar Secundario. La **Tabla 1.3** muestra el número de movimientos desde 1997 del Aeropuerto de Londres Luton.

Esperar a que un código transponder estuviera disponible, resultaba impráctico por lo que Estados Unidos y el Reino Unido comenzaron a trabajar en el desarrollo de un nuevo sistema radar que solucionara este problema, desarrollando el Radar Secundario Modo S, el cual es capaz de manejar casi 17 millones de códigos transponder y mucha más información que el Radar Secundario. Con el desarrollo de este sistema el problema quedó resuelto, en cuanto a la asignación de códigos transponders a más de 4,096 aeronaves.

² Este sistema se aborda en la sección 1.2.2.1.1

	Número de pasajeros	Número de movimientos
1997	3,238,458	63,586
1998	4,132,818	70,667
1999	5,284,810	79,423
2000	6,190,499	84,745
2001	6,555,155	83,707
2002	6,486,770	80,924
2003	6,797,175	85,302
2004	7,535,614	94,379
2005	9,147,776	107,892
2006	9,425,908	116,131
2007	9,927,321	120,238
2008	10,180,734	117,859

Tabla 1.3: Estadísticas del Aeropuerto de Londres Luton

Fuente: United Kingdom Civil Aviation Authority

En El Salvador CEPA (Comisión Ejecutiva Portuaria Autónoma) tiene a cargo la construcción, administración y operación del Aeropuerto Internacional El Salvador (AIES), pero quien responde ante la OACI es la AAC (Autoridad de Aviación Civil). Con el uso de los sistemas de Radar Primario y Secundario se logra un control del tráfico aéreo bastante eficiente, ya que las aeronaves no sobrepasan al mes, los 4,096 códigos transponder que maneja el Radar Secundario, como muestra la **Tabla 1.4** en las estadísticas del Aeropuerto Internacional El Salvador del año 2010.

Mes	Aterrizajes	Estacionamientos
ENERO	1,468	1,025
FEBRERO	1,441	1,310
MARZO	1,365	1,296
ABRIL	1,458	991
MAYO	1,339	960
JUNIO	1,429	843
JULIO	1,474	813
AGOSTO	1,573	791
SEPTIEMBRE	1,458	873
OCTUBRE	1,443	940
DICIEMBRE	3,013	1,744
Total	17,461	11,586

Tabla 1.4: Estadísticas del Aeropuerto Internacional El salvador del año 2010

Fuente: <http://www.aeropuertoelsalvador.gob.sv>

A medida que los sistemas radar se hagan más eficientes, los amplificadores de potencia de RF, tendrán nuevos requerimientos por la necesidad de reducir el tamaño, peso y costo de los sistemas radar, que se traduce en una ventaja económica. La eficiencia de un radar permite un mejor control del tráfico aéreo a los controladores de tráfico aéreo (separaciones verticales y horizontales de aeronaves) y una mayor seguridad para los pasajeros que viajan

en los aviones comerciales, debido a que se pueden evitar accidentes o indicar situaciones de emergencia. Una ventaja económica para las aerolíneas aéreas es que a mayor seguridad en las aeronaves utilizando sistemas radar más eficientes, mayor será el flujo de pasajeros y por tanto mayor serán sus ingresos, por el incremento de vuelos. Una ventaja social para los pasajeros al utilizar radares más eficientes es que el flujo de aeronaves será cada vez mayor debido al control estricto que se tienen en la vigilancia continua de las aeronaves, que permite especificar muchas más rutas aéreas y por tanto tener en vuelo muchas más aeronaves.

1.4 RADARES AERONAUTICOS

1.4.1 Historia de radares aeronáuticos

El rápido desarrollo del radar propiciado por la Segunda Guerra Mundial, supuso su aplicación para el control del tráfico aéreo por sus siglas en inglés ATC (Air Traffic Control), que es el que se encarga del control del tráfico aéreo de aeronaves en un determinado país. El radar también se utilizó para la vigilancia continua, proporcionar la posición de aeronaves en vuelo, seguridad y mejorar la eficiencia del sistema de gestión de tráfico aéreo por sus siglas en inglés de ATM (Air Traffic Management). El Radar Primario sirvió al control del tránsito aéreo durante cincuenta años, ya que el tráfico crecía lo suficiente como para complicar la discriminación de los ecos (reflexiones de ondas electromagnéticas) en pantalla, es así como los ingenieros desarrollaban el sistema que lo complementaría. El Radar Primario de Vigilancia no puede identificar a las aeronaves, por lo que se desarrolló el sistema IFF (Identification Friend / Foe), que significa “identificación amigo / enemigo”, su primer uso fue militar y permitía identificar aviones del propio bando (equipados con transponder) de los del bando contrario. El transponder o respondedor es un equipo de a bordo de la aeronave, capaz de responder a las interrogaciones hechas por la estación radar en tierra y que básicamente consta de receptor, decodificador, codificador y transmisor, este equipo permite identificar a una aeronave con un código transponder de cuatro dígitos octales de 4096 disponibles (2^{12}), que el piloto ha introducido ya que ha sido proporcionado por el ATC, para poder identificar a la aeronave. Entre 1950 y 1957 se establece como frecuencia de interrogación 1030 MHz y como frecuencia de respuesta 1090 MHz, con diferentes Modos de interrogación (Modos 1, 2 y 3). El Departamento de Defensa de Estados Unidos ofrece el IFF Mark X en el año de 1953 para uso civil como un sistema común para control de tráfico aéreo. Así gracias a los esfuerzos de organizaciones civiles y militares se produce el ATCRBS por sus siglas en inglés (Air Traffic Control Radar Beacon System) que significa sistema de seguimiento radar para control del tráfico aéreo. Este sistema radar permitía conocer la altitud de una aeronave, por lo que los blancos detectados por este sistema pasan a ser blancos tridimensionales. Al comienzo de los años 60 el sistema ATCRBS se implanta en los Estados Unidos y en 1967 aparece como obligatorio el uso de transponders para las aeronaves a una altitud de 18,000 pies que se encuentren operando en el sistema de control de tráfico aéreo. En 1962 el sistema ATCRBS aparece recomendado por la OACI en su anexo 10 de Normas y Métodos y se le denomina SSR (Secondary Surveillance Radar) que significa Radar Secundario de Vigilancia. El sistema SSR, es un equipo de características especiales, ya que es una tecnología desarrollada como parte de la estrategia de control de

tránsito aéreo y vigilancia radar. Las denominaciones de Primario y Secundario aparecieron en este momento porque el Radar Secundario se instaló como segundo radar en el mismo emplazamiento del Primario existente y montando la nueva antena sobre la del Primario, complementándolo. El SSR convencional utiliza el concepto de Ventana deslizante que en inglés se denomina como “SLIDING WINDOW” y tiene que ver con el proceso de extracción, en este proceso se busca un número de respuestas con códigos coincidentes en una serie consecutiva de interrogaciones, que se igualan según la distancia. Pero esta técnica presenta problemas en el cálculo exacto del acimut y genera confusión sincrónica por lo que se buscó una solución para mejorar el cálculo. Esta imprecisión da lugar a errores en las pistas de los blancos radar en la pantalla radar o PPI (Indicador de Posición de Plan), ya que no siguen líneas continuas, sino que más bien son quebradas, por lo que los controladores aéreos indican a los pilotos que se debe mantener una mayor distancia de seguridad entre las aeronaves de una misma ruta, desaprovechando el espacio aéreo.

El radar SSR tiene tres principales limitaciones que lo hacen poco adecuado para el uso futuro, una de tipo funcional que es la identificación única de una aeronave mediante un código transponder y dos de tipo técnico que son el solape de respuestas y el cálculo de la posición.

Una vez identificadas las limitaciones del SSR, a comienzos de los años 70 el Reino Unido y Estados Unidos se pusieron en marcha de forma coordinada para llevar a cabo lo que hasta entonces habían hecho de forma separada desarrollar el Modo selectivo “S”, por el que desde tierra se pudiera seleccionar individualmente a la aeronave interrogada para eliminar las respuestas solapadas del SSR convencional. En 1977 tras el desarrollo de un sistema experimental se iniciaron los ensayos en Estados Unidos. Eurocontrol estableció en febrero de 1980, un grupo de trabajo, en este grupo estaba la OACI (Organización de Aviación Civil Internacional), IATA (Asociación de Transporte Aéreo Internacional), IFALPA (Federación Internacional de Asociaciones de Pilotos de Línea Aérea), lo que ayudó a la unificación de los criterios operativos del sistema.

Una vez recibido el visto bueno del Grupo de Expertos sobre mejoras del Radar Secundario de Vigilancia y Sistemas Anticolisión de la OACI, las características básicas del sistema SSR en Modo S se incluyeron en el año de 1987 en el Anexo 10 Volumen IV de la OACI (Telecomunicaciones aeronáuticas). Desde entonces la OACI como la FAA establecieron que los aviones de nueva fabricación fueran equipados con transponder Modo S y pusieron en marcha a escala global diversos programas para facilitar su implementación.

Una técnica aparecida en la década de los años 80 que se conoce como monopolso consiste en que solo se necesita un impulso de cualquier respuesta del transponder de a bordo de la aeronave para determinar, mediante radiogoniometría, el acimut de un blanco dado. El Radar Secundario pasa así a denominarse MSSR (Monopulse Secondary Surveillance Radar) que significa Radar Secundario de Vigilancia Monopulso. Al aplicar esta técnica se reducen los efectos de la confusión sincrónica con lo que el sistema deja de saturarse. Con esto se mejora la función de vigilancia y hay un aumento de la capacidad del sistema ATC, sin modificación alguna de los transponders de a bordo.

En Estados Unidos, se reguló que los aviones matriculados después de 1989 deberían llevar un transponder Modo S capaz de soportar también la operación TCAS II (Sistema Anticolisión que utiliza interrogaciones SSR para proporcionar a la tripulación información que contiene un aviso de alerta de tránsito como una ayuda para el principio de “ver y eludir” versión II). En Centroamérica se instalaron los primeros Radares a mediados de la década de 1990. En 1994 se exigió la implantación de transponder Modo S en todos los aviones comerciales. En Europa, Eurocontrol exigió su implantación total a partir de enero de 2005, cifra que se consiguió al inicio del siglo XXI.

Con el desarrollo de la Navegación Aérea, los sistemas radar adquirieron gran importancia para diversas tareas, entre ellas la más importante brindar seguridad a las aeronaves en su recorrido desde un punto de origen hasta un punto de destino, por lo que aparecen fabricantes de radares comerciales como INDRA, ALENIA, etc.

El sistema radar IRS-20 MP/L es la tercera generación de radares secundarios de INDRA y cumple los requisitos establecidos por la OACI y las prestaciones exigidas por Eurocontrol para los Sistemas de Vigilancia y Navegación Aérea. Este sistema supone una solución revolucionaria de bajo coste y altas prestaciones que ofrece una gran flexibilidad para adaptarse a las necesidades de cada situación. El IRS-20 MP/L emplea la técnica monopulso y de extracción de código mediante la utilización de algoritmos muy avanzados lo que garantiza una gran precisión en la determinación del acimut del blanco. Puede trabajar de manera aislada o asociado a un Radar Primario.

El Aeropuerto Internacional El Salvador (AIES) cuenta con equipos especializados, como Sistema Meteorológico Integrado, que ha sentado un precedente a nivel centroamericano en cuanto a información aeronáutica, acorde a los estándares de la OACI. Posee un Radar Primario, con canal meteorológico incorporado, con alcance de 80 millas náuticas, es decir con capacidad para detectar movimientos de aeronaves en un radio de 148.18 kilómetros. Posee un Radar Secundario Monopulso con un radio de alcance de 255 millas náuticas, equivalentes a 472.31 kilómetros. El radar posee un moderno sistema de administración aérea con capacidad para manejar hasta 1,000 objetos simultáneamente. El Salvador posee además de estos radares en el AIES un Radar Secundario Monopulso que se encuentra ubicado sobre la carretera panorámica, este sistema es el radar modelo IRS-20 MP/L, que empezó a funcionar a partir del año 2009.

En el 2010 con la instalación de tres radares meteorológicos. El Salvador se convierte en el primer país de Centroamérica en contar con el sistema de cobertura total para monitorear la estimación de lluvia local.

A medida que la Navegación Aérea se hace más segura, se visualiza que los sistemas de tratamiento de datos radar deben manejar la información de las aeronaves de una manera mucho más eficiente y abierta, mediante un protocolo que permita el intercambio de datos radar entre diferentes regiones, aumentando así la seguridad de las aeronaves, mediante la vigilancia de las mismas.

1.4.2 Función de radares aeronáuticos

Entre las principales funciones de los radares aeronáuticos tenemos:

- **Servicio de control de Tránsito Aéreo:** El SSR permite el servicio de control de Tránsito Aéreo, debido a que puede manejar 4096 códigos transponder diferentes que permiten la identificación de aeronaves, aunque el SSR en modo S puede manejar más de 16 millones de códigos transponder diferentes.
- **Servicio de información de vuelo:** Debido a que son utilizados por los controladores aéreos, para monitorear los vuelos.
- **Vigilancia de aeronaves:** Los radares aeronáuticos nos permiten la vigilancia continua de aeronaves equipadas con transponder y también aquellas que no lo poseen utilizando un radar primario.
- **Identificación de aeronaves:** Permite la identificación de las aeronaves el SSR, MSSR y el SSR Modo S, debido a que a cada aeronave se le asigna un código transponder, que es proporcionado por el ATC.
- **Brindar seguridad a las aeronaves (altitud mínima de seguridad):** Los radares aeronáuticos brindan seguridad a las aeronaves, ya que los controladores aéreos pueden saber la posición y altitud de una aeronave, con lo que se puede evitar accidentes o situaciones de emergencia.
- **Asistencia radar a aeronaves en aproximación final:** Al fallar los sistemas ILS (sistema de aterrizaje por instrumentos) o de ser requerido, los radares aeronáuticos pueden brindar asistencia radar a las aeronaves en aproximación final hacia la pista, para que las aeronaves puedan aterrizar.
- **Establecimiento de separación radar entre aeronaves sucesivas:** Los radares aeronáuticos permiten la separación tanto horizontal como vertical de aeronaves a los controladores aéreos, debido a que se conoce el rango, acimut y altitud de las aeronaves.

1.5 CARACTERISTICAS TECNICAS DE RADARES AERONAUTICOS

1.5.1 Radar Primario de Vigilancia (PSR)

El Radar Primario de Vigilancia, por sus siglas en inglés PSR (Primary Surveillance Radar) posee una antena que gira 360° y emite un impulso de radio, que se refleja en el blanco y regresa a la antena del radar en forma de eco, apareciendo en la pantalla radar como una señal luminosa.

La **Figura 1.18** muestra el principio de funcionamiento de un radar primario básico. La antena radar ilumina a la aeronave (blanco) con una señal de radio, el blanco la refleja y entonces es captada por un dispositivo receptor. La señal eléctrica que capta la antena se

denomina “eco radar”. La señal radar es generada por un transmisor potente y recibida por un receptor altamente sensible.

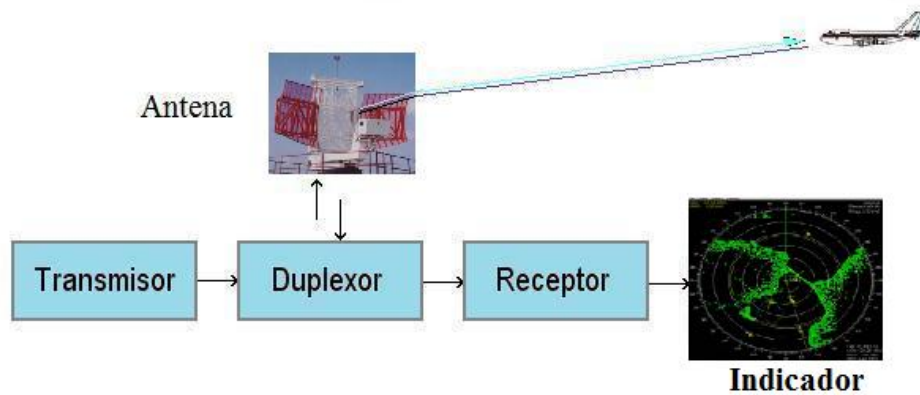


Figura 1.18: Principio de funcionamiento de un radar primario básico

La explicación detallada de cada uno de los bloques de la **Figura 1.18** se presenta a continuación:

ANTENA RADAR: Es la que convierte la energía del transmisor en ondas electromagnéticas en el espacio, con la distribución y eficiencia requeridas. Este proceso se aplica en forma inversa para la trayectoria de recepción. El sistema de antena para su presentación se divide en tres grandes partes, la primera es el conjunto del reflector este se construye en forma compuesta o módulos para reducir significativamente el peso y al mismo tiempo incrementar la rigidez de todo el conjunto, el haz principal usado para transmitir la señal radar y recibirla esta nominalmente dirigido a $+3.0^\circ$ de elevación con relación al horizonte y el único haz auxiliar de recepción esta nominalmente dirigido a 2.0° en elevación por encima del haz principal. El reflector puede ser inclinado manualmente en elevación entre -3.0° y $+6.0^\circ$ para que logre ajustarse al terreno. La segunda es el conjunto de alimentación que hace referencia a las estructuras de tres canales de señal que son el principal de guía de onda, el canal auxiliar de coaxial y un canal de prueba y la tercera es el conjunto del pedestal que soporta los dos motores eléctricos con sus respectivas acometidas eléctricas. La **Figura 1.19** muestra el Patrón de radiación de una antena real de Radar Primario, del tipo parabólica. Este tipo de antenas poseen alta ganancia y lóbulos laterales relativamente pequeños.

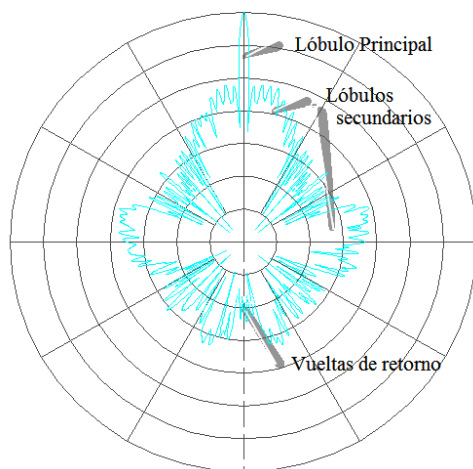


Figura 1.19: Patrón de radiación de una antena real de Radar Primario
Fuente: <http://www.radartutorial.eu/06.antennas/an10.de.html>

La **Figura 1.20** muestra la antena de un Radar de Vigilancia Primario (PSR) del fabricante INDRA modelo ASR12.



Figura 1.20: PSR INDRA ASR12

TRANSMISOR: Es un amplificador de potencia, es el que produce los pulsos de energía de RF de corta duración y alta potencia que la antena radia al espacio. Durante esta emisión el receptor queda aislado de la antena, debido a un conmutador (transmisor-receptor). Puede ser de diferentes tipos como un Klystron (Tubo de electrones), un TWT (Traveling-Wave Tube) o un CFA (Crossed Field-Amplifier). La aplicación que tenga el radar definirá la potencia que debe tener el amplificador que se utilice, para no cometer el error de invertir en transmisores caros. El transmisor utilizado en una estación de vigilancia aérea, puede tener una potencia promedio del orden de los kilowatts.

DUPLEXOR: Es el que alterna la antena entre el transmisor y el receptor, de modo que solo es necesario usar una antena. Este switching se requiere para evitar que los pulsos de

alta potencia del transmisor destruyan el receptor altamente sensible. El duplexor puede ser un aislador de estado sólido o alguna forma de dispositivo de descarga en gas.

RECEPTOR: Es el que amplifica y demodula las señales de RF recibidas y genera señales de video a la salida. Generalmente es una unidad totalmente en estado sólido. Utiliza un conjunto de amplificadores de alta potencia por sus siglas en inglés HPAA, En tres etapas, como un mezclador/divisor radial que ayuda de manera eficiente a las salidas de alta potencia en forma individual.

INDICADOR: Es el que presenta al operador una visualización continua y fácil de entender de la posición relativa de los blancos radar. Generalmente es conocido por sus siglas en inglés con el nombre de PPI, que significa Indicador de Posición de Plan. La **Figura 1.21** muestra un PPI de Radar Primario.

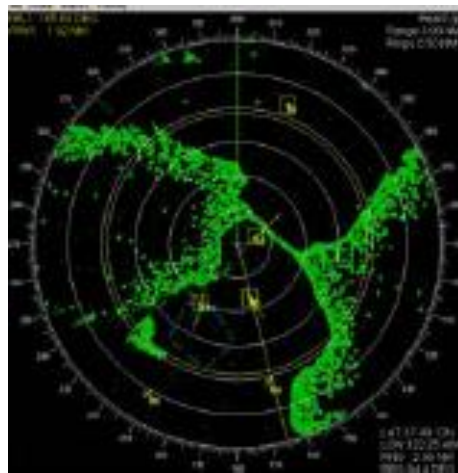


Figura 1.21: Indicador de Posición de Plan de un Radar Primario

El Radar Primario no puede identificar a las aeronaves, lo cual resulta un requisito indispensable cuando se utiliza como un Sistema de Vigilancia Aérea. Por ello su uso en el futuro será cada vez menor, como parte del Sistema de Control de Tránsito Aéreo. Normalmente se utiliza en combinación con un Radar Secundario, siendo ubicadas las antenas en el mismo emplazamiento, a esta combinación de PSR + SSR se le llama Multiradar.

1.5.2 Radar Secundario de Vigilancia (SSR)

El radar Secundario de Vigilancia consiste en una antena que gira 360° enviando al espacio libre, señales de radio. La relación existente entre la estación de tierra y la aeronave es del tipo activo, es decir que la aeronave deja de ser un blanco pasivo, para convertirse en un blanco activo, ya que la estación radar en tierra interroga a una frecuencia de 1030 MHz, y las aeronaves equipadas con un transponder responden a esta interrogación a una frecuencia de 1090 MHz.

Actualmente el más utilizado es el de tipo pulsante. La **Figura 1.22** muestra el periodo de los pulsos de un radar de onda pulsante.

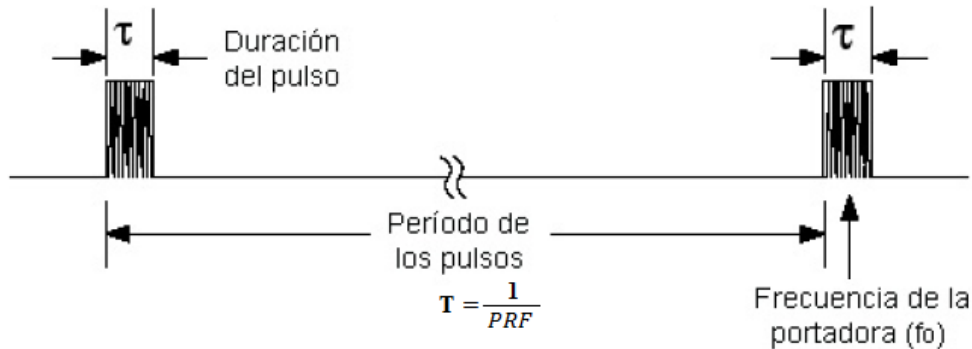


Figura 1.22: Periodo de los pulsos de un radar de onda pulsante

Los pulsos de transmisión permiten definir el concepto de Tasa de Repetición de Pulsos o PRF (abreviatura que viene del inglés Pulse Repetition Frequency), que se conoce también como frecuencia de repetición de pulso.

La ecuación que define el PRF es:

$$PRF = \frac{1}{T}$$

En donde T, es el tiempo de repetición del pulso, conocido por sus siglas en inglés como PRT. Este periodo de tiempo es en el cual el radar espera recibir una respuesta del blanco (pequeños pulsos de RF). El intervalo de tiempo entre dos pulsos de transmisión, determina el alcance máximo de un radar por la siguiente ecuación:

$$R_{max} = \frac{C_0 T}{2}$$

Poniendo el periodo de repetición del pulso en función del PRF nos queda la siguiente ecuación:

$$T = \frac{1}{PRF}$$

Sustituyendo esta ecuación en la ecuación de R_{max} nos queda finalmente la siguiente ecuación:

$$R_{max} = \frac{C_0}{2 PRF}$$

Así mediante la especificación del PRF, se puede saber que rango máximo tendrá el radar. Por ejemplo si un radar tiene un PRF = 323.8 Hz su alcance máximo sería:

$$R_{max} = \frac{3 \times 10^8}{\frac{2 \times 323.8}{1.853 \times 10^3}} = 250 \text{ NM}$$

1.5.2.1 Transponder

El transponder es un equipo de panel con un display, que puede ser mecánico o digital, que se encuentra instalado en la aeronave, en donde aparece el código de 4 dígitos octales, que el piloto introduce y que fue asignado por el ATC, normalmente tiene un selector para cada dígito y un selector con las posiciones OFF, SBY, ON, ALT y TST. En la **Figura 1.23** se muestra un transponder mecánico.

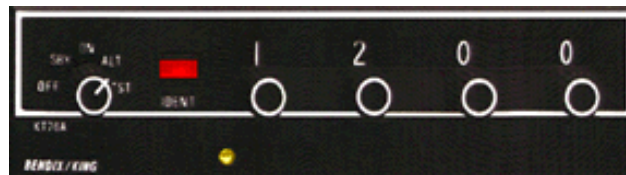


Figura 1.23: Transponder mecánico

Los transponder con pantalla digital, proporcionan además del código información del estado del equipo así como de la altitud que se transmite. En la **Figura 1.24** se muestra un transponder digital.



Figura 1.24: Transponder digital

La información de altitud, se expresa como nivel de vuelo en el transponder digital, es decir la altura sobre la superficie isobárica de 1013.2 HPa. El sistema se completa con una antena, que normalmente es de forma triangular, que se instala en la parte baja del avión.

En la posición SBY (Standby) el transponder está alimentado pero no tiene activada la respuesta a las interrogaciones, hasta que el equipo alcance la temperatura de trabajo. El procedimiento normal es mantener el transponder en SBY hasta haber despegado. independientemente de que se seleccione el código de respuesta (4 dígitos octales) al recibirlo del controlador.

Al despegar se debe poner el selector en posición ON, para que el transponder responda a las interrogaciones del SSR. La razón por la que no se pone antes en la posición ON es evitar ecos indeseados en la pantalla radar del controlador aéreo.

En la posición ALT, el transponder además de responder a las interrogaciones del SSR proporciona información del nivel de vuelo, siendo esta la posición que se utiliza en vuelo, a menos que el controlador indique lo contrario.

En la posición TST (test) el equipo no responde a las interrogaciones del SSR, es una posición para comprobar el equipo, conectándose toda la iluminación del mismo (para comprobar si está funcionando bien) y efectúa distintas rutinas de chequeo internas.

En la posición GND (ground) para algunos equipos, sirve para que se desconecte automáticamente el transponder cuando el tren de aterrizaje está bajado, para otros rechaza las respuestas en Modo A y C.

El transponder lleva un pulsador IDT o Ident, que no se pulsa a menos que sea necesario y que lo indique el controlador. Este pulsador genera una señal especial o SPI (Supplemental Pulse Identification) que hace que el eco de la aeronave en la pantalla quede iluminado de una forma más brillante durante 15 a 30 segundos, lo cual le ayuda al controlador para identificar mejor la aeronave.

Existen los siguientes tipos ó Modos para el SSR: Modo 1, Modo 2, Modo 3/A, Modo B, Modo C, Modo D. La **Tabla 1.5** es una breve descripción de cada uno de los Modos utilizados por el SSR³.

Modo		Aplicación
Militar	Civil	
1		Se utiliza para la identificación militar 32 códigos de 4096.
2		Se utiliza para identificar a un avión individual, con un número antes del despegue.
3	A	El Modo 3 o Modo A son el mismo modo de interrogación y se utiliza para identificar a una aeronave con un código de 4 dígitos octales de 4096 posibles.
	B	Reservado para uso futuro
	C	Es el que proporciona la altitud de la aeronave
	D	Reservado para uso futuro

Tabla 1.5: Modos de interrogación
Fuente: <http://dspace.epn.edu.ec/handle/123456789/910>

³ Abreviatura en inglés para indicar Radar Secundario de Vigilancia.

1.5.2.2 Antena del SSR

La antena del SSR es del tipo de arreglo de dipolos, lo que significa que no es tan directiva como la del PSR que es del tipo parabólica, por lo que la antena del SSR va a tener una ganancia menor y el ancho del haz es más ancho que la del PSR. En el SSR se busca que las respuestas sean las más cercanas al eje de la antena (bore sight). La antena radia un patrón de radiación horizontal denominado Suma (Σ), que es el haz de interrogación. Las respuestas se reciben por un patrón direccional denominado delta (Δ) en el caso del radar MSSR⁴. En la **Figura 1.25** se muestran los patrones de radiación Suma (Σ) y Diferencia (Δ) típicos de un MSSR.

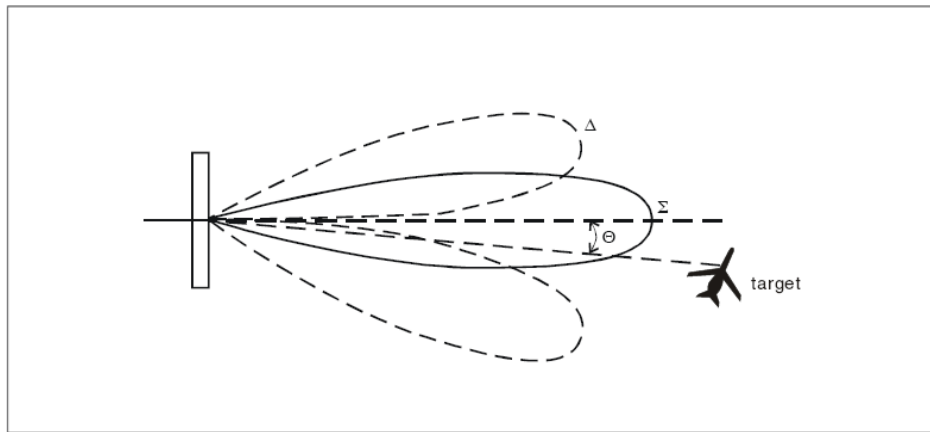


Figura 1.25: Patrones de radiación Suma (Σ) y Diferencia (Δ)

Fuente: Manual on the Secondary Surveillance Radar (SSR) Systems, Third Edition 2004

En la **Figura 1.26** se muestra el haz de control y de interrogación de la estación de tierra. La función del haz de control es que los blancos que están muy cercanos a la antena no capten los lóbulos secundarios del haz de interrogación, para evitar que a la estación radar en tierra lleguen respuestas equivocadas.

⁴ Abreviatura en ingles de Radar Secundario de Vigilancia Monopulso

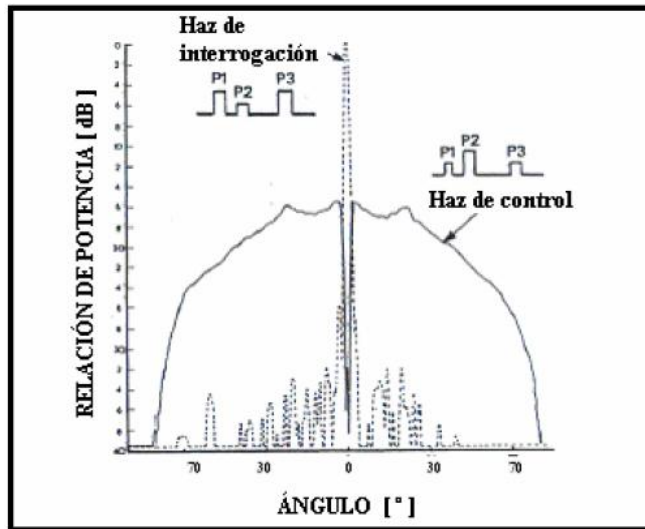


Figura 1.26: Haz de control y de interrogación de la estación de tierra
Fuente: <http://www.tsc.uniovi.es/Rad/Trabajos/Radar%20Secundario.pdf>

En la **Figura 1.27** se muestran los haces en el MSSR en donde se observa la presencia del haz de Diferencia (Δ).

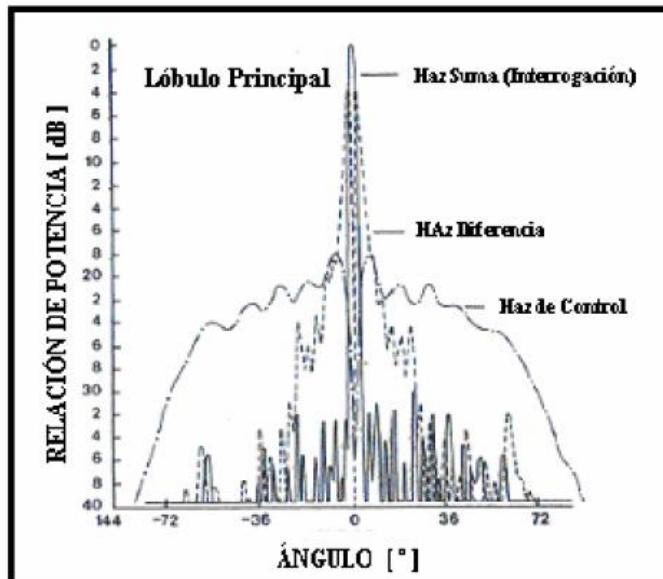


Figura 1.27: Haces en el MSSR
Fuente: Secondary Surveillance Radar, Michael C. Stevens

1.5.2.3 Transmisor del SSR

Es el que amplifica y modula lo que le manda el Coder (interrogación codificada) con la frecuencia de RF para enviarla a la antena, frecuencia de transmisión de 1030 MHz.

Normalmente el Radar Secundario trabaja con la mitad del PRF⁵ del Radar Primario, PRF de transmisión 50-450 Hz, un valor típico es 190 Hz, con un máximo de 282 Hz para 250 MN, potencia de transmisión típica de 500 watts (57 dBm⁶).

1.5.2.4 Receptor del SSR

El receptor del SSR es del tipo superheterodino con un mezclador, oscilador local, amplificador de FI y detector. Es el que filtra, amplifica y demodula los impulsos de la respuesta que se recibe a una frecuencia de 1090 MHz, con una potencia de 125 a 500 W (21 a 27 dBW⁷).

1.5.2.5 Coder del SSR

La función del Coder o Codificador es codificar la interrogación que será amplificada y modulada con ayuda del transmisor con la frecuencia de RF para enviarla al transponder.

1.5.2.6 Decoder del SSR

El Decoder o Decodificador decodifica y descifra la respuesta según la información deseada.

La **Figura 1.28** muestra el principio de funcionamiento de un Radar Secundario básico.

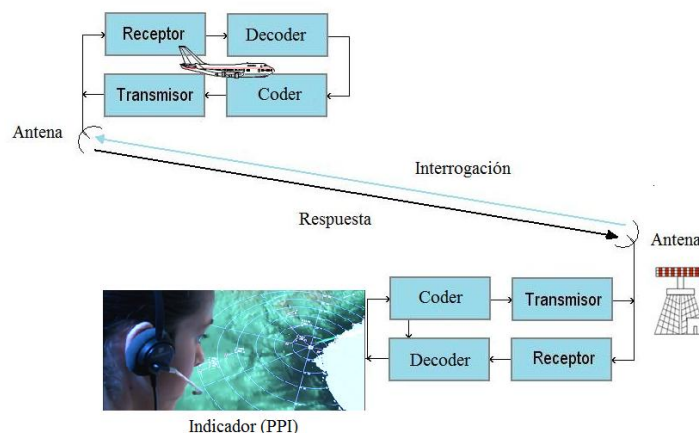


Figura 1.28: Principio de funcionamiento de un Radar Secundario básico

El SSR está formado por una estación en tierra que se llama interrogador y de un transponder que se encuentra a bordo de la aeronave, este equipo proporciona un enlace de datos de dos vías, con distintas frecuencias de transmisión y recepción.

⁵ Siglas en ingles que significan frecuencia de repetición de pulso

⁶ Unidad de medida de potencia

⁷ Unidad de medida de potencia

El procesamiento es de la siguiente forma:

1. En el interrogador de tierra se sincroniza la señal, se genera un modo de interrogación el cual es codificado en el Coder (Codificador), luego el transmisor modula estos impulsos con la frecuencia de RF y los envía a la antena para transmitirlos hacia la aeronave (máximo 2 kW según la OACI).
2. En el transponder de la aeronave se capta la interrogación en la antena, el receptor amplifica y demodula los impulsos de la interrogación, el Decoder los decodifica y descifra la pregunta según la información deseada e induce al Coder a preparar la respuesta conveniente, lo que el Coder hace es codificar la respuesta que será amplificada y modulada con ayuda del transmisor con la frecuencia de RF para enviarla a la estación de tierra.
3. En el interrogador de tierra el receptor (que detecta una señal mínima detectable de -88 dBm según la OACI) filtra, amplifica y demodula los impulsos de la respuesta, que contienen información de modo y código que son decodificados por el Decoder y finalmente en el Indicador del ATC⁸ se representa la información obtenida de la aeronave.

El sistema Radar Secundario Monopulso MSSR (Monopulse Secondary Surveillance Radar) tiene como fin la transmisión-recepción de interrogaciones-respuestas empleando la técnica monopulso. El vídeo obtenido es procesado para extraer de él la información sintética correspondiente a cada blanco (distancia, acimut, códigos de identificación especial, etc.). Esta información es formateada para poder ser usada por el sistema de proceso de datos en el formato ASTERIX⁹.

Una vez determinada la presencia del blanco, se procede a generar un mensaje por blanco y vuelta de antena en el que se determina la distancia, acimut, códigos SPI y emergencias de acuerdo con el formato ASTERIX.

La salida de datos de los blancos, se realiza por medio de dos líneas de la red LAN¹⁰. La LAN-1 está conectada al extractor 1, enviando la información ASTERIX y la LAN-2 está conectada al extractor 2 enviando también información ASTERIX. En la información ASTERIX, el blanco además de su propia información incluye el dato de la hora universal UTC con que se detectó.

El SSR proporciona, además de la información de rango y acimut:

- **Altitud:** El SSR obtiene el valor de altitud de la aeronave por medio del Modo C, las respuestas al Modo C, se generan automáticamente al estar asociado el transponder o también conocido como respondedor al altímetro del avión (cabe notar que un grupo de 1278 claves se destinan para cifrar la información de altitud). Las respuestas de altitud contienen la información de altitud en código gray

⁸ Siglas en ingles que significan Control de Tránsito Aéreo

⁹ Protocolo de intercambio de datos radar ASTERIX de Eurocontrol

¹⁰ Siglas en ingles que significan Red de trabajo de Area Local

en niveles separados cada 100 pies. Así este código, al cambiar un solo pulso entre un nivel y el siguiente, permite detectar fácilmente posibles errores.

- **Código transponder:** El SSR obtiene el código transponder asignado a la aeronave por medio del Modo 3/A.

El transponder mas utilizado en aviación privada y deportiva es el compatible con Modo A y C, aunque en los aviones comerciales ya se está extendiendo el uso del Modo S, el cual permite añadir información codificada que mejora la identificación y la información sobre la aeronave. La asignación de claves SSR las realiza el Centro de Control de Tránsito Aéreo (ATC) de acuerdo con la normatividad internacional vigente sobre navegación aérea.

Existen unos códigos internacionales sobre peligros o fallas a bordo de la aeronave, estos son los siguientes:

7500 INTERFERENCIA ILICITA. Se utiliza para notificar al Centro de Control que la aeronave se encuentra en situación de interferencia ilícita o secuestro.

7600 FALLA DE RADIO. Una vez se tiene la certeza de que el receptor de radio tiene una falla se seleccionará esta clave para informar de tal suceso al Centro de Control y se proseguirá según los procedimientos de seguridad establecidos.

7700 EMERGENCIA. Salvo indicación contraria del ATC, es la clave que se seleccionará en el transponder para indicar que la aeronave se encuentra en una situación de emergencia que puede cubrir las antes mencionadas. El uso de estas claves, está limitado a situaciones reales como las descritas.

Existen otros códigos especiales como:

- Código 0000 para ser asignado, mediante acuerdos regionales, para uso general.
- Código 2000 para la solicitud de asignación de código.
- Código 1215 para búsqueda satelital de emergencia.
- Código 1200 código no decodificado del 2700
- Código 2700 para navegación comercial.

En La **Figura 1.29** se muestra la combinación de ambos sistemas PSR + SSR del fabricante INDRA, generalmente usada en aeropuertos en área terminal, a la que se le denomina Multiradar.



Figura 1.29: Multiradar PSR + SSR INDRA

1.5.3 Radar Secundario de Vigilancia Modo S

El Modo S es un sistema avanzado de interrogación asociado al Radar Secundario de Vigilancia, que se caracteriza por poder interrogar de un Modo Selectivo (de ahí su denominación) a las aeronaves equipadas con un transponder compatible (este equipo es muy similar al transponder SSR con la diferencia de que procesa mucha más información).

La **Figura 1.30** muestra el principio de funcionamiento de un Radar Secundario básico Modo S que es similar al SSR, en lo que cambia es en el tipo de información que transmite y recibe, en lo que se refiere a la identificación de las aeronaves y otros datos.

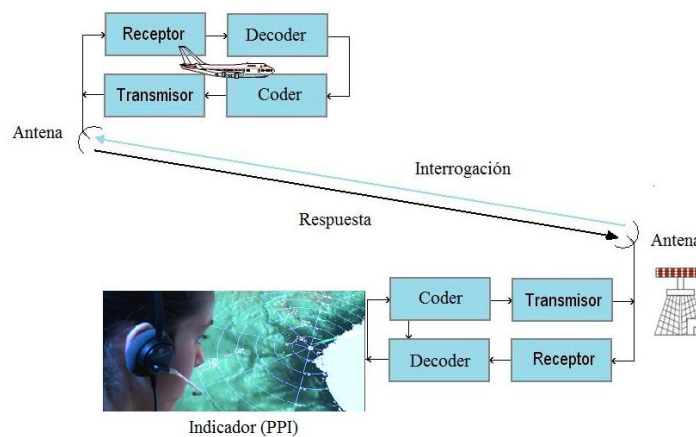


Figura 1.30: Principio de funcionamiento de un Radar Secundario Modo S básico

El principio de funcionamiento del radar Secundario Modo S básico es similar al MSSR, en lo único que cambia es en el tipo de información que este puede proporcionar, que es mayor que el MSSR y la forma de interrogar de manera selectiva.

La información básica que proporciona un SSR en Modo S además del rango y acimut es:

- Altitud
- Código transponder
- Velocidad del avión
- Evolución de la altitud (se rebaja de 100 pies a 25 pies)

Cuando el SSR en Modo S envía la interrogación (a una frecuencia de 1030 MHz), el transponder de la aeronave la examina para verificar su propia dirección y tras reconocerla, transmite la respuesta (a una frecuencia de 1090 MHz) oportuna. Por su parte, las restantes aeronaves de la zona ignorarán toda interrogación que no vaya dirigida a ellas. Con esta tecnología las interrogaciones y las respuestas de los aviones se secuencian de forma individual, con lo que los problemas de respuestas solapadas (FRUIT y Garbling) son eliminados. La **Figura 1.31** muestra la compatibilidad entre los Modos A/C y el Modo S del SSR.

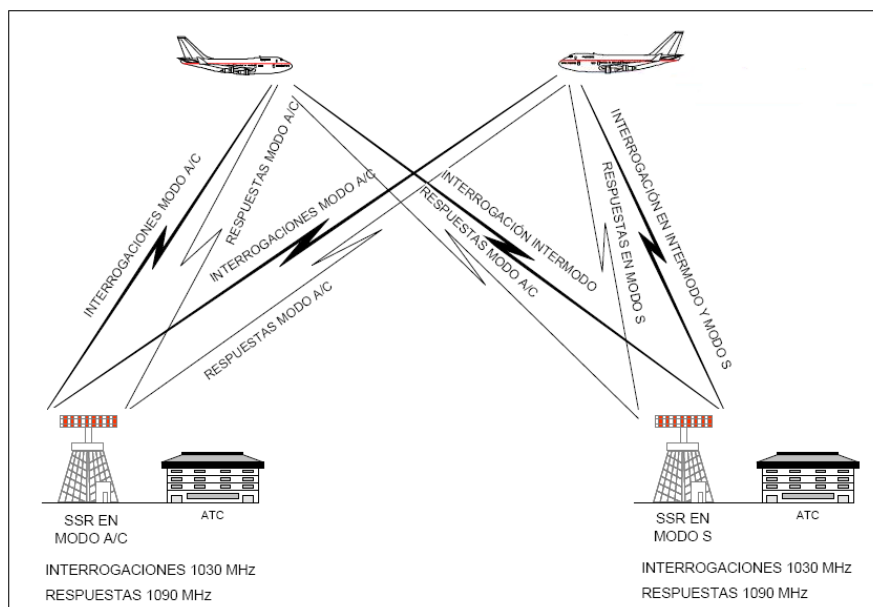


Figura 1.31: Compatibilidad entre los Modos A/C y el Modo S del SSR

A nivel operativo, el Modo S permite al ATC conocer y trabajar con: velocidad indicada, velocidad sobre el suelo, rumbo, régimen de ascenso o descenso y el nivel que mantiene o al que está ascendiendo o descendiendo un avión sin tener que preguntárselo al piloto. Esto no solo ahorra una gran cantidad de tiempo, sino también carga de trabajo en la forma reducción de comunicaciones entre pilotos y controladores, especialmente útil en la fase de aproximación.

1.6 ORGANISMOS NORMATIVOS DE SISTEMAS RADAR

La UIT (Unión Internacional de Telecomunicaciones) es la que dicta el Reglamento de Radiocomunicaciones, es decir asigna las frecuencias en las cuales van a trabajar determinados sistemas radioeléctricos. La UIT establece los procesos para el uso de las frecuencias para los sistemas radioeléctricos aeronáuticos.

La Organización de la Aviación Civil Internacional es actualmente una agencia especializada dependiente de la ONU que tiene como objetivos reforzar el aspecto civil de la aeronáutica y facilitar el desarrollo homogéneo del sistema de navegación aérea.

Una de las primeras tareas que acomete la OACI es la normalización de los procedimientos de control del tráfico aéreo, el diseño y operación de radioayudas y el uso adecuado de las frecuencias del espectro aeronáutico. Los documentos de la OACI se organizan en Anexos al Convenio. Cada Anexo trata un tema diferente y todos ellos tienen una orientación técnica y normativa. A la fecha son 18, el que se refiere a los sistemas radar es el Anexo 10 de Telecomunicaciones Aeronáuticas Volumen IV (Radares de Vigilancia y Sistemas para Evitar Colisiones)

La OACI desarrolla importantes proyectos y programas que buscan estimular el desarrollo seguro, ordenado y eficiente de la aviación civil internacional. Entre estos proyectos se mencionan:

- FANS (Future Air Navigation System): El grupo de trabajo *Sistema Futuro de Navegación Aérea* fue creado en el año 1985 con la misión de buscar la manera de utilizar técnicas basadas en los satélites para mejorar las comunicaciones, navegación y vigilancia. El desarrollo del sistema Galileo por parte de la Unión Europea es un punto básico para el desarrollo de este programa.
- FEATS (Future European Air Traffic System): El Sistema Futuro de Tráfico Aéreo Europeo es la adaptación del FANS a las condiciones europeas.

En la actualidad, EUROCONTROL desarrolla diversas actividades a través de los diferentes emplazamientos físicos donde se encuentran sus dependencias. La CEAC ha decidido pasarle a EUROCONTROL el desarrollo de sus programas (EATCHIP, APATSI, etc.). Durante un tiempo los pilares básicos de EUROCONTROL pasaron a ser la estandarización del equipamiento y la percepción de tarifas debido a la negativa de los ingleses, franceses y alemanes a que EUROCONTROL gestione su espacio aéreo en los años 70.

FAA es la Autoridad Federal de Aviación y es un órgano del gobierno de Estados Unidos que supervisa la industria de la aviación tanto para regularla como para promoverla. Forma parte del Departamento de Transporte y por eso es también responsable de la seguridad de la aviación civil.

Entre las funciones de la FAA podemos mencionar que promueve la seguridad, fomenta los avances de la tecnología aeronáutica, controla el tráfico aéreo, tanto para los aviones civiles como para los militares, investiga el desarrollo de la aeronáutica, estudia las formas de

reducir el ruido de las aeronaves y de alteración del medio ambiente y regula el espacio aéreo del territorio de Estados Unidos para evitar colisiones.

1.7 CONCEPTO DE INTEGRACION DE INFORMACION O SISTEMAS ABIERTOS

Cuando se comenzó la fabricación y comercialización de los primeros radares cada fabricante de radares, implementaba tecnologías cerradas, de tal forma que el concepto de integración de sistemas era muy difícil de realizar; es decir que cada radar que era fabricado tenía su propio protocolo de transmisión de datos que no era el mismo para otros sistemas radar. Esto cambió con el surgimiento de nuevas normas de parte de los organismos reguladores de este tipo de servicios como son OACI, UIT, EUROCONTROL y FAA, de modo que se han desarrollado códigos “abiertos” de transmisión de datos para sistemas radar, con lo cual surge el concepto de sistemas abiertos. Uno de los códigos abiertos más utilizados a nivel mundial es el protocolo ASTERIX de Eurocontrol, el cual facilita la integración de sistemas y el intercambio de datos radar entre diferentes regiones, como por ejemplo Europa, lo que fortalece la seguridad del control de tráfico aéreo.

1.8 PARAMETROS PARA ESTABLECER LA CALIDAD DE UN SISTEMA RADAR

Alta confianza de las señales en Modos A y C significa que no se reciben dos respuestas de una misma aeronave al mismo tiempo que genere respuestas solapadas (garbling y FRUIT). La **Figura 1.32** ilustra las respuestas entrelazadas que genera el fenómeno de garbling.

Garbling es cuando dos o más respuestas (correspondientes a peticiones de la estación a distintos blancos) se solapan en el tiempo en la recepción.

FRUIT ocurre cuando un blanco responde a una petición, que no iba dirigida a él, de otros interrogadores localizados en una estación terrena cercana.

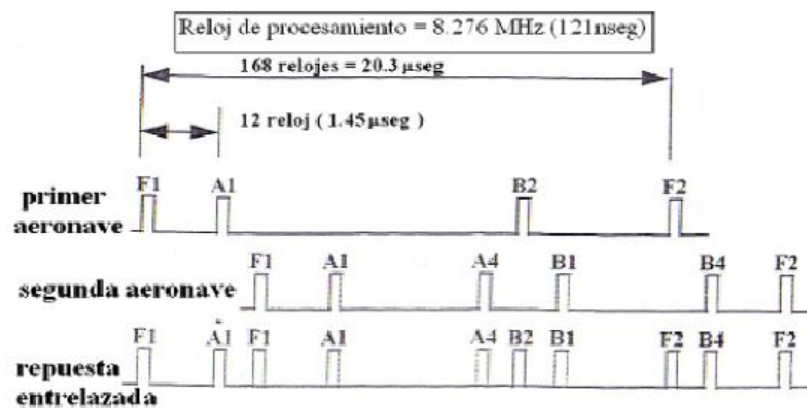


Figura 1.32: Respuestas entrelazadas (garbling)

Entre los parámetros para establecer la calidad de servicio de un sistema radar tenemos:

- **Probabilidad de Detección:** es la probabilidad de que por cada barrido de antena se produzca al menos una respuesta de blanco radar, con por lo menos datos de posición. A manera de cálculo podemos decir, que es la razón entre el número de detecciones validas de un blanco radar entre el número total de detecciones de ese blanco radar, esta definición es para un solo blanco individual. La Probabilidad de Detección Total, se define como la sumatoria de las Probabilidades de Detección individuales de los blancos radar considerados entre el número total de blancos radar. Según la OACI la Probabilidad de Detección para un Radar Primario debe ser como mínimo del 95 % y para un Radar Secundario del 97 %.
- **Falsos Blancos:** se espera que el número total de falsos blancos para un sistema radar sea el menor posible, en este caso se espera que sea de al menos menor que uno por barrido de antena.
- **Análisis de precisión:** se refiere al grado de precisión que debe tener un sistema radar en cuanto a rango y acimut. Para este tipo de pruebas se utiliza normalmente un transponder fijo, el objetivo es calibrar de una manera precisa la medición de rango y acimut del sistema radar.
- **Confianza de los Modos 3/A y C:** el nivel de alta confianza para un sistema radar en cuanto a la identificación de la aeronave mediante un código transponder se refiere (Modo 3/A) debe ser mayor al 99 %, en el caso de la altitud de la aeronave como es menos confiable que el rango y acimut, es la que generalmente presenta baja confianza y se espera que también sea al menos del 99 % o mayor.

1.9 ALGUNOS REQUISITOS DE OACI PARA SISTEMAS DE RADAR CIVILES

- La Probabilidad de Detección para un Radar Primario debe ser como mínimo del 95 % y para un Radar Secundario del 97 %.
- El PRF de transmisión del Radar Secundario debe de estar entre 50-450Hz (Típico 190 Hz, Máximo 282 Hz para 250 MN).
- La Potencia de transmisión del Radar Secundario debe ser por lo menos 500 watts (57 dbm).
- La Potencia de transmisión de un transponder de Radar Secundario debe estar entre 125 y 500 watts (21 a 27 dbW).
- El Radar Secundario debe ser capaz de reconocer de manera inmediata los códigos de emergencia 7500, 7600 y 7700 del Modo A.
- El radar Secundario debe ser capaz de procesar un mínimo de 1000 respuestas por segundo (lo recomendado es 1200 respuestas por segundo).
- Alta confianza de Código en Modo A y C para un Radar Secundario, de al menos 99 %.

1.10 BIBLIOGRAFIA

<http://www.tecnoradar.es/tiposderadares.html>

<http://www.24flotilla.com/foro/>

http://seminarios.iua.edu.ar/material/29_abril_exposicion_3.pdf

http://www.icao.int/icao/en/anb/cns/wrc2003/icao_itu_wrc2003_sp.pdf

http://www.prensalibre.com/internacionales/Salvador-primer-centroamericano-radares-climaticos_0_370163183.html

<http://www.aac.gob.sv/>

<http://www.aeropuertoelsalvador.gob.sv/>

<http://www.oni.escuelas.edu.ar/2008/CORDOBA/1324/trabajo/radarmeteorologico.html>

<http://www.scribd.com/doc/3024316/Funciones-de-RADAR>

http://www.comsoft.aero/download/atc/product/spanish/asterix_spanish.pdf

<http://www.leonart.tve.es/071207/ciencia/pdf/ciencia.pdf>

<http://www.milespps.com/36375/principios-basicos-del-radar/>

http://www.uclm.es/gestionaeronautica/pdf/programas/2curso/Fundamentos_Navegacion_Aerea.pdf

<http://aero.us.es/na/files0809/T2NA.pdf>

http://es.wikipedia.org/wiki/Navegaci%C3%B3n_a%C3%A9rea

http://www.atcevents.com/atc10/Website/PDF/INDRA_SURVEILLANCE_PSR.pdf

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/soriano_m_jc/apendiceJ.pdf

www.11-share.com/databank/descarga.php?file_id=87

www.11-share.com/databank/descarga.php?file_id=92

www.11-share.com/databank/descarga.php?file_id=92

<http://rt001282.eresmas.net/infovuelo.com/trepas.html>

<http://www.cfired.org.ar/Default.aspx?nId=7618>

edutechne.com.ar/forumsur/anteparabolic.doc

<http://will.zoomblog.com/archivo/2006/03/06/radar.html>

<http://www.radartutorial.eu/07.waves/wa04.es.html>

http://ec.digaden.edu.mx/cesnav/moodldata/34/CAP.5/radar_secundario_UNO.pdf

http://biblioteca.usac.edu.gt/tesis/08/08_0153_EO.pdf

<http://www.tsc.uniovi.es/Rad/Trabajos/Radar%20Secundario.pdf>

<http://www.radartutorial.eu/06.antennas/an05.en.html>

Secondary Surveillance Radar, Michael C. Stevens

Anexo 10, Telecomunicaciones Aeronáuticas, Volumen 4, Sistemas de vigilancia y anticolidión, 4ta edición, julio 2007.

Doc. 9684, Manual sobre el sistema del radar secundario de vigilancia (SSR), OACI, Tercera edición, 2004

CAPITULO II

PROTOCOLO ASTERIX DE EUROCONTROL

Introducción

Hace algunos años, cada fabricante de radares, implementaba tecnologías cerradas, de tal forma que el concepto de integración de sistemas era muy difícil de realizar; esto cambió con el surgimiento de nuevas normas de parte de los organismos reguladores de este tipo de servicios (OACI, EUROCONTROL y FAA), de modo que se han desarrollado códigos “abiertos” de transmisión de datos para sistemas radar. Uno de los códigos abiertos más utilizados a nivel mundial es el Asterix de Eurocontrol.

El Protocolo Asterix de Eurocontrol es un protocolo estándar diseñado por EUROCONTROL para el intercambio de información entre sensores radar y centros de control (ATC Systems) mediante una estructura de mensajes. Se definió a fines de los ochenta en Europa.

Su acrónimo corresponde con “**All Purpose S**tructure **E**urocontrol **S**urveillance **I**nformation **E**xchange”, que significa Estructura Todo Propósito para el Intercambio de Información de Vigilancia de Eurocontrol, siendo su desarrollo responsabilidad del Surveillance Task Force for Radar Data Exchange de Eurocontrol (STFRDE). Corresponde a este organismo, la asignación y publicación relativa al contenido de determinados campos. Dicho protocolo codifica la información relacionada con las aeronaves detectadas por un radar, dentro de una trama de transmisión de datos que perfectamente puede ser transmitida en una red tipo LAN.

En el presente capítulo se mostrará la importancia que tiene el Protocolo Asterix de Eurocontrol, debido a que facilita la integración de sistemas y el intercambio de datos radar entre diferentes regiones, lo que fortalece la seguridad del control de tráfico aéreo. Se definirá: sus objetivos; organización de los datos Asterix; la estructura del mensaje; identificación de la fuente de datos; descripción de las categorías 001 y 002; la aplicación de Asterix a los radares INDRA y ALENIA; como es el envío de datos radar; captura de datos Asterix con Wireshark; ejemplo de una trama Asterix y una breve descripción de analizadores comerciales de software libre (WRTADS y RBAT).

2.1 OBJETIVOS DEL PROTOCOLO ASTERIX DE EUROCONTROL

- ASTERIX permite implementar el fácil intercambio de datos de vigilancia dentro y entre países lo que siempre ha sido el principal objetivo del control de tráfico aéreo (ATC).
- ASTERIX es uno de los códigos abiertos más utilizados a nivel mundial de Eurocontrol, el cual facilita la integración de sistemas y el intercambio de datos Radar entre diferentes regiones, lo que fortalece la seguridad del control de tráfico aéreo.
- Su propósito es permitir una significativa transferencia de información entre dos entidades de aplicación usando una representación estándar de los datos que se intercambian.
- La transmisión de mensajes ASTERIX puede hacer uso de cualquier medio de comunicación disponible, por ejemplo. líneas seriales síncronas/asíncronas, LAN (TCP/IP, UDP/IP), etc.
- La definición de las capas inferiores de comunicaciones está fuera del alcance del estándar, las cuales serán acordadas entre las entidades de intercambio de datos.

2.2 ORGANIZACION DE LOS DATOS ASTERIX

La estructura ASTERIX parte de los siguientes conceptos básicos:

- Data Categories (Categorías de Datos)
- Data Item (Item de Datos)
- Data Field (Campo de Dato)
- User Application Profile (Perfil de Aplicación de Usuario)
- Data Block (Bloque de Dato)
- Registers (Registros)

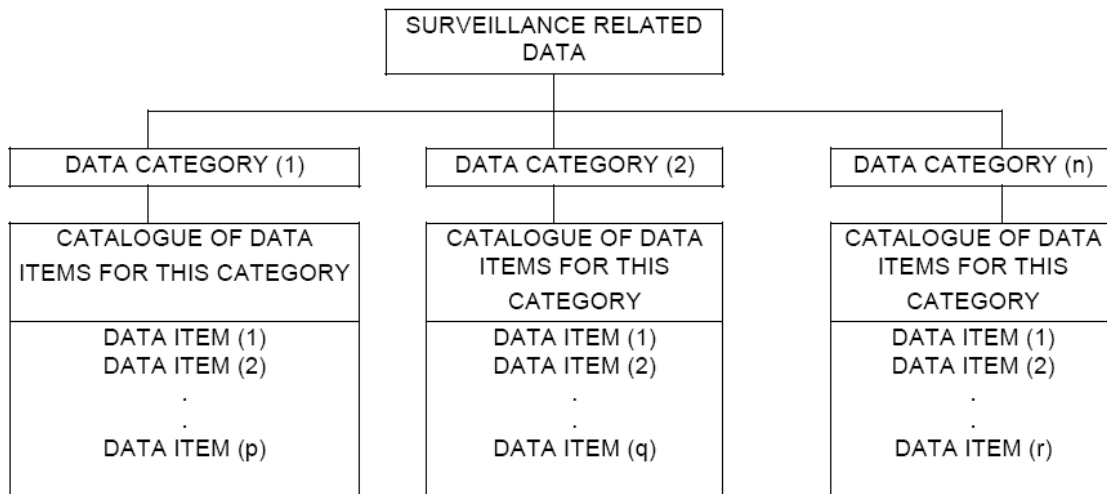


Figura 2.1: Estructura del Protocolo Asterix

	USER APPLICATION PROFILE							
	DATA FIELD (1)	DATA FIELD (2)	DATA FIELD (3)	DATA FIELD (4)	DATA FIELD (5)	DATA FIELD (6)	DATA FIELD (7)	DATA FIELD (8)
DATA ITEM (1)		x						
DATA ITEM (2)						x		
DATA ITEM (3)	x							
DATA ITEM (4)				x				
DATA ITEM (5)								
...								
DATA ITEM (x)			x					
DATA ITEM (x + 1)								x
...								
DATA ITEM (q)					x			

Tabla 2.1: Organización de los Datos del Protocolo Asterix

2.2.1 Data Categories (Categorías de Datos)

- Los datos a intercambiar sobre un medio de comunicación entre diferentes usuarios deben ser estandarizados y clasificados en categorías.
- Es la información que puede ser transmitida y codificada, cuyos datos serán estándar para todos los usuarios de Asterix.

El propósito de esta clasificación facilita la:

- Identificación de los datos.
- El envío de los datos.
- El establecimiento de una cierta jerarquía basado en su prioridad.

Pueden ser definidas hasta 256 categorías de datos, de acuerdo con la siguiente clasificación y su utilización:

- Categorías de 000 a 127 para aplicaciones civiles y militares estándar.
- Categorías de 128 a 240 reservadas para aplicaciones militares especiales.
- Categorías de 241 a 255 para aplicaciones civiles y militares no estándar .
- La lista actualizada de categorías es publicada en la página de Eurocontrol (Excepto especiales aplicaciones militares).

Data Categories (Ejemplos)

Categoría 001: Información de blancos radar desde una cabecera a un sistema de proceso de datos radar.

Categoría 002: Mensajes de Servicio Radar.

Categoría 008: Información monoradar de blancos meteorológicos.

Categoría 034: Nueva versión de Cat 002, SSR Modo S.

Categoría 048: Nueva versión de Cat 001 y Cat 016, SSR Modo S.

2.2.2 Data Item (Item de Datos)

- Es la unidad más pequeña de información de cada categoría.
- Para cada categoría se define un conjunto de Items de Datos que constituye el catálogo de Items de Datos.
- Aplicaciones que impliquen el intercambio de información de una determinada categoría de datos, harán uso únicamente de los Items de Datos normalizados en el catálogo.
- Cada Item de Datos tiene una referencia única que lo identifica de forma inequívoca.

La referencia simbólica consta de ocho caracteres de la forma siguiente:

I**nnn** / **AAA**, donde:

- **I**, indica que es un Item de Datos.
- **nnn**, es un número de tres dígitos decimales que indica la categoría de datos a la que pertenece.
- **AAA**, es un número de tres dígitos decimales que indica el número de Item de Datos.

Ejemplo:

I048/010 Data Source Identifier

2.2.3 Data Field (Campo de Dato)

- Es la implementación física de un Item de Datos para propósitos de comunicaciones, es decir cada Item de Datos es asignado a un slot llamado Data Field.
- Tienen una longitud igual a un número entero de octetos y tiene asignado un único Field Reference Number (FRN).
- La correspondencia entre los Item de Datos y Data Fields se estandariza para cada aplicación por medio del User Application Profile.

2.2.4 User Application Profile (UAP, Perfil de Aplicación de Usuario)

- Es el mecanismo por el cual se estandariza para cada aplicación la correspondencia entre los Item de Datos y los Data Fields haciendo uso de la estructura de los mensajes ASTERIX.
- La UAP es una tabla de control vinculada al programa de empaquetado/desempaquetado de los sistemas de procesamiento de mensajes.
- En esencia define cuáles de los Item de Datos catalogados serán utilizados, su longitud, su asignación a los Data Fields y los requisitos específicos que deben ser normalizados para la transmisión exitosa e interpretación de los mensajes.
- Con este mecanismo es fácil optimizar la eficiencia de la transmisión.
- Con este mecanismo es fácil optimizar la transmisión, sin necesidad de modificar el programa, teniendo en cuenta la frecuencia de aparición de determinados Item de Datos.
- Además, permite flexibilidad de elección entre las distintas representaciones lógicas de la información.
- Los spare bits del UAP debieran ser puestos a cero.
- La UAP será única por cada categoría.

Eurocontrol ha estipulado para las distintas categorías un estándar parametrizable, es decir, cada usuario define los campos que son obligatorios, los que son optativos o los no utilizables.

2.3 ESTRUCTURA DEL MENSAJE

2.3.1 Data Block (Bloque de Dato)

- Es una unidad de información que contiene uno o más registros, todos ellos con información de la misma categoría.

Está compuesto por:

- Un octeto de datos llamado Categoría (**CAT**), indicando a qué categoría pertenecen los datos transmitidos,
- Un campo de 2 octetos que indican el tamaño del bloque (**LEN**), incluyendo los campos de CAT y LEN .
- Uno o más registros que contienen los datos de la misma categoría.
- Cada registro es de longitud variable pero con límite de octetos definido. La longitud siempre será un múltiplo de un octeto.
- El tamaño máximo de un bloque de datos será de mutuo acuerdo entre las fuentes de datos y usuarios.

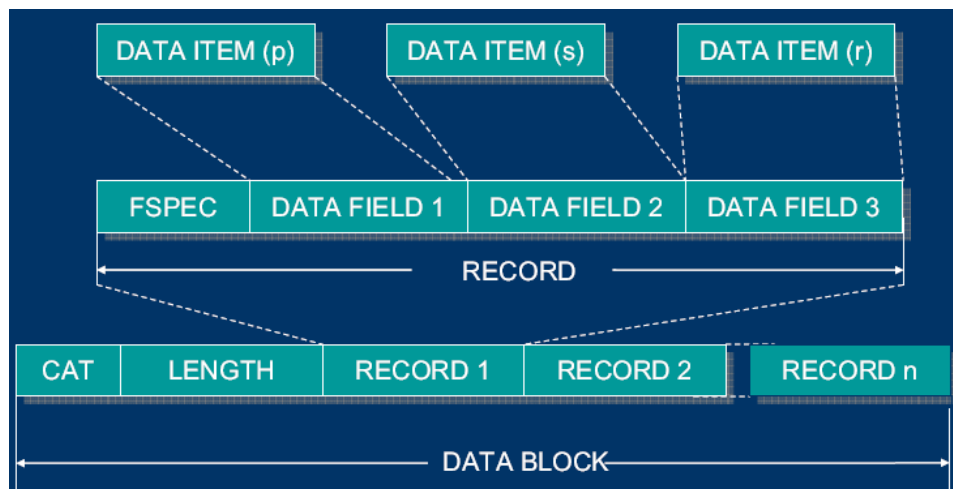


Figura 2.2: Estructura de un Bloque de Datos Asterix

2.3.2 Registers (Registros)

- Es un conjunto ordenado de Data Field de la misma categoría transmitidos dentro de un data block, el cual esta precedido por el campo Field Specification (FSPEC), donde se indica los Items de Datos presentes de acuerdo con el UAP.

Un registro está constituido por:

- Un campo de Field Specification (FSPEC) de longitud variable, que indica los Data Fields que están presentes en el registro y su orden.

- Un número variable de Data Fields, asociado con un Item de Datos, identificado por el Field Referencial Number (FRN).

Existe por otra parte, una publicación del User Application Profile que es el procedimiento que asigna Item de Datos a Data Block.

El tamaño de los Data Fields puede ser fijo o variable dependiendo de los siguientes:

- Longitud fija campos de datos, incluyen un número fijo de octetos.
- Longitud extendida, por ser de longitud variable deberá contener una parte fundamental indicando su longitud seguida inmediatamente por un número de partes secundarias, cada una de longitud predeterminada. La presencia de los campos se indicará mediante el establecimiento de un bit (menos significativo 1) del último octeto, llamado indicador de extensión de campo.
- Longitud de los campos de datos explícitos se iniciará con un solo indicador de longitud del octeto dando la longitud total del campo incluyendo el mismo.
- Campos de datos repetitivos, comprenderá un campo de un octeto de repetición Indicador para señalar la presencia de N Subcampos consecutivos, cada uno de la misma longitud predeterminada.
- Campos de datos compuestos estará integrado por un subcampo primario, seguida de datos subcampos. El subcampo primario determina la presencia o ausencia de los subcampos de datos posterior. Consta de una primera parte de un octeto extensible utilizando el mecanismo de extensión de campo. La definición, estructura y formato de los subcampos de datos son parte de la descripción de los correspondientes datos compuestos de elementos.

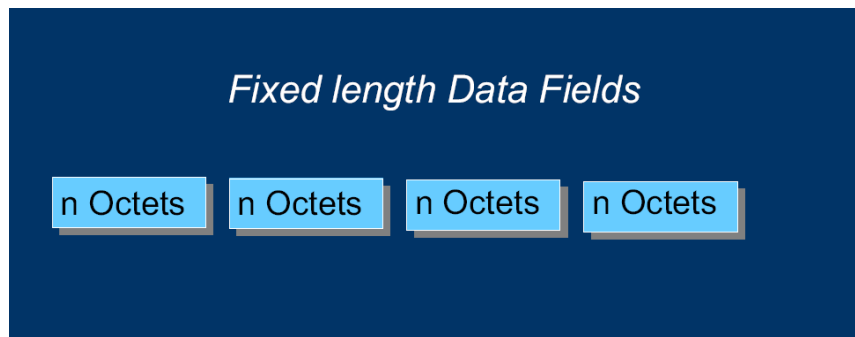


Figura 2.3: Estructura de campos de datos de longitud fija

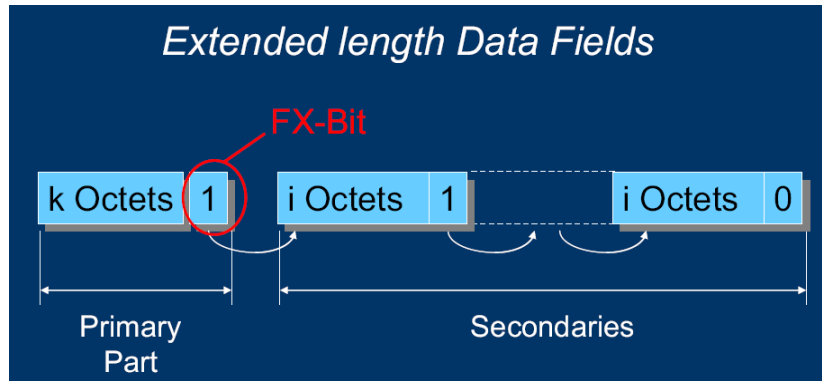


Figura 2.4: Estructura de campos de datos de longitud extendida

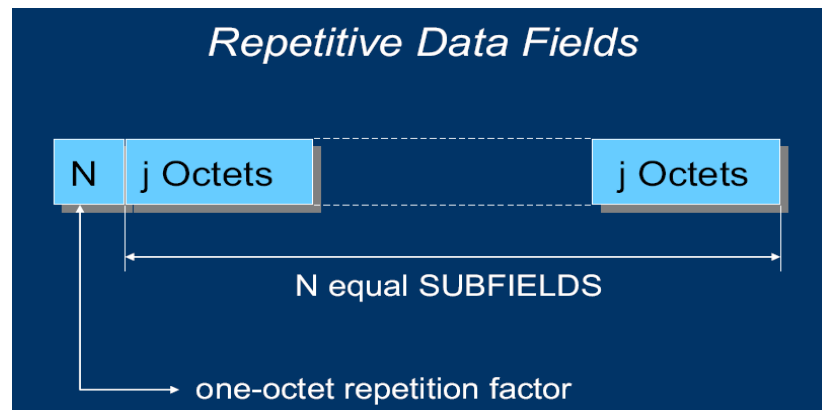


Figura 2.5: Estructura de campos de datos repetitivos

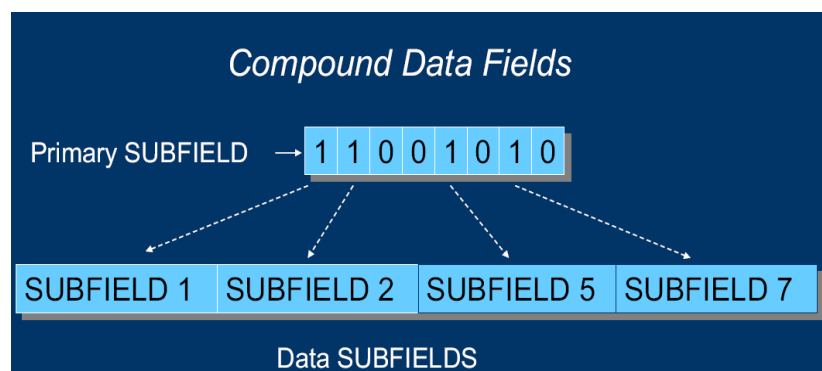


Figura 2.6: Estructura de campos de datos compuestos

El encadenamiento de los campos dentro de un registro, se puede hacer mediante una Organización Secuencial Ordered Field Sequencing (OFS) o una Organización aleatoria Random Field Sequencing (RFS).

2.3.3 Organización Secuencial

La OFS es el método estándar y consiste en utilizar el FSPEC como una tabla de contenidos secuenciales en forma de bits, donde cada bits indica la presencia o ausencia de un Data Field asignado.

El FSPEC es un número variable de Data Fields en orden creciente de FRN. La relación entre los bits del FSPEC, los Data Fields y los Item de Datos está en el UAP.

2.3.4 Field Specification (FSPEC)

El FSPEC es una tabla de contenidos en forma de secuencia de bits donde cada bits indica la presencia o ausencia de un Data Field determinado.

Existe la posibilidad de utilizar Data Field no estándar. Para ello se habilita un bit que indica la presencia de un propósito especial (SP). Por otra parte y en este mismo campo, otro bit indica la utilización de una organización Aleatoria (RFS).

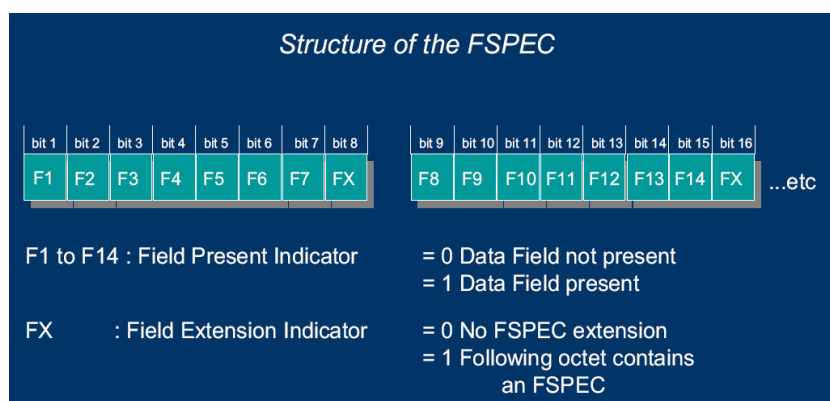


Figura 2.7: Estructura de la especificación de campo (FSPEC)

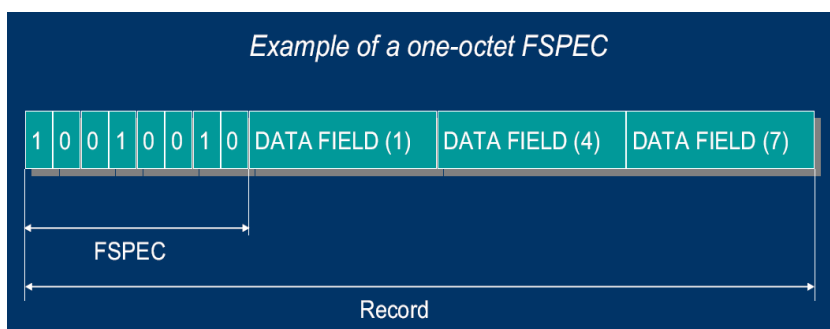


Figura 2.8: Ejemplo de un octeto FSPEC

State/Territory Estado/Territorio	SAC Code Format Formato Código SAC								Hexadecimal SAC Code Código SAC Hexadecimal
	B7	B6	B5	B4	B3	B2	B1	B0	
Anguilla (United Kingdom)	1	1	1	0	0	0	0	0	E0
Antigua & Barbuda	1	1	1	0	0	0	0	1	E1
Argentina	1	1	1	0	0	0	1	0	E2
Aruba (Netherlands)	1	1	1	0	0	0	1	1	E3
Bahamas	1	1	1	0	0	1	0	0	E4
Barbados	1	1	1	0	0	1	0	1	E5
Belize	1	1	1	0	0	1	1	0	E6
Bolivia	1	1	1	0	0	1	1	1	E7
Brasil	1	1	1	0	1	0	0	0	E8
Cayman Islands (United Kingdom)	1	1	1	0	1	0	0	1	E9
Chile	1	1	1	0	1	0	1	0	EA
Colombia	1	1	1	0	1	0	1	1	EB
Costa rica	1	1	1	0	1	1	0	0	EC
Cuba	1	1	1	0	1	1	0	1	ED
Dominica	1	1	1	0	1	1	0	1	EE
Dominican Republic	1	1	1	0	1	1	1	1	EF
Ecuador	1	1	1	1	0	0	0	0	F0
El Salvador	1	1	1	1	0	0	0	1	F1

Tabla 2.2: Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM

Guadeloupe, French Antilles (France)	0	0	0	0	1	0	0	0	08
Martinique, French Antilles (France)	0	0	0	0	1	0	0	0	08
French Guiana (France)	0	0	0	0	1	0	0	0	08
Grenada	1	1	1	1	0	1	0	1	F5
Guatemala	1	1	1	1	0	1	1	0	F6
Guyana	1	1	1	1	0	1	1	1	F7
Haiti	1	1	1	1	1	0	0	0	F8
Honduras	1	1	1	1	1	0	0	1	F9
Jamaica	1	1	1	1	1	0	1	0	FA
Mexico	1	1	1	1	1	0	1	1	FB
Montserrat (United Kingdom)	1	1	1	1	1	1	0	0	FC
Netherland Antilles (Netherlands)	1	1	1	1	1	1	0	1	FD
Nicaragua	1	1	1	1	1	1	1	0	FE
Panama	1	1	1	1	1	1	1	1	FF
Paraguay	1	1	0	1	0	0	0	0	D0
Peru	1	1	0	1	0	0	0	1	D1

Tabla 2.3: Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM

Puerto Rico (United States)	1	1	0	1	0	0	0	1	D2
Saint Kitts and Nevis	1	1	0	1	0	0	1	1	D3
Saint Lucia	1	1	0	1	0	1	0	0	D4
Saint Vincent and Grenadines	1	1	0	1	0	1	0	1	D5
Saint Maarten	1	1	0	1	0	1	1	0	D6
Suriname	1	1	0	1	0	1	1	1	D7
Tortola	1	1	0	1	1	0	0	0	D8
Trinidad and Tobago	1	1	0	1	1	0	0	1	D9
Turks and Caicos Is. (United Kingdom)	1	1	0	1	1	0	1	0	DA
United States (For sharing with CAR Region)	1	1	0	1	1	0	1	1	DB
Uruguay	1	1	0	1	1	1	0	0	DC
Venezuela	1	1	0	1	1	1	0	1	DD
Virgin Island (United Kingdom)	1	1	0	1	1	1	1	0	DE
Virgin Island (United States)	1	1	0	1	1	1	1	1	DF

Tabla 2.4: Plan de asignación de código SAC ASTERIX para las regiones CAR/SAM

2.5 DESCRIPCION DE LA CATEGORIA 001

Es aquella donde la información de blancos radar va desde una cabecera a un sistema de proceso de datos radar.

Se pueden transmitir mensajes de plots o pistas, o una combinación de ambos. El flujo de información es unidireccional, de la cabecera a los sistemas de proceso radar.

En la **Figura 2.10** se observa la estructura de un bloque de datos Asterix y en la **Figura 2.11** los Items de Datos estándar de categoría 001.

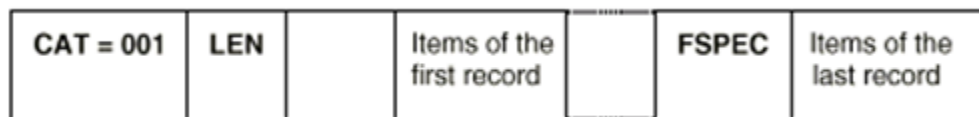


Figura 2.10: Bloque de Datos de categoría 001

Data Item Ref. No.	Description	System Units
I001/010	Data Source Identifier	N.A.
I001/020	Target Report Descriptor	N.A.
I001/030	Warning/Error Conditions	N.A.
I001/040	Measured Position in Polar Coordinates	RHO: 1/128 NM THETA: $360^\circ/(2^{16})$
I001/042	Calculated Position in Cartesian Coordinates	X, Y: 1/64 NM
I001/050	Mode-2 Code in Octal Representation	N.A.
I001/060	Mode-2 Code Confidence Indicator	N.A.
I001/070	Mode-3/A Code in Octal Representation	N.A.
I001/080	Mode-3/A Code Confidence Indicator	N.A.
I001/090	Mode-C Code in Binary Representation	1/4 FL
I001/100	Mode-C Code and Code Confidence Indicator	N.A.
I001/120	Measured Radial Doppler Speed	(2^{-14}) NM/s
I001/130	Radar Plot Characteristics	N.A.
I001/131	Received Power	1 dBm
I001/141	Truncated Time of Day	1/128 s
I001/150	Presence of X-Pulse	N.A.
I001/161	Track/Plot Number	N.A.
I001/170	Track Status	N.A.
I001/200	Calculated Track Velocity in Polar Coordinates	Speed: (2^{-14}) NM/s Heading: $360^\circ/(2^{16})$
I001/210	Track Quality	N.A.

Tabla 2.5: Items de Datos estándar de categoría 001

Así tenemos el UAP Standard para información de plot el cual se aprecia en la **Tabla 2.6** y el UAP para pistas se aprecia en la **Tabla 2.7**.

FRN	Data Item	Information	Length in Octets
1	I001/010	Data Source Identifier	2
2	I001/020	Target Report Descriptor	1+
3	I001/040	Measured Position in Polar Coordinates	4
4	I001/070	Mode-3/A Code in Octal Representation	2
5	I001/090	Mode-C Code in Binary Representation	2
6	I001/130	Radar Plot Characteristics	1+
7	I001/141	Truncated Time of Day	2
FX	-	Field Extension Indicator	-
8	I001/050	Mode-2 Code in Octal Representation	2
9	I001/120	Measured Radial Doppler Speed	1
10	I001/131	Received Power	1
11	I001/080	Mode-3/A Code Confidence Indicator	2
12	I001/100	Mode-C Code and Code Confidence Indicator	4
13	I001/060	Mode-2 Code Confidence Indicator	2
14	I001/030	Warning/Error Conditions	1+
FX	-	Field Extension Indicator	-
15	I001/150	Presence of X-Pulse	1
16	-	Spare	-
17	-	Spare	-
18	-	Spare	-
19	-	Spare	-
20	-	Reserved for SP Indicator	-
21	-	Reserved for Random Field Sequencing (RFS) Indicator (RS-bit)	-
FX	-	Field Extension Indicator	-

Tabla 2.6: UAP estándar para información de Plot

FRN	Data Item	Information	Length in Octets
1	I001/010	Data Source Identifier	2
2	I001/020	Target Report Descriptor	1+
3	I001/161	Track/Plot Number	2
4	I001/040	Measured Position in Polar Coordinates	4
5	I001/042	Calculated Position in Cartesian Coordinates	4
6	I001/200	Calculated Track Velocity in polar Coordinates	4
7	I001/070	Mode-3/A Code in Octal Representation	2
FX	-	Field Extension Indicator	-
8	I001/090	Mode-C Code in Binary Representation	2
9	I001/141	Truncated Time of Day	2
10	I001/130	Radar Plot Characteristics	1+
11	I001/131	Received Power	1
12	I001/120	Measured Radial Doppler Speed	1
13	I001/170	Track Status	1+
14	I001/210	Track Quality	1+
FX	-	Field Extension Indicator	-
15	I001/050	Mode-2 Code in Octal Representation	2
16	I001/080	Mode-3/A Code Confidence Indicator	2
17	I001/100	Mode-C Code and Code Confidence Indicator	4
18	I001/060	Mode-2 Code Confidence Indicator	2
19	I001/030	Warning/Error Conditions	1+
20	-	Reserved for Special Purpose Indicator (SP)	-
21	-	Reserved for RFS Indicator (RS-bit)	-
FX	-	Field Extension Indicator	-
22	I001/150	Presence of X-Pulse	1

Tabla 2.7: UAP estándar para información de Pista

2.6 DESCRIPCION DE ITEMS DE DATOS ESTANDAR DE LA CATEGORIA 001

2.6.1 ITEM I001/010 Identificación Fuente de Datos

Definición: Identificación de la estación radar de la cual son recibidos los datos.

Formato: Item de Datos de dos octetos de longitud fija.

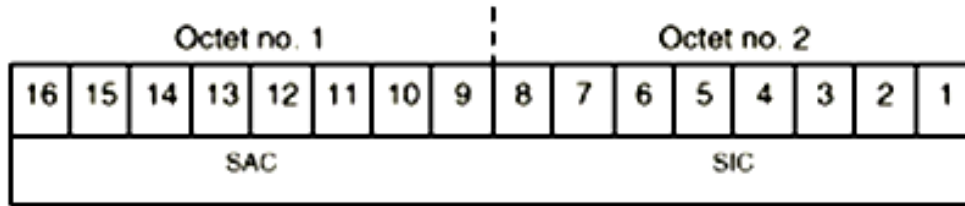


Figura 2.11: Estructura de los dos octetos de longitud fija del Item de datos I001/010

2.6.1.1 Descripción de los bits del Item de Datos I001/010

Bits-16/9 (SAC) = Sistema de código de área

Bits-8/1 (SIC) = Sistema de código de identificación

2.6.2 ITEM I001/020 Descripción del informe de blanco

Definición: Tipo y características del dato radar es transmitido por una estación radar.

Formato: Item de Dato de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto si es necesario.

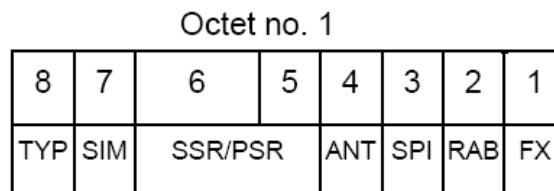


Figura 2.12: Estructura de los dos octetos de longitud fija del Item de datos I001/020

2.6.2.1 Descripción de los bits de la primera parte del Item de Datos I001/020

Bit-8 (TYP) = 0 Plot

= 1 Pista

Bit-7 (SIM) = 0 Plot o Pista actual

= 1 Plot o Pista simulada

Bits-6/5 (SSR/PSR) Detección radar en la última vuelta de antena, es como sigue:

= 00 Ninguna detección

- = 01 Detección solo radar primario
- = 10 Detección solo radar secundario
- = 11 Detección combinada radar primario y radar secundario

Bit-4 (ANT) = 0 Informe de blanco de antena 1

= 1 Informe de blanco de antena 2

Bit-3 (SPI) = 0 Por Defecto

= 1 Identificación de posición especial

Bit-2 (RAB) = 0 Por Defecto

= 1 Plot o Pista de un transponder fijo

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

Nota: Bit-7 (SIM) es usado para identificar un informe de blanco simulado como el producido por un simulador de tráfico.

Octet no. 1

8	7	6	5	4	3	2	1
TST	DS1/DS2	ME	MI	0	0	0	FX

Figura 2.13: Estructura del octeto de primera extensión del Item de datos I001/020

2.6.2.2 Descripción de los bits de la primera extensión del Item de Datos I001/020

Bit-8 (TST) = 0 Por defecto

= 1 Indicador de blanco Test

Bits-7/6 (DS1/DS2) = 00 Por defecto

= 01 Interferencia ilícita (código 7500)

= 10 Falla de radio-comunicación (código 7600)

= 11 Emergencia (código 7700)

Bit-5 (ME) = 0 Por defecto

= 1 Emergencia militar

Bit-4 (MI) = 0 Por defecto

= 1 Identificación militar

Bits-3/2 Bits Spare puestos a cero

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

2.6.3 ITEM I001/030 Condiciones de Alarma y Error

Definición: Condiciones alarma/error detectado por una estación radar para el informe de blanco envuelto.

Formato: Item de Datos de longitud comprendido por una primera parte de un octeto, seguido por extensiones de un octeto si es necesario.

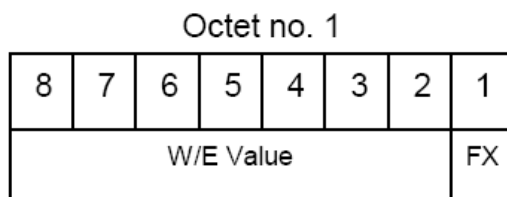


Figura 2.14: Estructura del octeto de primera parte del Item de datos I001/030

2.6.3.1 Descripción de los bits de la primera parte del Item de Datos I001/030

Bits-8/2 (W/E Value) Primer valor de condición alarma/error

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos (por ejemplo segundo valor de condición alarma/error)

Notas

1. Los valores de condición alarma/error de 0-63 están reservados para uso común estándar, donde los valores de 64-127 son aplicación dependiente.
2. El siguiente set de valores comunes de W/E es definida:
 - W/E = 0 No hay condición de alarma ni error.
 - W/E = 1 Replica solapada.
 - W/E = 2 Reflección.
 - W/E = 3 Replica de lóbulo secundario.
 - W/E = 4 Plot cortado.
 - W/E = 5 Respuesta de segunda vez.
 - W/E = 6 Angeles.
 - W/E = 7 Vehículos terrestres.
3. El siguiente set de valores es dependiente de las aplicaciones W/E están reservados:
 - W/E = 64 Posible código erróneo en Modo-3/A.
 - W/E = 65 Posible información errónea de altitud, transmitido con la credibilidad código c chequeo de fallos junto con el código Modo-C en notación binaria.
 - W/E = 66 Posible plot fantasma MSSR.
 - W/E = 80 Plot fijo PSR.
 - W/E = 81 Plot lento PSR.
 - W/E = 82 Plot de baja calidad PSR.

2.6.4 ITEM I001/040 Posición medida en Coordenadas Polares

Definición: Posición medida de una aeronave en coordenadas polares local.

Formato: Item de Datos de longitud fija de cuatro octetos.



Figura 2.15: Estructura de los cuatro octetos del Item de datos I001/040

2.6.4.1 Descripción de los bits del Item de Datos I001/040

Bit-17 (LSB) = 1/128 NM Rango máximo = 512 NM

Bit-1 (LSB) = $360^\circ / (2^{16}) = 0.0055^\circ$

Nota: Cuando es expresado en 16 bits, con signo o sin signo el acimut tiene el mismo valor.

2.6.5 ITEM I001/042 Posición calculada en Coordenadas Cartesianas

Definición: Posición calculada de una aeronave en coordenadas cartesianas.

Formato: Item de Datos de longitud fija de cuatro octetos.

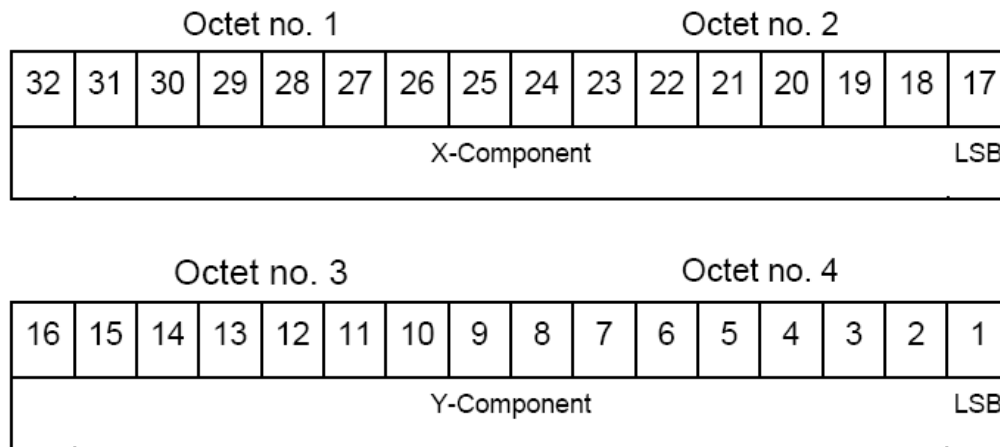


Figura 2.16: Estructura de los cuatro octetos del Item de datos I001/042

2.6.5.1 Descripción de los bits del Item de Datos I001/042

Bit-17 (LSB) = $2^{-(6+f)}$ donde f es el factor de escala aplicado, modificando la unidad de cuantización estándar. Rango máximo = $2^{(9+f)}$ NM.

Bit-1 (LSB) = $2^{-(6+f)}$. Rango máximo = $2^{(9+f)}$ NM.

Notas

1. Una unidad de cuantización por defecto, de 1/64 NM es obtenido para un valor de $f = 0$.
2. Valores negativos son expresados en la forma de complemento a dos, el bit-32 y el bit-16 debería ser puesto a 0 para valores positivos y 1 para valores negativos.

2.6.6 ITEM I001/050 Modo 2 en representación Octal

Definición: Respuesta a la interrogación de Modo-2.

Formato: Item de Datos de longitud fija de dos octetos.

Octet no. 1							Octet no. 2								
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
V	G	L	0	A4	A2	A1	B4	B2	B1	C4	C2	C1	D4	D2	D1

Figura 2.17: Estructura de los dos octetos del Item de datos I001/050

2.6.6.1 Descripción de los bits del Item de Datos I001/050

Bit-16 (V) = 0 Código valido

= 1 Código no valido

Bit-15 (G) = 0 Por defecto

= 1 Código solapado

Bit-14 (L) = 0 Código Modo-2 es derivado de la respuesta del transponder

= 1 Código Modo-2 Silencioso es provisto por un tracker local

Bit-13 Bit Spare puesto a 0

Bits-12/1 Código Modo-2 en representación octal.

Notas

1. Los datos Modo-2 silencioso (bit-14 puesto a uno) es usado cuando el plot contiene código que no es del Modo-2 o los códigos Modo-2 del plot y pista son diferentes.
2. Los Bits-16/15 no tienen significado en el caso de un Modo-2 silencioso y son puestos a 0 para una pista calculada.

2.6.7 ITEM I001/060 Indicador de confianza del Modo 2

Definición: Nivel de confianza para cada bit de una respuesta de Modo-2 es provisto por una estación SSR monopulso.

Formato: Item de Datos de longitud fija de dos octetos.

Octet no. 1								Octet no. 2							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	QA4	QA2	QA1	QB4	QB2	QB1	QC4	QC2	QC1	QD4	QD2	QD1

Figura 2.18: Estructura de los dos octetos del Item de datos I001/060

2.6.7.1 Descripción de los bits del Item de Datos I001/060

Bits-16/13 Bits spare puestos a 0

Bits-12/1 (QXi) = 0 pulso Xi de calidad alta

= 1 pulso Xi de calidad baja

Nota: Este Item de Datos es transmitido solo si por lo menos un pulso es de baja calidad.

2.6.8 ITEM I001/070 Modo 3/A en representación Octal

Definición: Código Modo-3/A convertido en representación octal.

Formato: Item de Datos de longitud fija de dos octetos.

Octet no. 1								Octet no. 2							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
V	G	L	0	A4	A2	A1	B4	B2	B1	C4	C2	C1	D4	D2	D1

Figura 2.19: Estructura de los dos octetos del Item de datos I001/070

2.6.8.1 Descripción de los bits del Item de Datos I001/070

Bit-16 (V) = 0 Código valido

= 1 Código no valido

Bit-15 (G) = 0 Por defecto

= 1 Código solapado

Bit-14 (L) = 0 Código Modo-3/A es derivado de la respuesta del transponder

= 1 Código Modo-3/A Smoothed es provisto por un tracker local

Bit-13 Bit Spare puesto a 0

Bits-12/1 Respuesta Modo-3/A en representación octal.

Notas

1. El detector de señales de un código solapado cuando en por lo menos dos respuestas están sobrepuestas. Los datos Modo-3/A silencioso (bit-14 puesto a uno) son usados en el caso de la ausencia del código Modo-3/A información en el plot, o en el caso de una diferencia entre el plot y pista información código Modo-3/A.
2. Los Bits-16/15 no tienen significado en el caso de un Modo-3/A silencioso y son puestos a 0 para una pista calculada.

2.6.9 ITEM I001/080 Indicador de confianza del Modo 3/A

Definición: Nivel de confianza para cada bit de una respuesta Modo-3/A es provisto por una estación SSR monopolso.

Formato: Item de Datos de longitud fija de dos octetos.

Octet no. 1								Octet no. 2							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	QA4	QA2	QA1	QB4	QB2	QB1	QC4	QC2	QC1	QD4	QD2	QD1

Figura 2.20: Estructura de los dos octetos del Item de datos I001/080

2.6.9.1 Descripción de los bits del Item de Datos I001/080

Bits-16/13 Bits Spare puestos a 0

Bits-12/1 (QXi) = 0 pulso Xi calidad alta

= 1 pulso Xi calidad baja

Nota: Este Item de Datos es transmitido solo si en por lo menos un pulso es de baja calidad.

2.6.10 ITEM I001/090 Código Modo C en representación Binaria

Definición: Altitud Modo-C convertido en representación binaria.

Formato: Item de Datos de longitud fija de dos octetos.

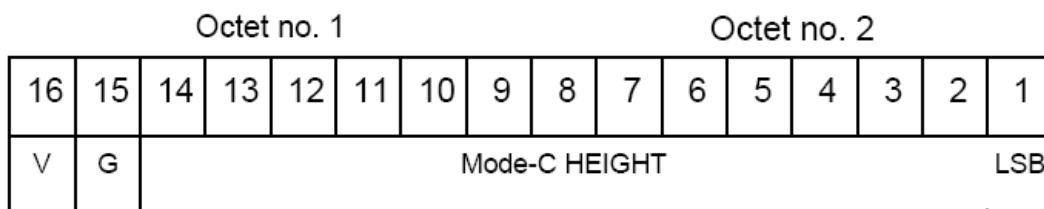


Figura 2.21: Estructura de los dos octetos del Item de datos I001/090

2.6.10.1 Descripción de los bits del Item de Datos I001/090

Bit-16 (V) = 0 Código valido

= 1 Código no valido

Bit-15 (G) = 0 Por defecto

= 1 Código solapado

Bit-1 (LSB) = 1/4 FL = 25 ft

Notas

1. El detector de señales de un código solapado se da cuando en por lo menos dos respuestas están sobrepuestas.
2. La máxima altitud que puede ser representada es 204775 ft. Prácticamente el máximo valor valido es 126750 ft (refiérase al Anexo 10 de la OACI).

3. Los valores negativos son expresados en la forma de complemento a dos, el bit-14 es puesto a 0 para valores positivos y 1 para valores negativos.

2.6.11 ITEM I001/100 Indicador de confianza del Modo C

Definición: Altitud Modo-C en notación Gray es recibido del transponder junto con el nivel de confianza por cada bit de respuesta es provisto por una estación SSR monopulso.

Formato: Item de Datos de longitud fija de cuatro octetos.

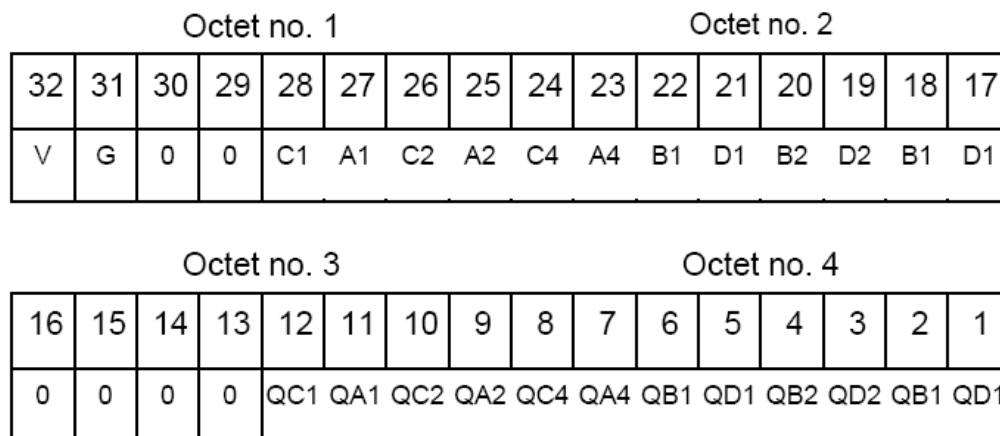


Figura 2.22: Estructura de los cuatro octetos del Item de datos I001/100

2.6.11.1 Descripción de los bits del Item de Datos I001/100

Bit-32 (V) = 0 Código valido

= 1 Código no valido

Bit-31 (G) = 0 Por defecto

= 1 Código solapado

Bits-30/29 Bits Spare puestos a 0

Bits-12/1 (QXi) = 0 pulso Xi calidad alta

= 1 pulso Xi calidad baja

Notas

1. Este Item de Datos es solamente transmitido si al menos un pulso es de baja calidad.
2. El detector de señales de un código solapado se da cuando en por lo menos dos respuestas están sobrepuestas.

2.6.12 ITEM I001/120 Velocidad Radial Doppler Medida

Definición: Componente radial de la velocidad tierra es medida por el significado de los bancos de filtro Doppler en procesadores de señal radar.

Formato: Item de Datos de longitud fija de un octeto.

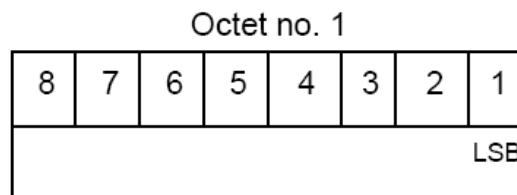


Figura 2.23: Estructura del octeto del Item de datos I001/120

2.6.12.1 Descripción de los bits del Item de Datos I001/120

Bit-1 (LSB) = $2^{(-14+f)}$ NM/s.

Notas

1. Una unidad de cuantización por defecto de 14.0625 kt y un máximo de +/- 1800 kt es obtenido para un valor de f=6.
2. Los valores negativos son expresados en la forma de complemento a dos, bit-8 es puesto a 0 para valores positivos y 1 para valores negativos.

2.6.13 ITEM I001/130 Características del Plot radar

Definición: Información adicional sobre la calidad del informe de blanco.

Formato: Item de Datos de longitud variable, comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesarias.

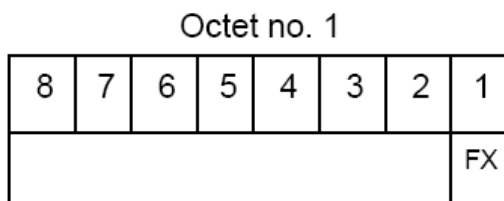


Figura 2.24: Estructura del octeto de la primera parte del Item de datos I001/130

2.6.13.1 Descripción de los bits del Item de Datos I001/130

Bits-8/2 (V) = Indicador. El significado actual de los bits es aplicación dependiente.

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

Nota: Este Item de Datos puede contener parámetros tales como longitud de corrida del plot (radar primario y secundario). Diferencias entre el radar primario y secundario acimut derivado, amplitud del pulso, etc.

2.6.14 ITEM I001/131 Potencia recibida

Definición: Medición de la potencia recibida.

Formato: Item de Datos de longitud fija de un octeto.

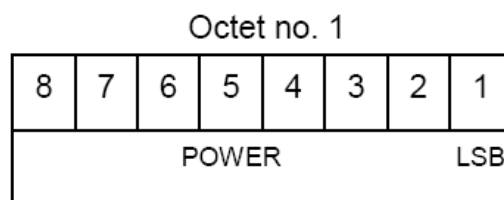


Figura 2.25: Estructura del octeto del Item de datos I001/131

2.6.14.1 Descripción de los bits del Item de Datos I001/131

Bits-7/1 (POWER) = Valor decimal logarítmico de la potencia recibida en dBm.

Bit-1 (LSB) = 1 dBm.

Notas

1. Potencia es el valor medido de la potencia recibida en la suma patrón para un plot.
2. Los valores negativos son expresados en la forma de complemento a dos, bit-8 es puesto a 0 para valores positivos y 1 para valores negativos.

2.6.15 ITEM I001/141 Tiempo Truncado del Día

Definición: El tiempo Real absoluto se expresa como Tiempo Universal Coordinado. Hora (UTC).

Formato: Item de Datos de longitud fija de dos octetos.

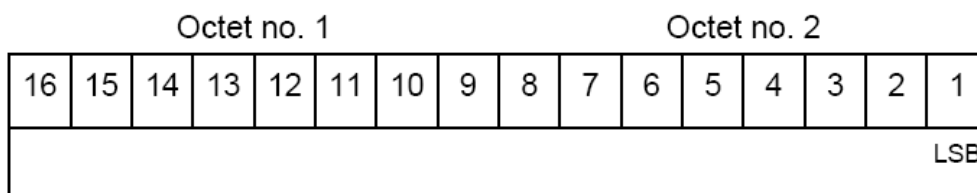


Figura 2.26: Estructura de los dos octetos del Item de datos I001/141

2.6.15.1 Descripción de los bits del Item de Datos I001/141

Bit-1 (LSB) = 2^{-7} s = 1/128 s

Notas

1. El intercambio de este Item de Datos permite la fácil derivación del valor correcto de tiempo UTC.
2. El valor de tiempo del día es puesto a 0 cada día a la medianoche.
3. Para el tiempo de manejo en aplicaciones de transmisión radar, refiérase a la parte 1, párrafo 5.4 del protocolo Asterix de Eurocontrol.

2.6.16 ITEM I001/150 Presencia del Pulso X

Definición: Presencia del Pulso X para los varios modos aplicados en el interlace de interrogación paralelo.

Formato: Item de Datos de longitud fija de un octeto.

Octet no. 1							
8	7	6	5	4	3	2	1
XA	0	XC	0	0	X2	0	0

Figura 2.27: Estructura del octeto del Item de datos I001/150

2.6.16.1 Descripción de los bits del Item de Datos I001/150

Bit-8 (XA) = 0 Por defecto

= 1 Pulso X recibido en respuesta Modo-3/A

Bit-7 Bit Spare puesto a 0

Bit-6 (XC) = 0 Por defecto

= 1 Pulso X recibido en respuesta Modo-C

Bits-5/4 Bits Spare puestos a 0

Bit-3 (X2) = 0 Por defecto

= 1 Pulso X recibido en respuesta Modo-2

Bits-2/1 Bits Spare puestos a 0

Nota: Este Item de Datos es transmitido solo si en por lo menos un pulso X ha sido recibido en una respuesta Modo-A, Modo-2 o Modo-C.

2.6.17 ITEM I001/161 Numero de Pista/Plot

Definición: Un valor entero representando una única referencia a una grabación Pista/Plot con un archivo particular Pista/Plot.

Formato: Item de Datos de longitud fija de dos octetos.

Octet no. 1								Octet no. 2							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
TRACK/PLOT NUMBER (max. 65 535)															

Figura 2.28: Estructura de los dos octetos del Item de datos I001/161

Nota: La diferenciación entre el número de una Pista y Plot está implícito en cada uno o es hecho vía el Descriptor de informe de blanco (Item de Datos I001/020).

2.6.18 ITEM I001/170 Estado de la Pista

Definición: Estado de la Pista derivado de la información radar del primario y/o secundario.

Formato: Item de Datos de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesarias.

Octet no. 1							
8	7	6	5	4	3	2	1
CON	RAD	MAN	DOU	RDPC	0	GHO	FX

Figura 2.29: Estructura del octeto de la primera parte del Item de datos I001/170

2.6.18.1 Descripción de los bits de la primera parte del Item de Datos I001/170

Bit-8 (CON) = 0 Pista confirmada

= 1 Pista en fase de inicialización

Bit-7 (RAD) = 0 Pista de radar primario

= 1 SSR/Pista combinada

Bit-6 (MAN) = 0 Por defecto

= 1 Aeronave maniobrada

Bit-5 (DOU) = 0 Por defecto

= 1 Plot dudoso para asociación de pista

Bit-4 (RDPC) Cadena de Procesamiento de Datos Radar

= 0 RDP Cadena 1

= 1 RDP Cadena 2

Bit-3 Bit Spare puestos a 0

Bit-2 (GHO) = 0 Por defecto

= 1 Pista fantasma

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

Notas

1. El bit-2 (GHO) es usado para señalar que la pista es sospechosa y fue generada por un blanco falso.
2. El bit-4 (RDPC) es usado para señalar la discontinuidad de los números de pista.

Octet no. 1

8	7	6	5	4	3	2	1
TRE	0	0	0	0	0	0	FX

Figura 2.30: Estructura del octeto de la primera extensión del Item de datos I001/170

2.6.18.2 Descripción de los bits de la primera extensión del Item de Datos I001/170

Bit-8 (TRE) = 0 Por defecto

= 1 Ultimo informe para una Pista

Bits-7/2 Bits Spare puestos a 0

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

2.6.19 ITEM I001/200 Velocidad de la pista calculada en coordenadas polares

Definición: Velocidad de la pista calculada expresada en coordenadas polares.

Formato: Item de Datos de longitud fija de cuatro octetos.

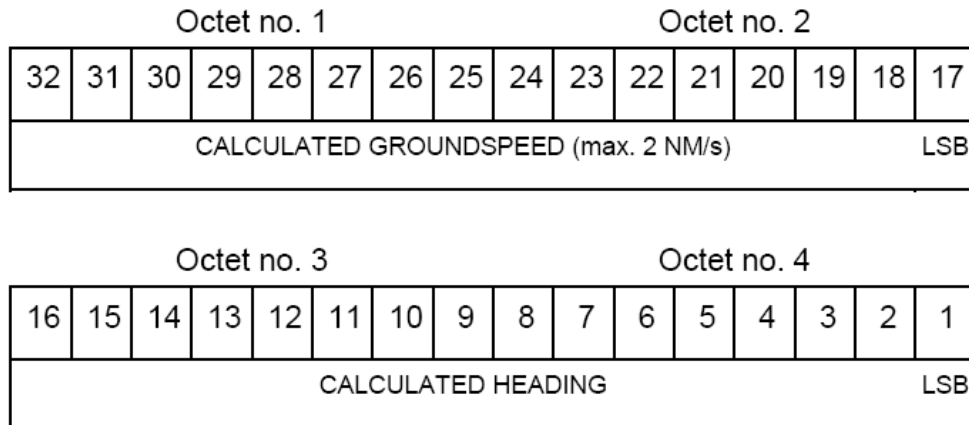


Figura 2.31: Estructura de los cuatro octetos del Item de datos I001/200

2.6.19.1 Descripción de los bits del Item de Datos I001/200

Bit-17 (LSB) = 2^{-14} MN/s = 0.22 kt

Bit-1 (LSB) = $360^\circ / (2^{16}) = 0.0055^\circ$

2.6.20 ITEM I001/210 Calidad de la pista

Definición: Calidad de la pista relativa.

Formato: Item de Datos de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesarias.

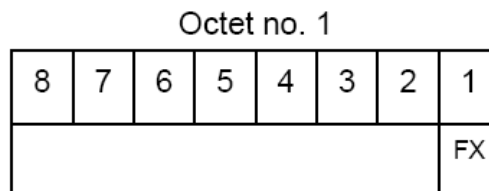


Figura 2.32: Estructura del octeto del Item de datos I001/210

2.6.20.1 Descripción de los bits del Item de Datos I001/210

Bits-8/2 Indicador de calidad

Bit-1 (FX) = 0 Fin de Item de Datos

 = 1 Presenta primera extensión del Item de Datos

Nota: El significado actual de los bits es dependiente de la aplicación.

2.7 DESCRIPCION DE LA CATEGORIA 002

Son Mensajes de Servicio Radar que contienen información necesaria para la correcta interpretación de la información radar recibida.

El flujo de información es unidireccional, de la cabecera a los sistemas de proceso radar.

Un simple UAP ha sido normalizado y debería ser usado para transmitir mensajes de servicio de una estación radar al usar sistemas RDP, en la **Figura 2.33** se muestra la estructura de un bloque de datos Asterix de categoría 002 y en la **Tabla 2.8** se muestra los Items de Datos estándar de categoría 002.

CAT = 002	LEN	FSPEC	Items of the first record		FSPEC	Items of the last record
------------------	------------	--------------	---------------------------	--	--------------	--------------------------

Figura 2.33: Bloque de Datos de categoría 002

Data Item Ref. No.	Description	System Units
I002/000	Message Type	N.A.
I002/010	Data Source Identifier	N.A.
I002/020	Sector Number	$360^\circ/(2^8)$
I002/030	Time of Day	1/128 s
I002/041	Antenna Rotation Period	1/128 s
I002/050	Station Configuration Status	N.A.
I002/060	Station Processing Mode	N.A.
I002/070	Plot Count Values	N.A.
I002/080	Warning/Error Conditions	N.A.
I002/090	Collimation Error	Range: 1/128 NM Azimuth: $360^\circ/(2^{16})$
I002/100	Dynamic Window - Type 1	RHO: 1/128 NM THETA: $360^\circ/(2^{16})$

Tabla 2.8: Items de Datos estándar de categoría 002

2.8 DESCRIPCION DE ITEMS DE DATOS ESTANDAR DE LA CATEGORIA 002

2.8.1 ITEM I002/000 Tipo de Mensaje

Definición: Este Item de Datos permite para un manejo más conveniente de los mensajes en el lado del receptor más lejos definiendo el tipo de transacción.

Formato: Item de Datos de longitud fija de un octeto.



Figura 2.34: Estructura del octeto del Item de datos I002/000

2.8.1.1 Descripción de los bits del Item de Datos I002/000

Bits-8/1 Tipo de mensaje

Notas

1. En aplicación donde las transacciones de varios tipos son intercambiadas, el tipo de mensaje de Item de Datos facilita el manejo del propio mensaje en el lado del receptor .
2. Los valores del tipo de mensaje de 1-127 están reservados para uso común estándar, donde los valores 128-255 son dependientes de la aplicación.
3. El siguiente set de tipos de mensaje están estandarizados para registros de la categoría 002:
 - 001, Mensaje de Norte
 - 002, Mensaje de Sector
 - 003, Mensaje de Sur
 - 008, Activación de filtrado de zona ciega
 - 009, Detener el filtrado de zona ciega

2.8.2 ITEM I002/010 Identificación Fuente de Datos

Definición: Identificación de la estación radar de la cual son recibidos los datos.

Formato: Item de Datos de longitud fija de dos octetos.

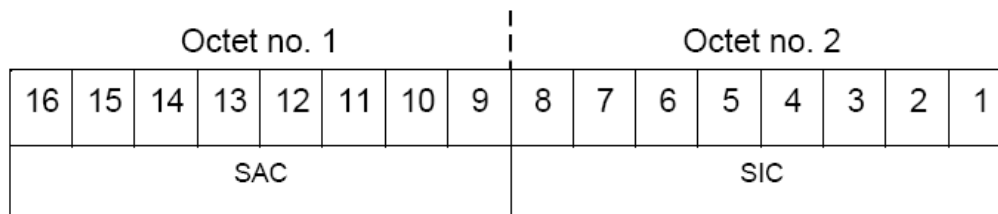


Figura 2.35: Estructura de los dos octetos del Item de datos I002/010

2.8.2.1 Descripción de los bits del Item de Datos I002/010

Bits-16/9 (SAC) Sistema de Código de Area

Bits-8/1 (SIC) Sistema de Código de Identificación

2.8.3 ITEM I002/020 Número de Sector

Definición: Ocho bits más significativos del acimut de la antena definiendo un particular sector de acimut.

Formato: Item de Datos de longitud fija de un octeto.

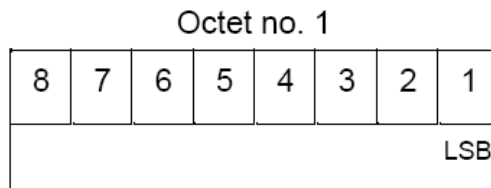


Figura 2.36: Estructura del octeto del Item de datos I002/020

2.8.3.1 Descripción de los bits del Item de Datos I002/020

Bits-8/1 = Ocho bits más significativos del acimut de antena

Bit-1 (LSB) = $360^\circ / (2^8) = 1.41^\circ$

Nota: El uso del acimut de antena como número de sector tiene la desventaja de ser independiente del número de sectores implementados.

2.8.4 ITEM I002/030 Tiempo de Día

Definición: El tiempo Real absoluto se expresa como Tiempo Universal Coordinado. Hora (UTC).

Formato: Item de Datos de longitud fija de tres octetos.

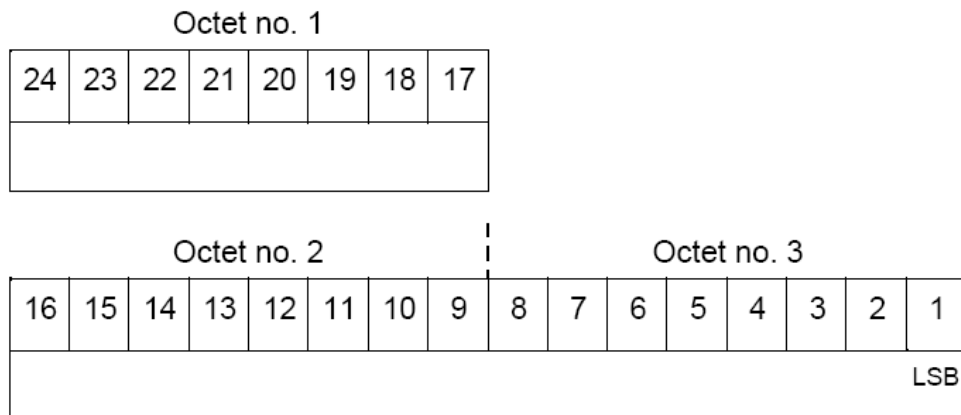


Figura 2.37: Estructura de los tres octetos del Item de datos I002/030

2.8.4.1 Descripción de los bits del Item de Datos I002/030

Bit-1 (LSB) = 2^{-7} s = 1/128 s

Notas

1. El valor del Tiempo de Día es puesto a cero cada día a la media noche.
2. El Item de Datos I002/030 puede tener varios significados lógicos. En un mensaje particular, el significado lógico es implícito de su contexto. (Por ejemplo en un mensaje de Norte esto representa el tiempo que la antena del radar marca el Norte, en un mensaje de sector esto representa el tiempo que la antena del radar marca el Sector).

2.8.5 ITEM I002/041 Período de rotación de antena

Definición: El Período de rotación de antena es medido entre dos consecutivas marcaciones de norte o durante el promedio de un periodo de tiempo.

Formato: Item de Datos de longitud fija de dos octetos.

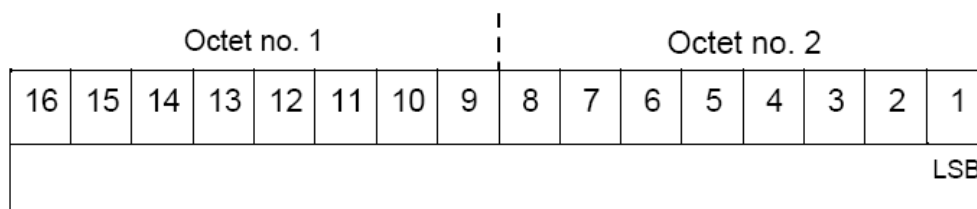


Figura 2.38: Estructura de los dos octetos del Item de datos I002/041

2.8.5.1 Descripción de los bits del Item de Datos I002/041

Bit-1 (LSB) = 2^{-7} s = 1/128 s

2.8.6 ITEM I002/050 Estado de configuración de la estación

Definición: Información concerniente al uso y estado de algún hardware vital componentes del sistema radar.

Formato: Item de Datos de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesario.

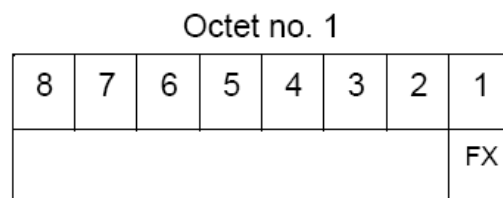


Figura 2.39: Estructura del octeto del Item de datos I002/050

2.8.6.1 Descripción de los bits del Item de Datos I002/050

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

Nota: Debido a la diversidad en diseño de hardware y el requerimiento presente y futuro de las estaciones radar, es impráctico para atender la definición de bits individuales.

2.8.7 ITEM I002/060 Modo de procesamiento de la estación

Definición: Detalles concernientes al estado presente con respecto a los parámetros de procesamiento y opciones.

Formato: Item de Datos de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesario.

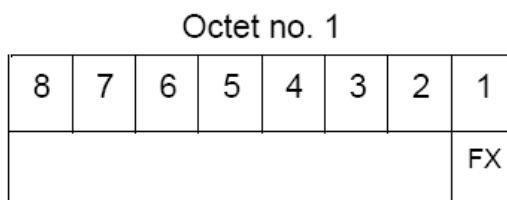


Figura 2.40: Estructura del octeto de la primera parte del Item de datos I002/060

2.8.7.1 Descripción de los bits del Item de Datos I002/060

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos

Notas

1. Información típica convista con este Item de Datos incluye tipo de inter alia de polarización en uso, Indicador de Blanco Movil (MTI) en uso y/o definición del rango al cual es aplicado MTI, presencia de condiciones de sobrecarga y el tipo de reducción de carga medida en uso.
2. Solamente la estructura de este Item de datos está definida, ningún intento es hecho al estandarizar su contenido, en servicio no para estorbar cada aplicación o desarrollo futuro.

2.8.8 ITEM I002/070 Valor Contador de Plots

Definición: Valores de conteo de plot, acorde a varias categorías de plot, para el ultimo escaneo total de antena o para el último sector procesado.

Formato: Item de Datos Repetitivo, iniciando con un octeto Indicador de Repetición de Campo (REP) seguido por al menos un contador de plot de dos octetos de longitud.

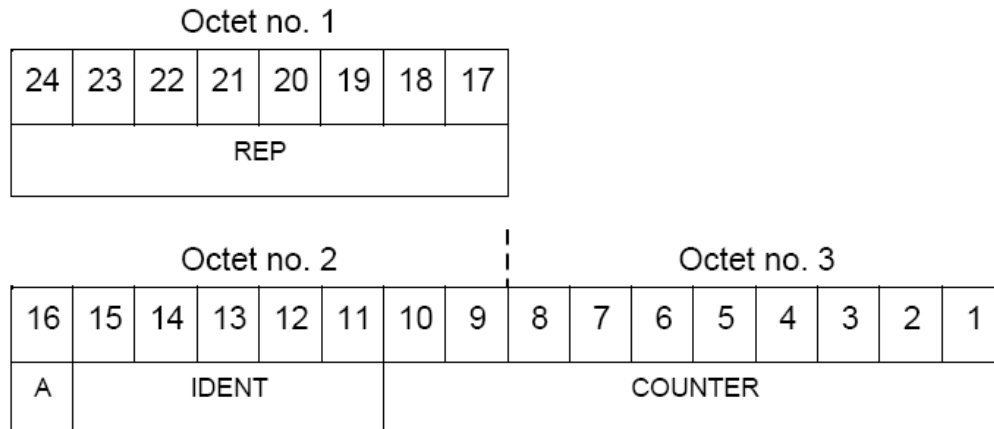


Figura 2.41: Estructura de los octetos del Item de datos I002/070

2.8.8.1 Descripción de los bits del Item de Datos I002/070

Bits-24/17 (REP) Factor de Repetición

Bit-16 (A) Identificación Aérea:
 = 0 Contador para antena 1
 = 1 Contador para antena 2

Bits-15/11 (IDENT) Código de Identificación de Categoría de plot de 5 bits, es como sigue:
 = 1 Plots solo primarios;
 = 2 Plots solo SSR;
 = 3 Plots combinados

Bits-10/1 (COUNTER) Valor contado de 10 bits

2.8.9 ITEM I002/080 Condiciones de Alarma/Error

Definición: Condiciones de Alarma/Error afectando la funcionalidad del mismo sistema radar.

Formato: Item de Datos de longitud variable comprendido por una primera parte de un octeto, seguido por extensiones de un octeto de ser necesario.

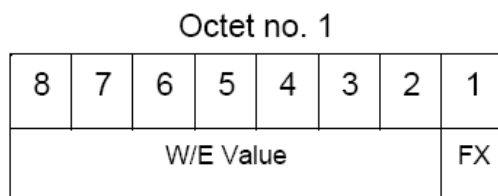


Figura 2.42: Estructura del octeto del Item de datos I002/080

2.8.9.1 Descripción de los bits del Item de Datos I002/080

Bits-8/2 Primer valor de condición Alarma/Error

Bit-1 (FX) = 0 Fin de Item de Datos

= 1 Presenta primera extensión del Item de Datos (por ejemplo segundo valor de condición Alarma/Error)

Nota: Los valores de condición Alarma/Error de 1-63 son reservados para uso estándar común, donde los valores de 64-127 son dependientes de la aplicación.

2.8.10 ITEM I002/090 Error de colimación

Definición: Diferencia promedio en rango y en acimut para la posición del blanco primario con respecto a la posición del blanco SSR es calculado por la estación radar.

Formato: Item de Datos de longitud fija de dos octetos.

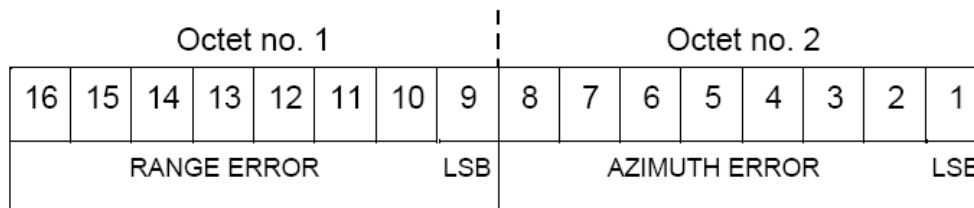


Figura 2.43: Estructura de los dos octetos del Item de datos I002/090

2.8.10.1 Descripción de los bits del Item de Datos I002/090

Bit-9 (LSB) = 1/128 NM

Bit-1 (LSB) = $360^\circ / (2^{(16-f)})$

Notas

1. Una unidad de cuantización por defecto de 0.022° y entre un rango de -2.8125° y 2.7905° es obtenido para un valor de $f=2$.
2. Los valores negativos son expresados en la forma de complemento a dos, bit-16 y bit-8 son puestos a 0 para valores positivos y 1 para valores negativos.

2.8.11 ITEM I002/100 Ventana Dinámica – Tipo 1

Definición: La activación de señales de una cierta función de filtrado selectivo y en un sistema de coordenadas polares las respectivas áreas geográficas.

Formato: Item de Datos de longitud fija de ocho octetos.

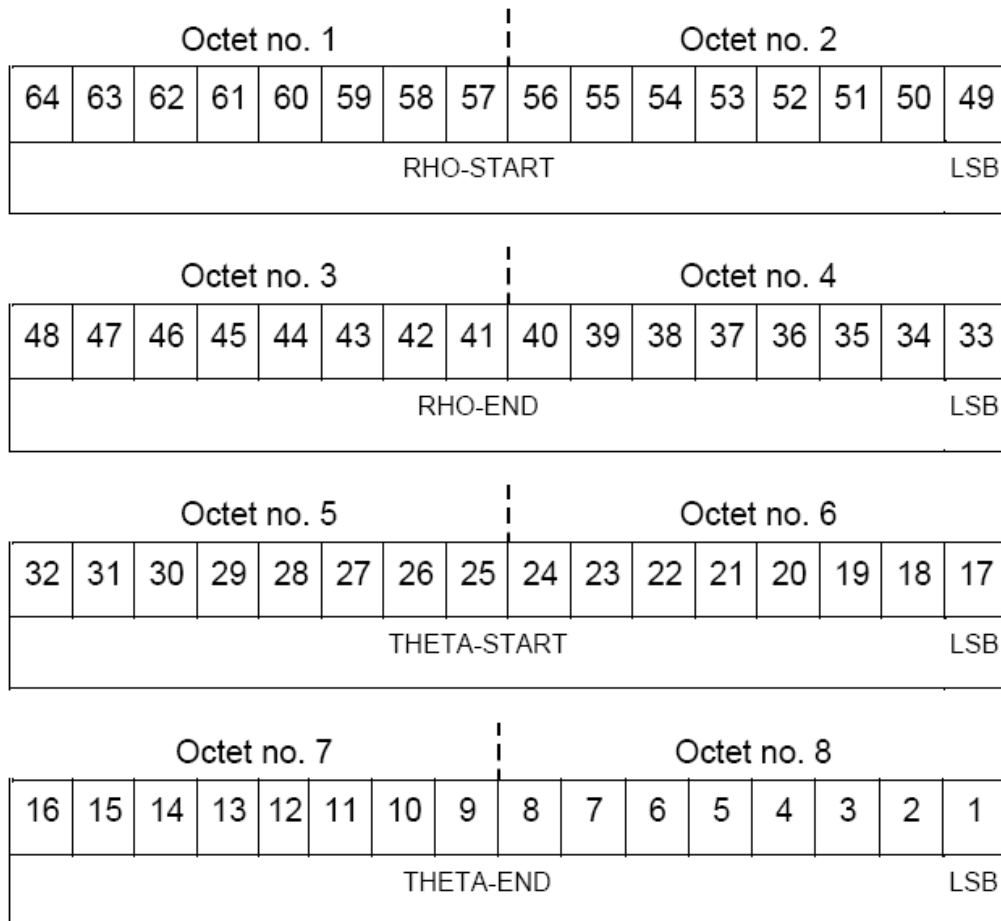


Figura 2.44: Estructura de los ocho octetos del ítem de datos I002/100

2.8.11.1 Descripción de los bits del Item de Datos I002/100

Bit-49 (LSB) = 1/128 NM Rango maximo = 512 NM

Bit-33 (LSB) = 1/128 NM Rango maximo = 512 NM

Bit-17 (LSB) = $360^\circ / (2^{16}) = 0.0055^\circ$

Bit-1 (LSB) = $360^\circ / (2^{16}) = 0.0055^\circ$

Nota: El significado lógico de la ventana polar es definido por su contexto, dado por el tipo de mensaje (Item de Datos I002/000) en el registro concerniente.

2.9 APLICACION DE ASTERIX AL RADAR INDRA IRS 20MP/L

En vista de que la fuente de datos es el radar secundario INDRA IRS 20MP/L, de San Salvador Ilopango, se detalla la aplicación del protocolo Asterix de Eurocontrol en las siguientes Tablas definiendo los Item de Datos que son obligatorios, los disponibles y los opcionales.

FRN	Data Item	Descripción del Data Item	Longitud en Octetos	Uso
1	I001/010	Data Source Identifier	2	obligatorio
2	I001/020	Target Report Descriptor	1+	obligatorio
3	I001/040	Measured Position in Polar Coordinates	4	obligatorio
4	I001/070	Mode-3/A Code in Octal Representation	2	si disponible
5	I001/090	Mode-C Code in Binary Representation	2	si disponible
6	I001/130	Radar Plot Characteristics	1+	obligatorio
7	I001/141	Truncated Time-of-Day	2	obligatorio
FX	-	Field Extension Indicator	-	1 / 0
8	I001/050	<i>Mode-2 Code in Octal Representation</i>	2	<i>No usado</i>
9	I001/120	<i>Measured Radial Doppler Speed</i>	1	<i>No usado</i>
10	I001/131	<i>Received Power</i>	1	<i>No usado</i>
11	I001/080	Mode-3/A Code Confidence Indicator	2	optional
12	I001/100	Mode-C Code and Confidence Indicator	4	optional
13	I001/060	<i>Mode-2 Code Confidence Indicator</i>	2	<i>No usado</i>
14	I001/030	<i>Warning/Error Conditions</i>	1+	<i>No usado</i>
FX	-	Field Extension Indicator	-	0

Tabla 2.9: Tabla de Aplicación de Usuario para Plots en Categoría 001

FRN	Data Item	Descripción del Data Item	Longitud en Octetos	Uso
1	I002/010	Data Source Identifier	2	obligatorio
2	I002/000	Message Type	1	obligatorio
3	I002/020	Sector Number	1	No usado
4	I002/030	Time of Day	3	obligatorio
FX	-	Field Extension Indicator	-	0

Tabla 2.10: Tabla de Aplicación de Usuario para Marcación Norte en Categoría 002

2.10 APLICACION DE ASTERIX A RADARES ALENIA

En la **Tabla 2.10** se detalla la aplicación del protocolo Asterix de Eurocontrol para los radares ALENIA que existen en Centroamérica.

Field Reference Number	Status in F-Spec	Data Item	Name	Length in bytes
1	0 / 1	I001/010	Data source identifier	0 / 2
2	1	I001/020	Target report descriptor	1 / 2
3	1	I001/040	Measured position in polar coordinates	4
4	0 / 1	I001/070	Mode-3/A code in octal representation	0 / 2
5	0 / 1	I001/090	Mode-C code in binary representation	0 / 2
6	0 / 1	I001/130	Radar Plot Characteristics	0 / 1
7	0	I001/141	Truncated time of day	0
FX	1	-	Field Extension Indicator	-
8	0 / 1	I001/050	Mode-2 code	0 / 2
9	0	-	not used	0
10	0	-	not used	0
11	0	-	not used	0
12	0	-	not used	0
13	0	-	not used	0
14	0	-	not used	0
FX	1	-	Field Extension Indicator	-
15	0	-	not used	0
16	0	-	not used	0
17	0	-	not used	0
18	0	-	not used	0
19	0	-	not used	0
20	0	-	Reserved for SP Indicator	-
21	1	-	Reserved for RFS Indicator	-
FX	0	-	Field Extension Indicator	-

Tabla 2.11: Tabla de Aplicación de Usuario para Plots en Categoría 001 ALENIA

2.11 ENVIO DE DATOS RADAR

Después que se han procesado los Datos de las diferentes estaciones radar y se han llevado al formato ASTERIX es necesario enviarlos al centro de control para su explotación. Es cuando los datos ya procesados se envían a los diferentes operadores de control de tránsito aéreo y control radar.

Para esta función se utiliza el protocolo HDLC (High-Level Data Link Control) Control de Enlace de Datos de Alto Nivel de ISO. En la **Figura 2.45** se muestra una Trama HDLC.

FLAG	DIRECCIÓN	CONTROL	DATOS	FCS		FLAG
1 Byte	1 Byte	1 Byte	Variable	2 Bytes		1 Byte
7E	00	03	Asterix	*	*	7E

Figura 2.45: Trama HDLC

2.12 EJEMPLO DE UNA TRAMA ASTERIX

01 00 12 FE F1 03 20 14 92 67 6B 09 BB 02 38 28 1E 91

CATEGORIA = 01 = 1 LONGITUD DE LA TRAMA = 0012 = 18 octetos

FSPEC = FE SAC = F1 = El salvador SIC = 03

DESCRIPCION DEL INFORME DE BLANCO = 20

- **Detección secundaria.**
- **Reporte de blanco de antena 1**

RANGO O DISTANCIA = 14 92= 41.14 NM

ACIMUT = 67 6B = 145.43°

CODIGO TRANSPONDER = 09 BB = 4673

ALTITUD DE LA AERONAVE = 02 38 = 14200 ft

CARACTERISTICAS DEL PLOT RADAR = 28

TIEMPO DE DIA TRUNCADO: 1E 91 = 00:01:01

2.14 EVALUADORES COMERCIALES DE ASTERIX

Las herramientas software WRTADS y RBAT se utilizan para evaluar el desempeño general de los sensores radar, siendo utilizadas en el mantenimiento y optimización de los sistemas de vigilancia por los especialistas radar y de automatización de los sistemas de procesamiento de datos de vigilancia, la FAA los utiliza para evaluar el desempeño de sus sistemas radar.

Estas herramientas son gratis y solo tienen que ser descargadas de la página de la FAA y ser instaladas en una computadora con Windows para su utilización.

2.14.1 Evaluador comercial WRTADS



WRTADS generalmente se utiliza para la captura de los datos radar en el formato Asterix, para luego hacer una evaluación gráfica rápida sobre los blancos radar. La opción setup de WRTADS permite al usuario seleccionar y configurar sitios radar que están siendo recibidos o a través del relevo avanzado interface PCMCIA o un enlace de red. En la interface Card Group Box, se chequea la Black Box si se usa el enlace de relevo avanzado o Ethernet si se usa un enlace de red. La fuente de tiempo para los mensajes registrados es corrientemente limitada a tiempo de PC, almacenada en 1/10 decimas de segundo. El número de sitios especifican cuántos sitios el usuario desea configurar en el sistema y Select Site especifica cual sitio será desplegado activamente. El botón Config Site muestra una ventana Site Info para configurar el sitio que es seleccionado. El Real-Time RBAT checkbox habilita los datos a ser enviados al programa Real-Time RBAT.

La **Figura 2.47** nos muestra la caja de Dialogo del SETUP de WRTADS.

2.14.1.1 Explicación de algunas opciones de WRTADS

Display Congelado

Display Congelado de Live Data

Esta opción permite al usuario pausar el display live data, también durante la grabación, mientras que no detiene el proceso de grabación. El icono Stop Live  Display se usa. Después este icono es clickeado, El icono Go  tomara su lugar en la barra de herramientas.

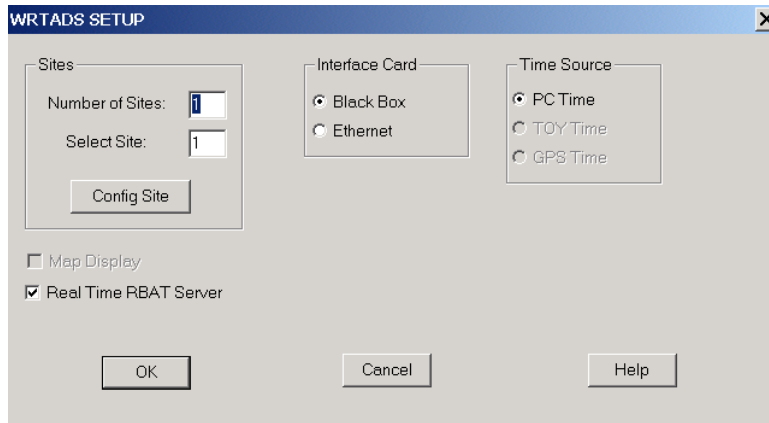


Figura 2.47: Caja de Dialogo del SETUP de WRTADS

Limpiar El Display (Reset)


El Display puede ser limpiado de todos los datos clickeando en el Icono Reset Display.




Esto también retorna el display a la posición central sin zoom y limpia escaneo acumulado e información de blanco siendo mantenido por la aplicación.

Rastros de blanco (Target trails)

El usuario tiene una elección de ningún rastro, 10 rastros, o rastros continuos. Hay tres iconos usados para el control de rastros de blanco.

El icono diez rastros 

El icono Rastro continuo 

Y el icono apagar rastros 

2.14.1.2 Barra de herramientas de WRTADS



Seleccionar el archivo grabado para Playback



Pausar el archivo Playback
























Retroceder archivo playback



Detener adelantamiento rápido de archivo playback





















Adelantamiento rápido de archivo playback










-  Resumen Playback después de pausa
-  Detener playback
-  Iniciar grabación Live Data
-  Detener grabación Live Data
-  Reset Display
-  Zoom en Display
-  Zoom fuera Display
-  Display 10 rastros
-  Display rastros continuos
-  Apagar rastros
-  Seleccionar filtros de blanco
-  Apagar el procesamiento Live Data
-  Resumen del procesamiento Live Data
-  Etiquetar blancos desplegados, etc. Pone bloques de datos
-  Ventana Display Data, con formato o crudo
-  Setup para Windows RTADS
-  Muestra la ventana sumaria
-  Display Congelado de Live Data
-  Resume el display de Live Data
-  Ayuda para Windows RTADS
-  Salir de RTADS para Windows

2.14.1.3 Leyenda de símbolo de blanco







ATCRBS Modo S

		Modo valido 3A y Modo C
		Modo valido 3A y Modo C, búsqueda reforzada, Discreta
		Modo valido 3A y Modo C, búsqueda reforzada
		Modo valido 3A y Modo C, Discreta
		Modo C valido
		Modo 3A valido
		Código 7700
		Código 7600
		Modo 3A y Modo C Invalido

2.14.1.4 Simbolos de seguimiento miscelaneo

	CPME Modo S
	Identificador de Posición Especial
	CPME ATCRBS
	Identificador de Posición Especial ATCRBS Modo S
	Papagayo ATCRBS
	Sector Sync Faro
	RTQC Faro
	Transpondedor Modo S Ilegal
	Falso blanco de seguimiento

2.14.1.5 Simbolos de busqueda

-  Búsqueda Correlacionada
-  Búsqueda Descorrelacionada
-  Blanco Test de Búsqueda
-  Sector Sync de Búsqueda
-  RTQC Descorrelacionado de Búsqueda
-  RTQC Correlacionado de Búsqueda

En la **Figura 2.48** se logra apreciar la caja de Dialogo de Record de WRTADS, en la cual el usuario puede marcar el numero de sitios radar a grabar, las opciones del archivo en el cual se grabaran los datos Asterix, comentarios, configuración de grabación Asterix, Tamaño del tráiler de grabación y versión de la grabación Asterix.

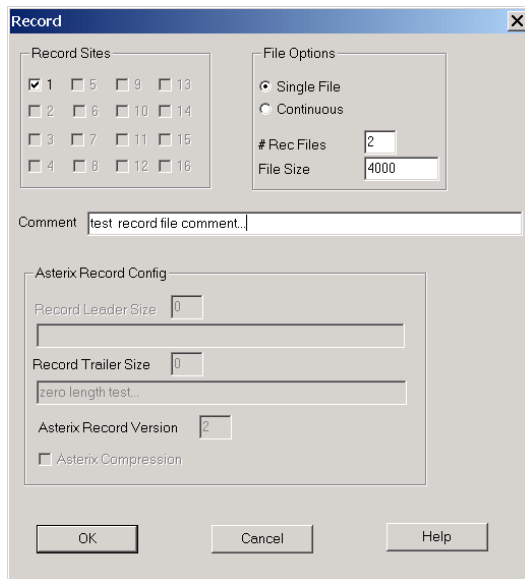


Figura 2.48: Caja de Dialogo de Record de WRTADS

En la **Figura 2.49** se muestra la ventana principal de WRTADS, en ella en la parte superior se observa la barra de herramientas compuesta por un contador de tiempo y cada uno de los iconos antes mencionados, se puede observar en el centro la PPI, a la izquierda un contador de escaneo o barrido, en el cual se cuentan el numero de barridos de antena, numero de

Atajo al sub menu item Merge Results bajo el menu ítem Program. Selecciona un programa para combinar los resultados.



Atajo al resumen sub menu item bajo el Historial sub menu item bajo el Plot menu ítem.



Atajo al Print Program bajo el sub menu item Display Results bajo el menu ítem Program.



Atajo al Scroll program bajo el sub menu item Display Results bajo el menu ítem Program.



Atajo al View program bajo el sub menu item Display Results bajo el menu ítem Program.



Atajo al sub menu item Utilities bajo el menu item Program. Selecciona una utilidad a convertir un archivo SGF o ASR11 ASTERIX EDC, pone por defecto los valores de entrada o convierte latitud y longitud.



Atajo al sub menu item Execute bajo el menu item Mode. Ejecuta el programa Ahora y cualquier otro en la cola.



Atajo al sub menu item Queue bajo el menu item Mode. Encola el programa hasta ejecutar el modo de entrada.



Atajo para el sub menu ítem Help Topics bajo el Help menu ítem. Despliega los tópicos de ayuda.



Atajo al sub menú ítem Tutorial bajo el menu item Help. Despliega el Tutorial.



Atajo al sub menu item About bajo el menu item Help. Despliega la información acerca de la versión de RBAT.

En la **Figura 2.50** se presenta la ventana principal de RBAT, en donde en la parte superior podemos observar la barra de herramientas principal con cada uno de los iconos descritos anteriormente.



Figura 2.50: Ventana principal de RBAT

2.14.2.2 Barra de herramientas de Real Time RBAT

La barra de herramientas es una secuencia de mapas de bits que son atajos para ítems en el Menú para Real Time RBAT. Sólo algunos de los ítems del menú tienen atajos. La barra de herramientas contiene los mapas de bits siguientes:



Atajo al conjunto menú ítem. Usado para seleccionar el conjunto para desplegar y trazar los datos.



Atajo para el sub menu item resumen bajo el Display menu ítem.



Atajo al resumen sub menu item bajo el Historial sub menu item bajo el Plot menu ítem.



Atajo para el Print menu item. Imprime el display seleccionado o Plot.



Atajo para el sub menu ítem Help Topics bajo el Help menu ítem. Despliega los tópicos de ayuda.



Atajo al sub menú ítem Tutorial bajo el menu item Help. Despliega el Tutorial.



Atajo al sub menú ítem bajo el Help menu item. Despliega acerca de la información de la versión Real Time RBAT.

2.14.2.3 Tipos de análisis de RBAT

En RBAT cada tipo de análisis está asociado a un programa, que analiza los datos en un archivo de extracción de entrada, una corta descripción de algunos de los programas a continuación:

- **Análisis de vigilancia:** Analiza el desempeño de Seguimiento y Búsqueda usando un trazador de Seguimiento y Búsqueda integrado. Las estadísticas son dadas por la pista, posición y en conjunto.
- **Pista plot Movil:** Grafica los mensajes que están en un usuario especificando intervalo de rango y acimut alrededor de la traza seleccionada. Esto se hace repetidamente mientras el usuario permanece entrando pistas. Este programa es usado para examinar el ambiente de problema de las pistas.
- **Resumen de blanco falso de seguimiento:** Identifica blancos falsos y los agrupa en la hendidura, una vuelta de anillo, reflexión downlink, PRF, reflexión uplink, ATCRBS/Mode S y otras categorías. Las reflexiones uplink son usadas para calcular la ubicación y orientación de los reflectores.
- **Impresión y Grafico de vigilancia:** Imprime, grafica y cuenta los mensajes. Tiempo, rango, acimut, altitud, ángulo de elevación, id ATCRBS, id Modo S, numero de archivo de vigilancia, Modo 2, canal, ATCRBS/Mode S y los filtros de tipo de mensaje están disponibles.
- **Seguimiento y cobertura radar:** Calcula el Seguimiento y cobertura radar para la celda. La cobertura es definida como la altitud mínima y ángulo de elevación para la celda. Una celda se define como una región del espacio aéreo.
- **Análisis de conflicto:** Analiza los conflictos. Dos trazas están en conflicto si están mutuamente en una especificación de usuario como el rango, acimut y ventanas de altitud (opcional). Las mismas estadísticas que son dadas para análisis de vigilancia son provistas para cada conflicto. Los gráficos de los conflictos son provistos también.
- **Análisis de búsqueda duplicado:** Identifica pares de mensajes de Búsqueda duplicados. Un mensaje de Búsqueda es un duplicado si tiene la misma altura, rango y acimut como otro mensaje de Búsqueda o si este tiene la misma altura y su rango y acimut esta dentro de las ventanas especificadas por el usuario alrededor de un radar reforzando el mensaje de Seguimiento.
- **Vigilancia mejorada:** Determina los registros GICB que son usados por transponders Modo S. Este programa requiere un archivo extracción de entrada ASTERIX.

- **Filtrado:** Filtrado del archivo extracción de entrada y crea un archivo extracción de salida que contiene todos los mensajes que pasan el filtro. Tiempo, rango, acimut, altitud, ángulo de elevación, id ATCRBS, id Mode S, numero de traza, numero de archivo de vigilancia, Modo 2, canal, ATCRBS/Modo S, tipo de mensaje y filtros de campo de bit están disponibles.
- **Exactitud de transponder fijo:** Calcula la exactitud del rango y acimut usando los mensajes de Seguimiento para un transpondedor fijo específico. El rango y acimut del transpondedor fijo son entradas por el usuario.

2.14.2.4 Análisis de Vigilancia

Como lo que interesa es la probabilidad de detección de blancos radar, este análisis inicia con la probabilidad de detección. Esta es dividida en tres categorías nombradas Seguimiento, Búsqueda y Total. La probabilidad de detección de Seguimiento es el porcentaje de veces que un mensaje de Seguimiento correlaciona con la pista. Correlación es el proceso de igualar un informe con su respectiva pista. Una pista es una secuencia de informes de una simple aeronave, donde un informe puede ser de Seguimiento, Búsqueda o un mensaje AIMS. Para algunas vueltas de antena el informe fallara, estos son costas. Así la definición de una pista es una secuencia de actualizaciones de una aeronave; la mayor parte de estas actualizaciones son informes pero algunos son costas. Para las vueltas de antena sin una igualación de mensaje, la pista se establece como costa; etc, un mensaje estimado es usado. El mensaje estimado es la mejor conjetura donde la aeronave esta en esa vuelta de antena. Para algunas vueltas de antena el problema opuesto ocurre, en lugar de ningún mensaje que iguale la pista hay múltiples mensajes que igualan la pista. La correlación debe igualar el mejor mensaje a la pista en cada vuelta de antena. Ahora todos los términos usados para definir la probabilidad de detección de Seguimiento están definidos. Es así una medida de con qué frecuencia el sistema de radar ve una aeronave que tiene un transponder expresada como un porcentaje. Esta definición es para una aeronave individual. Cuando todas las aeronaves son tomadas en conjunto, la probabilidad de detección a nivel de sistema es definida como la suma de todas las probabilidades de detección a nivel de pista. Esto parece directo pero existe una captura. Esta captura es en la forma de una suposición oculta. La suposición oculta es que todas las aeronaves son trazadas. Si una aeronave no es vista a todo por el radar, no tendrá una pista y así la probabilidad de detección a nivel de sistema no la tomara en cuenta. Esto también es verdadero si la aeronave es a veces vista pero no bastante para empezar una pista. Así el resultado de la probabilidad de detección de Seguimiento posiblemente exagere a nivel de sistema.

Las estadísticas a nivel de sistema son divididas en ocho categorías que se mencionan a continuación:

- Discreto
- No Discreto

- ATCRBS
- Modo S
- Seguimiento
- Búsqueda,
- Total
- Modo 4

En la **Figura 2.51** podemos apreciar la forma en la cual se dividen las diferentes categorías, pudiendo identificar cuáles son las categorías principales.

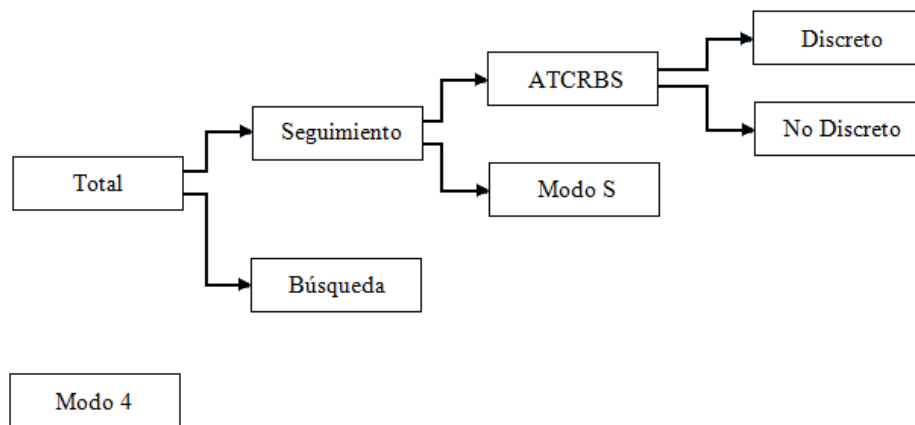



Figura 2.51: División en categorías de las estadísticas a nivel de sistema de RBAT

La categoría Total contiene las pistas de Seguimiento y Búsqueda pero no pistas Modo 4. Las categorías de Seguimiento y Búsqueda contienen las pistas de Seguimiento y Búsqueda respectivamente. La categoría de Seguimiento se parte en dos subcategorías, es decir ATCRBS y Modo S. La categoría ATCRBS también se parte en dos subcategorías, es decir discreta y no discreta. La categoría discreta contiene todas las pistas ATCRBS que son discretas. Esto significa que para cada actualización de pista el id ATCRBS es discreto. Un id ATCRBS es discreto si al menos uno de los seis bits menos significativos del id de ATCRBS es uno. Así el id ATCRBS, el cual está expresado en octal, no termina en 00. La categoría no discreta son todas las pistas en la categoría ATCRBS que no están en la categoría discreta.

Después se considera la fiabilidad de identidad y estadísticas de confianza. La fiabilidad de identidad es el porcentaje de veces que el código Modo 3/A del correlacionado mensaje de Seguimiento casa los bits de alta confianza en el código Modo 3/A código de la pista. El código de identidad es el código de 12 bits Modo 3/A. La confianza de pista es el código de 12 bits Modo 3/A confianza para la pista. La confianza de pista es construida usando el Modo 3/A confianza de código de los mensajes de Seguimiento que correlacionó a la pista en el pasado. Un bit es seleccionado en la confianza de pista siempre que el bit ha sido puesto en un correlacionado mensaje de Seguimiento al menos una vez en el pasado. Si un bit en la confianza de pista es seleccionado, el correspondiente bit en el código de identidad

para la pista es llamado un bit de alta confianza. Lo mismo es verdadero para el mensaje de Seguimiento. No existe ninguna penalización si un bajo bit de confianza no logra hacer juego. La confianza de identidad es el porcentaje de veces cuando el código del Modo 3/A hace juego (como se definió anteriormente) que la confianza de código del Modo 3/A del correlacionado mensaje de Seguimiento es confianza alta. Con esto se está midiendo con qué frecuencia el código correcto de identidad es visto y con qué frecuencia el código de identidad se ve, cuando es correcto, es confianza alta. Aquí de nuevo la fiabilidad de identidad y estadísticas de confianza son dependientes del trazador. Si el trazador correlaciona mal el mensaje de Seguimiento, son afectadas.

La última consideración es la fiabilidad de altitud y confianza. La fiabilidad de altitud es el porcentaje de veces cuando la pista tiene una altura positiva que la altura del correlacionado mensaje de Seguimiento (o mensaje de Búsqueda con la altura 3D) está dentro de la ventana de altura (en pies) de la altura de la pista. La ventana de altura es 800 pies durante 4.7 segundos tiempo de vuelta de antena y proevaluado para otros tiempos de examinación. Así esto es 1600 pies para un tiempo de examinación de 9.4 segundos. La confianza de altura es el porcentaje de veces cuando la altura hace juego que la confianza de altura del correlacionado mensaje de Seguimiento es confianza alta. Estamos midiendo con qué frecuencia la correcta altitud es vista y con qué frecuencia la altitud se ve, cuando es correcta, es de alta confianza. Aquí, aún más así que para la fiabilidad de identidad, el valor que la altura debe tener es problemático. El trazador sigue a la aeronave en altura así como en rango y acimut; Sin embargo, los datos de altura son menos confiables que el rango y datos de acimut. Esto hace el trazado de altura más difícil.

Existen también otras vías de categorizar las estadísticas, esencialmente variaciones en un tema. Para ello hay que examinar las salidas de texto. Existe también una salida de plot. Esto se ve usando el plot del programa. Para iniciar este programa el usuario elije el ítem del menú Program y se elije el sub menú ítem Display Results. Cuatro programas son desplegados. Se hace click en Plot y click en OK. Una alternativa es hacer click en el botón bit map  de la barra de herramientas.

El primer plot es la probabilidad de detección de Seguimiento, contra rango y altitud. La **Figura 2.52** es la grafica en 3D de la Probabilidad de Detección.

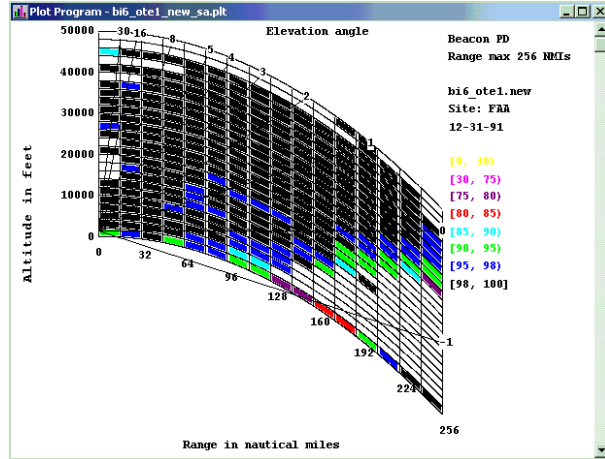


Figura 2.52: Grafica en 3D de la Probabilidad de Detección

Los ejes horizontales (ejes x) están curvos para mostrar la curvatura de la tierra. La probabilidad de detección de Seguimiento es calculada para cada celda de rango y altura, que es coloreada usando un rango especificado de valores. Si la celda es blanca, existen datos insuficientes para esa celda. Las líneas del ángulo de elevación son dibujadas a lo largo hacia la derecha y parte superior del plot. La línea une el origen de radar y el límite del plot. La línea no traza el camino de la radiación electromagnética que tiene inclinación en la atmósfera pero sólo muestra el límite de la línea de mira. La onda tiene inclinación debido a la difracción en la atmósfera. Esto se modela usando una esfera cuyo radio es 4/3 el radio de la tierra.

En la **Figura 2.53** se observa la ventana de resultados del cálculo de la Probabilidad de Detección donde salen los cálculos de las principales categorías de RBAT. La **Figura 2.54** muestra los gráficos de barras de la Probabilidad de Detección vs altitud en pies de las tres categorías más importantes de RBAT las cuales son Seguimiento (Beacon), Búsqueda (Search) y Total.

Target	Track	-Probability	Detect-	--Search	PD--	Id	Id	Alt	Alt	Md2	Md2	Nus	ATCRBS	Radar	Search	Range	Amath
Category	Life	Beacon	Search	Total	LogMtl	Mli	Rel	Conf	Rel	Conf	Rel	Conf	Mts	Reinf	Reinf	Collim	Error
DISCRETE	76995	95.34	0.00	95.34			99.83	100.00	99.32	100.00					0.00	0.02	0.19
SEM DISCRETE	1356	88.27	0.00	88.27			99.92	100.00	99.68	99.89					0.00	0.04	0.25
ATCRBS	78251	95.22	0.00	95.22			99.83	100.00	99.32	100.00					0.00	0.02	0.20
BEACON	78251	95.22	0.00	95.22			99.83	100.00	99.32	100.00					0.00	0.02	0.20
TOTAL	78251	95.22	0.00	95.22			99.83	100.00	99.32	100.00					0.00	0.02	0.20

Number of tracks included in the statistics = 618
 ATCRBS=618 (100.00 %) Mode S=0 (0.00 %) Mode2=0 (0.00 %) Search=0 (0.00 %) Mode 4=0 (0.00 %)
 Number of scans = 10831
 Uncorrelated reports: beacon = 3461 search = 0 Mode 4 = 0
 Coasted tracks: beacon = 3745 search = 0 Mode 4 = 0
 Uncorrelated reports per scan: beacon = 0.32 search = 0.00 Mode 4 = 0.00
 Coasted tracks per scan: beacon = 0.35 search = 0.00 Mode 4 = 0.00
 Uncorrelated reports plus immature track updates per scan: beacon = 0.48 search = 0.00 Mode 4 = 0.00

Gap*	NumGaps	PctCoasts	PctGaps
1	1762	45.45	65.92
2	597	31.88	23.12
3	283	22.67	10.96

Average gap size = 1.450
 * A gap is defined as the number of consecutive coasts

Figura 2.53: Ventana de resultados del cálculo de la Probabilidad de Detección

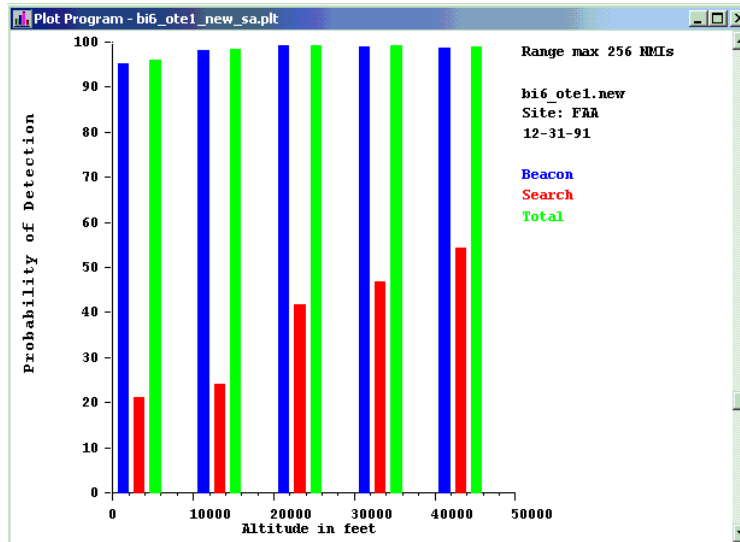


Figura 2.54: Gráficos de barras de la Probabilidad de Detección vs altitud en pies

2.14.2.5 Entradas de usuario del Análisis de Vigilancia

Las siguientes dos Figuras (Figura 2.55 y Figura 2.56) muestran algunas de las cajas de dialogo de RBAT de entrada de usuario:

Site Parameters: ELW

Radar type, beacon, search
 ATCBI6 En Route, ATCBI6, ARSR1/2 or FPS or BOS

Timing message
 Beacon RTQC Beacon Sector Mark
 Search RTQC Search Sector Mark

Mode 2
 Mode 2 SFN # Hits

Elevation of antenna above MSL (FT) 150.0 Use adjusted time
 Maximum range (NMs) 256.0 Is beacon delay on
 Latitude in DD:MM:SS 039:35:20.73 N
 Longitude in DD:MM:SS 074:41:53.80 W
 Magnetic declination (DEGs) 0.0

Cancel Back OK Help

Figura 2.55: Caja de Dialogo de los parámetros de sitio de RBAT

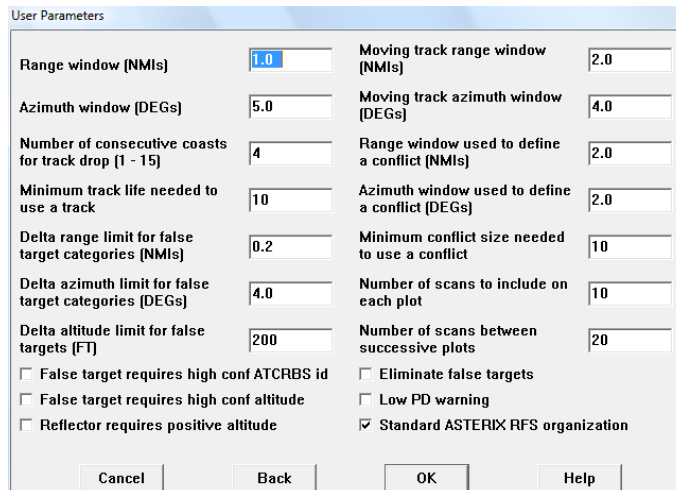


Figura 2.56: Caja de Dialogo de los parámetros de usuario de RBAT

Las cajas de diálogo siguientes son desplegadas al usuario:

- **Selección del Sitio:** Contiene los sitios que se graban en el archivo extracción. Se despliega si hay más de un sitio. En ese caso la caja de diálogo contiene un checkbox para cada sitio. Si el sitio se verifica, ese sitio se analizará; de lo contrario será ignorado. El valor por defecto es que todos los sitios están chequeados.
- **Origen del Multi Sitio:** Contiene la latitud, longitud y altitud de un sitio central. Esto sólo se muestra si un archivo de extracción multi sitio se usa y la caja "Múltiples sitios referenciados a un origen" esta chequeada.
- **Parámetros de Sitio:** Contiene el tipo de radar, mensaje cronometrado, Modo 2, uso de ajuste de tiempo, es retraso de seguimiento encendido, la elevación de antena sobre el nivel del mar, rango máximo, latitud, longitud, declinación magnética, estado on/off de transponder fijo, tipo de transponder fijo, rango de transponder fijo, acimut de transponder fijo, el id ATCRBS de transponder fijo y id Modo S del transponder fijo.
- **Parámetros de Usuario:** Contiene el número de prueba, mínima (y máxima) altitud para plots de altitud, tamaño de muestra de celda mínimo, uso de fiabilidad inversa y definiciones de confianza, ventana de rango, ventana de acimut, número de costas consecutivas o gota de pista, mínima vida de pista necesaria para usar una pista, elimina blancos falsos, advertencia de baja PD y organización normalizada ASTERIX RFS.
- **Filtrado:** Contiene el tiempo, rango, acimut, altitud, ángulo de elevación y filtros de número de pista.

- **Opciones de Plot:** Contiene los estados de on/off, colores y símbolos para cada una de las selecciones siguientes. [0, 40), [40, 75), [75, 80), [80, 85), [85, 90), [90, 95), [95, 98), y [98, 100) son intervalos de probabilidad de detección y su color es seleccionable. ATCRBS reforzado, Modo S reforzado, Modo 2, Búsqueda, Modo 4, ATCRBS sin reforzar, Modo S sin reforzar y Seguimiento radar son tipos de blanco y tienen colores seleccionables. Los informes falsos, inicios de pista, actualizaciones de pista y costas de pista son tipos de informe y tienen símbolos seleccionables. Por ejemplo, un inicio de pista de una pista reforzada ATCRBS usa el color para ATCRBS reforzado y el símbolo para el inicio de pista. La última selección es "sólo inicios tarde". Esto significa que sólo pistas que se comenzaron más de tres exámenes la primera vez en el archivo de extracción son trazadas. La razón para esto es diferenciar entre esas pistas que estaban en el espacio aéreo cuando se grabaron los datos de comienzo y aquéllos que se comienzan después.

2.14.2.6 Salida de texto del Análisis de Vigilancia

Hay una salida de advertencia de baja Probabilidad de Detección (PD), un mensaje de salida descorrelacionado, una salida de inscripción, una salida de historial, una salida resumen y una salida total. Si la opción sin correlación se selecciona, los informes sin correlación son enlistados. Si la opción de la inscripción se selecciona, los informes correlacionados y pistas costeadas son enlistados.

El mensaje de advertencia de baja PD es una salida en cada examinación que la probabilidad de detección de seguimiento está debajo de un umbral. El umbral depende de la carga de blanco. Tiene 75 % de 10 a 49 blancos, 90 % de 50 a 99 blancos y 95% para 100 o más blancos.

El mensaje descorrelacionado y el formato de salida de inscripción es el mismo y podemos enlistar los títulos de columna más importantes:

- Tiempo de día – tiempo de día en HH:MM:SS.
- Rango – el rango en millas náuticas.
- Azmth – el acimut en grados.
- Velocidad – la velocidad en millas náuticas / hora. La velocidad se calcula usando el informe actual y el informe más viejo guardado en la pista (normalmente se guardan 5 informes).
- Tipo – ATCRBS (A), Modo S (S), Modo 2 (2), búsqueda (R), o Modo 4 (4). El tipo de campo es Modo 2 si y sólo si el blanco se traza como una pista Modo 2.
- SFN–el número de archivo de vigilancia. Este campo sólo se usa si el parámetro de usuario Modo 2 es SFN.
- RT– el tipo de informe (el presente en el hit de seguimiento campo de cuenta si el modo es Modo S).
- Indx – índice de actualización con respecto a una pista. El primer correlacionado mensaje de Seguimiento, Búsqueda o AIMS. Para cada pista es asignado un

índice de 1. Cada subsiguiente correlacionado mensaje de Seguimiento, Búsqueda, o AIMS o la costa de pista se asigna un índice uno mayor que el anterior. Este campo está en blanco para los mensajes sin correlación.

- FTg – categoría de blanco falso. Si el blanco es real, este campo está en blanco; por otra parte es la categoría de blanco falso por el resumen de Blanco Falso de seguimiento determinada por Almenara el Resumen Designado Falso y escrito para el archivo de extracción.
- St – estado de campo. Si éste es un informe, este campo es iniciado (IN), actualización (UP), o está en blanco. Dos mensajes de Seguimiento que correlacionen son necesarios para comenzar una pista de Seguimiento y tres mensajes de Búsqueda (AIMS) que correlacionen son necesarios para comenzar una pista de Búsqueda (AIMS). Para estos mensajes, este campo es IN. Para un informe que se usa para actualizar una pista, este campo está UP. Para informes que no ponen en correlación a una pista, este campo está en blanco. Si esta es una pista costeadada, este campo es CO.
- Pista – el Análisis de Vigilancia el número de pista interior. Este campo está en blanco si y sólo si éste es un informe sin correlación.
- Mds-Id–el id Modo S en hexadecimal. Este campo sólo se usa si el formato del archivo es ASTERIX.
- A-Cd– el código Modo 3/A en octal. Si el código Modo 3/A no empareja la pista, hay un asterisco en el código Modo 3/A.
- Conf–Si éste es un mensaje de Seguimiento, este campo es la confianza del código Modo 3/A en octal. Si éste es un mensaje de Búsqueda, este campo es la confianza del radar. Por otra parte está en blanco.
- RptAlt – La altitud si el informe en pies. Si la altitud no empareja la pista, hay un asterisco después la altitud.
- TrkAlt–La altitud de la pista en pies. Este campo está en blanco a menos que éste es un mensaje de Seguimiento (o mensaje de Búsqueda con altura 3D) que es correlacionado a una pista.
- Conf – Confianza de código Modo C en octal.
- 2-Cd– Código Modo 2 en octal.
- Conf –Confianza de código Modo 2 en octal.
- ARTS – ARTS 3A calidad del informe y pista en decimal, respectivamente.
- Ampltde–la amplitud del informe y rastrea en decimal, respectivamente.
- ElAng– Angulo de elevación en grados.
- Wx–el tiempo nivelado para la región en la que el informe esta. El nivel de tiempo es ninguno (espacio en blanco), luz (L), medio (M), o pesado (H).
- AIMS–D1/D2 bits.

Hay otras ocho salidas, es decir la categoría de blancos total, la tabla de corrección de colimacion, el resumen de cobertura, el resumen de velocidad, el historial de blanco, el historial de posición, el historial de SFN y el resumen multi sitio. Para la categoría blancos totales, los blancos se agrupan en ATCRBS blancos discretos, ATCRBS blancos no-discretos, blancos ATCRBS, blancos Modo S, sólo blancos Modo 2, blancos de Seguimiento, blancos de Búsqueda, Total y blancos Modo 4. Un blanco es discreto si y sólo si todos los ids ATCRBS de la pista son discretos. El resumen de la tabla de corrección de

colimación y el resumen de cobertura tiene una opción de solo totales. Para los totales sólo la salida de la tabla de corrección de colimación, se dan el promedio de rango y colimación de acimut y tamaño de la muestra. Para los totales sólo salida del resumen de cobertura, la cobertura se da como una función solo de rango, como una función solo de acimut, como una función solo de altitud y como una función solo de ángulo de elevación.

Entre las cabeceras de columna más importantes de la categoría de blancos totales tenemos:

- Categoría de blanco
- Vida de Pista
- Probabilidad de detección de Seguimiento
- Probabilidad de detección de Búsqueda
- Probabilidad de detección Total
- Definiciones Normal (fiabilidad de identidad, confianza de identidad)
- Definiciones Inversa (fiabilidad de identidad, confianza de identidad)
- ATRBS reforzado
- Radar reforzado
- Colimación de Búsqueda
- Error de rango
- Error de acimut

Las estadísticas AIMS contienen las siguientes cabeceras:

- AIMS – AIMS (D1, D2) bits de 0 a 3. Para los archivos ASTERIX los bits AIMS FOE/FRI son usados en lugar de los bits D1/D2. Los bits ASTERIX FOE/FRI son mapeados en los bits D1/D2 como sigue: 0 (ninguna interrogación Modo 4) y 3 (ninguna respuesta) son mapeados en AIMS no presente, 1 (blanco amistoso) es mapeado en 3 (verdadero) y 2 (blanco desconocido) es mapeado en 0 (nulo).
- Conteo – El número de veces que los bits AIMS están presentes e igualan al valor en la columna AIMS.
- Porcentaje – El porcentaje de veces que los bits AIMS están presentes e igualan al valor en la columna AIMS.

2.14.2.7 Salida de plot del Análisis de Vigilancia

Hay cuatro tipos de plots. El primer tipo de plots contiene plots bidimensionales de la probabilidad de detección de Seguimiento, Búsqueda o Total o proporción de refuerzo radar. Los plots de la probabilidad de detección de Búsqueda y Total están presentes si y sólo si los mensajes de Búsqueda están presentes en el archivo de extracción. La proporción de refuerzo de radar sólo se grafica como una función de rango y altitud. Estos plots contienen cualquier celda de rango y altitud o celdas de rango y acimut. La probabilidad de detección está en blanco para celdas que no contienen ningún dato. Por otra parte la celda es coloreada basada en un intervalo de valores de porcentaje. Las graficas de rango y altitud

muestran la curvatura de la superficie de la tierra y las líneas de ángulo de elevación. Los plots de rango y acimut son plots polares y son divididos en plots separados por cada 10000 pies entre 0 y 50000 pies.

El segundo tipo de plot contiene plots de una dimensión de la probabilidad de detección de Seguimiento, Búsqueda y Total. Las estadísticas de la probabilidad de detección de Búsqueda y Total son presentadas si y solo si los mensajes de Búsqueda están presentes en el archivo de extracción. Estos plots contienen la probabilidad de detección versus altitud, versus ángulo de elevación, versus rango, versus acimut y versus velocidad. La probabilidad de detección se dibuja como una barra y la barra es azul para Seguimiento, roja para Búsqueda, y verde para Total.

El tercer tipo de plot contiene plots bidimensionales de las pistas e informes falsos. Los plots son rango versus acimut, rango versus altitud y acimut versus altitud. Estos contienen las posiciones de los inicios de pista, actualizaciones, costas e informes falsos para ATCRBS reforzado, Modo S reforzado, Modo 2, Búsqueda, Modo 4, ATCRBS sin reforzar, Modo S sin reforzar y blancos de Seguimiento radar. Los colores y símbolos usados para plot son seleccionados por el usuario. Por ejemplo, un inicio de pista de una pista ATCRBS reforzado usa el color de ATCRBS reforzado y el símbolo para inicio de pista. Si “Late inits only” es seleccionado, solo pistas que se iniciaron más de tres exámenes después del tiempo inicial en el archivo extracción son ploteadas. La razón de esto es diferenciar entre esas pistas que estaban en el espacio aéreo cuando la grabación de datos empezó y aquellas que se inician después. Las pistas pasajeras no son ploteadas. Una pista pasajera es una pista que tiene una vida de pista menos que la vida de pista mínima especificada por el usuario. Las costas al final de la dirección de pista a una huella no son ploteadas. Si la opción mapa es seleccionada un mapa con el límite de estado y país es sobre puesto en los plots rango versus acimut.

El cuarto tipo de plot es presentado solo si el archivo extracción contiene más que un sitio y el usuario especifica un origen común. Contiene plots de la probabilidad de detección (PD) de Seguimiento y mejores sitios. Los plots son desplegados en una rejilla X/Y por cada 10000 pies de altitud en un intervalo de 0 a 50000 pies. Cada celda en la rejilla contiene también la mejor PD de Seguimiento o el mejor sitio. Las celdas que están en blanco significa que no contienen ningún dato. De otra manera la celda es coloreada basada en un intervalo de valores de PD de colores de sitio.

2.14.2.8 Archivo de extracción de salida del Análisis de Vigilancia

Este es un archivo opcional que es escrito si la opción “salida” es seleccionada. El archivo es similar al archivo extracción de entrada excepto que cada mensaje tiene un número de pista almacenado en él. Esos mensajes que no son graficados y esos mensajes que son graficados pero no son incluidos en las estadísticas tienen un número de pista de cero. Si los parámetros de usuario “mínima vida de pista necesaria para usar pista en estadísticas” es puesta a uno. Todos los mensajes graficados son incluidos en las estadísticas y por lo tanto escritos en el archivo de salida.

2.15 BIBLIOGRAFIA

<http://www.eurocontrol.int/asterix>

The Asterix Estandar, Eurocontrol

Eurocontrol standard document for Surveillance data exchange, part 1,2a,2b

Ayuda de la aplicación Software WIRESHARK

Ayuda de la aplicación Software WRTADS

Ayuda de la aplicación Software RBAT

CAPITULO III

DESCRIPCION DEL SISTEMA PROPUESTO

Introducción

Las herramientas software WRTADS y RBAT se utilizan para evaluar el desempeño general de los sensores radar, no obstante no brindan información detallada sobre el contenido de las tramas Asterix y su relación con los conceptos básicos radar, para que un estudiante que no esté familiarizado con estos conceptos, los puedan utilizar. Para la comprensión de conceptos básicos es importante contar con un sistema didáctico de fácil utilización orientado a la enseñanza-aprendizaje. Por lo anterior se desarrollo una aplicación software que se llamo Analizador de Protocolo Asterix de Eurocontrol (ADPADEC). Este software es capaz de capturar tramas Asterix que fluyan dentro de una red LAN y una vez capturadas se pueden desarrollar diferentes análisis; los resultados de dichos análisis se presentan de forma grafica para una mejor comprensión. Entre los análisis e información que se pueden realizar destacan: seleccionar tramas individuales y extraer de ellas la información que contienen a partir del análisis de cada uno de los bytes que la conforman, análisis de probabilidad de detección y falsos blancos, presentación de datos en forma grafica en una pantalla que simula un presentador de datos radar o PPI. En el presente capitulo se describirá de forma detallada el software desarrollado.

3.1 CARACTERISTICAS GENERALES DE LA APLICACION

El software ADPADEC permite:

- i. Capturar tramas ASTERIX de categorías 001 y 002 que fluyen dentro de una red LAN. El tiempo de captura es variable y está definido por el usuario.
- ii. Decodificar tramas basándose en la interpretación de códigos establecida en el protocolo Asterix de Eurocontrol.
- iii. Realizar diferentes cálculos de probabilidad de detección tanto para blancos individuales como para un conjunto de blancos.
- iv. Visualizar de manera grafica cada uno de los blancos capturados.
- v. Obtener estadísticas de los blancos capturados.
- vi. Capturar datos, utilizando el motor de captura **Tshark**, uno de los programas de Wireshark, que captura archivos de tráfico de red con la extensión *pcap*.
- vii. Convertir los archivos capturados, de formato *pcap* a *txt*, por medio del programa **Mergecap**, otro de los programas de **Wireshark**.

3.2 CRITERIOS DE DISEÑO DE LA APLICACION

El software didáctico desarrollado, debía cumplir con los siguientes criterios de diseño:

- Capacidad para capturar tramas Asterix que fluyan dentro de una red LAN.
- Capacidad para decodificar tramas Asterix, basándose en la interpretación de códigos establecida en el protocolo Asterix de Eurocontrol.
- Calcular la probabilidad de detección de códigos, probabilidad de detección total (conjunto de blancos) y probabilidad de detección de falsos blancos.
- Permitir visualizar de manera grafica cada uno de los blancos capturados, en una pantalla que simule un presentador radar.
- Calcular estadísticas de los blancos capturados, como por ejemplo el número de respuestas con Alta Confianza de Modo C y Alta Confianza de Modo 3/A.

Al analizarse los requerimientos establecidos, se concluyó que debían desarrollarse módulos, en un lenguaje de programación que permitiera el uso de interfaces graficas de usuario, que pudiera hacer los cálculos requeridos de manera rápida y permitiera el uso de otros programas de manera eficiente. Es por estas razones que el lenguaje elegido fue **Python**, ya que nos permite utilizar otros programas como recursos, que son mucho más eficientes en **Linux** y **Unix**, por lo que el software se diseño, para esos sistemas operativos. La versión de **Python** mas explotada, para la cual hay una gran cantidad de módulos desarrollados por diversos programadores, es la 2.6, por lo que el software se desarrolló en esa versión.

Los módulos desarrollados, se presentan a manera de algoritmos en forma de flujogramas en bloques.

3.2.1 Algoritmo del software ADPADEC

En la **Figura 3.1** se muestra el flujograma en bloques del algoritmo del software ADPADEC. Se aprecia que el algoritmo de interface grafica de usuario, es el que permite poder ejecutar cada uno de los seis tipos de algoritmos (tipos de análisis).

El flujograma ilustra que el algoritmo de procesamiento para análisis, permite realizar los análisis de probabilidad de detección, visualización radar y estadísticas, ya que el resultado es un archivo *txt* con tramas Asterix decodificadas, que es usado por los algoritmos que calculan probabilidad de detección, permiten la visualización radar y el cálculo de las estadísticas de los blancos. El resultado de una captura puede ser un archivo de formato *pcap* ó *txt*. El resultado del algoritmo de extracción y decodificación, es independiente del algoritmo de procesamiento para análisis.

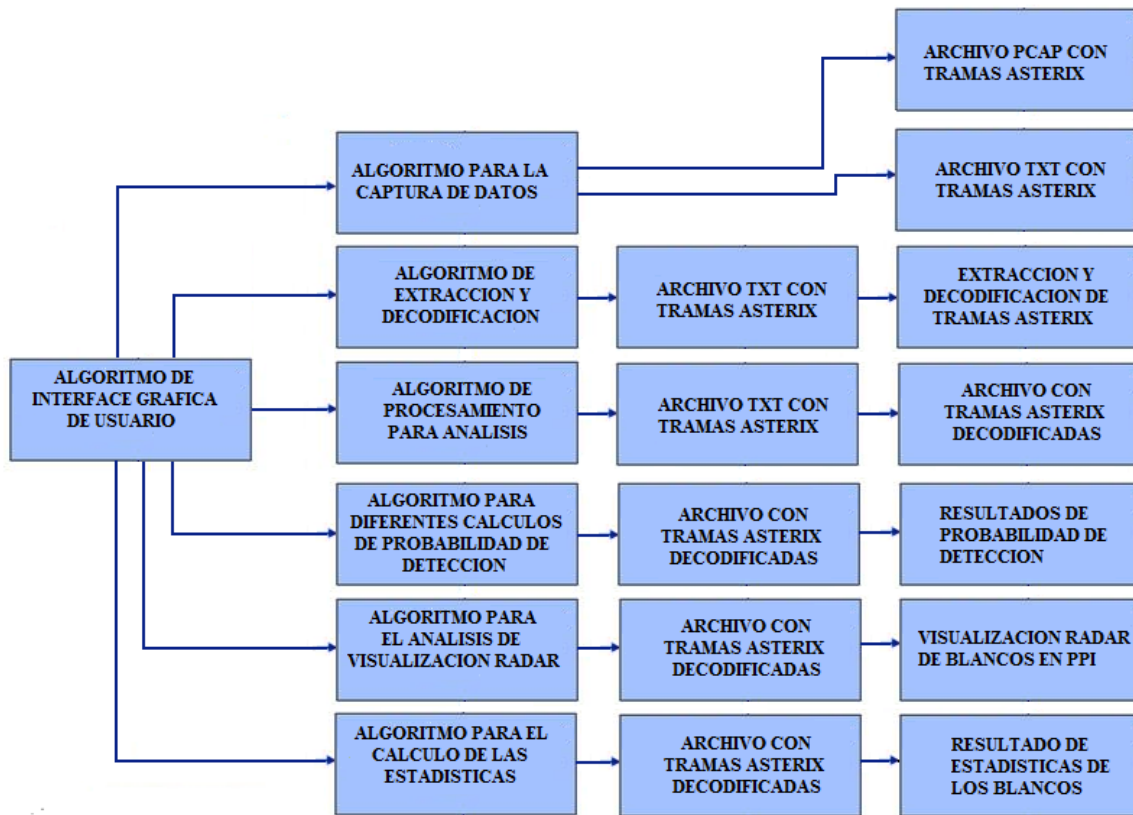


Figura 3.1: Flujograma en bloques del algoritmo del software ADPADEC

3.2.2 Algoritmo de interface grafica de usuario

El algoritmo de interface grafica de usuario, lo muestra el flujograma en bloques de la **Figura 3.2**, en donde cada bloque representa lo mínimo que debía contener, según los criterios de diseño de la aplicación.

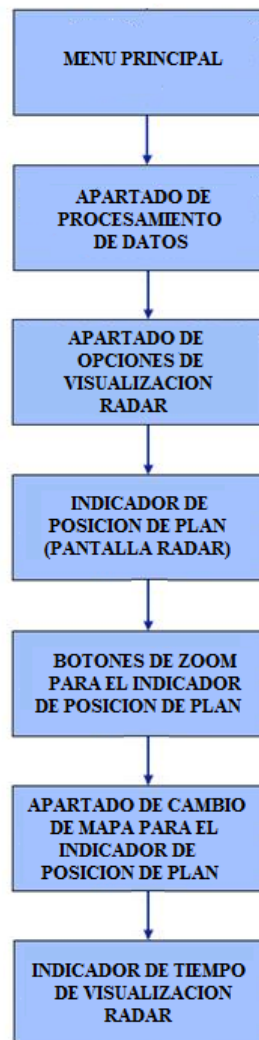


Figura 3.2: Flujograma en bloques del algoritmo de interface grafica de usuario

3.2.3 Algoritmo para la captura de datos

El algoritmo para la captura de datos, es el que se muestra en la **Figura 3.3**. En dicha Figura podemos observar, que lo primero que realiza el algoritmo, es programar el número de horas y minutos para la captura de datos, por parte del usuario. Una vez programado las horas y minutos, se tienen dos opciones: captura de datos en un archivo *pcap* ó captura y conversión de datos.

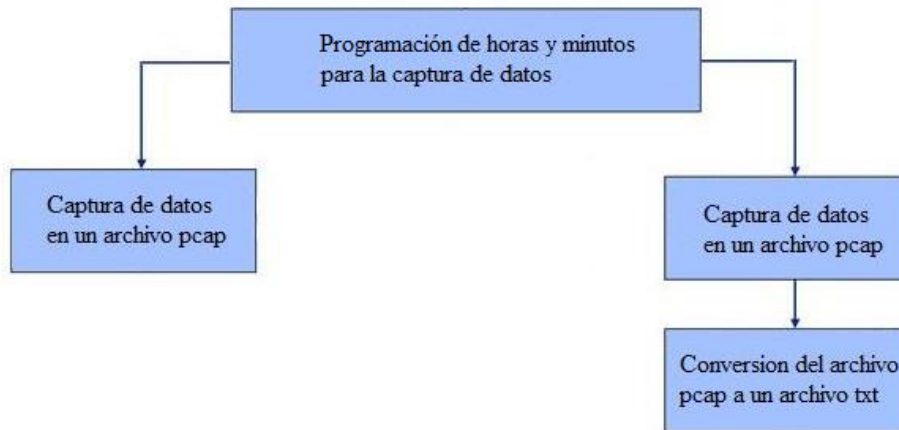


Figura 3.3: Flujograma en bloques del algoritmo para la captura de datos

3.2.4 Algoritmo de extracción y decodificación

El algoritmo de extracción y decodificación, se divide en: cinco algoritmos para categoría 01 y cinco algoritmos para categoría 02. El primer algoritmo para la búsqueda de tramas Asterix de categoría 01, caso uno es el que muestra el flujograma en bloques de la **Figura 3.4**. En esta Figura puede observarse que primero se busca en cada paquete: la categoría 01 que está compuesta de un octeto; la longitud de la trama Asterix que tendrá dos octetos; el FSPEC de un octeto; el código SAC hasta terminar la trama con el último octeto, según la longitud de la trama Asterix. Si no se cumple con la secuencia, desde la categoría hasta el código SAC, se vuelve a buscar desde la categoría. Es de hacer notar que el octeto de la categoría tendrá los números hexadecimales 0 y 1. Los dos octetos de la longitud de la trama Asterix, solo podrán tener números hexadecimales de 0 a F. El FSPEC para el caso 1 es de un octeto, en donde la primera posición del octeto, son números hexadecimales de 8 a F y la segunda posición del octeto, son números hexadecimales pares. La primera posición del octeto del código SAC, solamente admite los números hexadecimales E o F y la segunda posición del octeto los hexadecimales 1, 6, 9, C y E. La trama termina según su longitud con un último octeto, que puede tener en sus dos posiciones hexadecimales de 0 a F.

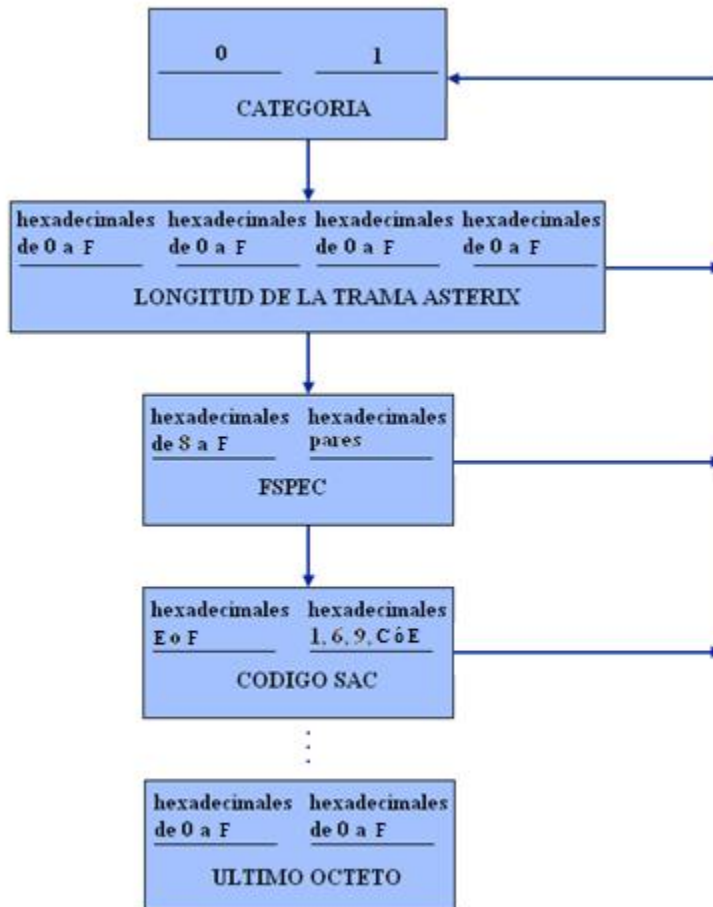


Figura 3.4: Caso 1 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01

El segundo algoritmo para la búsqueda de tramas Asterix de categoría 01 caso 2, es el que muestra el flujograma en bloques de la **Figura 3.5**, que como la **Figura 3.4**, se observa que la trama Asterix en el caso 2, únicamente cambia el FSPEC, que para este caso es de dos octetos. La primera posición del FSPEC, tendrá solamente hexadecimales de 8 a F. La segunda posición hexadecimales impares, la tercera posición hexadecimales de 0 a F y la cuarta posición hexadecimales pares.

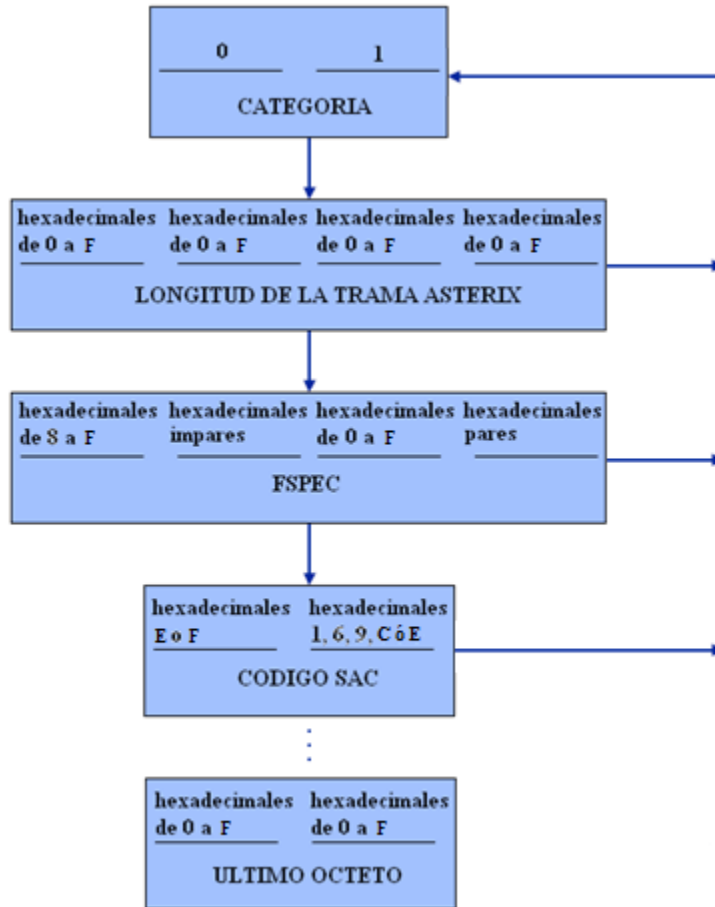


Figura 3.5: Caso 2 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01

El tercer algoritmo para la búsqueda de tramas Asterix de categoría 01 caso 3, es el que muestra el flujograma en bloques en la **Figura 3.6**, al igual que en la **Figura 3.5**, se observa que la trama Asterix en el caso 3, cambia en el FSPEC, que para este caso es de tres octetos. La primera posición del FSPEC, tendrá solamente hexadecimales de 8 a F, la segunda posición hexadecimales impares, la tercera posición hexadecimales de 0 a F, la cuarta posición hexadecimales impares, la quinta posición hexadecimales de 0 a F y la sexta posición hexadecimales pares.

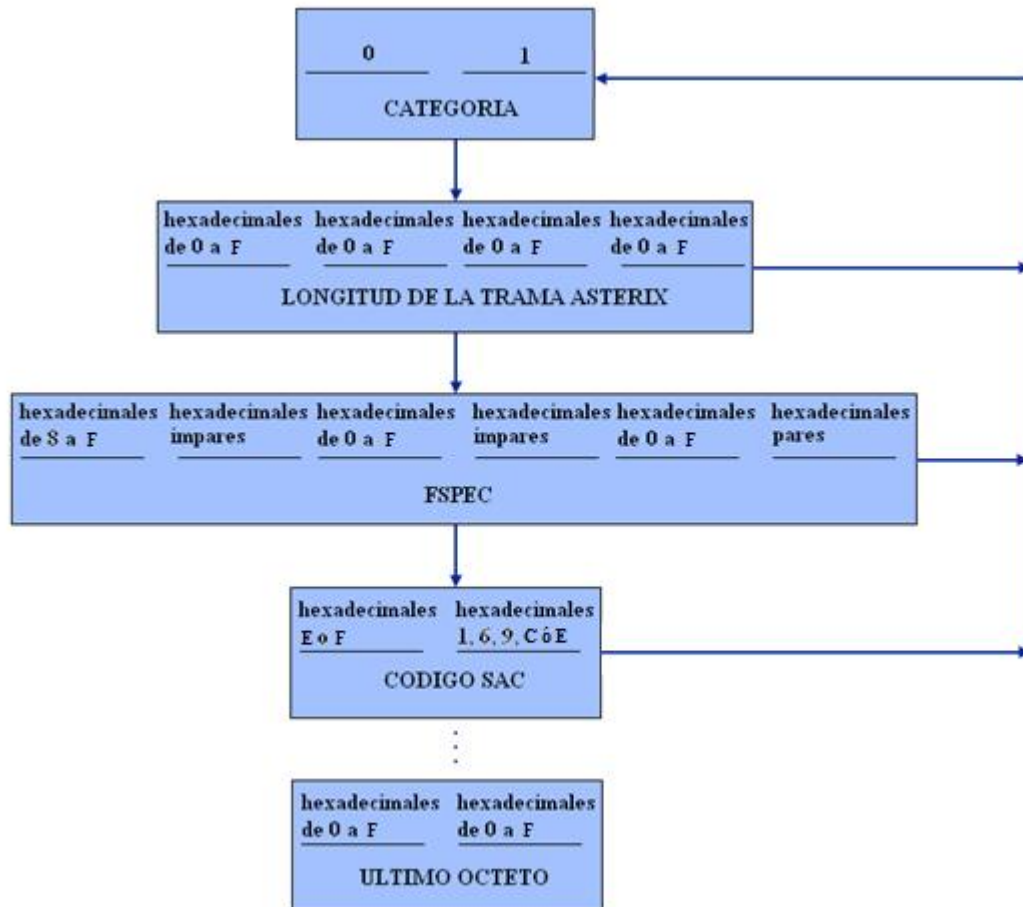


Figura 3.6: Caso 3 flujograma en bloques del algoritmo de búsqueda de tramas Asterix de categoría 01

Una vez se han encontrado las tramas Asterix, se procede a la decodificación de cada una, en base a la categoría Asterix. Para la categoría 01, se utilizó el cuarto algoritmo para la decodificación de plots radar, como muestra el flujograma en bloques en la **Figura 3.7**, en ella se muestra cada una de las partes, de las cuales se compone el algoritmo de decodificación de plots Asterix de categoría 01, en donde cada parte está compuesta por un número determinado de bytes u octetos. Es de hacer notar que el FIELD SPECIFICATION ó FSPEC, es el que define cuales son los Item de datos que se encuentran en la trama, por lo que su longitud máxima es de 3 bytes. El algoritmo solo contendrá, aquellos Items de Datos que han sido definidos por el FSPEC, por ejemplo una trama podría tener el Item de Datos Modo C, pero no tener el RADAR PLOT CHARACTERISTICS, que sería el siguiente Item de Datos en la secuencia, pero si tiene el Item de datos TRUNCATED TIME OF DAY, por lo que este sería el siguiente en la secuencia. La **Figura 3.7** es la secuencia si estuvieran todos los Items de Datos.

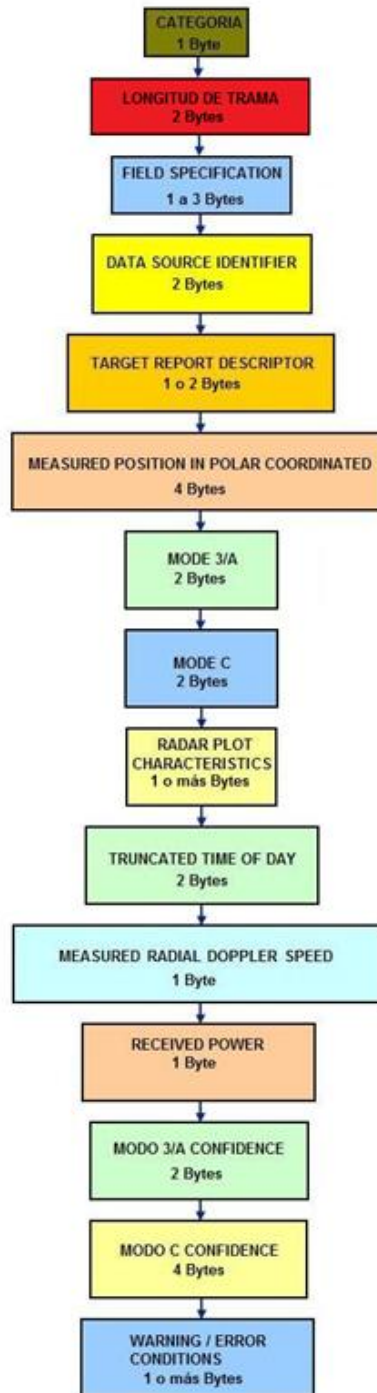


Figura 3.7: Flujograma en bloques del algoritmo de decodificación de plots Asterix categoría 01

Para la decodificación de pistas radar de categoría 01, se utiliza el quinto algoritmo que muestra el flujograma en bloques en la **Figura 3.8**. En este flujograma en bloques se puede apreciar, al igual que el flujograma en bloques de la **Figura 3.7**, cada una de las partes de las cuales se compone el algoritmo de decodificación de pistas Asterix de categoría 01, en donde cada parte está compuesta por un número determinado de bytes u octetos.

Para los casos 1, 2 y 3 el algoritmo de búsqueda de tramas Asterix de categoría 02, son similares a los casos 1, 2 y 3 del algoritmo de búsqueda de tramas Asterix de categoría 01, con la única diferencia que el primer octeto contiene los hexadecimales 0 y 2. En el caso de los algoritmos de decodificación de plots y pistas Asterix de categoría 02, son similares a los de la categoría 01, con la única diferencia que el primer octeto contiene los hexadecimales 0 y 2.

3.2.5 Algoritmo de procesamiento para análisis

Este algoritmo es el que se encarga del procesamiento y decodificación de datos radar para análisis, es decir que el archivo que se ha capturado con extensión *pcap*, es decodificado a un archivo de texto con tramas Asterix, para que los análisis de “PROBABILIDAD DE DETECCION”, “VISUALIZACION RADAR” y “ESTADISTICAS” puedan ser realizados.

Una vez que se ha realizado la transformación del archivo de formato *pcap* a *txt*, el algoritmo busca el primer mensaje de Norte que está en el archivo *txt*, el cual servirá para la sincronización de tiempo, partiendo del tiempo que tiene este mensaje (en horas, minutos y segundos), con el objetivo de incluir los blancos radar a partir de ese tiempo, hasta que finalice la captura de datos (tiempo registrado en el archivo de texto convertido). Para la búsqueda de este mensaje de Norte se utiliza un algoritmo similar, al de la búsqueda de tramas Asterix de Categoría 01 (**Figura 3.4**), con la única diferencia que este algoritmo comienza con el octeto de categoría 02, seguido de la longitud de la trama, el FSPEC de un octeto que define los Item de datos, código SAC, código SIC, código de mensaje de Norte (01 hexadecimal) y todos los octetos que completen la longitud de la trama Asterix que define el FSPEC.

Cuando el primer mensaje de Norte es encontrado, se procede a la decodificación de este mensaje, el algoritmo utilizado para la decodificación de este mensaje, es el flujograma en bloques que muestra la **Figura 3.9**, que puede decodificar otros tipos de mensajes de la categoría 02. En este algoritmo se puede apreciar que comienza siempre con el octeto de la categoría y que siempre debe cumplir con un cierto orden de octetos que identifican la trama del tipo de mensaje. Si no se cumple con los primeros seis bloques, con uno que falle, el algoritmo comienza nuevamente, con la búsqueda del primer bloque (el octeto de CATEGORIA).

Se hace la búsqueda y decodificación de tramas Asterix de categoría 01 con los algoritmos que se han visto anteriormente, con la diferencia de que los resultados se guardan, en un archivo de texto, a manera de tabla de datos con diez columnas y filas igual al número de tramas Asterix encontradas.

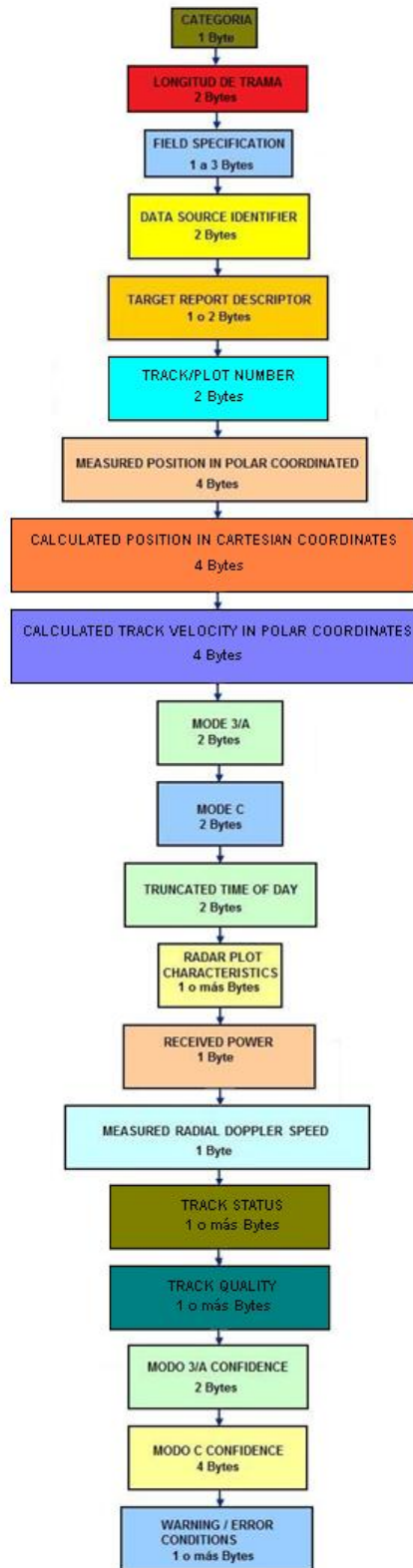


Figura 3.8: Flujograma en bloques del algoritmo de decodificación de pistas Asterix categoría 01

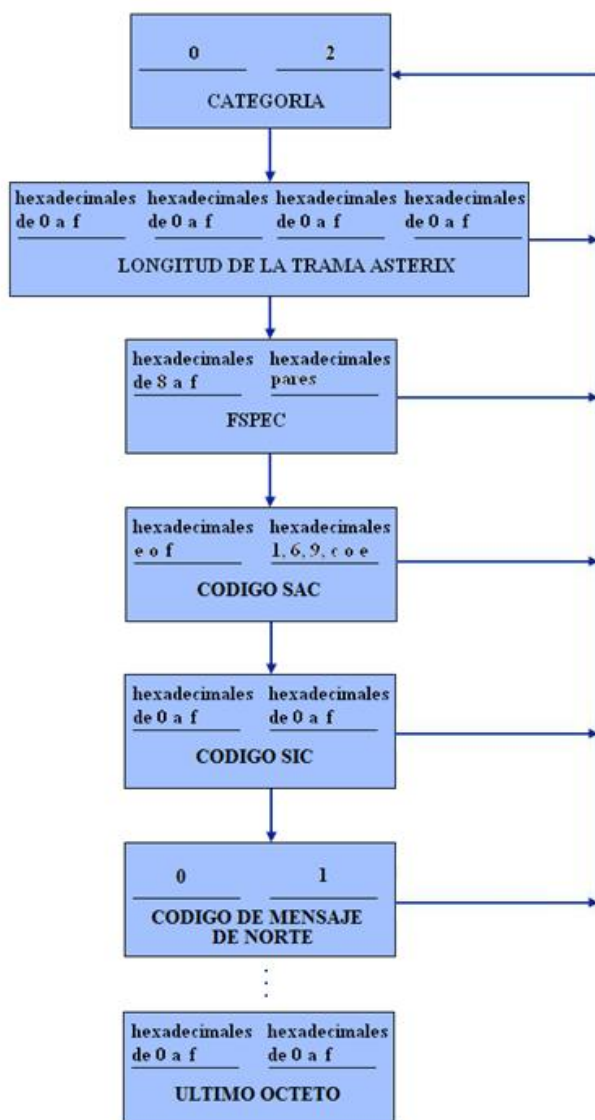


Figura 3.9: Flujograma en bloques del algoritmo para la decodificación de mensaje de Norte

3.2.6 Algoritmo para diferentes cálculos de probabilidad de detección

El algoritmo para el cálculo de la probabilidad de detección de códigos, toma en cuenta los siguientes criterios:

1. Se considera solo los plots y pistas con alta confianza de Modo A y Modo C, mayores que cero pies de altitud, pero menores que 50,000 pies de altitud (una aeronave comercial vuela en promedio a una altitud máxima de 49,000 pies, según normas de la OACI, Anexo 6: Operación de las Aeronaves Parte I: Transporte

Aéreo Internacional Comercial – Aeroplanos, debido a que se expone a la radiación cósmica).

2. Cuenta el número de plots y pistas con tiempo de captura diferente para un determinado blanco con un código transponder, eso será igual al número total de detecciones de ese blanco.
3. Si hay más de un plot y pista con diferente tiempo de captura con igual rango se toma solo un plot ó pista.
4. Verifica que el cambio en rango y acimut sea lógico y toma para las detecciones validas solo los plots y pistas con cambio en rango y acimut lógicos.

Una vez que los blancos cumplen los criterios anteriores, el algoritmo hace el cálculo de la probabilidad de detección, para un blanco radar aplicando la siguiente fórmula:

$$P_D = \frac{\text{Número de detecciones validas}}{\text{Número total de detecciones}} \times 100 \%$$

Hay que notar que el número de detecciones validas son todos aquellos plots y pistas para una determinada aeronave, que pasan por los cuatro criterios anteriormente mencionados. El número total de detecciones, son todas las detecciones del blanco (el número total de plots y pistas). Se espera que por cada vuelta de antena al menos haya una respuesta de la aeronave que genere en el PPI un plot radar. El algoritmo calcula la probabilidad de detección total con la siguiente fórmula:

$$P_D(\text{total}) = \frac{\sum P_D \text{ de los blancos radar}}{\text{Número total de blancos radar}} \times 100 \%$$

Para calcular la probabilidad de detección de falsos blancos, el algoritmo utiliza la siguiente fórmula:

$$P_D(\text{falsos blancos}) = \frac{\text{Número de respuestas de falsos blancos}}{\text{Número total de respuestas}} \times 100 \%$$

Se consideran falsos blancos los siguientes:

- Los blancos que solo aparecen una vez en el área de cobertura que tiene el radar.
- Los blancos que dejan al menos un plot que no forma una pista lógica (una pista lógica no es más que un historial de plots continuos, que describe el rumbo de una aeronave sobre un PPI).
- El código 0000, se considera falso blanco, porque debe ser asignado según acuerdos regionales de navegación aérea, Anexo 10 Volumen IV de la OACI.

3.2.7 Algoritmo para el análisis de visualización radar

El algoritmo utilizado para el análisis de visualización radar, es el que se muestra en la **Figura 3.10**.



Figura 3.10: Flujograma en bloques del algoritmo utilizado para el análisis de visualización radar

Se observa en la **Figura 3.10**, que el flujograma en bloques del algoritmo utilizado para el análisis de visualización radar, consta de tres procesos fundamentales:

- La búsqueda de pistas y plots en el archivo txt decodificado por el algoritmo procesamiento para análisis. Este archivo decodificado txt es el mismo que utiliza el algoritmo de probabilidad de detección, para obtener sus resultados y el algoritmo para las estadísticas.
- Sincronización de tiempo de pistas y plots con el primer mensaje de Norte.
- Mostrar pistas y plots en PPI.

3.2.8 Algoritmo para el cálculo de las estadísticas

El flujograma en bloques del algoritmo que calcula las estadísticas, del archivo de texto decodificado, se muestra en la **Figura 3.11**.

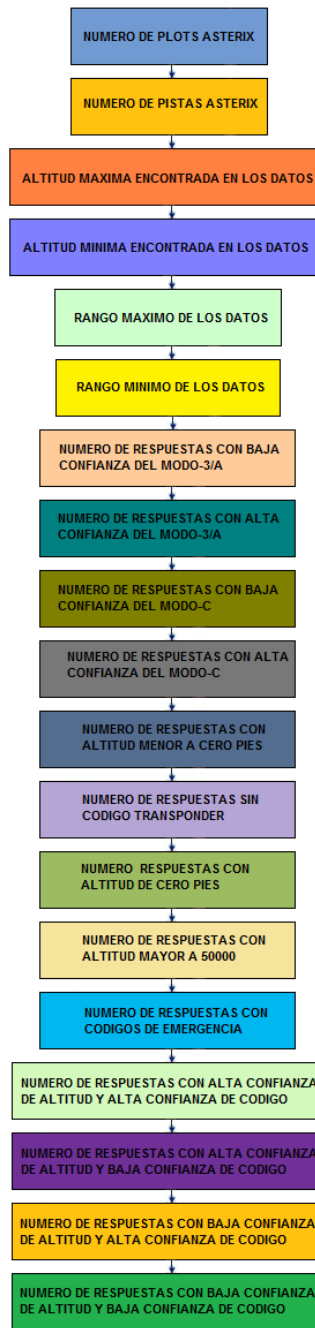


Figura 3.11: Flujograma en bloques del algoritmo para calcular las estadísticas

3.3 DESCRIPCION DE LA INTERFACE GRAFICA DE USUARIO

La interface grafica de usuario del software ADPADEC, se muestra en la **Figura 3.12**. Está diseñada de tal forma que el usuario pueda utilizarla fácilmente. El menú principal que se encuentra en la parte superior izquierda de la ventana principal, contiene cada uno de los seis tipos de análisis que realiza el software. Para ejecutar un determinado tipo de análisis, el usuario da doble click sobre la opción, de un determinado tipo de análisis en el menú principal.

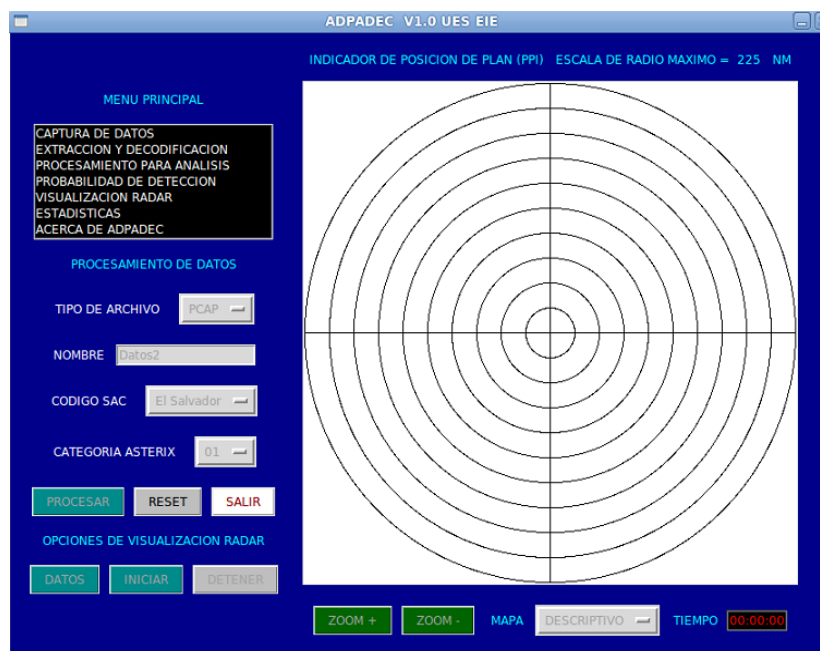


Figura 3.12: Interface grafica de usuario del software ADPADEC

La ventana principal contiene el apartado de “PROCESAMIENTO DE DATOS” que se activa dando doble click en la opción “EXTRACCION Y DECODIFICACION” ó “PROCESAMIENTO PARA ANALISIS” del menú principal, es donde el usuario especifica el tipo de archivo de la extracción y decodificación ó del procesamiento para análisis, que puede ser pcap o txt. Este apartado también contiene: el nombre del archivo, código SAC, categoría Asterix y los botones “PROCESAR”, “RESET” y “SALIR”. El apartado de “OPCIONES DE VISUALIZACION RADAR” contiene tres botones, el botón “DATOS” se activa dando doble click en la opción “VISUALIZACION RADAR” del menú principal. Al dar click sobre el botón “DATOS” aparece la ventana de “VISUALIZACION RADAR”, donde el usuario especifica los datos a ingresar para la visualización de los blancos en el PPI. El botón “INICIAR” da inicio al proceso de visualización radar y el botón “DETENER” finaliza el proceso. El Indicador de Posición de Plan (PPI) se encuentra en la parte derecha de la ventana principal. En la parte superior del PPI se indica además de su nombre la escala de radio máximo cuando los botones de ZOOM son presionados, ya sea el botón “ZOOM +” ó “ZOOM -”, que se encuentran en la parte inferior del PPI, en donde está también el menú “MAPA” y el indicador de tiempo de análisis.

3.4 DESCRIPCION DE LOS ANALISIS DE LA APLICACION

3.4.1 Captura de datos

Como se ha mencionado antes, el software ADPADEC utiliza para la captura de datos el programa **Tshark**, que captura archivos con extensión *pcap*, estos archivos contienen paquetes que fluyen por la red LAN. Contienen además su tiempo de captura y el tipo de paquete, así tenemos los siguientes tipos de paquetes: UDP, SSDP, ARP, ICMP, OSPF, STP, ICMPv6, IP, BROWSER, DHCP, NBNS, RIPv1, LLMNR, MDNS y CIGI.

El usuario da doble click en la ventana principal en “CAPTURA DE DATOS” y se despliega en pantalla la ventana de captura de datos que se muestra en la **Figura 3.13**, el usuario ingresa los siguientes datos que son:

- **NOMBRE DE ARCHIVO DE CAPTURA:** es el nombre que le da el usuario al archivo de captura de datos.
- **NUMERO DE ARCHIVO DE CAPTURA:** es el numero que el usuario le da al archivo de captura, con el propósito de tener un orden cuando existe más de un archivo con el mismo nombre.
- **TIEMPO DE CAPTURA DE DATOS:** en el campo “HORAS” se ingresa el número de horas de captura de datos, que puede sobrepasar las 24 horas. En el campo “MINUTOS” el usuario ingresa el numero de minutos de la captura de datos, que no puede sobrepasar los 59 minutos.

Una vez el usuario ha ingresado los datos del archivo de captura y el tiempo de captura, puede dar click en el botón “CAPTURAR” ó “CAPTURAR Y CONVERTIR”. La funcionalidad de los botones de la ventana de captura de datos es la siguiente:

- **CAPTURAR:** inicia el proceso de captura, según los datos ingresados, en los apartados “ARCHIVO DE CAPTURA DE DATOS” y “TIEMPO DE CAPTURA DE DATOS”.
- **CAPTURAR Y CONVERTIR:** inicia el proceso de captura, según los datos ingresados, en los apartados “ARCHIVO DE CAPTURA DE DATOS” y “TIEMPO DE CAPTURA DE DATOS”. Luego hace la conversión del archivo de captura de *pcap* a *txt*.
- **CERRAR:** cierra la ventana de captura de datos.
- **SALIR:** permite salir del sistema cerrando todas las ventanas.



Figura 3.13: Ventana de captura de datos del software ADPADEC

3.4.2 Extracción y decodificación

Este análisis es el que se encarga de la extracción y decodificación de los datos radar, que se encuentran en el archivo *pcap*, que contiene los paquetes con protocolo HDLC. Lo primero que realiza depende de que tipo de archivo va a procesar el usuario, así el usuario puede especificar en el apartado de “PROCESAMIENTO DE DATOS” de la ventana principal (**Figura 3.14**) los siguientes datos:

- **TIPO DE ARCHIVO:** este puede ser PCAP ó TEXT.
- **NOMBRE:** es el nombre del archivo a procesar.
- **CODIGO SAC:** es el sistema de código de área especificado en el Protocolo Asterix, que en este caso contiene en vez del numero hexadecimal el nombre de seis países Centroamericanos (El Salvador, Guatemala, Honduras, Belize, Nicaragua y Costa Rica).
- **CATEGORIA ASTERIX:** es la categoría Asterix que se va a procesar. El usuario puede elegir entre la 01 y 02.

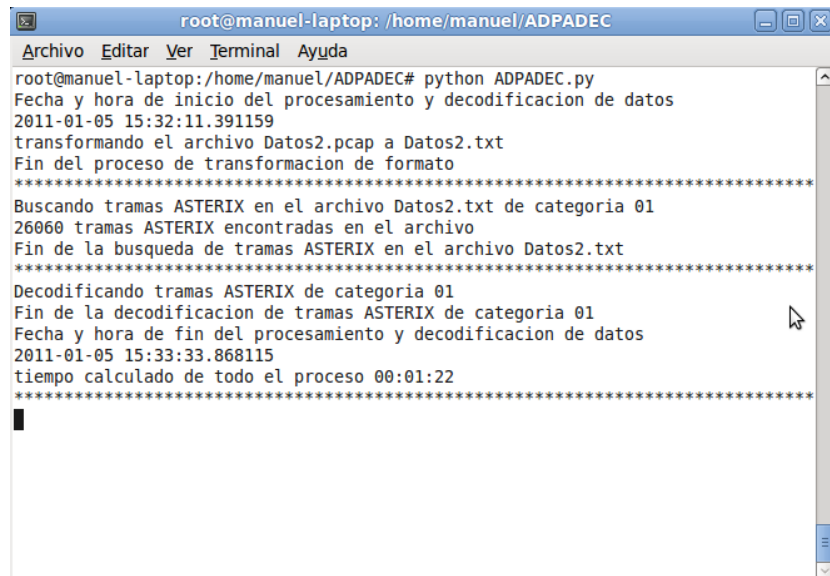


Figura 3.14: Apartado de procesamiento de datos

Una vez que el usuario ha especificado cada uno de los campos anteriores y da click en el botón “PROCESAR”, se inicia el proceso, ya sea de encontrar las tramas Asterix que se encuentran en el archivo con extensión *txt* ó convertir el archivo *pcap*, con el recurso **Mergecap** a un archivo de texto con extensión *txt*. Una vez que se encuentran las tramas Asterix de la categoría que especifico el usuario, que están en el archivo de texto, se procede a la decodificación de cada una de las tramas encontradas que se muestra en la **Figura 3.15**. Luego se muestra una ventana de resultados de la extracción y decodificación de cada una de las tramas Asterix, como se muestra en la **Figura 3.16**, en donde se puede apreciar que en la parte izquierda se encuentra una ventana con tramas Asterix. Si el usuario da doble click a una trama, los resultados de la decodificación le aparecen en el área de resultados, en la parte derecha de la ventana.

La funcionalidad de los botones del apartado “PROCESAMIENTO DE DATOS” es la siguiente:

- **PROCESAR:** inicia el proceso de la extracción y decodificación, según los datos ingresados, en el apartado “PROCESAMIENTO DE DATOS”.
- **RESET:** reinicia los tipos de análisis del sistema.
- **SALIR:** permite salir del sistema cerrando todas las ventanas.



```
root@manuel-laptop: /home/manuel/ADPADEC
Archivo Editar Ver Terminal Ayuda
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
Fecha y hora de inicio del procesamiento y decodificacion de datos
2011-01-05 15:32:11.391159
transformando el archivo Datos2.pcap a Datos2.txt
Fin del proceso de transformacion de formato
*****
Buscando tramas ASTERIX en el archivo Datos2.txt de categoria 01
26060 tramas ASTERIX encontradas en el archivo
Fin de la busqueda de tramas ASTERIX en el archivo Datos2.txt
*****
Decodificando tramas ASTERIX de categoria 01
Fin de la decodificacion de tramas ASTERIX de categoria 01
Fecha y hora de fin del procesamiento y decodificacion de datos
2011-01-05 15:33:33.868115
tiempo calculado de todo el proceso 00:01:22
*****
```

Figura 3.15: Informe de la extracción y decodificación de datos en consola Linux

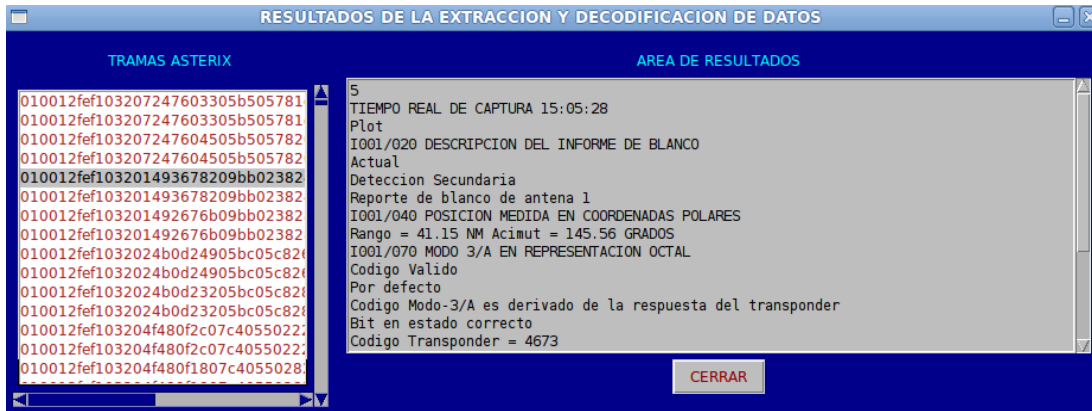


Figura 3.16: Ventana de resultados de la extracción y decodificación de datos

3.4.3 Procesamiento para análisis

Este análisis es el que se encarga del procesamiento y decodificación de datos radar para análisis, es decir que transforma el archivo que se ha capturado, en archivo de texto, para que los análisis de “PROBABILIDAD DE DETECCION”, “VISUALIZACION RADAR” y “ESTADISTICAS” puedan ser realizados. El usuario especifica en la ventana principal en el apartado “PROCESAMIENTO DE DATOS” los mismos datos ingresados para el análisis de “EXTRACCION Y DECODIFICACION”, excepto la categoría Asterix, la **Figura 3.17** muestra los datos que se ingresan.



Figura 3.17: Apartado de procesamiento de datos

La **Figura 3.18** muestra en consola **Linux**, el informe del procesamiento para análisis, que en realidad es procesamiento y decodificación de datos para análisis. Inicia con la fecha y hora en la cual se está realizando el análisis; luego hace una búsqueda y decodificación de la trama Asterix del primer mensaje de Norte en el archivo de texto, para la sincronización de tiempo de análisis; Se hace la búsqueda de tramas Asterix de categoría 01; La

decodificación de tramas Asterix de categoría 01 y finaliza el análisis, indicando el tiempo que se ha llevado en realizarlo.

```
root@manuel-laptop: /home/manuel/ADPADEC
Archivo Editar Ver Terminal Ayuda
Fecha y hora de inicio del procesamiento y decodificación de datos para analisis
2011-01-05 16:00:36.943090
transformando el archivo Datos2.pcap a Datos2.txt
Fin del proceso de transformacion de formato
*****
Buscando trama ASTERIX del primer mensaje de Norte
En el archivo Datos2.txt de categoria 02
1 mensaje de Norte encontrado en el archivo
Fin de la busqueda del primer mensaje de Norte
En el archivo Datos2.txt
Decodificando el mensaje de Norte
Fin de la decodificacion del mensaje de Norte
*****
Buscando tramas ASTERIX en el archivo Datos2.txt de categoria 01
26060 tramas ASTERIX encontradas en el archivo
Fin de la busqueda de tramas ASTERIX en el archivo Datos2.txt
Decodificando tramas ASTERIX de categoria 01
Fin de la decodificacion de tramas ASTERIX de categoria 01
Fin del procesamiento y decodificacion para analisis
Fecha y hora de fin del procesamiento y decodificacion de datos para analisis
2011-01-05 16:02:44.956072
tiempo calculado de todo el proceso 00:02:07
*****
```

Figura 3.18: Informe del procesamiento para análisis en consola Linux

3.4.4 Probabilidad de detección

El análisis de probabilidad de detección, calcula la probabilidad de detección de cada uno de los blancos radar, la probabilidad de detección total y la probabilidad de detección de falsos blancos.

3.4.4.1 Ventana de resultados de probabilidad de detección

La ventana de resultados de probabilidad de detección se activa, si el usuario da doble click en el menú principal, en la opción “PROBABILIDAD DE DETECCION”.

La **Figura 3.19** muestra la ventana de resultados de probabilidad de detección, en la cual podemos identificar: dos apartados; seis botones; un área de resultados y un área de gráficos de barras.

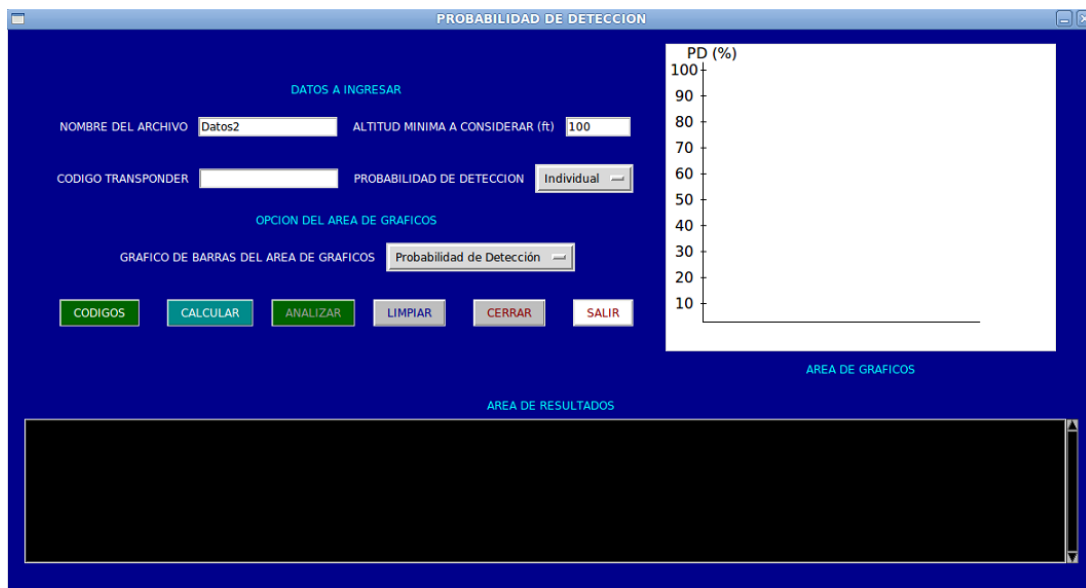


Figura 3.19: Ventana de resultados de probabilidad de detección

3.4.4.2 Apartado de datos a ingresar de probabilidad de detección

La **Figura 3.20** muestra en mayor detalle el apartado de “DATOS A INGRESAR” de la ventana de probabilidad de detección, que contiene los siguientes campos:

- **NOMBRE DEL ARCHIVO:** en este campo se especifica el nombre del archivo de texto decodificado, al cual se le quiere hacer el cálculo de probabilidad de detección.
- **ALTITUD MINIMA A CONSIDERAR:** en este campo el usuario especifica la altitud mínima en pies que se va a considerar para el análisis de los blancos radar.
- **CODIGO TRANSPONDER:** en este campo se especifica el código transponder al cual se le desea calcular la probabilidad de detección, en el menú de “PROBABILIDAD DE DETECCION” la opción “Individual” debe ser seleccionada.
- **PROBABILIDAD DE DETECCION:** este menú contiene tres opciones: “Individual” que especifica que la probabilidad de detección a calcular es la de un blanco radar; “Total” que especifica que la probabilidad de detección a calcular es la sumatoria de las probabilidades de detección de cada blanco radar dividida entre el número total de blancos radar y “Falsos Blancos” para calcular la probabilidad de detección de falsos blancos.

DATOS A INGRESAR

NOMBRE DEL ARCHIVO ALTITUD MINIMA A CONSIDERAR (ft)

CODIGO TRANSPONDER PROBABILIDAD DE DETECCION

Figura 3.20: Apartado de datos a ingresar de la ventana de probabilidad de detección

3.4.4.3 Apartado de opción del área de gráficos y botones

El apartado de “OPCION DEL AREA DE GRAFICOS” contiene el menú “GRAFICO DE BARRAS DEL AREA DE GRAFICOS” el cual tiene las siguientes opciones:

- **Probabilidad de Detección:** esta opción permite poder visualizar un grafico de barras ya sea de la probabilidad de detección individual o la probabilidad de detección total.
- **Probabilidad de Detección vs Rango:** esta opción permite poder visualizar tres gráficos de barras de probabilidad de detección vs rango.
- **Probabilidad de Detección vs Altitud:** esta opción permite poder visualizar tres gráficos de barras de probabilidad de detección vs altitud.

En la **Figura 3.21** se muestra el apartado “OPCION DEL AREA DE GRAFICOS” que contiene el menú “GRAFICO DE BARRAS DEL AREA DE GRAFICOS” y seis botones. La funcionalidad al dar click, sobre cada uno de los seis botones, es la siguiente:

- **CODIGOS:** muestra en el área de resultados cuantos códigos transponder diferentes contiene el archivo de texto decodificado y cuáles son esos códigos transponder.
- **CALCULAR:** realiza el proceso de cálculo de la probabilidad de detección (individual, total ó falsos blancos), mostrando resultados, en el área de resultados y el o los gráficos, en el área de gráficos.
- **ANALIZAR:** normalmente este botón esta desactivado, pero una vez hecho el cálculo de probabilidad de detección para un blanco radar, este se activa y si el usuario da click sobre él en el área de resultados, para un blanco radar identificado con un determinado código transponder, aparecen como una tabla de resultados datos de: numero de trama; código transponder; altitud; rango; acimut; código de confianza del Modo C; código de confianza de código Modo 3/A; código de emergencia y tipo de trama Asterix.
- **LIMPIAR:** limpia el área de resultados y el área de gráficos, de la ventana de resultados de probabilidad de detección.
- **CERRAR:** cierra la ventana de resultados de probabilidad de detección.
- **SALIR:** permite salir del sistema cerrando todas las ventanas.

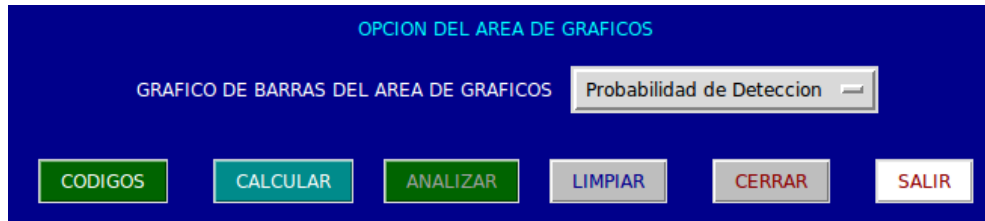


Figura 3.21: Apartado de opción del área de gráficos y botones

3.4.4.4 Area de resultados

El “AREA DE RESULTADOS” es donde se muestran los resultados del proceso de cálculo de la probabilidad de detección, si se calcula la probabilidad de detección de un blanco radar, esta nos proporciona información sobre:

- Número total de detecciones del blanco
- Numero de detecciones validas del blanco
- Probabilidad de Detección del blanco
- Tiempo calculado del proceso

Si se calcula la probabilidad de detección total, el “AREA DE RESULTADOS” nos proporciona información sobre:

- Código transponder del blanco y su Probabilidad de Detección
- Probabilidad de Detección Total
- Tiempo calculado del proceso

Si se calcula la probabilidad de detección de falsos blancos, el “AREA DE RESULTADOS” nos proporciona información sobre:

- Número total de respuestas
- Número de respuestas de falsos blancos
- Probabilidad de Detección de falsos blancos
- Tiempo calculado del proceso

La **Figura 3.22** muestra el área de resultados y lo obtenido al calcular la probabilidad de detección total.

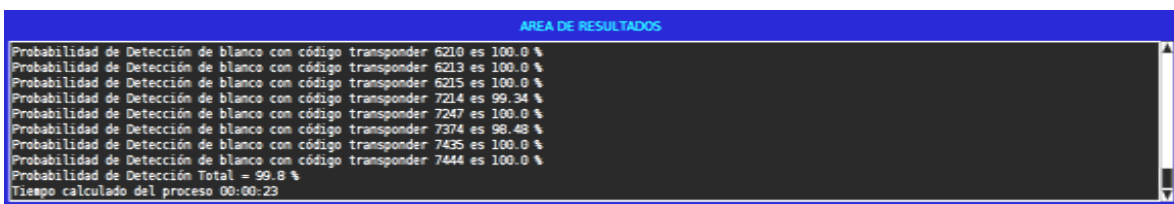


Figura 3.22: Area de resultados de la ventana de probabilidad de detección

El área de resultados también muestra información sobre el análisis de seguimiento de un blanco radar, que tiene un determinado código transponder, que se muestra en la **Figura 3.23**, esta información que se presenta como una tabla de resultados es con respecto a:

- **NUMERO DE TRAMA:** es el numero de trama que el sistema asigna, dependiendo del orden de aparición en el archivo de captura *pcap*.
- **CODIGO:** es el código transponder de la aeronave.
- **ALTITUD:** altitud de la aeronave en pies.
- **RANGO:** distancia a la cual se encuentra la aeronave de la estación radar.
- **ACIMUT:** angulo medido con respecto al norte GPS de la aeronave.
- **TIEMPO:** tiempo de captura de la trama Asterix, proporcionado por **Tshark**.
- **CONF. ALTITUD:** confianza de altitud de la aeronave. ACA son las siglas para describir Alta Confianza de Altitud. BCA son las siglas para describir Baja Confianza de Altitud.
- **CONF. CODIGO:** confianza de código transponder de la aeronave. ACC son las siglas para describir Alta Confianza de Código. BCC son las siglas para describir Baja Confianza de Código.
- **EMERGENCIA:** tipo de código de emergencia, que puede ser: Interferencia ilícita (código 7500); fallo de radio-comunicaciones (código 7600) y emergencia (código 7700).

AREA DE RESULTADOS											
# DE TRAMA	CODIGO	ALTITUD(ft)	RANGO(NM)	ACIMUT(GRADOS)	TIEMPO	CONF.	ALTITUD	CONF.	CODIGO	EMERGENCIA	TRAMA
4257	0116	9300	112.77	71.83	15:19:05	ACA		ACC		Ninguna	Plot
4301	0116	9500	112.66	71.79	15:19:11	ACA		ACC		Ninguna	Plot
4345	0116	9600	112.55	71.53	15:19:17	ACA		ACC		Ninguna	Plot
4389	0116	9800	112.45	71.51	15:19:23	ACA		ACC		Ninguna	Plot
4433	0116	10100	112.37	71.3	15:19:29	ACA		ACC		Ninguna	Plot
4477	0116	10300	112.29	71.19	15:19:35	ACA		ACC		Ninguna	Plot
4521	0116	10600	112.21	70.92	15:19:41	ACA		ACC		Ninguna	Plot
4567	0116	10700	112.14	70.74	15:19:47	ACA		ACC		Ninguna	Plot
4615	0116	10900	112.08	70.61	15:19:53	ACA		ACC		Ninguna	Plot

Figura 3.23: Area de resultados con información sobre seguimiento de un blanco

3.4.4.5 Area de gráficos

El área de gráficos de la ventana de probabilidad de detección, es la que muestra los gráficos de barras dependiendo de la opción especificada del menú “GRAFICO DE BARRAS DEL AREA DE GRAFICOS”, así se tienen tres opciones:

- **Probabilidad de Detección:** esta opción permite poder visualizar un grafico de barras, ya sea de la probabilidad de detección individual ó la probabilidad de detección total, la **Figura 3.24** muestra el grafico de barras de la probabilidad de detección de un blanco radar de un color cyan oscuro. El color del grafico de barras para la probabilidad de detección total es verde oscuro.

- **Probabilidad de Detección vs Rango:** esta opción permite poder visualizar tres gráficos de barras de probabilidad de detección vs rango, estos gráficos se dibujan sobre el área de gráficos cada vez que se calcula la probabilidad de detección total, en la **Figura 3.25** se muestran estos gráficos. Se dibujan tres gráficos: el primero de izquierda a derecha es el rango máximo y su probabilidad de detección (grafico verde); el segundo es el rango mínimo y su probabilidad de detección (grafico amarillo) y el último es la probabilidad de detección mínima a un rango máximo, para un determinado código transponder (grafico cyan).
- **Probabilidad de Detección vs Altitud:** esta opción permite poder visualizar tres gráficos de barras de probabilidad de detección vs altitud. La **Figura 3.26** muestra estos gráficos. Se dibujan tres gráficos: el primero de izquierda a derecha es la altitud máxima y su probabilidad de detección (grafico magenta); el segundo es la altitud mínima y su probabilidad de detección (grafico cyan) y el último es la probabilidad de detección mínima a una altitud máxima para un determinado código transponder (grafico naranja).

En el “AREA DE RESULTADOS” cada vez que se dibujan los gráficos sobre el “AREA DE GRAFICOS”, se muestran los resultados obtenidos sobre los gráficos y el tiempo que tardo el sistema, en hacer todo el proceso de cálculo, para mostrar todos los resultados.

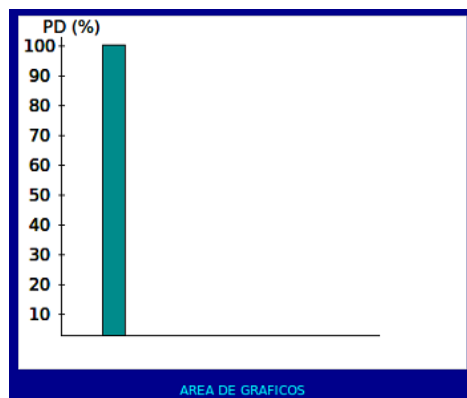


Figura 3.24: Grafico de barras de la probabilidad de detección de un blanco radar

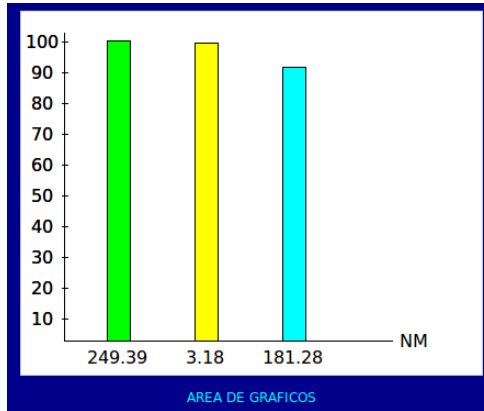


Figura 3.25: Gráficos de barras de la probabilidad de detección vs rango

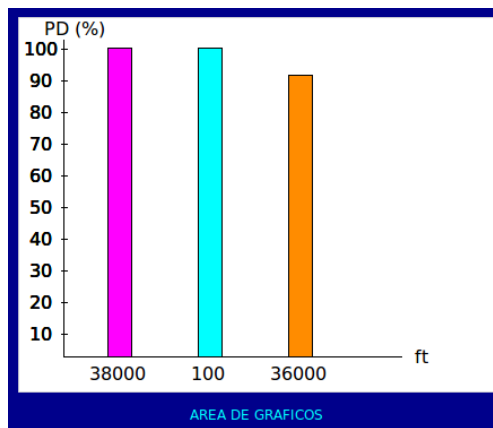


Figura 3.26: Gráficos de barras de la probabilidad de detección vs altitud

3.4.5 Visualización radar

El análisis de visualización radar, se encarga de mostrar al usuario en el PPI, cada uno de los blancos que el sistema radar ha detectado (como plots y pistas), que están en el archivo decodificado de texto, que el usuario ha especificado anteriormente pasando por el programa “PROCESAMIENTO PARA ANALISIS”.

3.4.5.1 Indicador de posición de plan

El Indicador de Posición de Plan es mejor conocido como PPI, en La **Figura 3.27** podemos ver el área del PPI.

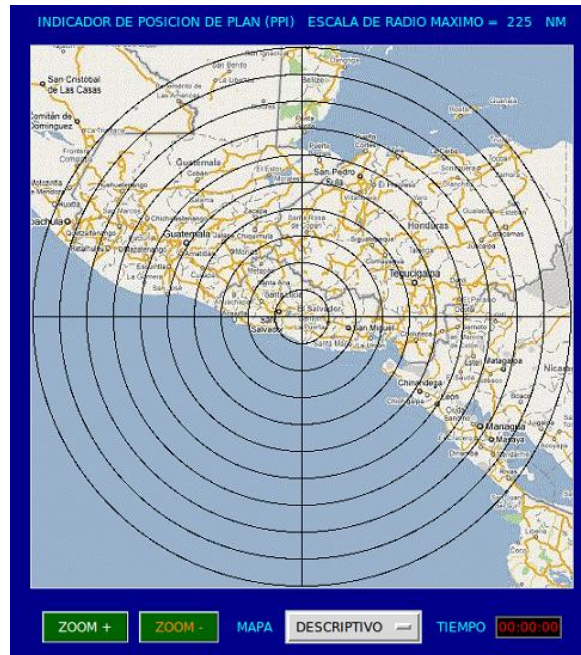


Figura 3.27: Area de Indicador de Posición de Plan

El área del PPI, contiene en la parte superior dos indicadores, el de la izquierda es para el nombre, indicando que es una ventana de visualización radar y el de la derecha indica la escala de radio máximo a la cual está el PPI, por ejemplo si marca 250 NM, el radio máximo del PPI es 250 NM (diez radios en aumentos de 25 NM, a partir de un radio mínimo de 25 NM). Para cambiar la escala de radio máximo del PPI, en la parte inferior izquierda del area del PPI, se encuentran dos botones, si el usuario da click en el botón “ZOOM +” la escala de radio máximo disminuye (acercamiento a la estación radar), con lo cual cambia el mapa, llegando hasta una escala de radio máximo del PPI de 6.25 NM. El botón “ZOOM -” aumenta la escala de radio máximo del PPI (alejamiento de la estación radar), llegando hasta 250 NM. En la parte inferior derecha del área del PPI, se encuentran el menú “MAPA” y el indicador de tiempo de análisis. El Menú “MAPA”, indica el mapa utilizado en el PPI, así el usuario puede escoger entre dos opciones: “DESCRIPTIVO” el cual muestra un mapa con detalles de municipios con sus respectivos nombres y “SATELITE” el cual nos muestra los detalles en cuanto a montañas, cerros, volcanes, etc. Cabe hacer notar que para cada mapa en el PPI, los plots y pistas tendrán un color determinado, para cada uno de sus datos, para una fácil visualización. El indicador de “TIEMPO”, es el que indica el tiempo en el cual se inicia la visualización radar, normalmente está en rojo indicando que se encuentra detenido, cuando se están visualizando plots y pistas en el PPI, este indicador cambia a color blanco.

3.4.5.2 Opciones de visualización radar

En la ventana principal del sistema ADPADEC, en el apartado “OPCIONES DE VISUALIZACION RADAR” se muestran tres botones, la función de cada uno de ellos se describe a continuación:

- **DATOS:** al dar click en este botón, se despliega la ventana de visualización radar.
- **INICIAR:** este botón sirve para iniciar la visualización de plots y pistas en el PPI, una vez se ha detenido la visualización con el botón “DETENER”.
- **DETENER:** sirve para detener la visualización de plots y pistas radar en el PPI.

La **Figura 3.28** muestra el apartado de “OPCIONES DE VISUALIZACION RADAR”.



Figura 3.28: Apartado de opciones de visualización radar

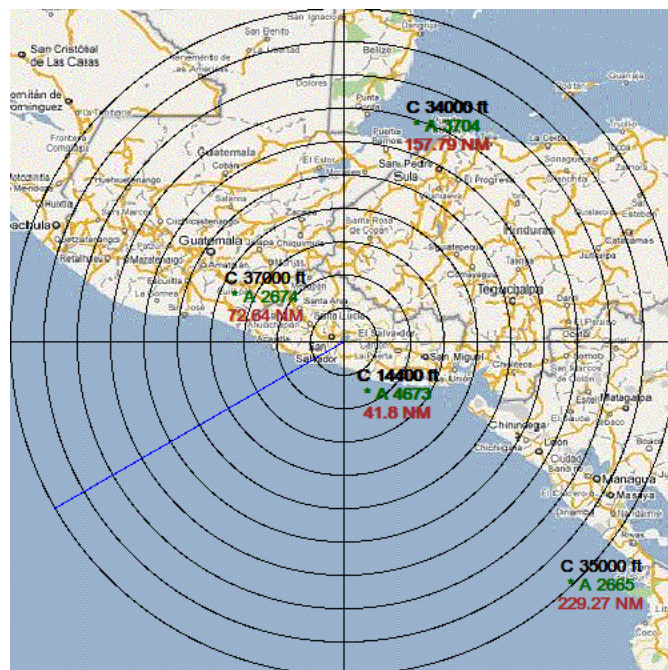


Figura 3.29: PPI con mapa descriptivo

En la **Figura 3.29** se muestra el PPI por defecto, el cual es un mapa descriptivo con detalles de municipios y lugares con sus respectivos nombres. Se observa la simbología utilizada por ADPADEC para representar los plots y pistas radar en el PPI. Una línea azul indica la posición del barrido de la antena del radar. La **Figura 3.30** muestra el PPI con mapa de satélite, la diferencia de este con el anterior está, en que este nos proporciona mayor detalle de montañas, cerros, volcanes y los nombres de algunos lugares. Se ha mencionado anteriormente que para cada mapa en el PPI, los plots y pistas tendrán un color determinado para cada uno de sus datos, para una fácil visualización en cada mapa.



Figura 3.30: PPI con mapa de satélite

La **Figura 3.31** muestra la ventana de visualización radar y en el apartado “DATOS A INGRESAR” el usuario especifica:

- **NOMBRE DEL ARCHIVO:** nombre del archivo capturado y que fue decodificado a un archivo de texto.
- **BLANCOS A VISUALIZAR:** esta opción permite al usuario especificar si desea visualizar en el PPI todos los blancos ó uno en específico. Si se escoge la opción “UNO” el campo de “CODIGO TRANSPONDER” deberá ser ingresado, especificando el código transponder a visualizar.
- **CODIGO TRANSPONDER:** se especifica cuando se ha seleccionado en el campo “BLANCOS A VISUALIZAR” la opción “UNO”, este deberá ser un código octal compuesto por cuatro números.
- **TIEMPO DE VISUALIZACION:** el usuario especifica en este menú, que medición de tiempo desea visualizar “UTC” ó “CA”. “UTC” es el tiempo por defecto de los datos, el cual es el Tiempo Universal Coordinado. “CA” es el tiempo medido en Centroamérica, que es el UTC menos seis horas.
- **HABILITAR PISTAS:** opción para habilitar la visualización de pistas en el PPI. La opción por defecto es “NO”.

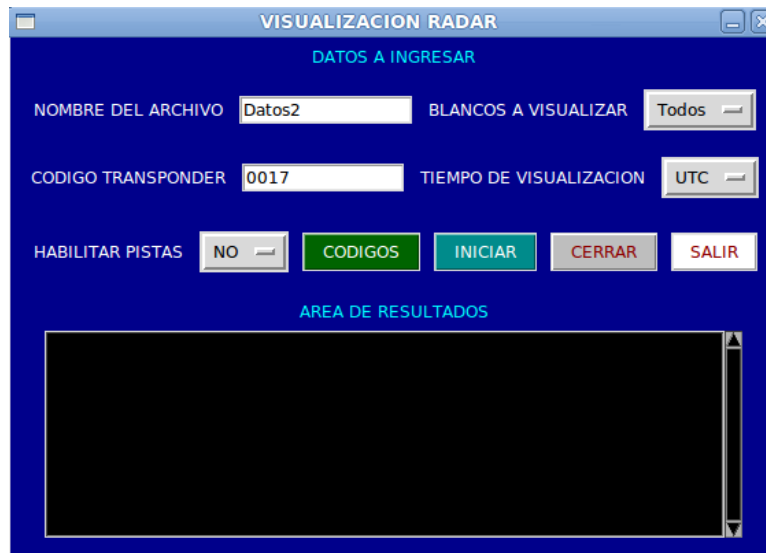


Figura 3.31: Ventana de ingreso de datos para visualización radar

A continuación la función de cada uno de los botones de la ventana de visualización radar:

- **CODIGOS:** muestra al usuario en el “AREA DE RESULTADOS” los códigos transponder que se encuentran en el archivo decodificado de texto, que el usuario ha previamente decodificado con el análisis de “PROCESAMIENTO PARA ANALISIS”.
- **INICIAR:** inicia la visualización radar, cerrando la ventana de visualización radar y dibujando los plots y pistas sobre el PPI, según las opciones especificadas por el usuario.
- **CERRAR:** cierra la ventana de visualización radar.
- **SALIR:** permite salir del sistema cerrando todas las ventanas.

En la **Figura 3.32** se muestra el PPI, se observa cada uno de los plots radar (ecos radar que representan a una aeronave), proporcionando así la siguiente información gráficamente:

- Posición del blanco (en rango y acimut)
- Altitud en pies
- Código transponder
- Rango en NM

Una pista no es más que un historial de plots, que indican el rumbo que lleva una aeronave.

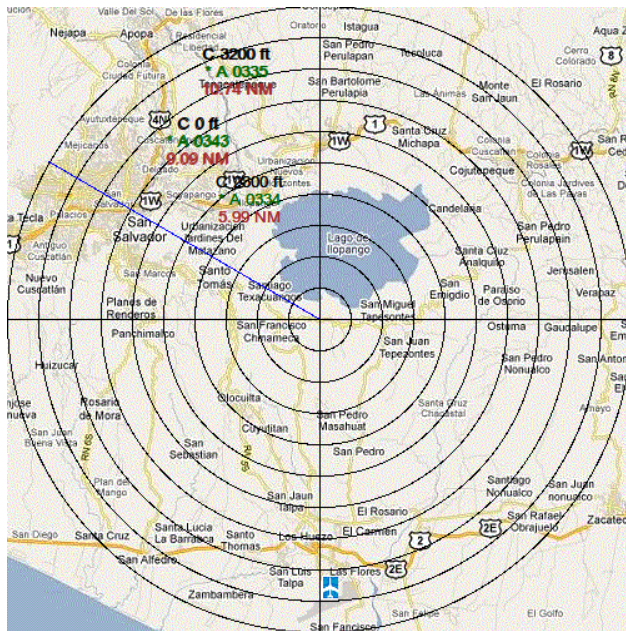


Figura 3.32: Plots radar en PPI

3.4.5.3 Simbología de plots radar

El sistema ADPADEC al igual que otros sistemas de procesamiento de datos radar, posee su simbología en cuanto a los plots radar que se muestran en el PPI. Esta simbología es la que sirve para poder identificar de manera fácil y rápida la información que nos interesa del blanco radar. La **Figura 3.33** muestra la simbología utilizada para un plot radar.



Figura 3.33: Simbología utilizada en el PPI para un plot radar

En la Figura anterior podemos apreciar que se utiliza la siguiente simbología:

- * : el símbolo “asterisco” es el que se utiliza para representar la posición de la aeronave.

- **A 2665:** la letra “A” indica el Modo A del protocolo Asterix, el cual proporciona el código transponder que el ATC le ha asignado a la aeronave y que el piloto introduce en el transponder. En este caso se trata del código 2665 y el color utilizado es verde.
- **C 35000 ft:** indica el Modo C del protocolo Asterix, es decir, la altitud que tiene la aeronave en pies. En este caso es de 35000 pies.
- **232.82 NM:** indica el rango o distancia a la cual se encuentra el blanco de la estación radar. La aeronave esta a 232.82 NM.

3.4.5.4 Simbología de plots radar de baja confianza

Cuando un plot radar es de baja confianza, ya sea en el Modo C ó en el Modo 3/A, su simbología cambia a un color rojo. La **Figura 3.34** muestra un plot radar con el código Modo C; código Modo 3/A y el símbolo que indica su posición en color rojo. El color rojo, es para una rápida identificación por parte del usuario, lo que indica que tanto la altitud como el código transponder, son de baja confianza.

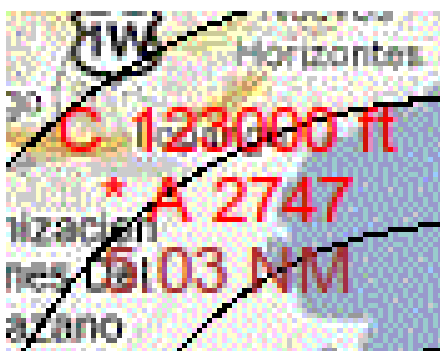


Figura 3.34: Simbología utilizada en el PPI para un plot radar de baja confianza

3.4.5.5 Simbología de plots para códigos de emergencia

El protocolo Asterix de Eurocontrol, define como se ha visto en el Capítulo II, unos códigos asignados en caso de que la aeronave presente una emergencia, ADPADEC utiliza la siguiente simbología en el símbolo que representa la posición del blanco radar:

- **I:** indica el código especial de Interferencia Ilícita (Código 7500).
- **R:** indica el código especial de Fallo de Radio-Comunicación (Código 7600).
- **E:** indica el código especial de Emergencia (Código 7700).

Un plot radar en el PPI, con código especial de emergencia 7700 se muestra en la **Figura 3.35**, cabe hacer notar, que cuando esto sucede, todos los datos del plot mostrados en el PPI

cambian a color rojo, con el objetivo de que el usuario se de cuenta rápidamente del código especial de emergencia.

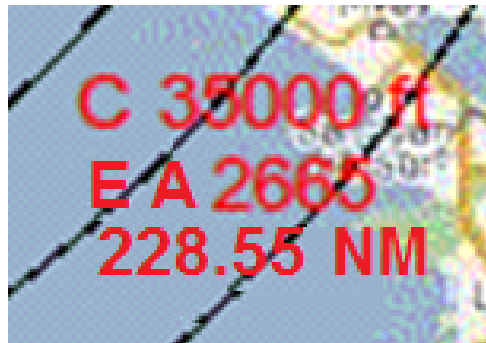


Figura 3.35: Simbología para un plot radar con código especial de emergencia

La simbología para una pista Asterix, es la misma que para un plot Asterix. En la **Figura 3.36**, se observa una pista en el PPI, que indica el rumbo que lleva la aeronave con código transponder 4673.

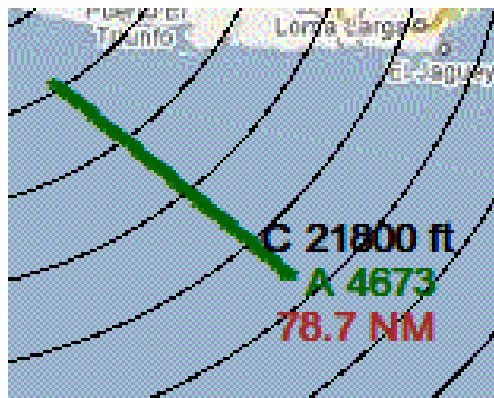


Figura 3.36: Simbología para una pista radar en PPI

3.4.6 Estadísticas

El análisis de estadísticas del archivo de texto decodificado, proporciona al usuario el siguiente tipo de información:

- Numero de plots Asterix
- Numero de pistas Asterix
- Altitud máxima encontrada en los datos.
- Altitud mínima encontrada en los datos.
- Rango máximo de los datos.
- Rango mínimo de los datos.

- Numero de respuestas con baja confianza del Modo -3/A.
- Numero de respuestas con alta confianza del Modo-3/A.
- Numero de respuestas con baja confianza del Modo-C.
- Numero de respuestas con alta confianza del Modo-C.
- Numero de respuestas con altitud menor a 0 pies.
- Numero de respuestas sin código transponder.
- Numero de respuestas con altitud de 0 pies.
- Numero de respuestas con altitud mayor a 50000.
- Numero de respuestas con códigos de emergencia.
- Numero de respuestas con alta confianza de altitud y alta confianza de código.
- Numero de respuestas con alta confianza de altitud y baja confianza de código.
- Numero de respuestas con baja confianza de altitud y alta confianza de código.
- Numero de respuestas con baja confianza de altitud y baja confianza de código.

La **Figura 3.37** muestra la ventana de estadísticas, la cual contiene:

- **AREA DE RESULTADOS:** es donde se muestran, todas la estadísticas que se han obtenido, al analizar el archivo especificado por el usuario en el apartado “NOMBRE DEL ARCHIVO” del “AREA DE GRAFICOS”.
- **AREA DE GRAFICOS:** es donde se muestran los gráficos de barras de porcentaje de cada uno de los cuatro casos en los que ADPADEC clasifica el nivel de confianza de los datos radar. Estos casos se especifican en mayor detalle en el área de resultados.
- **NOMBRE DEL ARCHIVO:** nombre del archivo que se capturo y que fue decodificado a un archivo de texto.
- **BOTONES:** esta área tiene cuatro botones: “ESTADISTICAS” que realiza el cálculo de estadísticas del archivo especificado por el usuario; “LIMPIAR” que limpia el “AREA DE RESULTADOS” como el “AREA DE GRAFICOS”; “CERRAR” que cierra la ventana de estadísticas y “SALIR” para salir del sistema.

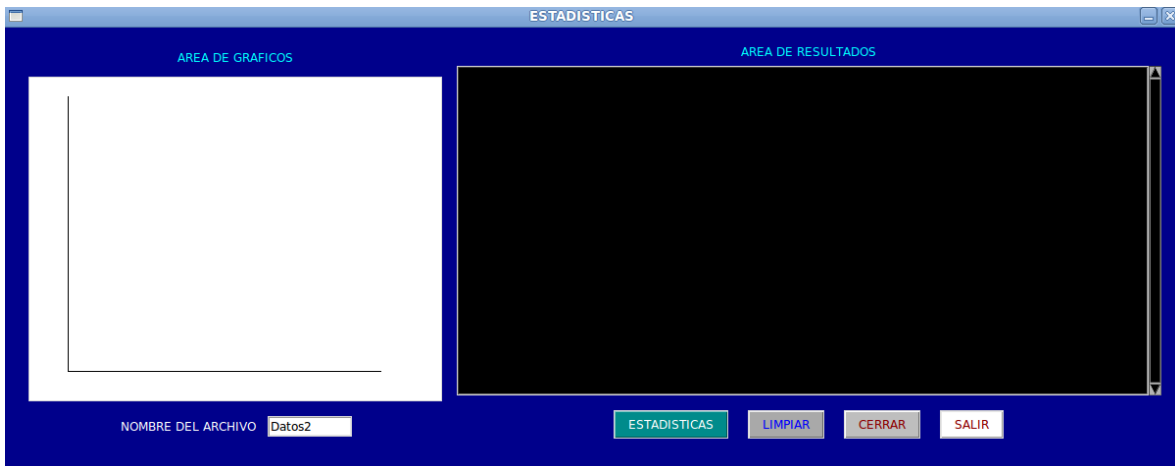


Figura 3.37: Ventana de estadísticas del software ADPADEC

3.5 VENTANA DE ERRORES

En todo sistema siempre se presentan errores, ya sea de que falle el sistema ó no funcione adecuadamente un determinado modulo, pero existen otro tipo de errores que son cometidos por el usuario, estos errores son los que la ventana de errores de ADPADEC presenta al usuario, estos son:

1. Error de nombre de archivo.
2. Error de especificación de un código transponder.
3. Error de tipo de dato.
4. Error de intervalo de un dato.

3.5.1 Error de nombre de archivo

Es aquel error que el usuario comete, cuando especifica el nombre de un archivo, ya sea porque este es incorrecto ó no existe el archivo.

3.5.2 Error de especificación de un código transponder

Es el error que el usuario comete, cuando especifica un código transponder, debido a las siguientes razones:

1. La longitud de los dígitos del código transponder, introducido por el usuario es errónea.
2. El código transponder que el usuario ha introducido, no está en el archivo de texto decodificado.
3. Los números que el usuario introduce del código transponder, no están en el rango de 0 a 7.

4. El usuario introduce letras u otros símbolos, que no son números, en el rango de 0 a 7.

3.5.3 Error de tipo de dato

Es el error que el usuario comete cuando especifica un dato que no es el adecuado, por ejemplo escribir un símbolo ó letras cuando los datos deben ser solo números.

3.5.4 Error de intervalo de un dato

Es cuando el usuario especifica un dato, que esta fuera de rango, por ejemplo los minutos de captura del archivo *pcap*, que no deben sobrepasar los 59 minutos.

En la **Figura 3.38**, se muestra la ventana de errores del software ADPADEC, en este caso se especifica el error de nombre de archivo.

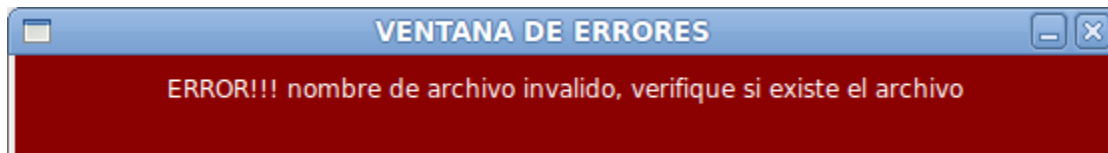


Figura 3.38: Ventana de errores del software ADPADEC

3.6 REQUERIMIENTOS DE HARDWARE DE LA APLICACION

- Computadora portátil ó de escritorio.
- Tarjeta Ethernet 10 / 100 (LAN).
- Espacio en disco duro, mayor a 5 GB.
- Cable Ethernet categoría 5 ó superior.

3.7 REQUERIMIENTOS DE SOFTWARE DE LA APLICACION

- Linux en una computadora portátil, de preferencia **Ubuntu 9.10 karmic koala** ó superior.
- **Python 2.6**, con la librería Tkinter.
- **Tshark** versión 1.2.2 ó superior.
- **Mergecap** versión 1.2.2 ó superior.
- Software ADPADEC, en carpeta personal de **Linux Ubuntu 9.10 karmic koala**.

NOTA: Para la instalación de cada uno de estos programas referirse a la distribución de Linux que se esté utilizando y a la ayuda de instalación que trae cada programa.

3.8 INSTRUCCIONES DE EJECUCION DEL SOFTWARE ADPADEC

Se asume que se está utilizando **Linux Ubuntu 9.10 karmic koala**, que se cumplen con los requerimientos de hardware y software.

1. Ejecutar el terminal de Linux como muestra la **Figura 3.39**.

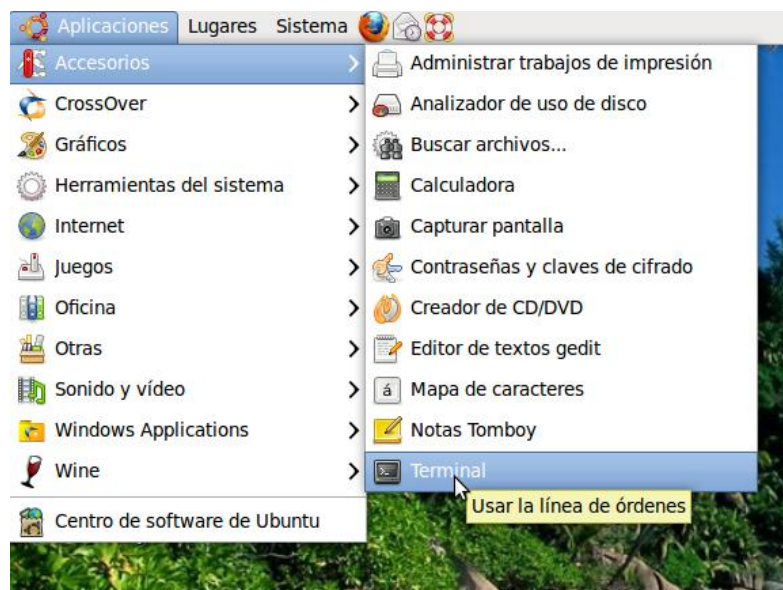


Figura 3.39: Ejecutando el terminal de Linux Ubuntu 10.0

2. En el terminal de **Linux** acceder a la carpeta ADPADEC y ejecutar las instrucciones para la ejecución del software como muestra la **Figura 3.40** y aparecerá la interface grafica que se muestra en la **Figura 3.41**. Para la captura de datos antes de ejecutar el software ADPADEC se deben tener permisos de super usuario.

```
Archivo Editar Ver Terminal Ayuda
manuel@manuel-laptop:~$ su
Contraseña:
root@manuel-laptop:/home/manuel# cd ADPADEC
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
█
```

Figura 3.40: Instrucciones en consola Linux para ejecutar ADPADEC



Figura 3.41: Interface grafica de usuario del software ADPADEC

3.9 LIMITANTES DEL SOFTWARE ADPADEC

- El tamaño de los archivos capturados no debe ser mayor a 600 MB, debido a que el procesamiento puede ser más lento.
- El Indicador de Posición de Plan, no puede expandirse.
- El Indicador de posición de plan para ciertos blancos, no puede mostrar datos de código transponder, altitud y rango, debido a la gran cantidad de información que se procesa en pocos segundos.
- No puede realizar análisis más complejos como: análisis de precisión, análisis de vigilancia mejorado y análisis de conflicto (que realizan los software comerciales como **RBAT**).
- No puede visualizar blancos en tiempo real.
- No puede analizar tramas Asterix de Radares Secundarios Modo S, debido a que no hay módulos para la categoría 048 y 034.

3.10 UTILIZACION DEL SOFTWARE ADPADEC

Para el uso de ADPADEC se han propuesto las siguientes guías de laboratorio:

- GUIA 1: CAPTURA DE DATOS RADAR
- GUIA 2: ANALISIS DE DATOS RADAR
- GUIA 3: VISUALIZACION DE DATOS RADAR

Estas se encuentran en el **Anexo A**.

3.11 BIBLIOGRAFIA

www.python.org/

Python 2.1 Bible. Dave Brueck and Stephen Tanner. Hungry Minds, Inc.

<http://docs.python.org/release/2.6.6/>

CAPITULO IV

APLICACION DEL SISTEMA PROPUESTO

Introducción

Cualquier software que se utilice para obtener resultados de sistemas radar, debe ser capaz de: capturar datos, decodificar datos y visualizar datos radar. En algunos casos el software es capaz de hacer análisis estadísticos, entre ellos la probabilidad de detección, que es un parámetro muy importante para establecer la calidad de un sistema radar, como lo es también la alta confianza de altitud y de código transponder. En este capítulo se muestran las pruebas de campo realizadas conectándose a un radar (ubicado sobre la carretera panorámica y administrado por autoridades del gobierno de El Salvador) y los resultados obtenidos con el software ADPADEC. Estos resultados fueron obtenidos de un modelo IRS-20MP/L, fabricado por la empresa Indra de España.

4.1 RESULTADOS DE LA CAPTURA DE DATOS RADAR

Las pruebas de captura de datos radar, se realizaron en el Centro de Control asociado al radar antes indicado. Se conectó una computadora portátil (laptop) con el software ADPADEC, a un equipo que tenía un puerto Ethernet, con una red LAN donde fluían tramas Asterix de Eurocontrol, a través de un cable Ethernet categoría 5. El equipo estaba perfectamente protegido de la lluvia y con aire acondicionado.

La **Figura 4.1** muestra una fotografía de cómo es la captura de tramas Asterix de la red LAN con PC, utilizando ADPADEC.

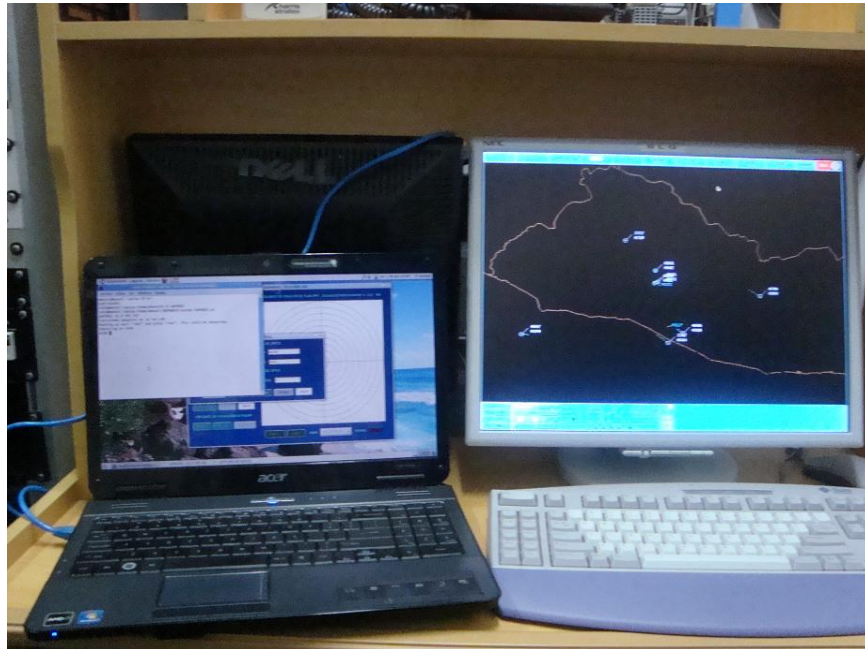


Figura 4.1: Fotografía de cómo es la captura de tramas Asterix con PC

La **Figura 4.2** muestra una fotografía con mayor detalle, como en consola Linux se capturan los paquetes con ADPADEC, donde se encuentran las tramas Asterix.

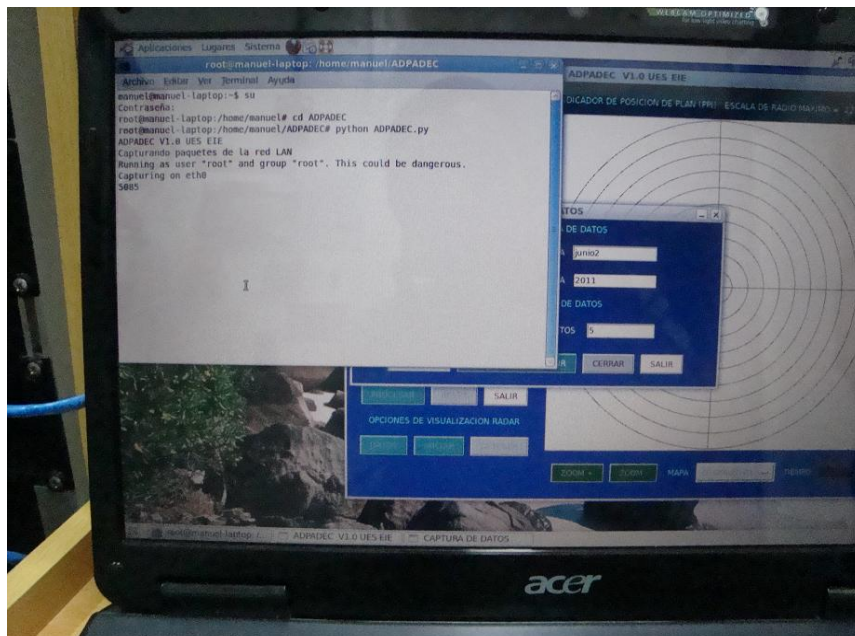


Figura 4.2: Fotografía con mayor detalle de la captura de tramas Asterix

Los resultados de las capturas realizadas se muestran en la **Tabla 4.1**.

Tiempo de captura (HH:MM:SS)	Promedio de paquetes capturados	Tamaño del archivo pcap (MB)
01:00:00	221,656	35.0
01:20:00	219,184	32.9
01:30:00	263,019	41.4
02:00:00	464,888	89.2

Tabla 4.1: Resultados de las capturas realizadas con ADPADEC

Las capturas se realizaron en días diferentes, por lo que el promedio de paquetes capturados está en función del tráfico aéreo. Según los resultados obtenidos, en un segundo se reciben 55 paquetes.

Cada una de las capturas se guardo en un directorio de nombre ADPADEC, con la extensión *pcap*.

4.2 RESULTADOS DE EXTRACCION Y DECODIFICACION DE DATOS RADAR

Una vez que se han capturado los datos en un archivo *pcap*, se realizo la extracción y decodificación de los datos radar para la categoría 01, que inicia generalmente con la conversión del archivo de formato *pcap* a *txt*. El primer archivo a analizar, fue el archivo de una hora de captura, que se nombró d12010.pcap, para este proceso de conversión ADPADEC utiliza el programa **Merg pcap**. El resultado de esta conversión es el archivo d12010.txt.

Una vez se obtiene el archivo d12010.txt, el algoritmo de extracción y decodificación, busca tramas Asterix de categoría 01. Para el archivo d12010.txt fueron encontradas 26060 tramas Asterix de categoría 01. Una vez se han encontrado las tramas Asterix, el algoritmo decodifica cada una de ellas y calcula el tiempo de todo el proceso. Los resultados de todo el proceso de extracción y decodificación del archivo d12010.pcap se muestran en la **Figura 4.3**. Es de hacer notar que el tiempo de la extracción y decodificación depende del tamaño del archivo *pcap*.

El resultado final del análisis de extracción y decodificación, es la ventana de resultados de extracción y decodificación, que para d12010.pcap se muestra en la **Figura 4.4**. En la parte izquierda de la ventana de resultados de extracción y decodificación, están todas las tramas Asterix de categoría 01, que fueron encontradas en el archivo d12010.txt. Al hacer doble click a la quinta trama Asterix, se observan los resultados, al decodificar la trama, que se muestran en el área de resultados, en la parte derecha de la ventana. Entre los datos que

aparecen están: el número de trama que es la 5, el tiempo real de captura UTC 15:05:28, la trama es un plot Asterix, fue detectada por un radar secundario, de una antena, posición de la aeronave (rango = 41.15 NM y acimut = 145.56 grados), modo 3/A valido, código derivado de la respuesta del transponder y código transponder 4673.

```

Fecha y hora de inicio de la extracción y decodificación de datos
2011-06-13 16:54:52.065379
transformando el archivo d12010.pcap a d12010.txt
Fin del proceso de transformación de formato
*****
Buscando tramas ASTERIX en el archivo d12010.txt de categoria 01
26060 tramas ASTERIX encontradas en el archivo
Fin de la búsqueda de tramas ASTERIX en el archivo d12010.txt
*****
Decodificando tramas ASTERIX de categoria 01
Fin de la decodificación de tramas ASTERIX de categoria 01
Fecha y hora de fin de la extracción y decodificación de datos
2011-06-13 16:56:16.987125
tiempo calculado de todo el proceso 00:01:24
*****

```

Figura 4.3: Informe de la extracción y decodificación de datos en consola Linux

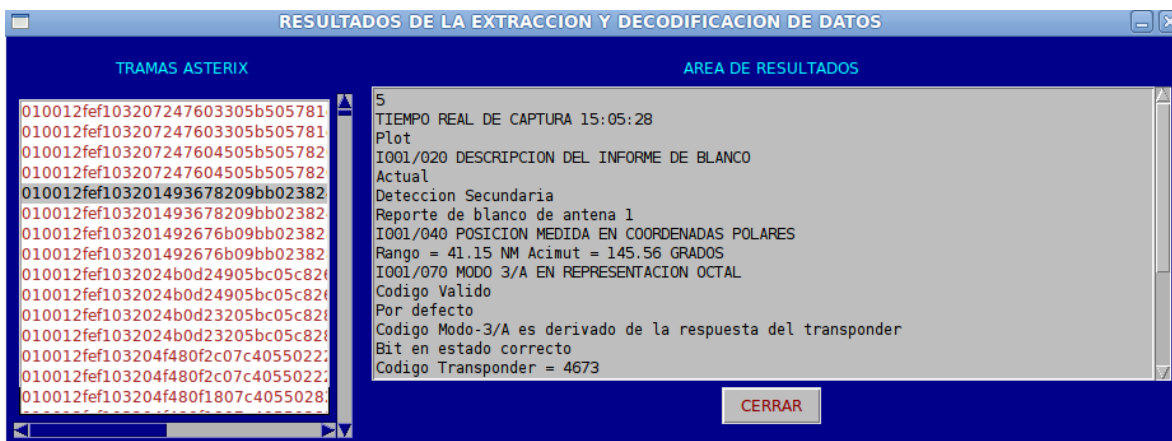


Figura 4.4: Ventana de resultados de la extracción y decodificación de datos

Se realizo el mismo procedimiento para las demás capturas, con los resultados obtenidos, se elaboro la **Tabla 4.2**.

Tiempo de captura (HH:MM:SS)	Promedio de paquetes capturados	Promedio de tramas capturadas de categoría 001	Promedio de tramas capturadas de categoría 002
01:00:00	221,656	26,060	2,391
01:20:00	219,184	21,702	1,596
01:30:00	263,019	32,638	1,794
02:00:00	464,888	66,523	4,782

Tabla 4.2: Resultados de las capturas realizadas con ADPADEC

Según los resultados obtenidos de la **Tabla 4.2**, en un segundo se recibe un promedio de 7 tramas Asterix de categoría 001, de un promedio de 55 paquetes por segundo. En el caso de tramas de categoría 002, se recibe un promedio de 0.50 tramas Asterix de categoría 002, de un promedio de 55 paquetes por segundo.

4.3 RESULTADOS DEL PROCESAMIENTO PARA ANALISIS DE DATOS

Se realizó el procesamiento para análisis de datos radar, para el archivo d12010.pcap. Los resultados obtenidos de este tipo de análisis se observan en la **Figura 4.5**. Esta Figura presenta el informe del procesamiento para análisis de datos radar, en consola **Linux**. Lo primero que realiza el algoritmo del procesamiento para análisis, es convertir el archivo d12010.pcap a d12010.txt. Una vez que se ha realizado la transformación del archivo *pcap* a *txt*, el algoritmo del procesamiento para análisis de datos, busca el primer mensaje de Norte que está en el archivo d12010.txt y lo decodifica, este mensaje servirá para la sincronización de tiempo, ya que a partir del tiempo que marque el primer mensaje de Norte, será el punto de partida para la inclusión de blancos radar, hasta el tiempo en que finalizó la captura de datos, que está registrado en el archivo de texto convertido. Se hace la búsqueda y decodificación de tramas Asterix de categoría 01, con los algoritmos que se han visto anteriormente, con la diferencia de que los resultados se guardan en un archivo de texto a manera de tabla de datos, con diez columnas y filas igual al número de tramas Asterix encontradas en el archivo de texto d12010.txt. El procesamiento para análisis finaliza, con la indicación de tiempo que se ha llevado en realizarlo.

```
Archivo Editar Ver Terminal Ayuda
Fecha y hora de inicio del procesamiento y decodificación de datos para análisis
2011-06-12 21:19:14.431430
transformando el archivo d12010.pcap a d12010.txt
Fin del proceso de transformación de formato
*****
Buscando trama ASTERIX del primer mensaje de Norte
En el archivo d12010.txt de categoria 02
1 mensaje de Norte encontrado en el archivo
Fin de la búsqueda del primer mensaje de Norte
En el archivo d12010.txt
Decodificando el mensaje de Norte
Fin de la decodificación del mensaje de Norte
*****
Buscando tramas ASTERIX en el archivo d12010.txt de categoria 01
26060 tramas ASTERIX encontradas en el archivo
Fin de la búsqueda de tramas ASTERIX en el archivo d12010.txt
Decodificando tramas ASTERIX de categoria 01
Fin de la decodificación de tramas ASTERIX de categoria 01
Fin del procesamiento y decodificación para análisis
Fecha y hora de fin del procesamiento y decodificación de datos para análisis
2011-06-12 21:21:04.868674
tiempo calculado de todo el proceso 00:01:49
*****
```

Figura 4.5: Informe del procesamiento para análisis en consola Linux

Se realizó el procesamiento para análisis para los demás archivos capturados, con el objetivo de poder realizar los análisis de probabilidad de detección, visualización radar y estadísticas.

4.4 RESULTADOS DE CALCULOS DE PROBABILIDAD DE DETECCION

Al realizar el análisis de probabilidad de detección al archivo d12010.txt, se obtuvo el siguiente resultado para la probabilidad de detección total que se muestra en la **Figura 4.6**, en la cual se puede apreciar, en el apartado “NOMBRE DEL ARCHIVO” se escribió el nombre del archivo d12010. La “ALTITUD MINIMA A CONSIDERAR (ft)” fue de 100 pies, debido a que el Aeropuerto Internacional El Salvador (AIES), tiene una altitud sobre el nivel del mar de aproximadamente 100 pies, esta es la altura a la cual aterrizan los aviones comerciales que llegan al AIES. En el apartado de “PROBABILIDAD DE DETECCION” se elige del menú la opción “Total” y se da click al botón “CALCULAR”. En el “AREA DE GRAFICOS” aparece el grafico de barras de la probabilidad de detección total. En el “AREA DE RESULTADOS” tenemos el cálculo de las probabilidades de detección, de cada una de las aeronaves que posee un código transponder, seguido de la probabilidad de detección total y el tiempo calculado del proceso. En el “AREA DE GRAFICOS” se dibuja el grafico de barras correspondiente a la probabilidad de detección total.

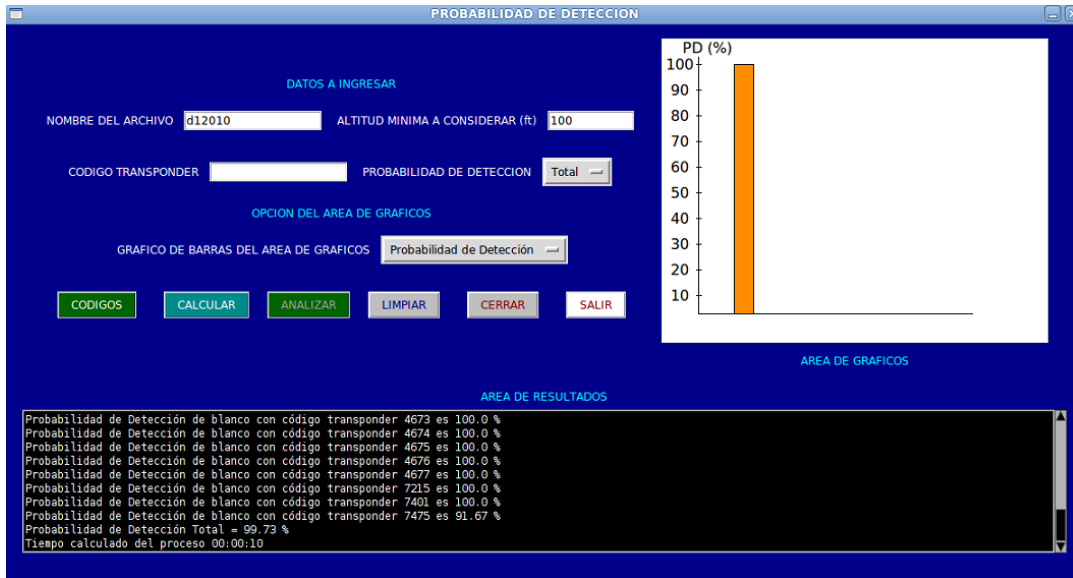


Figura 4.6: Resultados de probabilidad de detección para el archivo d12010.txt

Para calcular la probabilidad de detección de una aeronave, se debe ingresar el código transponder de esta, en el apartado “CODIGO TRANSPONDER”, pero antes se debe conocer cual es ese código transponder. Para conocer ese código transponder se da click en el botón “CODIGOS” y aparecen todos los códigos validos en el “AREA DE RESULTADOS” que se encuentran en el archivo d12010.txt que se muestran en la **Figura 4.7**. Una vez conociendo los códigos transponder, se puede calcular su probabilidad de detección, escribiendo su código en “CODIGO TRANSPONDER” y eligiendo del menú “PROBABILIDAD DE DETECCION” la opción “Individual”. El resultado obtenido para la aeronave con el código transponder 2346 se muestra en la **Figura 4.8**, en la cual se puede apreciar, en el “AREA DE RESULTADOS”:

- El número total de detecciones del blanco, que resulto ser de 192.
- El número de detecciones validas del blanco, que resulto ser de 192.
- La probabilidad de detección del blanco, que es del 100 %.
- El tiempo calculado del proceso, que es menos de un segundo.

Se puede calcular la probabilidad de detección de cada aeronave, ingresando el código transponder valido que se encuentra en el archivo d12010.txt.

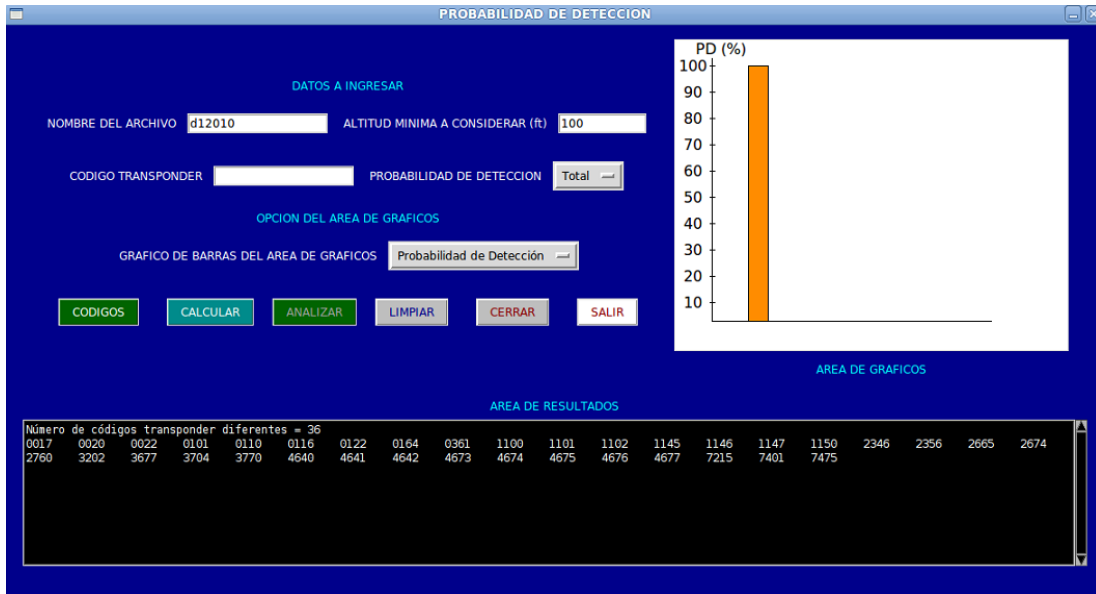


Figura 4.7: Códigos validos que se encuentran en el archivo d12010.txt

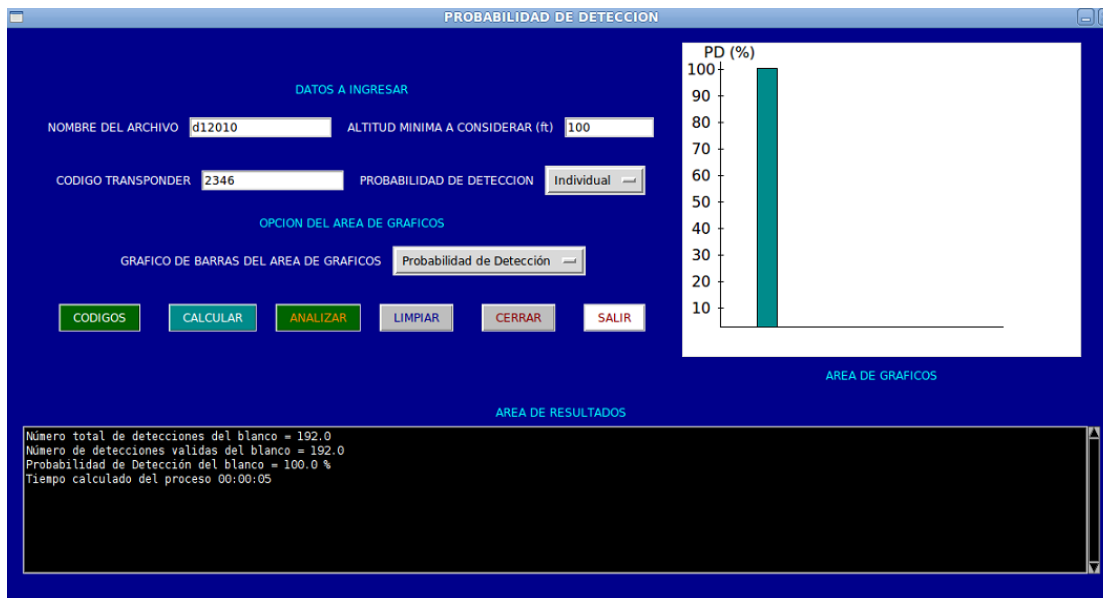


Figura 4.8: Resultados de probabilidad de detección para el código transponder 2346

Para la aeronave con código transponder 2346, se realizó un seguimiento dándole click al botón “ANALIZAR” del apartado de botones de la ventana de “PROBABILIDAD DE DETECCION”. Los resultados obtenidos se pueden apreciar en la **Figura 4.9**, en esta Figura en el “AREA DE RESULTADOS”, se observa las diez columnas de datos para la aeronave con código transponder 2346 que son: la primera columna es el numero de trama; la segunda el código transponder de la aeronave; la tercera la altitud en pies; la cuarta el rango en millas

náuticas; la quinta el acimut en grados; la sexta el tiempo UTC de captura del blanco; la séptima la confianza de altitud; la octava es la confianza de código transponder; la novena es el código de emergencia y la decima columna es el tipo de trama Asterix.

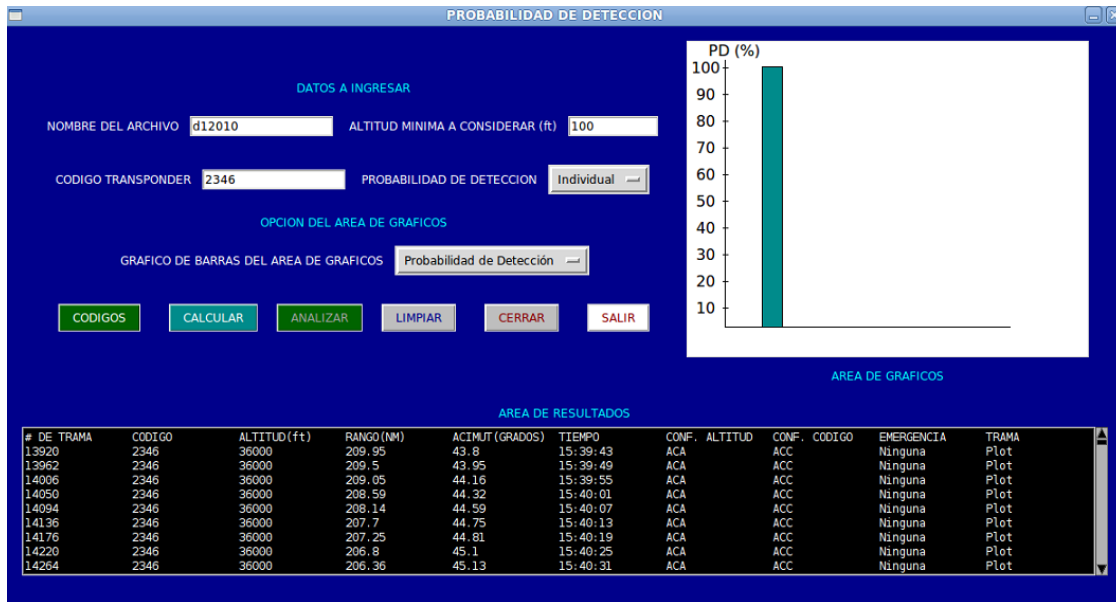


Figura 4.9: Análisis de seguimiento de la aeronave con código transponder 2346

Se obtuvieron los gráficos de barras de probabilidad de detección vs rango, seleccionando del menú “GRAFICO DE BARRAS DEL AREA DE GRAFICOS”, la opción “Probabilidad de Detección vs Rango”, los resultados se pueden apreciar en la **Figura 4.10**.

En la **Figura 4.10**, en el “AREA DE GRAFICOS” se dibujan los tres gráficos de barras correspondientes a la probabilidad de detección vs rango del archivo d12010.txt. El gráfico de barras verde corresponde al rango máximo obtenido de los datos y cuál es la probabilidad de detección obtenida a este rango. El gráfico de barras amarillo es el rango mínimo obtenido de los datos y la probabilidad de detección obtenida a este rango. El gráfico de barras magenta es el rango máximo de la probabilidad de detección mínima que el algoritmo encontró en el archivo d12010.txt. En el “AREA DE RESULTADOS” se muestran los resultados obtenidos: el rango máximo y su probabilidad de detección; el rango mínimo y su probabilidad de detección; el rango máximo de la probabilidad de detección mínima y el tiempo calculado del proceso.

Se obtuvieron los gráficos de barras de probabilidad de detección vs altitud, seleccionando del menú “GRAFICO DE BARRAS DEL AREA DE GRAFICOS”, la opción “Probabilidad de Detección vs Altitud”, los resultados se pueden apreciar en la **Figura 4.11**.

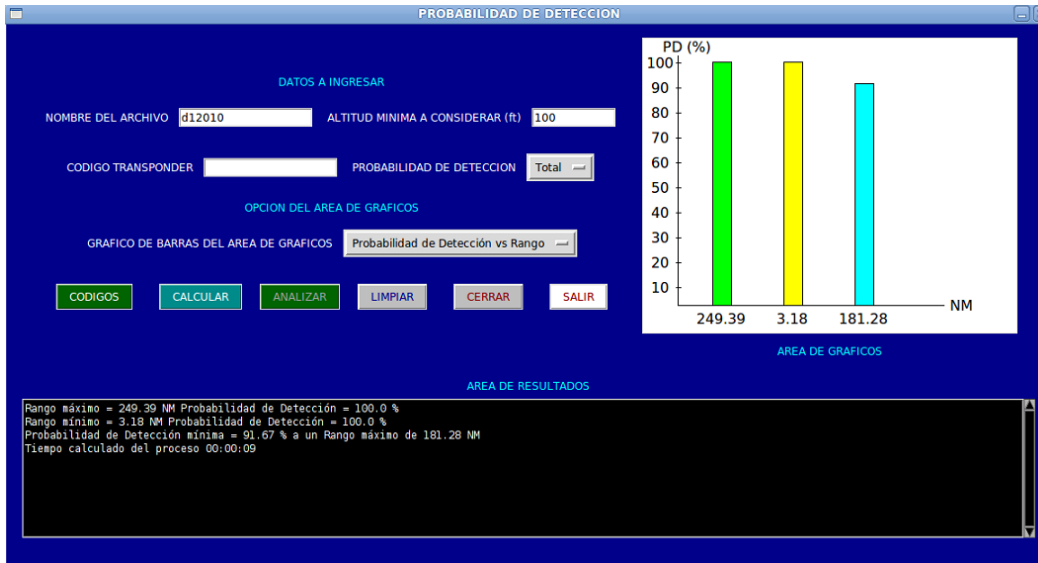


Figura 4.10: Resultados de Probabilidad de Detección vs Rango de d12010.txt

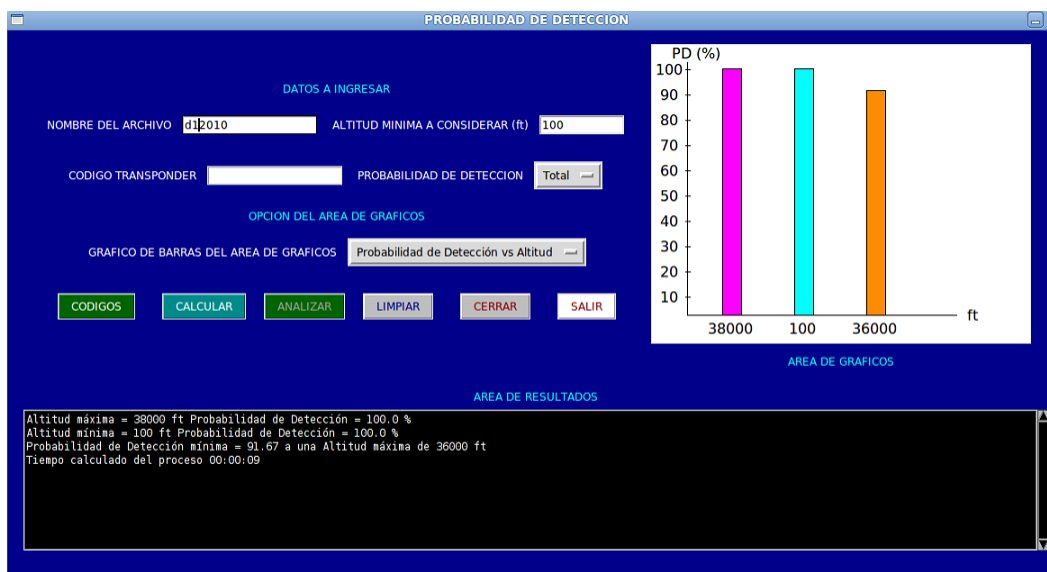


Figura 4.11: Resultados de Probabilidad de Detección vs Altitud de d12010.txt

En la **Figura 4.11**, en el “AREA DE GRAFICOS” se dibujan los tres gráficos de barras correspondientes a la probabilidad de detección vs altitud del archivo d12010.txt. El gráfico de barras magenta corresponde a la altitud máxima de los datos y la probabilidad de detección obtenida a esta altitud. El gráfico de barras cyan es la altitud mínima de los datos y la probabilidad de detección obtenida a esta altitud. El gráfico de barras naranja es la

altitud máxima de la probabilidad de detección mínima que el algoritmo encontró en el archivo d12010.txt. En el “AREA DE RESULTADOS” se muestran los resultados obtenidos: la altitud máxima y su probabilidad de detección; la altitud mínima y su probabilidad de detección; la altitud máxima de la probabilidad de detección mínima y el tiempo calculado del proceso.

En la **Figura 4.12**, se puede observar la ventana de “PROBABILIDAD DE DETECCION” y los resultados obtenidos al calcular la probabilidad de detección de falsos blancos para d12010.txt.

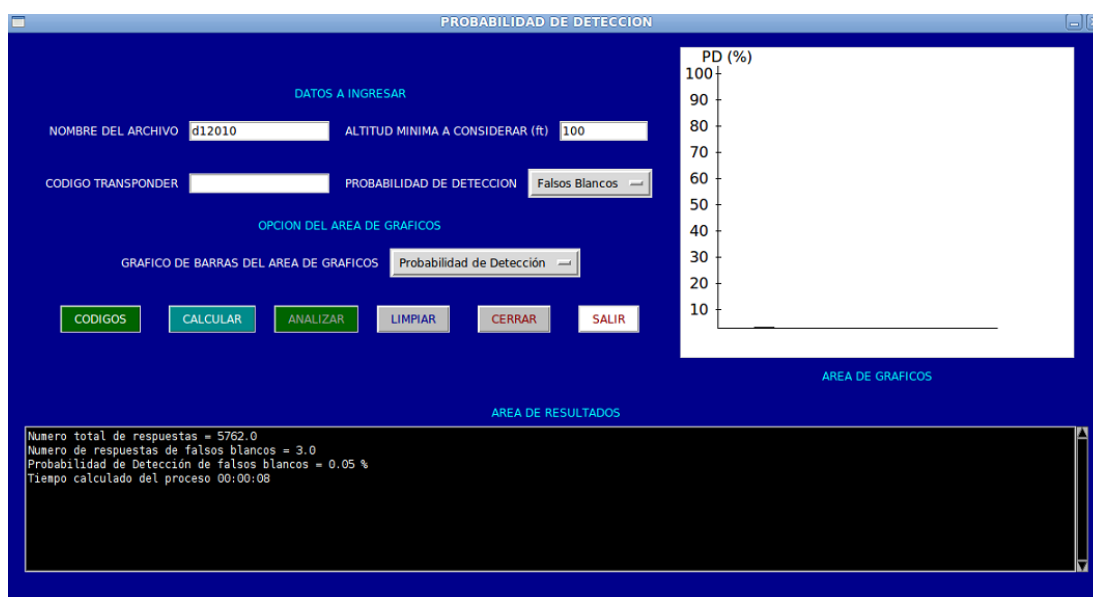


Figura 4.12: Resultados de probabilidad de detección de falsos blancos de d12010.txt

Repitiendo el mismo procedimiento para los demás archivos capturados, se obtuvieron los siguientes resultados de probabilidad de detección que muestra la **Tabla 4.3**.

Tiempo de captura (HH:MM:SS)	Probabilidad de detección total (%)	Probabilidad de detección de falsos blancos (%)
01:00:00	99.73	0.05
01:20:00	99.56	0.20
01:30:00	99.80	0.20
02:00:00	99.46	0.64

Tabla 4.3: Resultados del análisis de probabilidad de detección de d12010.pcap

4.5 RESULTADOS DE VISUALIZACION RADAR

En la ventana de “VISUALIZACION RADAR” se especificaron los siguientes datos:

- **NOMBRE DEL ARCHIVO:** el archivo a analizar d12010.
- **BLANCOS A VISUALIZAR:** como es todo se especifica “Todos”.
- **CODIGO TRANSPONDER:** se deja en blanco ó con el dato que aparece.
- **TIEMPO DE VISUALIZACION:** se elije en este caso “CA” que significa Centroamérica.
- **HABILITAR PISTAS:** se elije la opción “NO”.

Se da click en el botón “INICIAR”, desaparece la ventana de “VISUALIZACION RADAR” y en el PPI de la ventana principal del software, van apareciendo los plots de cada uno de los blancos que ha detectado el radar, que se encuentran en el archivo d12010.txt. Se hizo un acercamiento con el botón “ZOOM +”, a una escala de radio máximo de 100 NM, como se observa en la **Figura 4.13**.

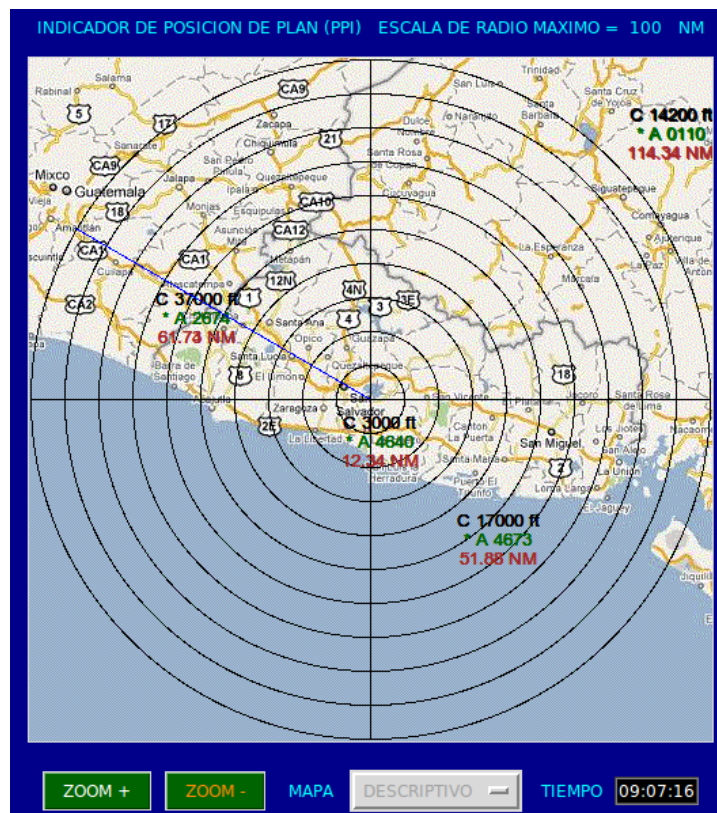


Figura 4.13: Visualización radar de d12010.txt a 100 NM

También las pistas Asterix son dibujadas en el PPI, pero en este caso por ser el radar IRS-20MP/L, el archivo d12010.txt solo contiene plots, lo que confirma este resultado es el análisis de estadísticas, que se realiza más adelante. El periodo de rotación de antena para el sistema radar IRS-20MP/L, resulto ser de 6 segundos.

Se cambio el mapa a “SATELITE” y en la **Figura 4.14**, se muestra que el blanco más cercano al radar esta a 11.18 NM (color cyan). Entre sus datos se observa en el PPI que la aeronave tiene el código transponder 4640 (color naranja); que su altitud es de 1100 pies (color amarillo) y que su posición se encuentra cercana al océano pacifico, en el Departamento de La Paz. Se observa en la **Figura 4.14** que a 09:06:22 hora de Centroamérica aparecen en el PPI cuatro blancos radar, tres de ellos fuera de El Salvador y uno entrando por el Océano Pacífico.



Figura 4.14: Resultados obtenidos de la visualización radar de d12011.txt

Al acercarse ó alejarse de la estación radar con los botones “ZOOM +” ó “ZOOM -” respectivamente, se observan los lugares en los cuales se encuentran las aeronaves y se pueden analizar los datos de cada uno de los blancos radar. Para las otras capturas la única diferencia, es que aparecen más o menos aeronaves sobre El Salvador y los países de Centroamérica cercanos.

4.6 RESULTADOS DE ESTADISTICAS

Para el archivo d12010.txt se obtuvieron los resultados que se muestran en el “AREA DE RESULTADOS” de la ventana de “ESTADISTICAS” en la **Figura 4.15**. Los resultados obtenidos son los que el algoritmo obtiene al analizar el archivo d12010.txt. En el “AREA DE GRAFICOS” se dibuja un grafico de barras de color verde que corresponde al Caso 1 (Numero de respuestas con Alta Confianza de Altitud y Alta Confianza de Código) que

resulta ser del 99.89 %. El grafico que no se alcanza a ver, es el grafico del Caso 3 (Numero de respuestas con Baja Confianza de Altitud y Alta Confianza de Código) que es del 0.11 %. Estas estadísticas pueden ser utilizadas por el usuario para ver el nivel de confianza de los datos, tanto en altitud como en código transponder de cada una de las aeronaves que el radar logra detectar.

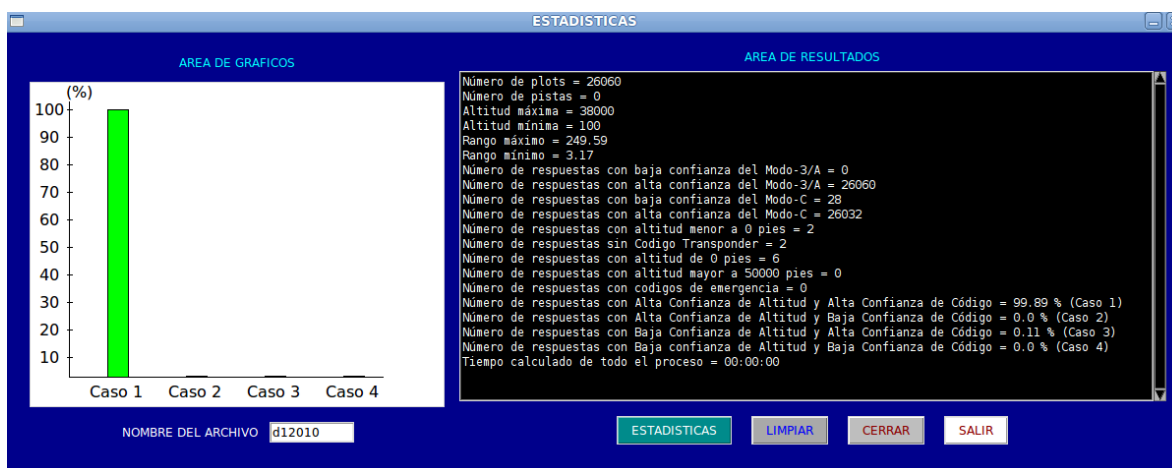


Figura 4.15: Resultados obtenidos del análisis de estadísticas de d12010.txt

Un resumen de los resultados de las estadísticas, obtenidas con ADPADEC de cada una de las capturas se muestra en las **Tablas 4.4 y 4.5**.

Tiempo de captura (HH:MM:SS)	Altitud máxima (pies)	Altitud mínima (pies)	Rango máximo (NM)	Rango mínimo (NM)
01:00:00	38000	100	249.59	3.17
01:20:00	71900	100	249.60	0.0
01:30:00	123000	100	249.35	0.0
02:00:00	41000	100	249.63	0.0

Tabla 4.4: Resultados de las estadísticas de cada una de las capturas

De los resultados obtenidos se puede decir que, la altitud máxima puede ser errónea, debido a que la aeronave puede tener descompuesto su altímetro ó es una respuesta solapada debido al garbling. La altitud mínima obtenida corresponde a la altitud en pies de la pista del AIES. El rango máximo casi alcanza las 250 NM.

Tiempo de captura (HH:MM:SS)	Número de respuestas con Baja Confianza de Altitud y Alta Confianza de Código (%)	Número de respuestas con Alta Confianza de Altitud y Baja Confianza de Código (%)	Número de respuestas con Alta Confianza de Altitud y Alta Confianza de Código (%)
01:00:00	0.11	0.00	99.89
01:20:00	0.06	0.18	99.75
01:30:00	0.45	0.44	98.98
02:00:00	0.05	0.00	99.95

Tabla 4.5: Resultados de las estadísticas de cada una de las capturas

De los resultados obtenidos se puede decir que, la captura que presenta los mejores resultados, es la de dos horas, ya que el número de respuestas con alta confianza de altitud y alta confianza de código es del 99.95 %. El dato que presenta más respuestas de baja confianza es la altitud.

CONCLUSIONES GENERALES

- Los sistemas de radar civiles deben cumplir con las recomendaciones de la OACI, para que puedan ser utilizados en los sistemas de vigilancia de navegación aérea civiles.
- Los sistemas de Radar Primario y Secundario son muy necesarios para garantizar la seguridad de las aeronaves comerciales, ya que nos permiten poder realizar la vigilancia de las mismas.
- El Protocolo Asterix de Eurocontrol es uno de los códigos abiertos más utilizados a nivel mundial de Eurocontrol, ya que facilita el intercambio de datos entre estaciones radar y centros de control.
- El Protocolo Asterix de Eurocontrol permite la integración de sistemas, ya que se pueden compartir los datos con otros países, mejorando la vigilancia de las aeronaves.
- El software ADPADEC se diseñó para sistemas operativos Linux y Unix, debido a que utiliza los programas **Tshark** y **Mergecap**, que en sistemas operativos Windows, no son tan eficientes, como en Linux y Unix, ya que generalmente vienen instalados por defecto.
- El software ADPADEC es una herramienta desarrollada para el uso didáctico en el proceso de enseñanza-aprendizaje del protocolo Asterix de Eurocontrol y conceptos de radar básicos, ya que permite al estudiante ir analizando trama por trama del Protocolo Asterix de Eurocontrol de archivos *pcap*, desde la extracción hasta los análisis estadísticos y visualización radar.
- ADPADEC puede capturar y analizar tramas Asterix de Radares Primarios, Secundarios y la combinación de ambos (Multiradar), pero no de Radares Secundarios Modo S.

RECOMENDACIONES GENERALES

- Para la correcta ejecución y utilización del software ADPADEC, se deben cumplir con los requerimientos de hardware y software, establecidos en el Capítulo III.
- Para la utilización del software ADPADEC en prácticas de laboratorio, en la Escuela de Ingeniería Eléctrica, se pueden usar las guías propuestas en el Anexo A.
- En la captura de archivos *pcap* con ADPADEC, la computadora en la cual este funcionando el software, deberá tener un espacio en disco duro mayor a 5 GB, si se capturan archivos de más de dos horas.
- Para la captura de datos en archivos *pcap*, el usuario debe tener permisos de super usuario, debido a que ADPADEC utiliza el programa **Tshark**, que establece esta condición.
- El tamaño de los archivos capturados utilizando ADPADEC, no debe ser mayor a 600 MB, debido a que el procesamiento puede ser más lento.
- En el software ADPADEC, cuando se quiera realizar los análisis de probabilidad de detección, visualización radar y estadísticas, se debe realizar el procesamiento para análisis.

ANEXO A – GUIAS DE LABORATORIO

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERIA Y ARQUITECTURA

ESCUELA DE INGENIERIA ELECTRICA

SISTEMAS DE RADIONAVEGACION AEREA

GUIA DE LABORATORIO N # 1



CAPTURA DE DATOS RADAR

CATEDRATICO: Ing. Werner Meléndez

OBJETIVOS

- Proporcionar los procedimientos básicos para capturar tramas ASTERIX de Eurocontrol, desde una LAN por la cual fluya datos de ese tipo.
- Familiarizarse con el software ADPADEC para captura y análisis de datos bajo el protocolo ASTERIX.

PARTE I

1.1 REQUERIMIENTOS DE HARDWARE

- Computadora portátil.
- Tarjeta Ethernet 10 / 100 (LAN).
- Espacio en disco duro, mayor a 5 GB.
- Memoria USB con el software ADPADEC.
- Cable Ethernet categoría 5 ó superior.
- Tener acceso a un puerto Ethernet de un equipo, con una red LAN donde fluyan tramas ASTERIX de Eurocontrol, que este perfectamente protegido de la lluvia y de las condiciones climatológicas.

1.2 REQUERIMIENTOS DE SOFTWARE

- Linux en una computadora portátil, de preferencia Ubuntu 9.10 karmic koala ó superior.
- Python 2.6, con la librería Tkinter.
- Tshark versión 1.2.2 ó superior.
- Mergecap versión 1.2.2 ó superior.

NOTA: Para la instalación de cada uno de estos programas referirse a la distribución de Linux que se esté utilizando y a la ayuda de instalación que trae cada programa.

1.3 INSTRUCCIONES DE EJECUCION DEL SOFTWARE ADPADEC

Se asume que se está utilizando Linux Ubuntu 9.10 karmic koala, que se cumplen con los requerimientos de software y hardware.

3. Abrir la carpeta personal como muestra la **Figura 1.1**.
4. Conectar al puerto USB la memoria de datos.
5. Copiar la carpeta ADPADEC de la USB a la carpeta personal, como muestra la **Figuras 1.2**.

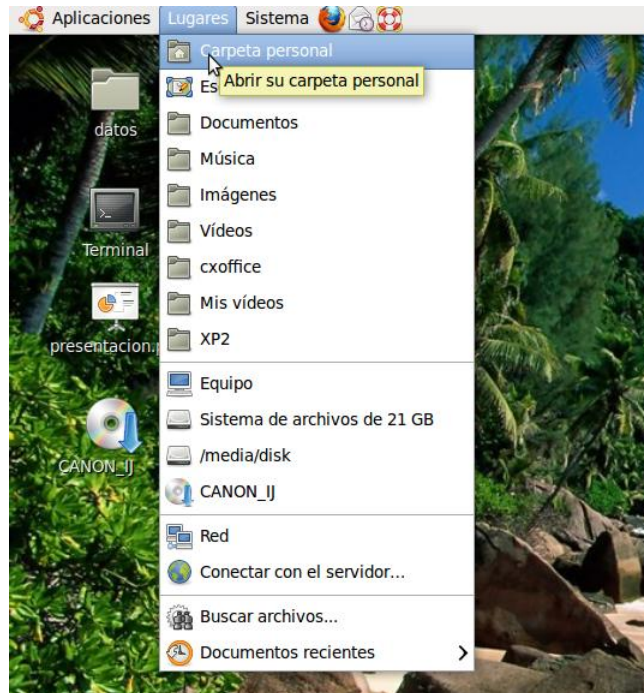


Figura 1.1: Abriendo la carpeta personal en Linux Ubuntu 10.0

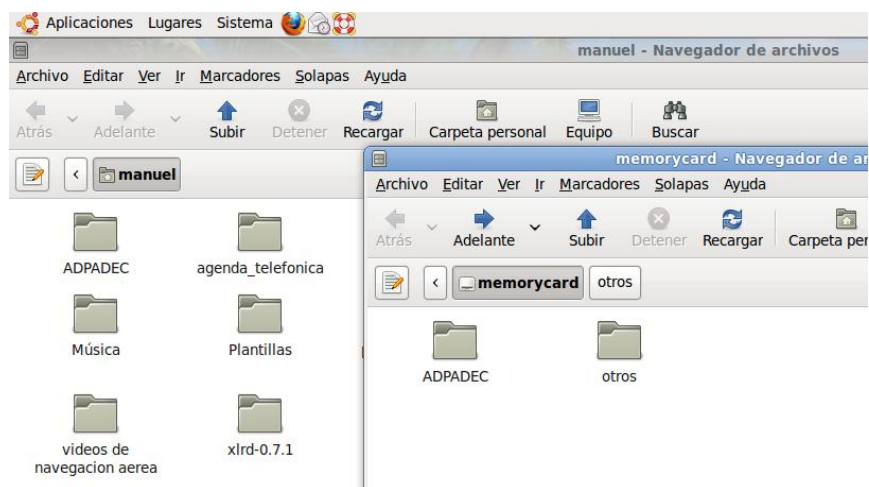


Figura 1.2: Copiando la carpeta ADPADEC de la USB al directorio personal

6. Ejecutar el terminal de Linux como muestra la **Figura 1.3**.

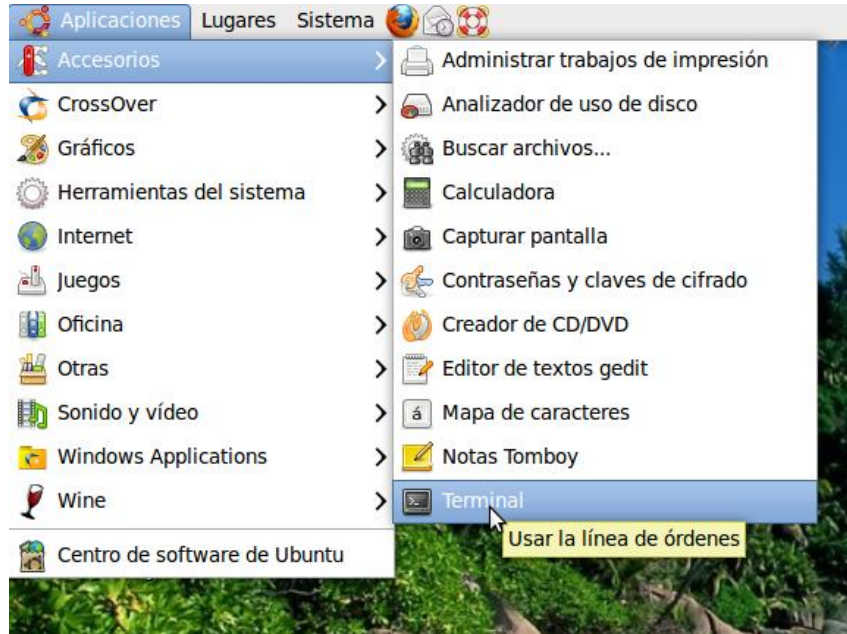


Figura 1.3: Copiando la carpeta ADPADEC de la USB al directorio personal

7. En el terminal de Linux acceder a la carpeta ADPADEC y ejecutar las instrucciones para la ejecución del sistema como muestra la **Figura 1.4** y aparecerá la interface grafica que se muestra en la **Figura 1.5**. Para la captura de datos antes de ejecutar el software ADPADEC se deben tener permisos de super usuario.

```
Archivo Editar Ver Terminal Ayuda
manuel@manuel-laptop:~$ su
Contraseña:
root@manuel-laptop:/home/manuel# cd ADPADEC
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
```

Figura 1.4: Instrucciones en consola Linux para ejecutar el software ADPADEC



Figura 1.5: Interface grafica de usuario del software ADPADEC

PARTE II

2.1 PROCEDIMIENTO PREVIO A INICIAR LA CAPTURA DE DATOS

1. Encender la computadora y cargar Linux.
2. Ejecutar las instrucciones de ejecución del software ADPADEC (sección 1.3).
3. Conectar el cable Ethernet categoría 5 o superior, a la tarjeta Ethernet de la computadora, donde está el software ADPADEC y al puerto Ethernet del equipo, a la red LAN donde fluyen tramas ASTERIX de Eurocontrol.

2.2 PROCEDIMIENTO PARA LA CAPTURA DE DATOS RADAR

1. Realizar el procedimiento previo a iniciar la captura de datos (sección 2.1).
2. Dar doble click en la ventana principal de ADPADEC en “CAPTURA DE DATOS”, que está marcado con rojo, en la **Figura 2.1** y se despliega en pantalla la ventana de captura de datos que se muestra en la **Figura 2.2**.

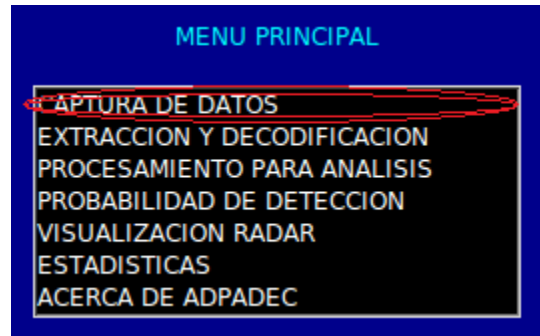


Figura 2.1: Menú principal del software ADPADEC



Figura 2.2: Ventana de captura de datos del software ADPADEC

3. Programar una captura de una hora, ingresando los datos en los campos en la ventana de captura de datos como muestra la **Figura 2.3**. Si se comete un error en la especificación del número de archivo de captura aparecerá la ventana de errores que se muestra en la **Figura 2.4**.



Figura 2.3: Ingreso de datos en los campos de la ventana de captura de datos

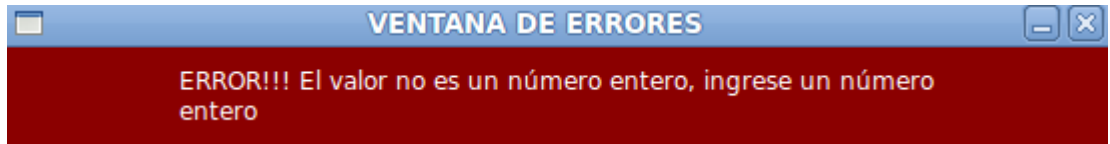


Figura 2.4: Error de tipo de dato

Otro error que se puede cometer, es el error de intervalo de un dato, cuando el usuario especifica un valor de minutos mayor a 59, la **Figura 2.5** muestra este tipo de error.

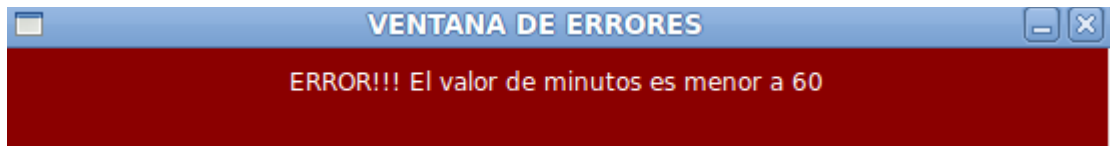


Figura 2.5: Error de intervalo de un dato

En el caso del nombre del archivo se recomienda solo utilizar letras y números, pero no símbolos, ya que esto hace que no se capture el archivo.

4. Una vez especificados los campos dar click en el botón “CAPTURAR”, que está marcado con rojo, en la **Figura 2.4**. Un ejemplo de lo que aparece en consola Linux al estar capturando paquetes de la red LAN, es lo que se observa en la **Figura 2.5**.



Figura 2.4: Marcación del botón “CAPTURAR”

```
root@manuel-laptop: /home/manuel/ADPADEC
Archivo Editar Ver Terminal Ayuda
manuel@manuel-laptop:~$ su
Contraseña:
root@manuel-laptop:/home/manuel# cd ADPADEC
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
ADPADEC V1.0 UES EIE
Capturando paquetes de la red LAN
Running as user "root" and group "root". This could be dangerous.
Capturing on eth0
4525
```

Figura 2.5: Información en consola Linux al realizar una captura con ADPADEC

5. Al finalizar la captura aparece: el número de paquetes capturados; el tiempo de inicio de la captura y el tiempo en el cual finalizo la captura. Cerrar el software ADPADEC dando click en el botón “SALIR”, que está marcado con rojo, en la **Figura 2.6**. El archivo capturado tendrá una extensión pcap y queda almacenado en la carpeta ADPADEC. Según los datos especificados, tendrá el nombre de Datos1.pcap.

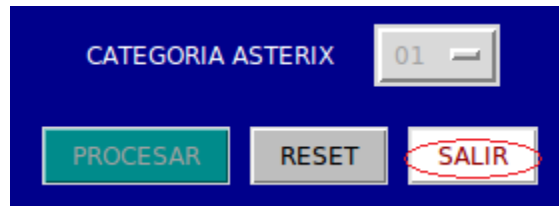


Figura 2.6: Marcación del botón “SALIR” en la ventana principal

NOTA: Para una explicación detallada del software ADPADEC consultar la tesis “Desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol”.

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA
SISTEMAS DE RADIONAVEGACION AEREA
GUIA DE LABORATORIO N # 2



ANALISIS DE DATOS RADAR

CATEDRATICO: Ing. Werner Meléndez

OBJETIVOS

- Familiarizarse con los conceptos básicos radar que aplica el software ADPADEC.
- Analizar con el software ADPADEC, datos radar de archivos pcap con tramas Asterix de Eurocontrol.

PARTE I

1.1 REQUERIMIENTOS DE HARDWARE

- Computadora portátil o de escritorio.
- Espacio en disco duro, mayor a 5 GB, para contener archivos pcap, de por lo menos 100 MB.
- Memoria USB con el software ADPADEC.

1.2 REQUERIMIENTOS DE SOFTWARE

- Linux en una computadora portátil o de escritorio, de preferencia Ubuntu 9.10 karmic koala o superior.
- Python 2.6, con la librería Tkinter.
- Tshark versión 1.2.2 o superior.
- Mergecap versión 1.2.2 o superior.

NOTA: Para la instalación de cada uno de estos programas referirse a la distribución de Linux que se esté utilizando y a la ayuda de instalación que trae cada programa.

1.3 INSTRUCCIONES DE EJECUCION DEL SOFTWARE ADPADEC

Se asume que se está utilizando Linux Ubuntu 9.10 karmic koala, que se cumplen con los requerimientos de software y hardware.

8. Abrir la carpeta personal como muestra la **Figura 1.1**.
9. Conectar al puerto USB la memoria de datos.
10. Copiar la carpeta ADPADEC de la USB a la carpeta personal, como muestra la **Figura 1.2**. Dentro de la carpeta ADPADEC, ya se encuentra el archivo d12010.pcap.

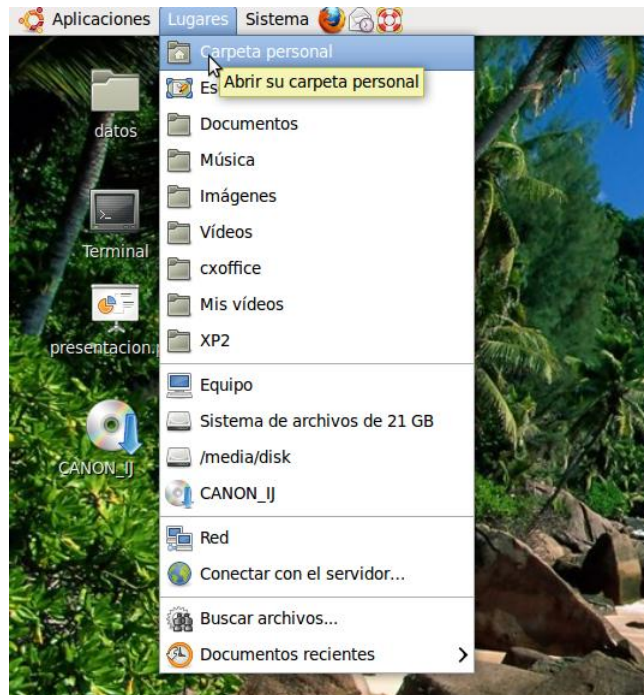


Figura 1.1: Abriendo la carpeta personal en Linux Ubuntu 10.0

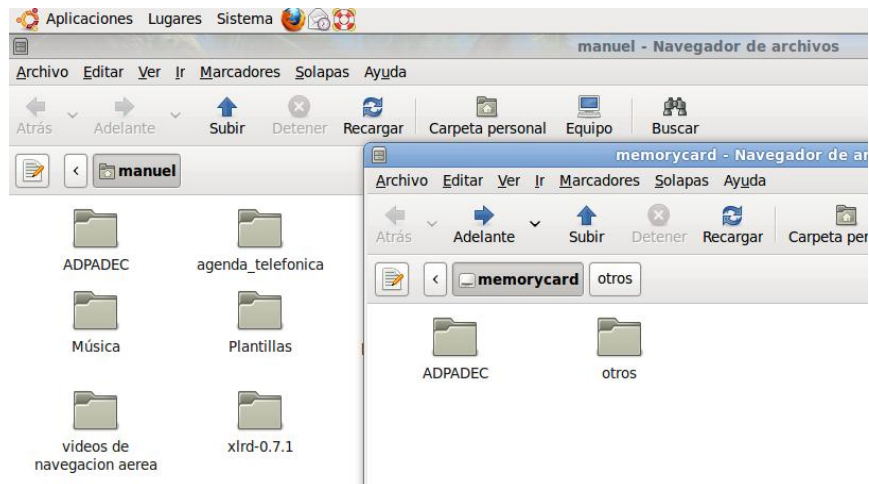


Figura 1.2: Copiando la carpeta ADPADEC de la USB al directorio personal

11. Ejecutar el terminal de Linux como muestra la **Figura 1.3**.

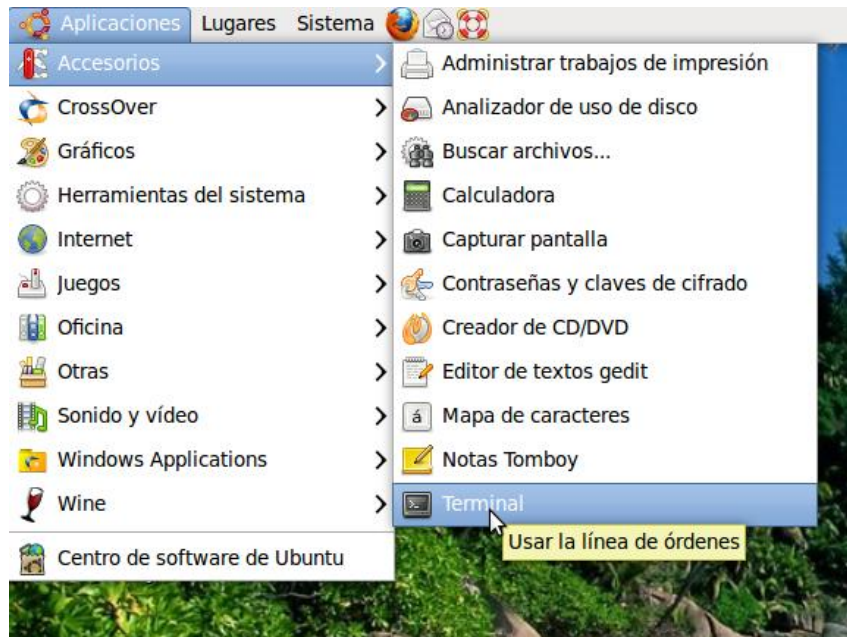


Figura 1.3: Copiando la carpeta ADPADEC de la USB al directorio personal

12. En el terminal de Linux acceder a la carpeta ADPADEC y ejecutar las instrucciones para la ejecución del software como muestra la **Figura 1.4** y aparecerá la interface grafica que se muestra en la **Figura 1.5**.

```
root@manuel-laptop:/home/manuel# cd ADPADEC
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
```

Figura 1.4: Instrucciones en consola Linux para ejecutar el software ADPADEC



Figura 1.5: Interface grafica de usuario del software ADPADEC

PARTE II

2.1 EXTRACCION Y DECODIFICACION DE TRAMAS ASTERIX

1. Dar doble click en la opción “EXTRACCION Y DECODIFICACION” del menú principal de ADPADEC, donde muestra lo marcado en color rojo, en la **Figura 2.1**. Se habilitaran las opciones del apartado “PROCESAMIENTO DE DATOS”.
2. Especificar los campos que se muestran en la **Figura 2.2**. Uno de los errores que el usuario puede cometer, es el nombre del archivo, ya que si el nombre del archivo no existe aparecerá la ventana de errores del software ADPADEC, con el error que muestra la **Figura 2.3**.

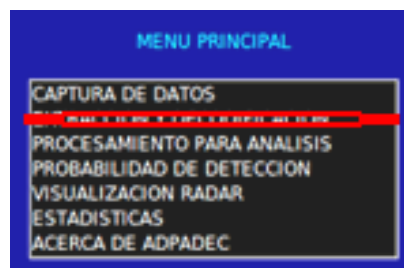


Figura 2.1: Menú principal del software ADPADEC

PROCESAMIENTO DE DATOS

TIPO DE ARCHIVO PCAP

NOMBRE d12010

CODIGO SAC El Salvador

CATEGORIA ASTERIX 01

Figura 2.2: Datos a ingresar en el apartado “PROCESAMIENTO DE DATOS”

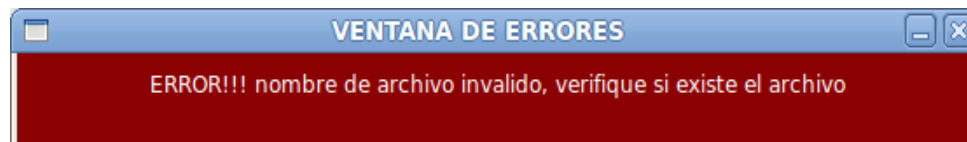


Figura 2.3: Error de nombre de archivo

3. Una vez que se ha especificado cada uno de los campos anteriores, dar click en el botón “PROCESAR”, que inicia el proceso de extracción y decodificación de tramas Asterix.
4. Observar los resultados obtenidos en consola Linux. Aparece una ventana de resultados de la extracción y decodificación de cada una de las tramas Asterix que se encontraban en el archivo d12010. Dar doble click en la quinta trama del área de tramas Asterix, los resultados de la decodificación aparecen en el área de resultados como muestra la **Figura 2.4**. Cada trama representa un blanco radar, es en esta parte donde se pueden apreciar los datos de cada uno de los blancos radar. Así se le puede dar doble click a cualquier otra trama y los resultados de la decodificación aparecen en el área de resultados.

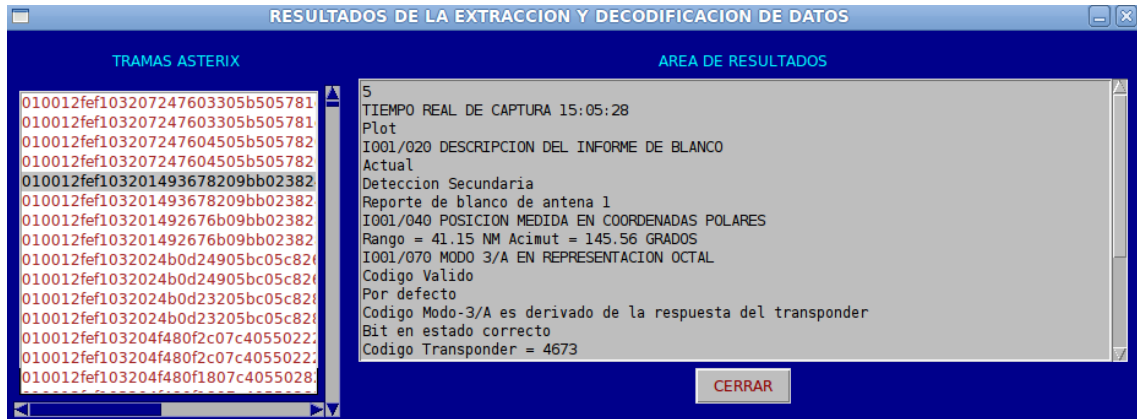


Figura 2.4: Ventana de resultados de la extracción y decodificación de datos de d12010.pcap

5. Repetir los pasos del 2 al 4, solamente cambiando en el apartado “PROCESAMIENTO DE DATOS”, en el campo “CATEGORIA ASTERIX”, 01 por 02, para obtener tramas Asterix de mensajes de Norte.
6. Cerrar la ventana de resultados de la extracción y decodificación.

PARTE III

3.1 PROCESAMIENTO PARA ANALISIS

1. Dar doble click en la opción “PROCESAMIENTO PARA ANALISIS” del menú principal de ADPADEC, donde muestra lo marcado en color rojo, en la **Figura 3.1**. Esto hace que se habiliten las opciones del campo “PROCESAMIENTO DE DATOS”, especificar los mismos datos ingresados para el análisis de “EXTRACCION Y DECODIFICACION”.

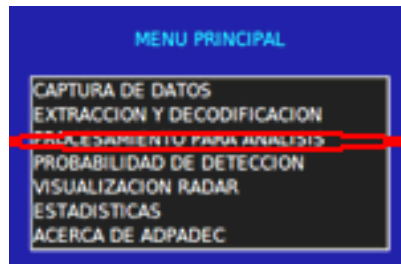


Figura 3.1: Menú principal del software ADPADEC

2. Una vez que se ha especificado cada uno de los campos anteriores, dar click en el botón “PROCESAR”, que inicia el procesamiento y decodificación para análisis de tramas Asterix.
3. Observar los resultados obtenidos en consola Linux. Es en esta parte donde se puede observar: La fecha y hora en la cual se está realizando el análisis; la búsqueda y decodificación de la trama Asterix del primer mensaje de Norte; la búsqueda y decodificación de tramas Asterix de categoría 01 y el tiempo que se llevo para

realizar este análisis. Este análisis se hace con el objetivo de decodificar el archivo d12010.txt, a un archivo decodificado de texto, para que los análisis de “PROBABILIDAD DE DETECCION”, “VISUALIZACION RADAR” y “ESTADISTICAS”, puedan ser realizados.

PARTE IV

4.1 PROBABILIDAD DE DETECCION TOTAL

1. Dar doble click en la opción “PROBABILIDAD DE DETECCION” del menú principal de ADPADEC, donde muestra lo marcado en color rojo, en la **Figura 4.1**, que hace que aparezca la ventana de resultados de probabilidad de detección.

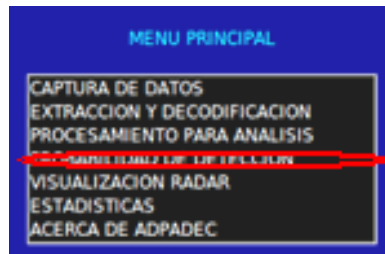


Figura 4.1: Menú principal del software ADPADEC

2. En la ventana de probabilidad de detección, en el apartado de “DATOS A INGRESAR”, especificar lo que se muestra en la **Figura 4.2**. Entre los errores que puede cometer el usuario están: error de nombre de archivo, que se muestra en la **Figura 4.3**; error de especificación de un código transponder, que muestran las **Figuras 4.4, 4.5, 4.6 y 4.7** y error de tipo de dato (cuando se ingresa un valor de altitud invalida), como muestran las **Figuras 4.8 y 4.9**.

Figura 4.2: Datos a ingresar en la ventana probabilidad de detección

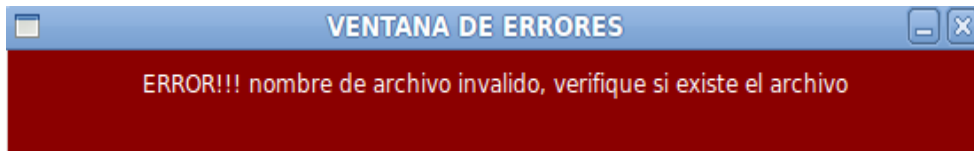


Figura 4.3: Error de nombre de archivo

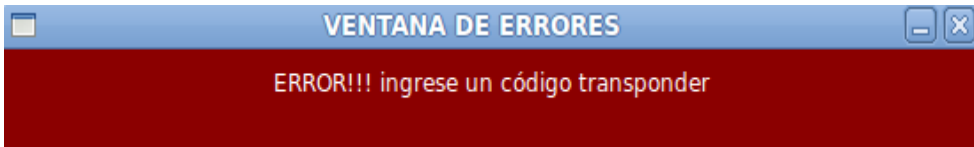


Figura 4.4: Error de especificación de un código transponder



Figura 4.5: Error de especificación de un código transponder

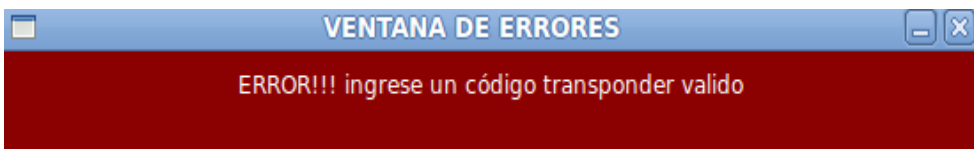


Figura 4.6: Error de tipo de dato

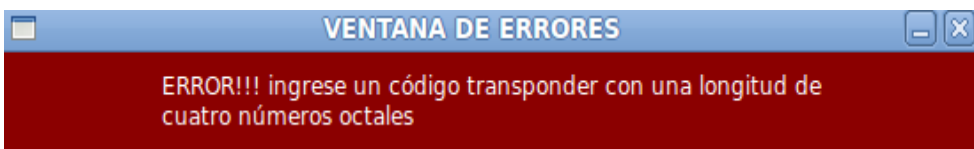


Figura 4.7: Error de especificación de un código transponder

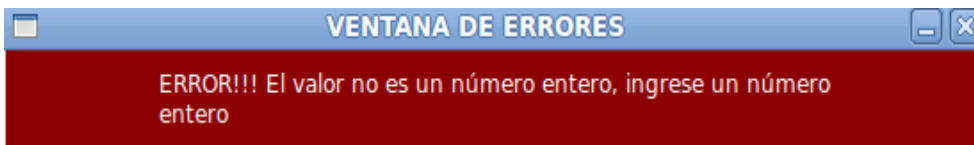


Figura 4.8: Error de tipo de dato

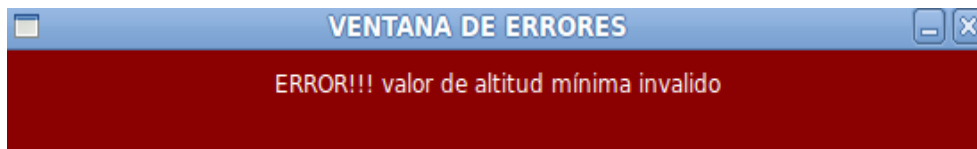


Figura 4.9: Error de tipo de dato

- Una vez que se ha especificado cada uno de los campos anteriores, dar click en el botón "CALCULAR", que está marcado con rojo en la **Figura 4.10**, que inicia el calculo de la probabilidad de detección total, del archivo decodificado de d12010.txt.



Figura 4.10: Apartado de opción del área de gráficos y botones

- Observar en el área de resultados que se tiene una probabilidad de detección total del 99.73 %. También se puede apreciar el cálculo de las probabilidades de detección de códigos. En el área de gráficos de barras, deberá aparecer un grafico de barras color naranja, que indica el porcentaje de la probabilidad de detección total calculada.
- Especificar lo que se muestra en la **Figura 4.2**. En el apartado "OPCION DEL AREA DE GRAFICOS" elegir la opción "Probabilidad de Detección vs Rango" para obtener los resultados de Probabilidad de Detección vs Rango y dar un click al botón "CALCULAR", observar el área de resultados y área de gráficos.
- En el apartado "OPCION DEL AREA DE GRAFICOS" elegir la opción "Probabilidad de Detección vs Altitud" para obtener los resultados de Probabilidad de Detección vs Altitud y dar un click al botón "CALCULAR", observar el área de resultados y área de gráficos.

4.2 PROBABILIDAD DE DETECCION DE CODIGOS

1. Calcular la probabilidad de detección del blanco con código transponder 2346 ingresando los datos como muestra la **Figura 4.11**, el resultado debería ser del 100 %.

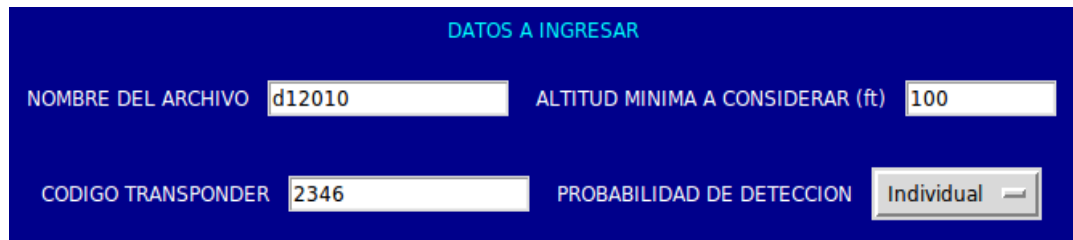


Figura 4.11: Datos a ingresar para calcular la probabilidad de detección del código 2346

2. Ahora se hará un análisis de seguimiento del blanco con código transponder 2346, dar click en el botón “ANALIZAR”, que está marcado con rojo en la **Figura 4.12**. Observar los resultados obtenidos en el área de resultados, esto se muestra en la **Figura 4.13**.



Figura 4.12: Apartado de opción del área de gráficos y botones



# DE TRAMA	CODIGO	ALTITUD(ft)	RANGO(NM)	ACIMUT(GRADOS)	TIEMPO	CONF.	ALTITUD	CONF.	CODIGO	EMERGENCIA	TRAMA
13920	2346	36000	209.95	43.8	15:39:43	ACA		ACC		Ninguna	Plot
13962	2346	36000	209.5	43.95	15:39:49	ACA		ACC		Ninguna	Plot
14006	2346	36000	209.05	44.16	15:39:55	ACA		ACC		Ninguna	Plot
14050	2346	36000	208.59	44.32	15:40:01	ACA		ACC		Ninguna	Plot
14094	2346	36000	208.14	44.59	15:40:07	ACA		ACC		Ninguna	Plot
14136	2346	36000	207.7	44.75	15:40:13	ACA		ACC		Ninguna	Plot
14176	2346	36000	207.25	44.81	15:40:19	ACA		ACC		Ninguna	Plot
14220	2346	36000	206.8	45.1	15:40:25	ACA		ACC		Ninguna	Plot
14264	2346	36000	206.36	45.13	15:40:31	ACA		ACC		Ninguna	Plot

Figura 4.13: Análisis de seguimiento del código 2346

4.3 PROBABILIDAD DE DETECCION DE FALSOS BLANCOS

1. Calcular la probabilidad de detección de falsos blancos ingresando los datos como muestra la **Figura 4.14**.

Figura 4.14: Datos a ingresar para calcular la probabilidad de detección de falsos blancos

- Una vez que se ha especificado cada uno de los campos anteriores, dar click en el botón “CALCULAR”, que está marcado con rojo en la **Figura 4.15**, que inicia el calculo de la probabilidad de detección de falsos blancos, del archivo decodificado de d12010.txt. El resultado deberá ser del 0.05 %.

Figura 4.15: Apartado de opción del área de gráficos y botones

- Cerrar la ventana de probabilidad de detección.

PARTE V

5.1 ESTADISTICAS

- Dar doble click en la opción “ESTADISTICAS” del menú principal de ADPADEC. Aparece la ventana de estadísticas que se muestra en la **Figura 4.16**.

Figura 4.16: Ventana de estadísticas

2. En el campo “NOMBRE DEL ARCHIVO”, ingresar el nombre d12010. Dar un click al botón “ESTADISTICAS”, observar los resultados obtenidos en el “AREA DE RESULTADOS” y en el “AREA DE GRAFICOS”. Se debe obtener un Número de respuestas con Alta Confianza de Altitud y Alta Confianza de Código (Caso 1) de 99.89 %, además de muchos otros resultados.

NOTA: Para una explicación detallada del software ADPADEC consultar la tesis “Desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol”.

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE INGENIERIA ELECTRICA
SISTEMAS DE RADIONAVEGACION AEREA
GUIA DE LABORATORIO N # 3



VISUALIZACION DE DATOS RADAR

CATEDRATICO: Ing. Werner Meléndez

OBJETIVOS

- Familiarizarse con la simbología que utiliza ADPADEC, para la presentación de plots y pistas en el PPI (Indicador de Posición de Plan) ó pantalla radar.
- Realizar el análisis de visualización radar, de cada uno de los blancos contenidos en el archivo decodificado de texto de d12010.txt, con el software ADPADEC.

PARTE I

1.1 REQUERIMIENTOS DE HARDWARE

- Computadora portátil ó de escritorio.
- Espacio en disco duro, mayor a 5 GB, para contener archivos pcap, de por lo menos 100 MB.
- Memoria USB con el software ADPADEC.

1.2 REQUERIMIENTOS DE SOFTWARE

- Linux en una computadora portátil o de escritorio, de preferencia Ubuntu 9.10 karmic koala ó superior.
- Python 2.6, con la librería Tkinter.
- Tshark versión 1.2.2 ó superior.
- Mergecap versión 1.2.2 ó superior.

NOTA: Para la instalación de cada uno de estos programas referirse a la distribución de Linux que se esté utilizando y a la ayuda de instalación que trae cada programa.

1.3 INSTRUCCIONES DE EJECUCION DEL SOFTWARE ADPADEC

Se asume que se está utilizando Linux Ubuntu 9.10 karmic koala, que se cumplen con los requerimientos de software y hardware.

13. Abrir la carpeta personal como muestra la **Figura 1.1**.
14. Conectar al puerto USB la memoria de datos.
15. Copiar la carpeta ADPADEC de la USB a la carpeta personal, como muestra la **Figura 1.2**. Dentro de la carpeta ADPADEC, ya se encuentra el archivo d12010.pcap.

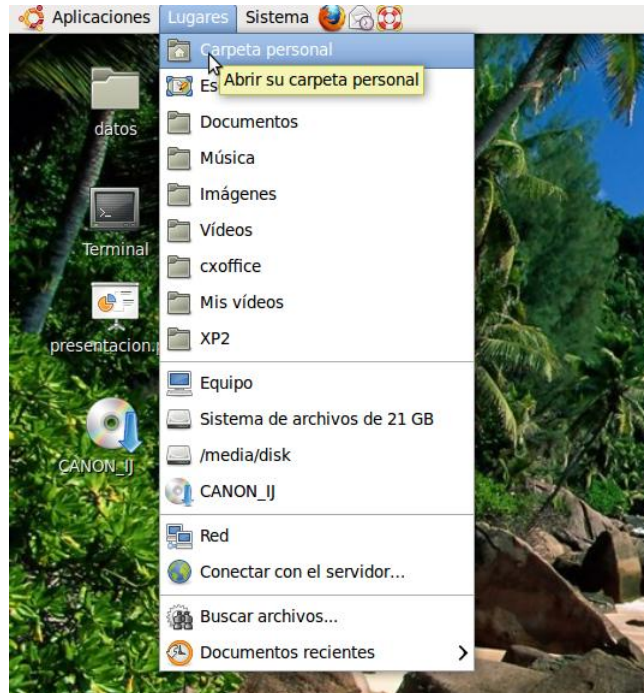


Figura 1.1: Abriendo la carpeta personal en Linux Ubuntu 10.0

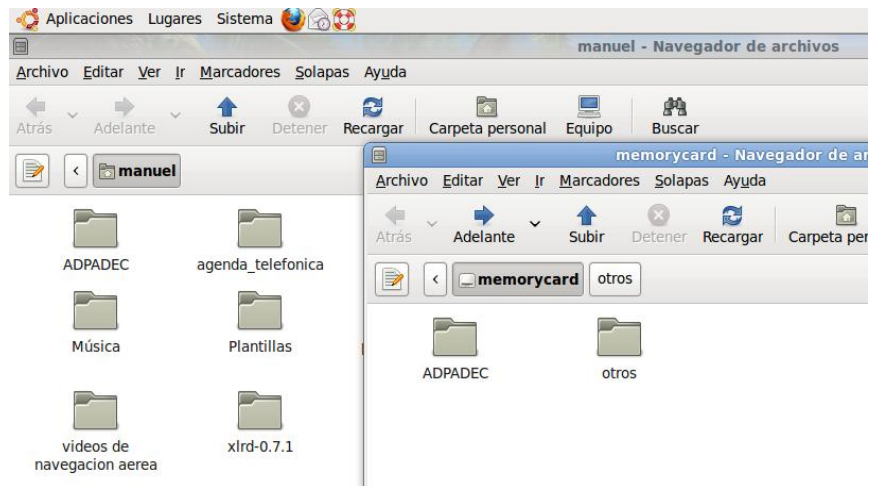


Figura 1.2: Copiando la carpeta ADPADEC de la USB al directorio personal

16. Ejecutar el terminal de Linux como muestra la **Figura 1.3**.

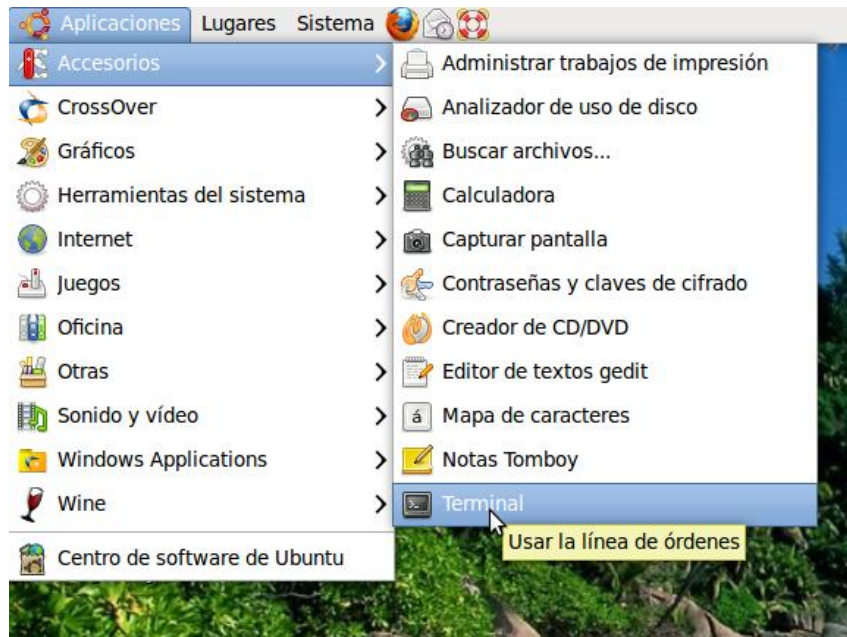


Figura 1.3: Copiando la carpeta ADPADEC de la USB al directorio personal

17. En el terminal de Linux acceder a la carpeta ADPADEC y ejecutar las instrucciones para la ejecución del sistema como muestra la **Figura 1.4** y aparecerá la interface grafica que se muestra en la **Figura 1.5**.

```
root@manuel-laptop:/home/manuel# cd ADPADEC
root@manuel-laptop:/home/manuel/ADPADEC# python ADPADEC.py
```

Figura 1.4: Instrucciones en consola Linux para ejecutar el sistema ADPADEC



Figura 1.5: Interface grafica de usuario del software ADPADEC

PARTE II

2.1 CONCEPTOS DE PLOTS Y PISTAS RADAR

Un plot, es un eco radar, que puede ser visualizado mediante una simbología propia del radar, en una vuelta de antena. Esta simbología debe ser clara, de manera que sea fácilmente entendible, para los usuarios que utilizan determinado radar.

Una pista es el historial de plots, es decir que determinado blanco tiene una pista que se forma por determinados plots en vueltas sucesivas de antena.

La **Figura 2.1** muestra la simbología utilizada para un plot radar por un determinado radar. Se puede apreciar el código transponder, que identifica a la aeronave el cual es el 4647; el nivel de vuelo representado por los números 0146 y el símbolo del plot que indica su posición (en rango y acimut).



Figura 2.1: Simbología utilizada para un plot radar por un determinado radar

2.2 SIMBOLOGIA DE PLOTS RADAR UTILIZADA POR ADPADEC

El software ADPADEC al igual que otros sistemas de procesamiento de datos radar, posee su simbología en cuanto a los plots radar que se muestran en el PPI. Esta simbología es la que sirve para poder identificar de manera fácil y rápida la información que nos interesa del blanco radar, así tenemos en la **Figura 2.2** la simbología utilizada para un plot radar.



Figura 2.2: Simbología utilizada en el PPI para un plot radar

En la Figura anterior, podemos apreciar que se utiliza la siguiente simbología:

- * : El símbolo “asterisco” es el que se utiliza para representar la posición de la aeronave.
- **A 2665**: La letra “A” indica el Modo A del protocolo Asterix, el cual proporciona el código transponder que el ATC le ha asignado a la aeronave y que el piloto introduce en el transponder. En este caso se trata del código 2665 y el color utilizado es verde.

- **C 35000 ft:** Indica el Modo C del protocolo Asterix, es decir, la altitud que tiene la aeronave en pies. En este caso es de 35000 pies.
- **229.27 NM:** Indica el rango o distancia a la cual se encuentra el blanco de la estación radar. La aeronave esta a 229.27 NM.

NOTA: Para una explicación detallada de la simbología utilizada por el software ADPADEC, consultar la tesis “Desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol”.

PARTE III

3.1 VISUALIZACION DE BLANCOS RADAR

3. Ejecutar el software ADPADEC (sección 1.3).
4. Dar doble click en la opción “VISUALIZACION RADAR” del menú principal de ADPADEC, donde muestra lo marcado en color rojo, en la **Figura 3.1**. Esto hace que se active el botón “DATOS”, en el apartado “OPCIONES DE VISUALIZACION RADAR”, dar un click al boton “DATOS”, que está marcado con rojo, en la **Figura 3.2**, que hace que se ejecute la ventana de visualización radar con sus opciones.

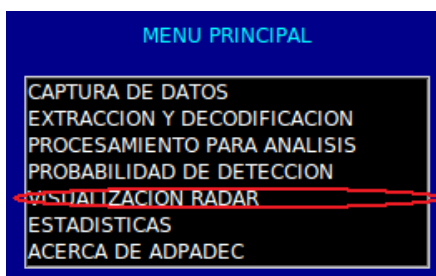


Figura 3.1: Menú principal del sistema ADPADEC



Figura 3.2: Apartado de opciones de visualización radar

5. Especificar los campos que se muestran en la **Figura 3.3**. Entre los errores que puede cometer el usuario están: error de nombre de archivo, que se muestra en la **Figura 3.4**; error de especificación de un código transponder, que muestran las **Figuras 3.5, 3.6, 3.7 y 3.8** y error de tipo de dato como muestra la **Figura 3.9**.



Figura 3.3: Datos a ingresar en la ventana de visualización radar

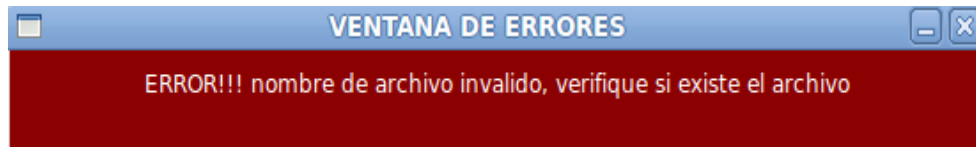


Figura 3.4: Error de nombre de archivo

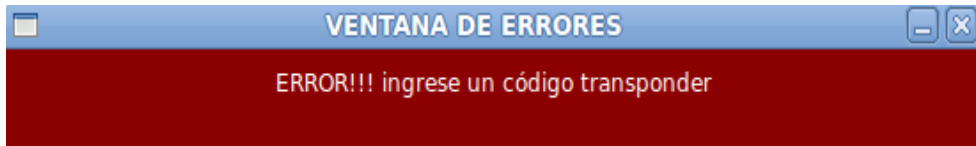


Figura 3.5: Error de especificación de un código transponder

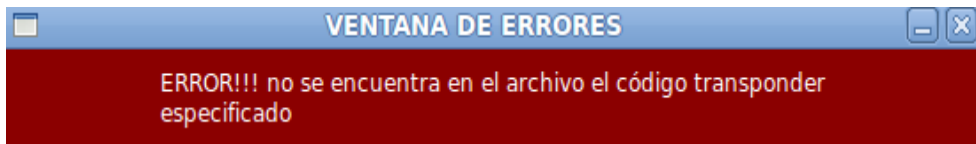


Figura 3.6: Error de especificación de un código transponder

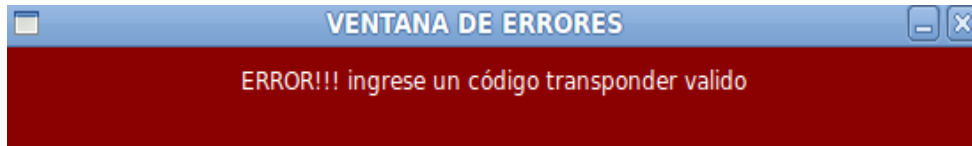


Figura 3.7: Error de tipo de dato

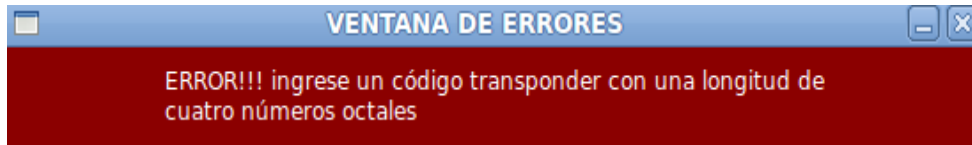


Figura 3.8: Error de especificación de un código transponder

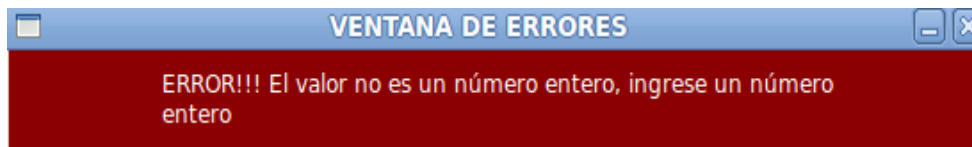


Figura 3.9: Error de tipo de dato

6. Una vez ingresados los datos en los campos, dar click en el botón "INICIAR", que inicia el análisis de visualización radar, que mostrara los plots en el PPI.
7. Observar como aparecen los plots en el PPI y como el indicador de tiempo (ubicado en la parte inferior derecha del área del PPI) cambia de color rojo a blanco, indicando su activación. Hacer una visualización de los blancos radar a una escala de radio máximo de 100 NM (dando click en los botones de "ZOOM +" y "ZOOM -"). Lo observado en el PPI debe corresponder a lo mostrado en la **Figura 3.10**.

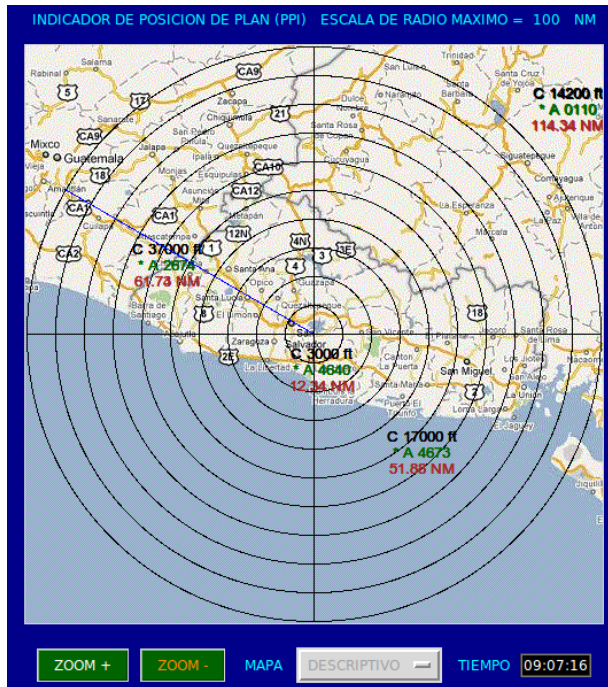


Figura 3.10: Visualización de plots en PPI a una escala de radio máximo de 100 NM

8. Dar click en el botón “DETENER” del apartado “OPCIONES DE VISUALIZACION RADAR” de la ventana principal, con lo cual se detiene la visualización de plots y pistas en el PPI.
9. Cambiar a mapa de satélite con el campo “MAPA” eligiendo la opción “SATELITE”. Dar click en el botón “INICIAR”. Lo observado en el PPI debe corresponder a lo mostrado en la **Figura 3.11**.



Figura 3.11: Visualización de plots en PPI a una escala de radio máximo de 100 NM

3.2 VISUALIZACION DE PISTAS

1. Hacer una visualización de los blancos radar a una escala de radio máximo de 100 NM en PPI.
2. Dar click en el botón “DATOS”, volverá a aparecer la ventana de visualización radar.
3. Especificar los campos que se muestran en la **Figura 3.12**.

Figura 3.12: Datos a ingresar en la ventana de visualización radar

4. Observar como aparecen las pistas en el PPI, durante los próximos minutos. El resultado obtenido a 09:10:56 debería ser como el mostrado en la **Figura 3.13**.

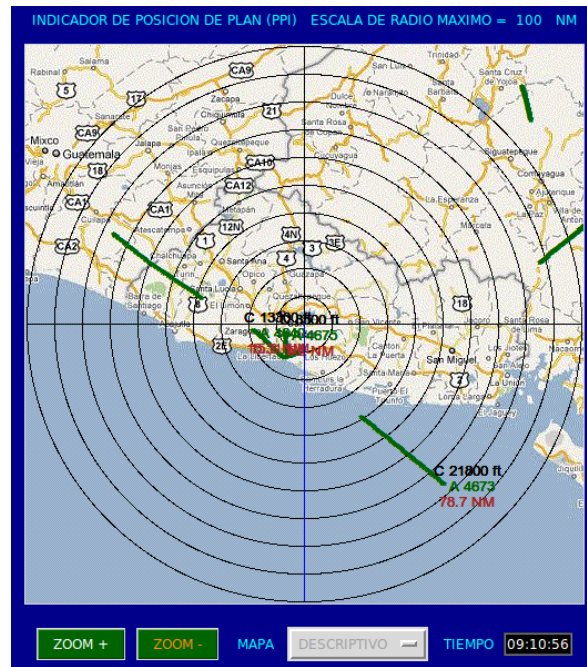


Figura 3.13: Visualización de pistas en PPI a una escala de radio máximo de 100 NM

5. Detener la visualización de pistas en PPI con el botón “DETENER”, de “OPCIONES DE VISUALIZACION RADAR” de la ventana principal.

3.3 VISUALIZACION DE UN BLANCO RADAR

1. Hacer una visualización de los blancos radar a una escala de radio máximo de 25 NM en PPI.
2. Dar click en el botón “DATOS”, volverá a aparecer la ventana de visualización radar.
3. Dar click en el botón “CODIGOS” y verificar que este el código 4640.
4. En el campo “NOMBRE DEL ARCHIVO” ingresar d12010.
5. En el campo “BLANCOS A VISUALIZAR” elegir la opción “UNO”
6. En el campo “CODIGO TRANSPONDER” ingresar el código 4640.
7. En el campo “TIEMPO DE VISUALIZACION” elegir la opción “UTC”.
8. En el campo “HABILITAR PISTAS” dejar la opción “NO”.

9. Dar click en el botón “INICIAR”, para visualizar el blanco en el PPI y su tiempo de aparición. El plot observado en el PPI debe corresponder a lo mostrado en la **Figura 3.14**. Notar que es una aeronave que va despejando del AIES, por el cambio que tiene en su altitud.

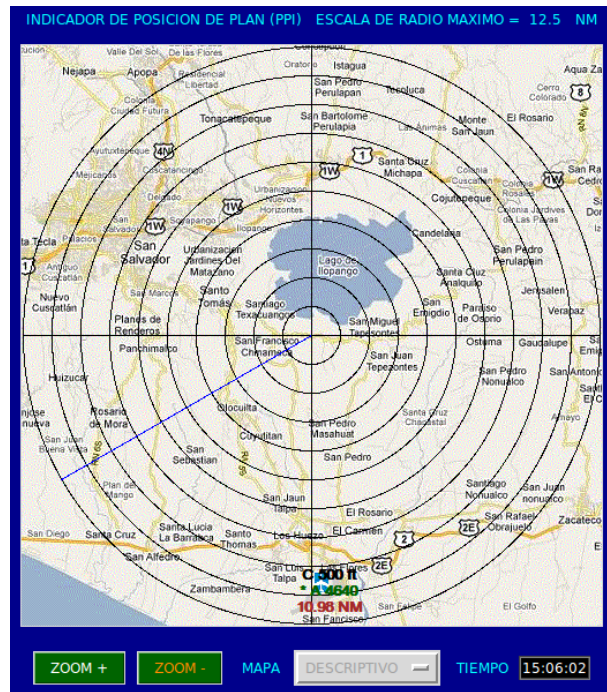


Figura 3.14: Visualización del blanco con código transponder 4640 en PPI

10. Dar click en el botón “SALIR”, de la ventana principal, para salir del sistema.

NOTA: Para una explicación detallada del análisis de visualización radar, del software ADPADEC, consultar la tesis “Desarrollo de un Analizador de Protocolo ASTERIX de Eurocontrol”.

ANEXO B – CODIGOS FUENTES DEL SOFTWARE

ADPADEC.py

```
#!/usr/bin/python

# -*- coding: cp1252 -*-

from GUI import *

class ADPADEC:

    def __init__(self):

        GUI()

if __name__=='__main__':

    A=ADPADEC()
```

GUL.py

```
#!/usr/bin/env python

# Este archivo usa el encoding: utf-8

import sys

import os

from math import *

from string import *

from Tkinter import *

import adpest

#from adpc01 import *

from cat01 import *

from cat02 import *

from lon_t import *

import mostrarb

import capturar

import imp

from Area_de_texto import *

from trama_asterix import *
```

```

from datetime import *
from adpdec import *
from cat01pd import *
from cat02pd import *
from estdpd import *

class GUI:
    def __init__(self):
        self.ventana=Tk()
        self.ventana.title("ADPADEC V1.0 UES EIE")
        self.ventana.maxsize(width=840,height=630)
        self.ventana.minsize(width=840,height=630)
        self.ventana.config(bg="dark blue")
        self.stop=0
        self.tipodgraf=StringVar()
        self.tipodgraf.set(unicode("Probabilidad de Detección","latin-1"))
        self.categoria=0 # para las categorias
        self.ldl1=[] # lista de longitudes de trama
        self.indices=[] # lista de indices de la trama
        self.cursor=[]
        self.cpdb1=[] # coordenadas polares de cada uno de los blancos
        self.cpdb2=[]
        self.blancos=[] # cada uno de los blancos
        self.confda=[] # para la confianza de altitud
        self.confdc=[] # para la confianza de codigo
        self.i=1
        self.temporal=[" "] # para la PD
        self.ro=0
        self.theta=0
        self.act_blanco=1 # variable para activar los blancos detectados
        self.time=500
        self.count=0

```

```

self.numdp = StringVar() # para la captura de paquetes

self.b1=0

self.rda=400 # variable para cambiar la rotacion de antena

self.paquetes=0 # variable para saber el numero de paquetes

self.avisos=""

self.ppidir = os.path.join(os.path.dirname(__file__),'ppi')

self.capdir = os.path.join(os.path.dirname(__file__),'decodificaciones')

self.numdb=0 # para controlar los blancos del SSR,PSR

self.cv1=0 # para controlar la ventana 1

self.cv2=0 # para controlar la ventana 2

self.cv3=0 # para controlar la ventana 3

self.cv4=0 # para controlar la ventana 4

self.cv5=0 # para controlar la ventana 5

self.cv6=0 # para controlar la ventana 6

self.var = IntVar()

self.var1=StringVar()

self.var2=StringVar()

self.var3=StringVar()

self.var3.set("Datos2")

self.var4=StringVar()

self.var6=StringVar()

self.var7=StringVar()

self.var8=StringVar() # para ID del blanco

self.var9=StringVar() # numero de vueltas

self.var10=StringVar()

self.numdbc=StringVar() # para el nuevo analisis de los blancos en PPI

self.numdbc.set("Todos")

self.var10.set("Terminal")

self.var11=StringVar()

self.var11.set("Individual")

self.var12=StringVar() # Tipo de procesamiento de los datos

self.var13=StringVar() # numero de vueltas de antena radar

```



```
self.var13.set("100")
self.tdcapd=StringVar()

self.var14=StringVar()
self.var14.set("POR NUMERO DE PAQUETES")
self.var15=StringVar() # archivo decodificado
self.var15.set("Datos2")
self.var16=StringVar()
self.var16.set("PCAP")
self.var17=StringVar()
self.var17.set("Datos2")
self.thoras=StringVar()
self.tminutos=StringVar()
self.thoras.set("0")
self.tminutos.set("1")
self.nomda=StringVar()
self.nomda.set("2")
self.mapa=StringVar()
self.mapa.set("DESCRIPTIVO")
self.var4.set("00:00:00")
self.var1.set("El Salvador") # default selection
self.var2.set("01") # default selection
self.cat=0 # para la categoria asterix
self.sac=" "
self.ta=" "
self.nb=0
self.horas=0
self.minutos=0
self.segundos=0
self.tiempodc=[]
self.tiempodc2=[] # para ordenar los tiempos
self.identord=[] # para ordenar las identidades
```

```

self.identda=[] # Para identificar a las aeronaves

self.altda=[] # Para guardar la altitud de cada aeronave

self.fallas=[] # Para saber si existen fallas

self.cursor2=[]

self.cursor3=[] # Para el efecto de la linea continua del radar

self.cursor4=[] # Para los graficos de barras

self.x=0

self.xx=0

self.pant1=[90,180,270,359.90]

self.pant2=[359.90,90,180,270]

self.tipodproc=["CAPTURA DE DATOS","EXTRACCION Y DECODIFICACION","PROCESAMIENTO PARA
ANALISIS","PROBABILIDAD DE DETECCION","VISUALIZACION RADAR","ESTADISTICAS","ACERCA DE
ADPADEC"]

self.captura=StringVar() # nombre del archivo de captura

self.captura.set("Datos")

self.actdes4=0

self.inform=StringVar()

self.inform.set("RESULTADOS DEL ANALISIS") # para el analisis de un codigo transponder su archivo

self.idr=["imdr1.gif","imdr2.gif","imdr3.gif","imdr4.gif","imdr5.gif","imdr6.gif","imdr7.gif","imdr8.gif","imdr9.gif","imdr10.gif","imdr11.gif","imdr12.gif"]

self.idr2=["radar1.gif","radar2.gif","radar3.gif","radar4.gif","radar5.gif","radar6.gif","radar7.gif","radar8.gif","radar9.gif","radar10.gif","radar11.gif","radar12.gif"]

self.eppi=[6.25,12.5,25,50,75,100,125,150,175,200,225,250] # las escalas del PPI

self.vpidr=0 # para controlar las imagenes del radar

self.escalamn=225 # para la escala de PPI

self.escdppi=StringVar()

self.escdppi.set(str(self.escalamn))

self.cb=0 # para graficar los blancos cada cierto tiempo

self.gdconf=StringVar() # grado de confianza de codigo y altitud

self.gdconf.set(unicode("Probabilidad de Detección","latin-1"))

self.adusb=StringVar()

self.adusb.set("Datos2") # variable para la visualizacion de uno o varios blancos

self.codtpv=StringVar()

```

```

self.codtpv.set("0017") # codigo transponder para su visualizacion
self.barrido=0 # tiempo de barrido de antena radar
self.angulos=[] # lista de angulos de barrido de antena radar
self.cdmapa=0 # variable para cambiar el mapa del PPI y los colores de los plots radar
self.visual=0 # variable para controlar el menu de visualizacion
self.actmenu=0 # variable para controlar todos los analisis
self.tidvisual=StringVar()
self.tidvisual.set("UTC")
self.pistas=StringVar()
self.pistas.set("NO")
self.elemvisual=0

agrupar=Frame(self.ventana,bg="dark blue")

Label(agrupar,text="MENU PRINCIPAL",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

caja2=Frame(agrupar,bg="dark blue")

self.tipoda=Listbox(caja2,width=30,height=7,bg="black",fg="white")

for i in self.tipodproc:
    self.tipoda.insert(END,i)

self.tipoda.pack(padx=5,pady=5,fill=BOTH,expand=1,side=TOP)

self.tipoda.bind("<Double-Button-1>",self.procesamiento)

caja2.pack(padx=5,pady=5,side=TOP)

Label(agrupar,text="PROCESAMIENTO DE DATOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

caja1=Frame(agrupar,bg="dark blue")

```

```

Marco0=Frame(caja1,bg="dark blue")
Label(Marco0,text="TIPO DE ARCHIVO ",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.tipda=OptionMenu(Marco0,self.var16,"PCAP","TEXT")
self.tipda.pack(padx=5,pady=5,side=LEFT)
Marco0.pack(padx=5,pady=5,side=TOP)

Marco1=Frame(caja1,bg="dark blue")
Label(Marco1,text="NOMBRE",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.entrada=Entry(Marco1,bg="white",fg="black",textvariable=self.var3,width=17)
self.entrada.pack(padx=5,pady=5,side=LEFT)
Marco1.pack(padx=5,pady=5,side=TOP)

Marco2=Frame(caja1,bg="dark blue")
Label(Marco2,text="CODIGO SAC ",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.codsac=OptionMenu(Marco2,self.var1,"El Salvador","Guatemala","Honduras","Belize","Nicaragua","Costa Rica")
self.codsac.pack(padx=5,pady=5,side=LEFT)
Marco2.pack(padx=5,pady=5,side=TOP)

Marco3=Frame(caja1,bg="dark blue")
Label(Marco3,text="CATEGORIA ASTERIX ",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.catast=OptionMenu(Marco3,self.var2,"01","02")
self.catast.pack(padx=5,pady=5,side=LEFT)
Marco3.pack(padx=5,pady=5,side=TOP)

Marco4=Frame(caja1,bg="dark blue")
self.bdp=Button(Marco4,text="PROCESAR",bg="dark cyan",fg="white",command=self.error_archivo)
self.bdp.pack(padx=5,pady=5,side=LEFT)
self.bdreset=Button(Marco4,text="RESET",bg="gray",fg="black",command=self.reset)
self.bdreset.pack(padx=5,pady=5,side=LEFT)
self.bdreset['state'] = DISABLED

```

```

Button(Marco4,text="SALIR",bg="white",fg="dark red",command=sys.exit).pack(padx=5,pady=5,side=LEFT)

Marco4.pack(padx=5,pady=5,side=TOP)

Label(caja1,text="OPCIONES DE VISUALIZACION RADAR",bg="dark
blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

Marco5=Frame(caja1,bg="dark blue")

self.menuppi=Button(Marco5,text="DATOS",bg="dark cyan",fg="white",command=self.ventana4)

self.menuppi.pack(padx=5,pady=5,side=LEFT)

self.menuppi['state'] = DISABLED

self.inppi=Button(Marco5,text="INICIAR",bg="dark cyan",fg="white",command=self.continuar)

self.inppi.pack(padx=5,pady=5,side=LEFT)

self.finppi=Button(Marco5,text="DETENER",bg="gray",fg="dark red",command=self.detener)

self.finppi.pack(padx=5,pady=5,side=LEFT)

self.inppi['state'] = DISABLED

self.finppi['state'] = DISABLED

Marco5.pack(padx=5,pady=5,side=TOP)

caja1.pack(padx=5,pady=5,side=TOP)

self.entrada['state'] = DISABLED

self.codsac['state'] = DISABLED

self.catast['state'] = DISABLED

self.bdp['state'] = DISABLED

self.tipda['state'] = DISABLED

agrupar.pack(padx=5,pady=5,side=LEFT)

Marco1=Frame(self.ventana,bg="dark blue")

cajadppi=Frame(Marco1,bg="dark blue")

```

```
Label(cajadppi,text="INDICADOR DE POSICION DE PLAN (PPI) ESCALA DE RADIO MAXIMO =",bg="dark blue",fg="cyan").pack(padx=2,pady=2,side=LEFT)
```

```
Label(cajadppi,textvariable=self.escdppi,bg="dark blue",fg="cyan").pack(padx=2,pady=2,side=LEFT)
```

```
Label(cajadppi,text=" NM",bg="dark blue",fg="cyan").pack(padx=2,pady=2,side=LEFT)
```

```
cajadppi.pack(padx=5,pady=5,side=TOP)
```

```
self.indicaciones=Canvas(Marco1,width=503,height=503,bg="white")
```

```
self.indicaciones.pack(padx=5,pady=5,side=TOP)
```

```
PPI=Frame(Marco1,bg="dark blue")
```

```
self.zoomb1=Button(PPI,text="ZOOM +",bg="dark green",fg="white",command=self.zoommas)
```

```
self.zoomb1.pack(padx=5,pady=5,side=LEFT)
```

```
self.zoomb2=Button(PPI,text="ZOOM -",bg="dark green",fg="dark orange",command=self.zoommen)
```

```
self.zoomb2.pack(padx=5,pady=5,side=LEFT)
```

```
self.zoomb1['state'] = DISABLED
```

```
self.zoomb2['state'] = DISABLED
```

```
Marco6=Frame(PPI,bg="dark blue")
```

```
Label(Marco6,text="MAPA",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=LEFT)
```

```
self.tdmap=OptionMenu(Marco6,self.mapa,"DESCRIPTIVO","SATELITE")
```

```
self.tdmap.pack(padx=5,pady=5,side=LEFT)
```

```
self.tdmap['state'] = DISABLED
```

```
Marco6.pack(padx=5,pady=5,side=LEFT)
```

```
Label(PPI,text="TIEMPO",bg="dark blue",fg="cyan").pack(padx=2,pady=2,side=LEFT)
```

```
self.etiqueta=Label(PPI,borderwidth=2,relief=GROOVE,textvariable=self.var4,bg="black",fg="red")
```

```
self.etiqueta.pack(padx=5,pady=5,side=LEFT)
```

```
PPI.pack(padx=5,pady=5,side=TOP)
```

```
self.arcos("black")
```

```
Marco1.pack(side=LEFT)
```

```
self.ventana.mainloop()
```

```
def reset(self):
```

```

self.bdreset['state'] = DISABLED
self.entrada['state'] = DISABLED
self.codsac['state'] = DISABLED
self.catast['state'] = DISABLED
self.bdp['state'] = DISABLED
self.tipda['state'] = DISABLED
self.menuppi['state'] = DISABLED
self.inppi['state'] = DISABLED
self.finppi['state'] = DISABLED
self.act_blanco=0
self.actmenu=0
self.zoomb2['state'] = DISABLED
self.zoomb1['state'] = DISABLED

def ventana1(self):
    if self.cv1==0:
        self.entrada['state'] = DISABLED
        self.codsac['state'] = DISABLED
        self.catast['state'] = DISABLED
        self.bdp['state'] = DISABLED
        self.tipda['state'] = DISABLED
        self.menuppi['state'] = DISABLED
        self.inppi['state'] = DISABLED
        self.finppi['state'] = DISABLED

        self.cajadd1 = Toplevel()
        self.cajadd1.config(bg="dark blue")
        self.cajadd1.title("CAPTURA DE DATOS")
        self.cajadd1.maxsize(width=550,height=250)
        self.cajadd1.minsize(width=550,height=250)
        self.cajadd1.bind("<Destroy>",self.habvent)

```

```
Label(self.cajadd1,text="ARCHIVO DE CAPTURA DE DATOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
```

```
Marco1=Frame(self.cajadd1,bg="dark blue")
```

```
Label(Marco1,text="NOMBRE DE ARCHIVO DE CAPTURA",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
```

```
self.enda=Entry(Marco1,bg="white",fg="black",textvariable=self.captura,width=15)
```

```
self.enda.pack(padx=5,pady=5,side=LEFT)
```

```
Marco1.pack(padx=5,pady=5,side=TOP)
```

```
Marco2=Frame(self.cajadd1,bg="dark blue")
```

```
Label(Marco2,text="NUMERO DE ARCHIVO DE CAPTURA",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
```

```
self.endp=Entry(Marco2,bg="white",fg="black",textvariable=self.nomda,width=15)
```

```
self.endp.pack(padx=5,pady=5,side=LEFT)
```

```
Marco2.pack(padx=5,pady=5,side=TOP)
```

```
Label(self.cajadd1,text="TIEMPO DE CAPTURA DE DATOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
```

```
Marco3=Frame(self.cajadd1,bg="dark blue")
```

```
Label(Marco3,text="HORAS",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
```

```
self.endh=Entry(Marco3,bg="white",fg="black",textvariable=self.thoras,width=12)
```

```
self.endh.pack(padx=10,pady=10,side=LEFT)
```

```
Label(Marco3,text="MINUTOS",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
```

```
self.endm=Entry(Marco3,bg="white",fg="black",textvariable=self.tminutos,width=12)
```

```
self.endm.pack(padx=10,pady=10,side=LEFT)
```

```
Marco3.pack(padx=5,pady=5,side=TOP)
```

```
Marco4=Frame(self.cajadd1,bg="dark blue")
```

```
self.botoncap=Button(Marco4,text="CAPTURAR",bg="dark cyan",fg="white",command=self.cerrar)
```

```
self.botoncap.pack(padx=5,pady=5,side=LEFT)
```

```
Button(Marco4,text="CAPTURAR Y CONVERTIR",bg="dark cyan",fg="white",command=self.cerrar2).pack(padx=5,pady=5,side=LEFT)
```



```

    Button(Marco4,text="CERRAR",bg="gray",fg="dark
red",command=self.cerrarv1).pack(padx=5,pady=5,side=LEFT)

    Button(Marco4,text="SALIR",bg="white",fg="dark red",command=sys.exit).pack(padx=5,pady=5,side=LEFT)

    Marco4.pack(padx=5,pady=5,side=TOP)

self.cv1=1

def ventana2(self,categoria):

    self.entrada['state'] = DISABLED
    self.codsac['state'] = DISABLED
    self.catast['state'] = DISABLED
    self.bdp['state'] = DISABLED
    self.tipda['state'] = DISABLED
    self.menuppi['state'] = DISABLED
    self.inppi['state'] = DISABLED
    self.finppi['state'] = DISABLED

    self.cajadd2 = Toplevel()
    self.cajadd2.config(bg="dark blue")
    self.cajadd2.title("RESULTADOS DE LA EXTRACCION Y DECODIFICACION DE DATOS")
    self.cajadd2.maxsize(width=925,height=320)
    self.cajadd2.minsize(width=925,height=320)
    self.cajadd2.bind("<Destroy>",self.cerrarv2)

    caja2=Frame(self.cajadd2,bg="dark blue")
    Label(caja2,text="TRAMAS ASTERIX",bg="dark blue",fg="cyan").pack(padx=5,pady=10,side=TOP)
    scrollbar = Scrollbar(caja2,bg="dark blue",orient=VERTICAL)
    scrollbar2 = Scrollbar(caja2,bg="dark blue",orient=HORIZONTAL)

self.Idtast=Listbox(caja2,width=30,height=20,bg="white",fg="brown",yscrollcommand=scrollbar.set,xscrollcommand=scrollbar2.set)

    scrollbar.config(command=self.Idtast.yview)

```

```

scrollbar.pack(side=RIGHT, fill=Y)

scrollbar2.config(command=self.ldtast.xview)
scrollbar2.pack(side=BOTTOM, fill=X)

self.ldtast.pack(padx=5,pady=5,fill=BOTH,expand=1,side=TOP)
if categoria=="01":
    archivo=open(os.path.join(self.capdir,self.var3.get()+"t01.txt"),"r+")
elif categoria=="02":
    archivo=open(os.path.join(self.capdir,self.var3.get()+"t02.txt"),"r+")
lineas=archivo.readlines()
archivo.close()

granlista=[]

for i in range(len(lineas)):
    if lineas[i].count(":")==0:
        granlista.append(lineas[i])

lista1=[]

for k in granlista:
    lista1.append(k)

x=0

for i in lista1: # para encontrar los enter
    if i=="\n":
        x=x+1

for i in range(x):
    lista1.remove('\n') # para remover los enter

for i in lista1:
    self.ldtast.insert(END,i[0:len(i)-1])

```

```

self.sac_cat()

self.ldest.bind("<Double-Button-1>",self.mensaje)

caja2.pack(padx=5,pady=5,side=LEFT)

caja3=Frame(self.cajadd2,bg="dark blue")
Label(caja3,text="AREA DE RESULTADOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
self.At1=Area_de_texto(caja3,87,15,"gray","black")
self.At1.act_des(1)

Button(caja3,text="CERRAR",bg="gray",fg="dark red",command=self.cerrar_v2).pack(padx=5,pady=5,side=TOP)
caja3.pack(padx=10,pady=10,side=LEFT)
self.cv2=1

def ventana3(self):
    self.entrada['state'] = DISABLED
    self.codsac['state'] = DISABLED
    self.catast['state'] = DISABLED
    self.bdp['state'] = DISABLED
    self.tipda['state'] = DISABLED
    self.menuppi['state'] = DISABLED
    self.inppi['state'] = DISABLED
    self.finppi['state'] = DISABLED
    self.cajadd3 = Toplevel()
    self.cajadd3.config(bg="dark blue")
    self.cajadd3.title("PROBABILIDAD DE DETECCION")
    self.cajadd3.maxsize(width=1175,height=610)
    self.cajadd3.minsize(width=1175,height=610)
    self.cajadd3.bind("<Destroy>",self.habvent3)

caja0=Frame(self.cajadd3,bg="dark blue")

```

```

caja1=Frame(caja0,bg="dark blue")
Label(caja1,text="DATOS A INGRESAR",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

Marco0=Frame(caja1,bg="dark blue")
Label(Marco0,text="NOMBRE DEL ARCHIVO",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.probdd=Entry(Marco0,bg="white",fg="black",textvariable=self.var15,width=18)
self.probdd.pack(padx=5,pady=5,side=LEFT)

Marco1=Frame(Marco0,bg="dark blue")
Label(Marco1,text="ALTITUD          MINIMA          A          CONSIDERAR          (ft)",bg="dark
blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.tipor=Entry(Marco1,bg="white",fg="black",textvariable=self.var13,width=11)
self.tipor.pack(padx=5,pady=5,side=LEFT)
Marco1.pack(padx=5,pady=5,side=TOP)

Marco0.pack(padx=5,pady=5,side=TOP)

Marco2=Frame(caja1,bg="dark blue")
Label(Marco2,text="CODIGO TRANSPONDER",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.codtrans=Entry(Marco2,bg="white",fg="black",textvariable=self.var8,width=18)
self.codtrans.pack(padx=5,pady=5,side=LEFT)

Marco3=Frame(Marco2,bg="dark blue")
Label(Marco3,text="PROBABILIDAD          DE          DETECCION",bg="dark
blue",fg="white").pack(padx=5,pady=5,side=LEFT)
self.tipodpd=OptionMenu(Marco3,self.var11,"Individual","Total","Falsos Blancos")
self.tipodpd.pack(padx=5,pady=5,side=LEFT)
Marco3.pack(padx=5,pady=5,side=TOP)

Marco2.pack(padx=5,pady=5,side=TOP)

Label(caja1,text="OPCION          DEL          AREA          DE          GRAFICOS",bg="dark
blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

```

```

Marco5=Frame(caja1,bg="dark blue")

Label(Marco5,text="GRAFICO DE BARRAS DEL AREA DE GRAFICOS",bg="dark
blue",fg="white").pack(padx=5,pady=5,side=LEFT)

self.tdconf=OptionMenu(Marco5,self.gdconf,unicode("Probabilidad de Detección","latin-1"),unicode("Probabilidad
de Detección vs Rango","latin-1"),unicode("Probabilidad de Detección vs Altitud","latin-1"))

self.tdconf.pack(padx=5,pady=5,side=LEFT)

Marco5.pack(padx=5,pady=5,side=TOP)

Marco4=Frame(caja1,bg="dark blue")

Button(Marco4,text="CODIGOS",bg="dark
green",fg="white",command=self.codigosdif).pack(padx=15,pady=15,side=LEFT)

Button(Marco4,text="CALCULAR",bg="dark
cyan",fg="white",command=self.cerrarv3).pack(padx=15,pady=15,side=LEFT)

self.botonda=Button(Marco4,text="ANALIZAR",bg="dark green",fg="dark orange",command=self.anact)

self.botonda.pack(padx=5,pady=5,side=LEFT)

Button(Marco4,text="LIMPIAR",bg="gray",fg="dark
blue",command=self.limatpd).pack(padx=15,pady=15,side=LEFT)

Button(Marco4,text="CERRAR",bg="gray",fg="dark
red",command=self.cerrarv3).pack(padx=15,pady=15,side=LEFT)

Button(Marco4,text="SALIR",bg="white",fg="dark red",command=sys.exit).pack(padx=15,pady=15,side=LEFT)

Marco4.pack(padx=5,pady=5,side=TOP)

self.botonda['state'] = DISABLED

caja1.pack(padx=5,pady=5,side=LEFT)

caja2=Frame(caja0,bg="dark blue")

self.resdest=Canvas(caja2,width=420,height=330,bg="white")

self.resdest.pack(padx=5,pady=5,side=TOP)

self.resdest.create_text(50,10,text="PD (%)",font="New_Times_Roman")

x=28

lista=["100","90","80","70","60","50","40","30","20","10"," "]

for i in range(0,10):

    self.resdest.create_line(38,x,44,x,fill="black")

```

```

self.resdest.create_text(20,x,text=lista[i],font="New_Times_Roman")

x=x+28

Label(caja2,text="AREA DE GRAFICOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
caja2.pack(padx=5,pady=5,side=LEFT)

caja0.pack(padx=5,pady=5,side=TOP)

Label(self.cajadd3,text="AREA DE RESULTADOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

self.At2=Area_de_texto(self.cajadd3,160,10,"black","white")
self.At2.act_des(1)

self.resdest.create_line(40,20,40,300,fill="black")
self.resdest.create_line(40,300,340,300,fill="black")

self.cv3=1

def ventana4(self):

self.menuppi['state'] = DISABLED
self.zoomb1['state'] = DISABLED
self.zoomb2['state'] = DISABLED
self.cajadd4 = Toplevel()
self.cajadd4.config(bg="dark blue")
self.cajadd4.title("VISUALIZACION RADAR")
self.cajadd4.maxsize(width=560,height=390)
self.cajadd4.minsize(width=560,height=390)
self.cajadd4.bind("<Destroy>",self.habvent4)
Label(self.cajadd4,text="DATOS A INGRESAR",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
Marco0=Frame(self.cajadd4,bg="dark blue")
Label(Marco0,text="NOMBRE DEL ARCHIVO",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)

```

```

Entry(Marco0,bg="white",fg="black",textvariable=self.adusb,width=15).pack(padx=5,pady=5,side=LEFT)
Label(Marco0,text="BLANCOS A VISUALIZAR",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
OptionMenu(Marco0,self.numdbc,"Todos","Uno").pack(padx=5,pady=5,side=LEFT)
Marco0.pack(padx=5,pady=5,side=TOP)

Marco2=Frame(self.cajadd4,bg="dark blue")
Label(Marco2,text="CODIGO TRANSPONDER",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
Entry(Marco2,bg="white",fg="black",textvariable=self.codtpv,width=14).pack(padx=5,pady=5,side=LEFT)
Label(Marco2,text="TIEMPO DE VISUALIZACION",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
OptionMenu(Marco2,self.tidvisual,"UTC","CA").pack(padx=5,pady=5,side=LEFT)
Marco2.pack(padx=5,pady=5,side=TOP)

Marco3=Frame(self.cajadd4,bg="dark blue")
Label(Marco3,text="HABILITAR PISTAS",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)
OptionMenu(Marco3,self.pistas,"NO","SI").pack(padx=5,pady=5,side=LEFT)
Button(Marco3,text="CODIGOS",bg="dark
green",fg="white",command=self.codigosdif2).pack(padx=5,pady=5,side=LEFT)
Button(Marco3,text="INICIAR",bg="dark cyan",fg="white",command=self.vradar).pack(padx=5,pady=5,side=LEFT)
Button(Marco3,text="CERRAR",bg="gray",fg="dark red",command=self.cerrarv4).pack(padx=5,pady=5,side=LEFT)
Button(Marco3,text="SALIR",bg="white",fg="dark red",command=sys.exit).pack(padx=5,pady=5,side=LEFT)
Marco3.pack(padx=10,pady=10,side=TOP)

Label(self.cajadd4,text="AREA DE RESULTADOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
self.At3=Area_de_texto(self.cajadd4,70,10,"black","white")

self.At3.act_des(1)

self.cv4=1

def ventana5(self,mensaje,color):
    self.cajadd5 = Toplevel()
    self.cajadd5.config(bg=color)
    self.cajadd5.title("VENTANA DE ERRORES")
    self.cajadd5.maxsize(width=550,height=50)

```

```

self.cajadd5.minsize(width=550,height=50)

self.cajadd5.bind("<Destroy>",self.habvent5)

msg=Message(self.cajadd5,text=mensaje,width=400,bg=color,fg="white")

msg.pack(padx=5,pady=5)

self.cv5=1

def ventana6(self):

    self.entrada['state'] = DISABLED

    self.codsac['state'] = DISABLED

    self.catast['state'] = DISABLED

    self.bdp['state'] = DISABLED

    self.tipda['state'] = DISABLED

    self.menuppi['state'] = DISABLED

    self.inppi['state'] = DISABLED

    self.finppi['state'] = DISABLED

    self.cajadd6 = Toplevel()

    self.cajadd6.config(bg="dark blue")

    self.cajadd6.title("ESTADISTICAS")

    self.cajadd6.maxsize(width=1200,height=450)

    self.cajadd6.minsize(width=1200,height=450)

    self.cajadd6.bind("<Destroy>",self.habvent6)

    caja1=Frame(self.cajadd6,bg="dark blue")

    Marco0=Frame(caja1,bg="dark blue")

    Label(Marco0,text="AREA DE GRAFICOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)

    self.grafp=Canvas(Marco0,width=420,height=329,bg="white")

    self.grafp.pack(padx=5,pady=5,side=TOP)

    self.grafp.create_line(40,20,40,300,fill="black")

    self.grafp.create_line(40,300,360,300,fill="black")

    Marcob1=Frame(Marco0,bg="dark blue")

    Label(Marcob1,text="NOMBRE DEL ARCHIVO",bg="dark blue",fg="white").pack(padx=5,pady=5,side=LEFT)

```



```

Entry(Marcob1,bg="white",fg="black",textvariable=self.var17,width=10).pack(padx=5,pady=5,side=LEFT)
Marcob1.pack(padx=5,pady=5,side=TOP)

Marco0.pack(padx=5,pady=5,side=LEFT)

Marco1=Frame(caja1,bg="dark blue")
Label(Marco1,text="AREA DE RESULTADOS",bg="dark blue",fg="cyan").pack(padx=5,pady=5,side=TOP)
self.At4=Area_de_texto(Marco1,100,22,"black","white")
self.At4.act_des(1)
Marcob2=Frame(Marco1,bg="dark blue")
Button(Marcob2,text="ESTADISTICAS",bg="dark
cyan",fg="white",command=self.errorarch).pack(padx=10,pady=10,side=LEFT)
Button(Marcob2,text="LIMPIAR",bg="dark
gray",fg="blue",command=self.bAt4).pack(padx=10,pady=10,side=LEFT)
Button(Marcob2,text="CERRAR",bg="gray",fg="dark
red",command=self.cerrarv6).pack(padx=10,pady=10,side=LEFT)
Button(Marcob2,text="SALIR",bg="white",fg="dark red",command=sys.exit).pack(padx=10,pady=10,side=LEFT)
Marcob2.pack(padx=5,pady=5,side=TOP)
Marco1.pack(padx=5,pady=5,side=LEFT)
caja1.pack(padx=5,pady=5,side=TOP)
self.cv6=1

def valdct(self,codv):
    lista=["0","1","2","3","4","5","6","7"]
    dig=0
    if len(codv)==0:
        self.ventana5(unicode("ERROR!!! ingrese un código transponder","latin-1"),"dark red")
    elif len(codv)>4:
        self.ventana5(unicode("ERROR!!! ingrese un código transponder con una longitud de cuatro números
octales","latin-1"),"dark red")
    else:
        codigo=codv
        for i in lista:
            if codigo[0:1]==i:

```

```

        dig=dig+1
        break
for i in lista:
    if codigo[1:2]==i:
        dig=dig+1
        break
for i in lista:
    if codigo[2:3]==i:
        dig=dig+1
        break
for i in lista:
    if codigo[3:4]==i:
        dig=dig+1
        break
if dig<4:
    self.ventana5(unicode("ERROR!!! ingrese un código transponder valido","latin-1"),"dark red")
return dig

def valdalt(self):
    var=0
    try:
        altitud=int(self.var13.get())
    except ValueError:
        self.ventana5(unicode("ERROR!!! valor de altitud mínima invalido","latin-1"),"dark red")
    else:
        if int(self.var13.get())>=0:
            altitud=int(self.var13.get())
            var=1
        else:
            self.ventana5(unicode("ERROR!!! valor de altitud mínima invalido","latin-1"),"dark red")
    return var

```

```

def procesamiento(self,event):
    a=self.tipoda.curselection()
    ab=int(a[0])+1
    if ab==1 and self.actmenu==0:
        self.actmenu=1
        self.bdreset['state'] = DISABLED
        self.ventana1()
    elif ab==2 and self.actmenu==0:
        self.actmenu=1
        self.bdreset['state'] = NORMAL
        self.habproc1()
    elif ab==3 and self.actmenu==0:
        self.actmenu=1
        self.bdreset['state'] = NORMAL
        self.habproc2()
    elif ab==4 and self.actmenu==0:
        self.actmenu=1
        self.bdreset['state'] = DISABLED
        if self.cv3==0:
            self.ventana3()
    elif ab==5 and self.actmenu==0:
        self.bdreset['state'] = NORMAL
        self.zoomb2['state'] = NORMAL
        self.inppi['state'] = DISABLED
        self.finppi['state'] = DISABLED
        self.menuppi['state'] = NORMAL
        self.tdmap['state'] = NORMAL
        self.visual=1
        self.actmenu=1
        self.zoomb1['state'] = NORMAL
        self.visualizar()
    elif ab==6 and self.actmenu==0:

```

```

self.actmenu=1

self.bdreset['state'] = DISABLED

if self.cv6==0:
    self.ventana6()

elif ab==7 and self.actmenu==0:
    self.actmenu=1
    self.creditos()

def cerrarv1(self):
    self.cajadd1.destroy()
    self.actmenu=0

def cerrarvn3(self):
    self.cajadd3.destroy()
    self.actmenu=0

def cerrarv3(self):
    for i in self.temporal:
        self.resdest.delete(i)
    self.temporal=[]
    if self.var11.get()=="Individual":
        digitos=self.valdct(self.var8.get())
        if digitos==4:
            val=self.valdalt()
            if val==1:
                res=self.vcodtrans(self.var8.get(),self.var15.get())
                if res==1:
                    if self.gdconf.get()==unicode("Probabilidad de Detección","latin-1"):
                        x=28
                        self.buscarct()
                elif res==0:

```

```
self.ventana5(unicode("ERROR!!! no se encuentra en el archivo el código transponder especificado", "latin-1"), "dark red")
```

```
if self.var11.get()=="Total":
```

```
    val=self.valdalt()
```

```
    if val==1:
```

```
        self.buscarct()
```

```
if self.var11.get()=="Falsos Blancos":
```

```
    val=self.valdalt()
```

```
    if val==1:
```

```
        self.buscarct()
```

```
def cerrarv4(self):
```

```
    self.cajadd4.destroy()
```

```
    self.cv4=0
```

```
def cerrarv5(self):
```

```
    self.cajadd5.destroy()
```

```
    self.cv5=0
```

```
def cerrarv6(self):
```

```
    self.cajadd6.destroy()
```

```
    self.cv6=0
```

```
    self.actmenu=0 # activar los otros analisis
```

```
def habvent4(self,event):
```

```
    self.cv4=0
```

```
def habvent5(self,event):
```

```
    self.cv5=0
```

```

def habvent6(self,event):

    self.cv6=0

    self.actmenu=0

def evv3(self):

    if self.var11.get()=="Individual":

        self.codtrans['state'] = NORMAL

        self.botonda['state'] = NORMAL

    elif self.var11.get()=="Total":

        self.codtrans['state'] = DISABLED

        self.botonda['state'] = DISABLED

    self.cajadd3.after(1,self.evv3)

def cerrar(self):

    self.cajadd1.destroy()

    self.cv1=0

    self.evdhym()

def cerrar2(self):

    self.cajadd1.destroy()

    self.cv1=0

    self.evdhym()

    self.decopcap(self.captura.get()+self.nomda.get())

def evdhym(self): # Para la validacion de las horas y minutos

    try:

        h=int(self.thoras.get())

        m=int(self.tminutos.get())

    except ValueError:

        self.ventana5(unicode("ERROR!!! El valor no es un número entero, ingrese un número entero","latin-1"),"dark red")

    else:

        if m>=60:

```

```

        self.ventana5("ERROR!!! El valor de minutos es menor a 60","dark red")

elif m<60:

    print("ADPADEC V1.0 UES EIE\nCapturando paquetes de la red LAN")

    self.ejecutar()

def buscarct(self):

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

    listadt=self.calcdt1()

    if self.var11.get()=="Individual":

        self.botonda['state'] = NORMAL

        self.inform.set("RESULTADOS DEL ANALISIS")

    try:

        archivo=open(os.path.join(self.capdir,self.var15.get()+"01dc.txt"))

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

        if self.gdconf.get()==unicode("Probabilidad de Detección","latin-1"):

objeto=estc01(os.path.join(self.capdir,self.var15.get()+"01dc.txt"),self.var8.get(),os.path.join(self.capdir,self.var15.get()+"
02dc.txt"),1,self.var13.get(),os.path.join(self.capdir,"Buffer.txt"),os.path.join(self.capdir,"Bufferc.txt"))

        uno=objeto.est1()

        pd=uno[1]

        dos=uno[0]

        fijs=uno[7]

        dos.reverse()

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

    self.grafbarras(self.resdest,pd,"dark cyan", " ",80)

    self.At2.act_des(0)

```

```

self.At2.borrar_texto()

proceso=self.calcdt2(listadt[1])

self.At2.insertar_texto("Tiempo calculado del proceso "+proceso[1]+"n")

if fijo==0:

    self.At2.insertar_texto(unicode("Probabilidad de Detección del blanco = ","latin-1")+dos[0])

    self.At2.insertar_texto(unicode("Número de detecciones validas del blanco = ","latin-1")+dos[1])

    self.At2.insertar_texto(unicode("Número total de detecciones del blanco = ","latin-1")+dos[2])

elif fijo==1:

    self.At2.insertar_texto(unicode("Blanco con código transponder ","latin-1")+dos[2]+" "+dos[1]+" "+dos[0])

    self.botonda['state'] = DISABLED

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

elif fijo==2:

    self.At2.insertar_texto(unicode("Blanco con código transponder ","latin-1")+dos[2]+" "+dos[1]+" "+dos[0]+unicode("para el cálculo de la Probabilidad de Detecciónn","latin-1"))

    self.botonda['state'] = DISABLED

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

elif fijo==3:

    self.At2.insertar_texto(unicode("Blanco con código transponder ","latin-1")+dos[2]+" "+dos[1]+" "+dos[0]+unicode("para el cálculo de la Probabilidad de Detecciónn","latin-1"))

    self.botonda['state'] = DISABLED

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Rango","latin-1"):

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

self.At2.act_des(0)

```



```

        self.At2.borrar_texto()

        self.At2.insertar_texto(unicode("El calculo de Probabilidad de Detección vs Rango y sus gráficos solo son
validos para la Probabilidad de Detección Total","latin-1"))

        self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Altitud","latin-1"):

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

    self.At2.act_des(0)

    self.At2.borrar_texto()

    self.At2.insertar_texto(unicode("El calculo de Probabilidad de Detección vs Altitud y sus gráficos solo son
validos para la Probabilidad de Detección Total","latin-1"))

    self.At2.act_des(1)

elif self.var11.get()=="Total":

    self.botonda['state'] = DISABLED

    self.inform.set("RESULTADOS DEL ANALISIS")

    try:

        archivo=open(os.path.join(self.cadir,self.var15.get()+"01dc.txt"))

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

objeto=estc01(os.path.join(self.cadir,self.var15.get()+"01dc.txt"),self.var8.get(),os.path.join(self.cadir,self.var15.get()+"
02dc.txt"),2,self.var13.get(),os.path.join(self.cadir,"Buffer.txt"),os.path.join(self.cadir,"Bufferc.txt"))

        uno=objeto.est1()

        pd=uno[1]

        dos=uno[0]

        dos.reverse()

if self.gdconf.get()==unicode("Probabilidad de Detección","latin-1"):

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

```

```

self.grafbarras(self.resdest,pd,"dark orange", " ",80)

self.At2.act_des(0)

self.At2.borrar_texto()

proceso=self.calcdt2(listadt[1])

self.At2.insertar_texto("Tiempo calculado del proceso "+proceso[1])

x=0

for i in dos:

    if x==0:

        self.At2.insertar_texto(unicode("Probabilidad de Detección Total = ", "latin-1")+i)

    else:

        self.At2.insertar_texto(unicode("Probabilidad de Detección de blanco con código transponder ", "latin-1")+i)

    x=x+1

self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Rango", "latin-1"):

    lista=[]

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

    rmax=uno[2]

    prob=uno[3]

    rmin=uno[5]

    minimo=min(rmin)

    maximo=max(rmax)

    probmin=min(prob)

    for i in range(len(rmin)):

        if rmin[i]==minimo:

            posdmin=i

    for i in range(len(rmax)):

        if rmax[i]==maximo:

            posdmax=i

    for i in range(len(prob)):

```

```

    if prob[i]==probmin:
        posdpmin=i

self.temporal.append(self.resdest.create_text(359,300,text="NM",font="New_Times_Roman")) # columna, fila
self.grafbarras(self.resdest,prob[posdmax],"green",str(maximo),80) # Rango maximo y su PD
self.grafbarras(self.resdest,prob[posdmin],"yellow",str(minimo),160) # Rango minimo y su PD
self.grafbarras(self.resdest,probmin,"cyan",str(rmax[posdpmin]),240) # PD minima y su rango

lista.append(unicode("Rango máximo = ","latin-1")+str(maximo)+unicode(" NM Probabilidad de Detección = ","latin-1")+str(prob[posdmax])+" % \n")

lista.append(unicode("Rango mínimo = ","latin-1")+str(minimo)+unicode(" NM Probabilidad de Detección = ","latin-1")+str(prob[posdmin])+" %\n")

lista.append(unicode("Probabilidad de Detección mínima = ","latin-1")+str(probmin)+unicode(" % a un Rango máximo de ","latin-1")+str(rmax[posdpmin])+" NM \n")

lista.reverse()

self.At2.act_des(0)

self.At2.borrar_texto()

proceso=self.calcdt2(listadt[1])

self.At2.insertar_texto("Tiempo calculado del proceso "+proceso[1])

for i in lista:
    self.At2.insertar_texto(i)

self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Altitud","latin-1"):

    lista=[]

    for i in self.temporal:
        self.resdest.delete(i)

    self.temporal=[]

    amax=uno[4]

    amin=uno[6]

    prob=uno[3]

    minimo=min(amin)

    maximo=max(amax)

    probmin=min(prob)

```

```

for i in range(len(amin)):
    if amin[i]==minimo:
        posdmin=i
for i in range(len(amax)):
    if amax[i]==maximo:
        posdmax=i
for i in range(len(prob)):
    if prob[i]==probmin:
        posdpmin=i

self.temporal.append(self.resdest.create_text(357,300,text="ft",font="New_Times_Roman")) # columna,fila
self.grafbarras(self.resdest,prob[posdmax],"magenta",str(int(maximo)),80) # Rango maximo y su PD
self.grafbarras(self.resdest,prob[posdmin],"cyan",str(int(minimo)),160) # Rango minimo y su PD
self.grafbarras(self.resdest,probmin,"dark orange",str(int(amax[posdpmin])),240) # PD minima y su rango

lista.append(unicode("Altitud máxima = ", "latin-1")+str(int(maximo))+unicode(" ft Probabilidad de Detección = ", "latin-1")+str(prob[posdmax])+" %\n")

lista.append(unicode("Altitud mínima = ", "latin-1")+str(int(minimo))+unicode(" ft Probabilidad de Detección = ", "latin-1")+str(prob[posdmin])+" %\n")

lista.append(unicode("Probabilidad de Detección mínima = ", "latin-1")+str(probmin)+unicode(" a una Altitud máxima de ", "latin-1")+str(int(amax[posdpmin]))+" ft\n")

lista.reverse()

self.At2.act_des(0)

self.At2.borrar_texto()

proceso=self.calcdt2(listadt[1])

self.At2.insertar_texto("Tiempo calculado del proceso "+proceso[1])

for i in lista:

    self.At2.insertar_texto(i)

self.At2.act_des(1)

elif self.var11.get()=="Falsos Blancos":

self.botonda['state'] = DISABLED

self.inform.set("RESULTADOS DEL ANALISIS")

try:

    archivo=open(os.path.join(self.capdir,self.var15.get()+"01dc.txt"))

```

```

except IOError:

    self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

else:

objeto=estc01(os.path.join(self.capdir,self.var15.get()+"01dc.txt"),self.var8.get(),os.path.join(self.capdir,self.var15.get()+"
02dc.txt"),2,self.var13.get(),os.path.join(self.capdir,"Buffer.txt"),os.path.join(self.capdir,"Bufferc.txt"))

    uno=objeto.anfb()

    total=uno[0]

    falsos=uno[1]

if self.gdconf.get()==unicode("Probabilidad de Detección","latin-1"):

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

    self.grafbarras(self.resdest,round((float(falsos)/float(total))*100,2),"dark orange"," ",80)

    self.At2.act_des(0)

    self.At2.borrar_texto()

    proceso=self.calcdt2(listadt[1])

    self.At2.insertar_texto("Tiempo calculado del proceso "+proceso[1])

    self.At2.insertar_texto(unicode("Probabilidad de Detección de falsos blancos = ","latin-
1")+str(round((float(falsos)/float(total))*100,2))+ "% "+ "\n")

    self.At2.insertar_texto(unicode("Numero de respuestas de falsos blancos = ","latin-1")+str(float(falsos))+ "\n")

    self.At2.insertar_texto(unicode("Numero total de respuestas = ","latin-1")+str(float(total))+ "\n")

    self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Rango","latin-1"):

    self.At2.act_des(0)

    self.At2.borrar_texto()

    self.At2.insertar_texto(unicode("Gráficos validos solo para Probabilidad de Detección Total","latin-1"))

    self.At2.act_des(1)

elif self.gdconf.get()==unicode("Probabilidad de Detección vs Altitud","latin-1"):

    self.At2.act_des(0)

    self.At2.borrar_texto()

    self.At2.insertar_texto(unicode("Gráficos validos solo para Probabilidad de Detección Total","latin-1"))

    self.At2.act_des(1)

```

```

def grafbarras(self,area,valor,color,texto,pos):

    x=299

    barra=int(valor*2.8)

    for i in range(0,barra-8):

        self.temporal.append(area.create_line(pos-1,x,pos,x,fill="black"))

        self.temporal.append(area.create_line(pos,x,pos+20,x,fill=color))

        self.temporal.append(area.create_line(pos+20,x,pos+21,x,fill="black"))

    x=x-1

    self.temporal.append(area.create_line(pos-1,x,pos+21,x,fill="black"))

    self.temporal.append(area.create_text(pos+8,315,text=texto,font="New_Times_Roman"))

def limatpd(self):

    self.At2.act_des(0)

    self.At2.borrar_texto()

    self.At2.act_des(1)

    self.inform.set("RESULTADOS DEL ANALISIS")

    for i in self.temporal:

        self.resdest.delete(i)

    self.temporal=[]

def codigosdif(self):

    try:

        archivo=open(os.path.join(self.capdir,self.var15.get()+"01dc.txt"))

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

        self.At2.act_des(0)

        self.At2.borrar_texto()

objeto=estc01(os.path.join(self.capdir,self.var15.get()+"01dc.txt"),self.var8.get(),os.path.join(self.capdir,self.var15.get()+"02dc.txt"),1,self.var13.get(),os.path.join(self.capdir,"Buffer.txt"),os.path.join(self.capdir,"Bufferc.txt"))

uno=objeto.codtd()

lista=uno[1]

```

```

lista.reverse()

self.At2.act_des(0)

for i in lista:
    self.At2.insertar_texto(i+"\t")

self.At2.insertar_texto(unicode("Número de códigos "+uno[0],"latin-1"))

self.At2.act_des(1)

def vradar(self):
    self.actmenu=1
    if self.vpidr==11:
        self.zoomb2['state'] = DISABLED
        self.zoomb1['state'] = NORMAL
    elif self.vpidr==0:
        self.vpidr=0
        self.zoomb1['state'] = DISABLED
        self.zoomb2['state'] = NORMAL
    else:
        self.zoomb1['state'] = NORMAL
        self.zoomb2['state'] = NORMAL
    self.cmapa()
    self.tdmap['state'] = DISABLED
    if self.numdbc.get()=="Uno":
        res=self.vcodtrans(self.codtpv.get(),self.adusb.get())
        if res==1:
            self.veraPPI()
        elif res==0:
            self.ventana5(unicode("ERROR!!! no se encuentra en el archivo el código transponder especificado","latin-1"),"dark red")
        elif self.numdbc.get()=="Todos":
            self.veraPPI()

def vcodtrans(self,codigo,archivo):

```

```

esta=0

objeto=estc01(os.path.join(self.cadir,archivo+"01dc.txt"),codigo,os.path.join(self.cadir,archivo+"02dc.txt"),1,self.var13
.get(),os.path.join(self.cadir,"Buffer.txt"),os.path.join(self.cadir,"Bufferc.txt"))

uno=objeto.codtd()

lista=uno[1]

for i in lista:

    if i==codigo:

        esta=1

        break

return esta

def codigosdif2(self):

    try:

        archivo=open(os.path.join(self.cadir,self.adusb.get()+"01dc.txt"))

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

objeto=estc01(os.path.join(self.cadir,self.adusb.get()+"01dc.txt"),self.var8.get(),os.path.join(self.cadir,self.var15.get()+
"02dc.txt"),1,self.var13.get(),os.path.join(self.cadir,"Buffer.txt"),os.path.join(self.cadir,"Bufferc.txt"))

        uno=objeto.codtd()

        lista=uno[1]

        lista.reverse()

        self.At3.act_des(0)

        self.At3.borrar_texto()

        for i in lista:

            self.At3.insertar_texto(i+"\t")

        self.At3.insertar_texto(unicode("Número de códigos ","latin-1")+uno[0])

        self.At3.act_des(1)

def anact(self):

    archivob=open(os.path.join(self.cadir,"Buffer.txt"),"r+")

    lineas=archivob.readlines()

```



```

archivob.close()

temporal=[]

x=0

for i in lineas:

    lda=i.split("\t")

    for j in lda:

        temporal.append(j)

    if x!=len(lda)-1:

        temporal.append("\t\t")

    x=x+1

x=0

temporal.reverse()

self.At2.act_des(0)

self.At2.borrar_texto()

for i in temporal:

    self.At2.insertar_texto(i)

    self.At2.insertar_texto("#
TRAMA\t\tCODIGO\t\tALTITUD(ft)\t\tRANGO(NM)\t\tACIMUT(GRADOS)\t\tTIEMPO\t\tCONF.
ALTITUD\t\tCONF. CODIGO\t\tEMERGENCIA\t\tTRAMA\n")

self.At2.act_des(1)

self.botonda['state'] = DISABLED

def ejecutar(self):

    inicio=str(datetime.today())

    try:

        int(self.nomda.get())

        int(self.thoras.get())

        int(self.tminutos.get())

    except ValueError:

        self.ventana5(unicode("ERROR!!! El valor no es un número entero, ingrese un número entero","latin-1"),"dark red")

    else:

        archivo1=open(os.path.join(self.captura.get()+self.nomda.get()+".txt"),"w")

        h=int(self.thoras.get())

```

DE

```

m=int(self.tminutos.get())
x=m*60 # minutos a segundos
y=h*3600 # horas a segundos
a=x+y

os.system("/usr/bin/touch "+self.captura.get()+self.nomda.get()+".pcap")

os.system("/usr/bin/tshark -i eth0 -aduration:"+str(a)+" -w "+self.captura.get()+self.nomda.get()+".pcap")

print("Paquetes capturados")

print("Fecha y hora de inicio de la captura de datos\n"+inicio+"\nFecha y hora de fin de la captura de
datos\n"+str(datetime.today()))

def error_archivo(self):

    if self.cv2==1:

        self.cajadd2.destroy()

        self.cv2=0

    try:

        if self.var16.get()=="PCAP":

            archivo=open(self.var3.get()+".pcap")

        elif self.var16.get()=="TEXT":

            archivo=open(os.path.join(self.capdir,self.var3.get()+".txt"))

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

        self.proc1()

def veraPPI(self):

    try:

        archivo=open(os.path.join(self.capdir,self.adusb.get()+"01dc.txt"),"r+")

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

        self.inppi['state'] = NORMAL

    else:

        self.inppi['state'] = DISABLED

```

```

self.finppi['state'] = NORMAL

self.cpdb1=[]
self.cpdb2=[]
self.tiempodc=[]
self.identda=[]
self.altda=[]
self.fallas=[]
self.confda=[]
self.confdc=[]
self.borrar_cursor(self.cursor)
self.borrar_cursor(self.cursor2)
self.borrar_cursor(self.cursor3)
cursor=[]
cursor2=[]
cursor3=[]
self.act_blanco=1
self.x=0
self.i=0
if self.numdbc.get()=="Uno":
    digitos=self.valdct(self.codtpv.get())
    if digitos==4:
        self.pp_PPI()
elif self.numdbc.get()=="Todos":
    self.pp_PPI()

def veramn(self):
    try:
        archivom=open(os.path.join(self.capdir,self.adusb.get()+"02dc.txt"),"r+")
    except IOError:
        self.ventana5(unicode("ERROR!!! la decodificación del primer Mensaje de Norte no se ha realizado","latin-1"))
    else:
        self.plotear1()

```

```

def pp_PPI(self):
    self.cerrarv4()

    archivo=open(os.path.join(self.capedir,self.adusb.get()+"01dc.txt"),"r+")

    lineas=archivo.readlines()

    archivo.close()

    self.angulos=[] # actualizar angulos

    if self.numdbc.get()=="Uno":

        listab=[]

        for i in lineas:

            lista=i.split("\t")

            if self.codtpv.get()==lista[1]:

                listab.append(i)

        x=0

        for i in range(len(listab)): # para capturar las coordenadas polares de cada uno de los blancos

            blancos=listab[i].split("\t")

            self.identda.append(blancos[1])

            self.altda.append(blancos[2])

            self.cpdb1.append(blancos[3])

            self.cpdb2.append(blancos[4])

            self.tiempodc.append(blancos[5])

            self.confda.append(blancos[6])

            self.confdc.append(blancos[7])

            self.fallas.append(blancos[8])

        nuevo=listab[0]

        h=nuevo.split("\t")

        hmn=h[5]

        x=0

        hm2=int(hmn[0:2])

        mm2=int(hmn[3:5])

        sm2=int(hmn[6:8])

```

```

self.horas=hm2

self.minutos=mm2

self.segundos=sm2

self.etiqueta.config(fg="white")

self.barrido=self.tdbant()

angulo=int(360.0/float(self.barrido))

self.angulos.append(0)

for i in range(1,self.barrido+1):

    if i==1:

        pv=90-angulo

        self.angulos.append(pv)

    else:

        pv=pv-angulo

        self.angulos.append(pv)

self.activar_blancos=1

self.act_blancos=1

self.reloj()

elif self.numdbc.get()=="Todos":

    x=0

    for i in lineas: # para capturar las coordenadas polares de cada uno de los blancos

        lista=i.split("\t")

        self.cpdb1.append(lista[3])

        self.cpdb2.append(lista[4])

        self.tiempodc.append(lista[5])

        self.identda.append(lista[1])

        self.altda.append(lista[2])

        self.confda.append(lista[6])

        self.confdc.append(lista[7])

        self.fallas.append(lista[8])

self.act_blancos=1

self.veramn()

```

```

def cerrar_v2(self):
    self.cajadd2.destroy()
    self.actmenu=0 # activar los otros analisis

def cerrarv2(self,event):
    self.actmenu=0

def borrar_cursor(self,lista):
    for i in range(len(lista)):
        self.indicaciones.delete(lista[i])

def borrar_cursor2(self,lista):
    for i in range(len(lista)):
        self.grafp.delete(lista[i])

def efecto(self,angulo):
    self.ro=250.00
    self.theta=float(angulo)
    xx=self.ro*cos(self.theta*pi/180)+252
    yy=252-self.ro*sin(self.theta*pi/180)
    if int(xx)==251:
        self.cursor3.append(self.indicaciones.create_line(252,252,int(xx)+1,int(yy),fill="blue"))
    else:
        self.cursor3.append(self.indicaciones.create_line(252,252,int(xx),int(yy),fill="blue"))

def detener(self):
    self.act_blanco=0
    self.finppi['state'] = DISABLED
    self.inppi['state'] = NORMAL
    self.menuppi['state'] = NORMAL
    self.tdmap['state'] = NORMAL

```

```

self.actmenu=0 # activar los otros analisis

self.etiqueta.config(fg="red")

def creditos(self):

    self.entrada['state'] = DISABLED

    self.codsac['state'] = DISABLED

    self.catast['state'] = DISABLED

    self.bdp['state'] = DISABLED

    self.tipda['state'] = DISABLED

    self.menuppi['state'] = DISABLED

    self.inppi['state'] = DISABLED

    self.finppi['state'] = DISABLED

    self.cajaddc = Toplevel()

    self.cajaddc.config(bg="dark blue")

    self.cajaddc.title("ACERCA DE ADPADEC V1.0 UES EIE")

    self.cajaddc.maxsize(width=402,height=150)

    self.cajaddc.minsize(width=402,height=150)

    color="dark blue"

    self.pres=Canvas(self.cajaddc,width=400,height=150,bg="white")

    self.pres.pack(padx=5,pady=5,side=LEFT)

    self.pres.create_text(120,10,text="UNIVERSIDAD DE EL SALVADOR",font=("Arial",10),fill="blue")

    self.pres.create_text(163,25,text="FACULTAD DE INGENIERIA Y
ARQUITECTURA",font=("Arial",10),fill="blue")

    self.pres.create_text(137,40,text="ESCUELA DE INGENIERIA ELECTRICA",font=("Arial",10),fill="blue")

    self.pres.create_text(196,55,text="TRABAJO DE GRADUACION DE INGENIERIA
ELECTRICA",font=("Arial",10),fill="blue")

    self.pres.create_text(142,70,text="PRESENTADO Y DESARROLLADO POR:",font=("Arial",10),fill="blue")

    self.pres.create_text(134,85,text="EDGAR MANUEL MORAZAN BONILLA",font=("Arial",10),fill="blue")

    self.pres.create_text(195,100,text="PARA OPTAR AL TITULO DE INGENIERO
ELECTRICISTA",font=("Arial",10),fill="blue")

    self.pres.create_text(173,115,text="DOCENTE DIRECTOR: ING. WERNER
MELENDEZ",font=("Arial",10),fill="blue")

```

```
self.pres.create_text(175,130,text="TIEMPO  
22/04/2011",font=("Arial",10),fill="blue")
```

DE

DESARROLLO:

22/02/2010

A

```
self.cajaddc.bind("<Destroy>",self.habvent6)
```

```
def plotear1(self):
```

```
self.angulos=[]
```

```
archivom=open(os.path.join(self.capdir,self.adusb.get()+"02dc.txt"),"r+")
```

```
lineasn=archivom.readlines()
```

```
archivom.close()
```

```
for i in lineasn:
```

```
    h=i.split("\t")
```

```
    hmn=h[1]
```

```
x=0
```

```
hm2=int(hmn[0:2])
```

```
mm2=int(hmn[3:5])
```

```
sm2=int(hmn[6:8])
```

```
self.horas=hm2
```

```
self.minutos=mm2
```

```
self.segundos=sm2
```

```
self.etiqueta.config(fg="white")
```

```
self.barrido=self.tdbant()
```

```
angulo=int(360.0/float(self.barrido))
```

```
self.angulos.append(0)
```

```
for i in range(1,self.barrido+1):
```

```
    if i==1:
```

```
        pv=90-angulo
```

```
        self.angulos.append(pv)
```

```
    else:
```

```
        pv=pv-angulo
```

```
        self.angulos.append(pv)
```



```

self.activar_blanco=1

self.reloj()

def decopcap(self,var1):

    print("transformando el archivo "+var1+".pcap a "+var1+".txt")

    os.system("/usr/bin/mergcap -F k12text -w decodificaciones/"+var1+".txt "+var1+".pcap")

    print(unicode("Fin del proceso de transformación de formato","latin-1"))

def buscart(self):

    print("*****")

    print("Buscando tramas ASTERIX en el archivo "+self.var3.get()+".txt de categoria "+self.var2.get())

    if self.var2.get()=="01":

        self.sac_cat()

objeto=adpdec(1,self.sac,os.path.join(self.cadir,self.var3.get()+".txt"),os.path.join(self.cadir,self.var3.get()+"t01.txt"),os
.path.join(self.cadir,self.var3.get()+"p01.txt"),os.path.join(self.cadir,"Paquetes.txt"))

    uno=objeto.decdb()

    elif self.var2.get()=="02":

        self.sac_cat()

objeto=adpdec(2,self.sac,os.path.join(self.cadir,self.var3.get()+".txt"),os.path.join(self.cadir,self.var3.get()+"t02.txt"),os
.path.join(self.cadir,self.var3.get()+"p02.txt"),os.path.join(self.cadir,"Paquetes.txt"))

    uno=objeto.decdb()

    print(str(uno-1)+" tramas ASTERIX encontradas en el archivo")

    print("Fin de la búsqueda de tramas ASTERIX en el archivo "+self.var3.get()+".txt")

    print("*****")

def proc1(self):

    listadt=self.calcdt1()

    if self.var12.get()=="PPD":

        print(unicode("Fecha y hora de inicio del procesamiento y decodificación de datos para
análisis\n"+str(listadt[0]),"latin-1"))

        if self.var16.get()=="PCAP":

```

```

        self.decopcap(self.var3.get())

self.sac_cat()

print("*****")

print("Buscando trama ASTERIX del primer mensaje de Norte\nEn el archivo "+self.var3.get()+".txt de categoria
02")

objeto=adpdec(0,self.sac,os.path.join(self.caddir,self.var3.get()+".txt"),os.path.join(self.caddir,self.var3.get()+"t02.txt"),os
.path.join(self.caddir,self.var3.get()+"p02.txt"),os.path.join(self.caddir,"Paquetes.txt"))

uno=objeto.decdb()

print(str(uno-1)+" mensaje de Norte encontrado en el archivo")

print("Fin de la busqueda del primer mensaje de Norte\nEn el archivo "+self.var3.get()+".txt")

print("Decodificando el mensaje de Norte")

objeto=c02pd(os.path.join(self.caddir,self.var3.get()+"t02.txt"),os.path.join(self.caddir,self.var3.get()+"02dc.txt"))

uno=objeto.adpcons()

print(unicode("Fin de la decodificación del mensaje de Norte", "latin-1"))

print("*****")

print("Buscando tramas ASTERIX en el archivo "+self.var3.get()+".txt de categoria 01")

self.sac_cat()

objeto=adpdec(1,self.sac,os.path.join(self.caddir,self.var3.get()+".txt"),os.path.join(self.caddir,self.var3.get()+"t01.txt"),os
.path.join(self.caddir,self.var3.get()+"p01.txt"),os.path.join(self.caddir,"Paquetes.txt"))

uno=objeto.decdb()

print(str(uno-1)+" tramas ASTERIX encontradas en el archivo")

print("Fin de la busqueda de tramas ASTERIX en el archivo "+self.var3.get()+".txt")

print("Decodificando tramas ASTERIX de categoria 01")

objeto=c01pd(os.path.join(self.caddir,self.var3.get()+"t01.txt"),os.path.join(self.caddir,self.var3.get()+"01dc.txt"),os.path.j
oin(self.caddir,self.var3.get()+"p01.txt"),os.path.join(self.caddir,"anconf.txt"))

uno=objeto.adpcons()

print(unicode("Fin de la decodificación de tramas ASTERIX de categoria 01", "latin-1"))

self.actmenu=0

print(unicode("Fin del procesamiento y decodificación para análisis", "latin-1"))

proceso=self.calcdt2(listadt[1])

print(unicode("Fecha y hora de fin del procesamiento y decodificación de datos para análisis\n", "latin-
1")+str(proceso[0]))

print("tiempo calculado de todo el proceso "+proceso[1])

```

```

print("*****")

if self.var12.get()=="PPA":
    print(unicode("Fecha y hora de inicio de la extracción y decodificación de datos\n", "latin-1")+str(datetime.today()))
    if self.var16.get()=="PCAP":
        self.decopcap(self.var3.get())
    self.buscart()
    if self.var2.get()=="01":
        print("Decodificando tramas ASTERIX de categoria "+self.var2.get())

objeto=c01(os.path.join(self.capdir,self.var3.get()+"t01.txt"),os.path.join(self.capdir,self.var3.get()+"cat01.txt"),os.path.joi
n(self.capdir,self.var3.get()+"p01.txt"),os.path.join(self.capdir,"anconf.txt"))

    uno=objeto.adpcons()
    print(unicode("Fin de la decodificación de tramas ASTERIX de categoria ", "latin-1")+self.var2.get())
    proceso=self.calcdt2(listadt[1])
    print(unicode("Fecha y hora de fin de la extracción y decodificación de datos\n", "latin-1")+str(proceso[0]))
    print("tiempo calculado de todo el proceso "+proceso[1])
    print("*****")
    self.ventana2(self.var2.get())
elif self.var2.get()=="02":
    print("Decodificando tramas ASTERIX de categoria "+self.var2.get())
    objeto=c02(os.path.join(self.capdir,self.var3.get()+"t02.txt"),os.path.join(self.capdir,self.var3.get()+"cat02.txt"))
    uno=objeto.adpcons()
    print(unicode("Fin de la decodificación de tramas ASTERIX de categoria ", "latin-1")+self.var2.get())
    proceso=self.calcdt2(listadt[1])
    print(unicode("Fecha y hora de fin de la extracción y decodificación de datos\n", "latin-1")+str(proceso[0]))
    print("tiempo calculado de todo el proceso "+proceso[1])
    print("*****")
    self.ventana2(self.var2.get())

def calcdt1(self):
    time1=datetime.today()
    tiempo1=str(time1).split(" ")

```

```

ddt1=tiempo1[1]
h1=int(ddt1[0:2])*3600
m1=int(ddt1[3:5])*60
s1=int(ddt1[6:8])
segundos1=h1+m1+s1
return [time1,segundos1]

```

```

def calcdt2(self,segundos1): # funcion que se utiliza para calcular el tiempo de los procesos

```

```

time2=datetime.today()
tiempo2=str(time2).split(" ")
ddt2=tiempo2[1]
h2=int(ddt2[0:2])*3600
m2=int(ddt2[3:5])*60
s2=int(ddt2[6:8])
segundos2=h2+m2+s2
segfinal=segundos2-segundos1
r1=float(segfinal)/3600.0
horas=int(r1)
r2=(r1-float(horas))*60
minutos=int(r2)
r3=(r2-float(minutos))*60
segundos=int(r3)
if horas<10:
    h="0"+str(horas)
else:
    h=str(horas)
if minutos<10:
    m="0"+str(minutos)
else:
    m=str(minutos)
if segundos<10:
    s="0"+str(segundos)

```

```
else:  
    s=str(segundos)  
    return [time2,h+": "+m+": "+s]
```

```
def sac_cat(self):  
    a=self.var1.get()  
    if a=="El Salvador":  
        self.sac="f1"  
        self.ta="ESA"  
    elif a=="Guatemala":  
        self.sac="f6"  
        self.ta="GUA"  
    elif a=="Honduras":  
        self.sac="f9"  
        self.ta="HON"  
    elif a=="Belize":  
        self.sac="e6"  
        self.ta="BEL"  
    elif a=="Nicaragua":  
        self.sac="fe"  
        self.ta="NIC"  
    elif a=="Costa Rica":  
        self.sac="ec"  
        self.ta="COS"
```

```
def habvent(self,event):  
    self.cv1=0  
    self.actmenu=0
```

```
def habvent3(self,event):  
    self.cv3=0  
    self.actmenu=0
```

```

def mensaje(self,event):
    self.At1.act_des(0)
    self.At1.borrar_texto()
    a=self.Idtast.curselection()
    self.At1.act_des(1)
    try:
        ab=str(int(a[0])+1)
    except IndexError:
        self.At1.act_des(1)
    else:
        self.At1.act_des(0)
        if self.var2.get()=="01":
            archivov=open(os.path.join(self.capedir,self.var3.get()+"cat01.txt"),"r+")
            lineas=archivov.readlines()
            archivov.close()
            x=0
            temporal=[] # lista para cada blanco
            for i in lineas:
                if
                i=="++++++\n":
                    numero=lineas[x+1]
                    if ab+"\n"==numero:
                        for j in range(x+1,len(lineas)):
                            if
                            lineas[j]=="*****\n":
                                temporal.append(lineas[j])
                            if
                            lineas[j]=="*****\n":
                                break
                x=x+1
            temporal.reverse()

```

```

for i in temporal:
    self.At1.insertar_texto(i)
self.At1.act_des(1)

elif self.var2.get()=="02":
    archivov=open(os.path.join(self.capedir,self.var3.get()+"cat02.txt"),"r+")
    lineas=archivov.readlines()
    archivov.close()
    x=0
    temporal=[] # lista para cada blanco
    for i in lineas:
        if
i=="++++++\n":
        numero=lineas[x+1]
        if ab+"\n"==numero:
            for j in range(x+1,len(lineas)):
                if
lineas[j]=="*****\n":
                break
            else:
                temporal.append(lineas[j])
            x=x+1
    temporal.reverse()
    for i in temporal:
        self.At1.insertar_texto(i)
    self.At1.act_des(1)

def habproc1(self):
    if self.cv2==1:
        self.cajadd2.destroy()
        self.cv2=0
    self.entrada['state'] = NORMAL

```

```

self.codsac['state'] = NORMAL

self.catast['state'] = NORMAL

self.bdp['state'] = NORMAL

self.tipda['state'] = NORMAL

self.var12.set("PPA") # Procesamiento para Analisis

def habproc2(self):

    self.entrada['state'] = NORMAL

    self.codsac['state'] = NORMAL

    self.catast['state'] = DISABLED

    self.bdp['state'] = NORMAL

    self.tipda['state'] = NORMAL

    self.var12.set("PPD") # Procesamiento para Probabilidad de Detección

def continuar(self):

    self.act_blanco=1

    self.actmenu=1

    self.cmapa()

    self.finppi['state'] = NORMAL

    self.inppi['state'] = DISABLED

    self.menuppi['state'] = DISABLED

    self.tdmapi['state'] = DISABLED

    self.etiqueta.config(fg="white")

    self.reloj()

def errorarch(self):

    try:

        archivo=open(os.path.join(self.cadir,self.var17.get()+"01dc.txt"),"r+")

    except IOError:

        self.ventana5("ERROR!!! nombre de archivo invalido, verifique si existe el archivo","dark red")

    else:

```



```

self.estadisticas()

def estadisticas(self):
    listadt=self.calcdt1()

objeto=estc01(os.path.join(self.capedir,self.var17.get()+"01dc.txt"),self.var8.get(),os.path.join(self.capedir,self.var17.get()+"02dc.txt"),1,self.var13.get(),os.path.join(self.capedir,"Buffer.txt"),os.path.join(self.capedir,"Bufferc.txt"))

uno=objeto.estadb()

total=uno[15]+uno[16]+uno[17]+uno[18]

self.cursor4.append(self.grafp.create_text(50,10,text="(%)",font="New_Times_Roman"))

x=28

lista=["100","90","80","70","60","50","40","30","20","10"," "]

for i in range(0,10):
    self.cursor4.append(self.grafp.create_line(38,x,44,x,fill="black"))
    self.cursor4.append(self.grafp.create_text(20,x,text=lista[i],font="New_Times_Roman"))

    x=x+28

caso1=uno[15]*100.00/total
caso2=uno[16]*100.00/total
caso3=uno[17]*100.00/total
caso4=uno[18]*100.00/total

self.At4.act_des(0)

proceso=self.calcdt2(listadt[1])

self.At4.insertar_texto("Tiempo calculado de todo el proceso = "+proceso[1]+\n")

self.At4.insertar_texto(unicode("Número de respuestas con Baja confianza de Altitud y Baja Confianza de Código = ", "latin-1")+str(round(caso4,2))+ " % (Caso 4)\n")

self.At4.insertar_texto(unicode("Número de respuestas con Baja Confianza de Altitud y Alta Confianza de Código = ", "latin-1")+str(round(caso3,2))+ " % (Caso 3)\n")

self.At4.insertar_texto(unicode("Número de respuestas con Alta Confianza de Altitud y Baja Confianza de Código = ", "latin-1")+str(round(caso2,2))+ " % (Caso 2)\n")

self.At4.insertar_texto(unicode("Número de respuestas con Alta Confianza de Altitud y Alta Confianza de Código = ", "latin-1")+str(round(caso1,2))+ " % (Caso 1)\n")

self.At4.insertar_texto(unicode("Número de respuestas con codigos de emergencia = ", "latin-1")+str(uno[12])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con altitud mayor a 50000 pies = ", "latin-1")+str(uno[4])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con altitud de 0 pies = ", "latin-1")+str(uno[5])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas sin Codigo Transponder = ", "latin-1")+str(uno[11])+"\n")

```

```

self.At4.insertar_texto(unicode("Número de respuestas con altitud menor a 0 pies = ", "latin-1")+str(uno[6])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con alta confianza del Modo-C = ", "latin-1")+str(uno[10])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con baja confianza del Modo-C = ", "latin-1")+str(uno[9])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con alta confianza del Modo-3/A = ", "latin-1")+str(uno[8])+"\n")

self.At4.insertar_texto(unicode("Número de respuestas con baja confianza del Modo-3/A = ", "latin-1")+str(uno[7])+"\n")

self.At4.insertar_texto(unicode("Rango mínimo = ", "latin-1")+str(uno[3])+"\n")

self.At4.insertar_texto(unicode("Rango máximo = ", "latin-1")+str(uno[2])+"\n")

self.At4.insertar_texto(unicode("Altitud mínima = ", "latin-1")+str(uno[1])+"\n")

self.At4.insertar_texto(unicode("Altitud máxima = ", "latin-1")+str(uno[0])+"\n")

self.At4.insertar_texto(unicode("Número de pistas = ", "latin-1")+str(uno[14])+"\n")

self.At4.insertar_texto(unicode("Número de plots = ", "latin-1")+str(uno[13])+"\n")

self.At4.act_des(1)

self.grafbarras(self.grafp,caso1,"green","Caso 1",80)

self.grafbarras(self.grafp,caso2,"red","Caso 2",160)

self.grafbarras(self.grafp,caso3,"blue","Caso 3",240)

self.grafbarras(self.grafp,caso4,"orange","Caso 4",320)

def bAt4(self):

self.At4.act_des(0)

self.At4.borrar_texto()

self.At4.act_des(1)

self.borrar_cursor2(self.cursor4)

self.borrar_cursor2(self.temporal)

def tdbant(self): # calcula el tiempo de barrido de antena

archivo=open(os.path.join(self.capdir,self.adusb.get()+"01dc.txt"), "r+")

lineas=archivo.readlines()

archivo.close()

identda=[]

```

```

tiempodc=[]

for i in lineas:
    lista=i.split("\t")
    identda.append(lista[1])
    tiempodc.append(lista[5])

restas=[]
x=0
y=0
tdc=[]

for k in identda:
    for i in identda:
        if i==k:
            tdc.append(tiempodc[x])

            x=x+1

if len(tdc)>1:
    for j in tdc:
        if y==0:
            h1=int(j[0:2])
            m1=int(j[3:5])
            s1=int(j[6:8])

            rr1=(h1*3600)+(m1*60)+s1

        else:
            h2=int(j[0:2])
            m2=int(j[3:5])
            s2=int(j[6:8])

            rr2=(h2*3600)+(m2*60)+s2

            if rr2>rr1:
                restas.append(rr2-rr1)

        y=y+1

    if y==8:
        break

```

```

        break
    return min(restas)

def reloj(self):
    if self.horas>23:
        self.horas=0
    if self.minutos==60:
        self.minutos=0
        self.horas=self.horas+1
    if self.segundos>59:
        self.segundos=0
        self.minutos=self.minutos+1

    if self.horas<10:
        h="0"+str(self.horas)
    else:
        h=str(self.horas)

    if self.minutos<10:
        m="0"+str(self.minutos)
    else:
        m=str(self.minutos)

    if self.segundos<10:
        s="0"+str(self.segundos)
    else:
        s=str(self.segundos)

    if self.tidvisual.get()=="UTC":
        if int(m)==60:
            m="00"
            h=str(self.horas+1)

```

```

self.var4.set(h+": "+m+": "+s)
elif self.tidvisual.get()=="CA":
    hor1=int(h)-6
    if hor1<10:
        hor2="0"+str(hor1)
    else:
        hor2=str(hor1)
    if int(m)==60:
        m="00"
        if hor1<10:
            hor2="0"+str(hor1+1)
        else:
            hor2=str(hor1+1)
    self.var4.set(hor2+": "+m+": "+s)
for i in range(int(len(self.tiempodc))):
    if self.tiempodc[i]==h+": "+m+": "+s:
        if self.x!=int(len(self.tiempodc)):
            objeto=mostrarb.mostrar_blanco(self.cdmapa,self.cursor2,self.indicaciones,[self.cpdb1[self.x],self.cpdb2[self.x]],self.act
            _blanco,self.eppi[self.vpidr],self.identda[self.x],self.altda[self.x],self.fallas[self.x],self.mapa.get(),self.confda[self.x],self.
            confdc[self.x],self.pistas.get()) #
            objeto.mbep()
            self.x=self.x+1
        self.segundos=self.segundos+1
        self.i=self.i+1
    for i in range(1,self.barrido+1):
        if self.i==i:
            self.borrar_cursor(self.cursor3)
            if self.i==self.barrido:
                self.efecto(self.angulos[self.i])
                self.borrar_cursor(self.cursor2)
                self.i=0
            else:

```

```

        self.efecto(self.angulos[self.i])

if self.act_blanco==1:
    self.ventana.after(1000,self.reloj)

def visualizar(self):
    self.vpidr=10

    self.escdpi.set(str(self.eppi[self.vpidr]))

    self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr2[self.vpidr]))

    self.indicaciones.create_image(252,252,image=self.img)

    self.arco("black")

def cmapa(self):
    if self.mapa.get()=="DESCRPTIVO":
        self.cdmapa=0

        self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr2[self.vpidr]))

        self.indicaciones.create_image(252,252,image=self.img)

        self.arco("black")
    elif self.mapa.get()=="SATELITE":
        self.cdmapa=1

        self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr[self.vpidr]))

        self.indicaciones.create_image(252,252,image=self.img)

        self.arco("white")

def zoommas(self):
    self.vpidr=self.vpidr-1

    self.zoomb2['state'] = NORMAL

    self.zoomb1['state'] = NORMAL

    if self.mapa.get()=="DESCRPTIVO":
        self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr2[self.vpidr]))

        self.indicaciones.create_image(252,252,image=self.img)

        self.arco("black")

```

```

elif self.mapa.get()=="SATELITE":
    self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr[self.vpidr]))
    self.indicaciones.create_image(252,252,image=self.img)
    self.arcos("white")
self.escdpi.set(str(self.eppi[self.vpidr]))
if self.vpidr==0:
    self.vpidr=0
    self.zoomb1['state'] = DISABLED
    self.zoomb2['state'] = NORMAL

def zoommen(self):
    self.vpidr=self.vpidr+1
    self.zoomb2['state'] = NORMAL
    self.zoomb1['state'] = NORMAL
if self.mapa.get()=="DESCRIPTIVO":
    self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr2[self.vpidr]))
    self.indicaciones.create_image(252,252,image=self.img)
    self.arcos("black")
elif self.mapa.get()=="SATELITE":
    self.img=PhotoImage(file=os.path.join(self.ppidir,self.idr[self.vpidr]))
    self.indicaciones.create_image(252,252,image=self.img)
    self.arcos("white")
self.escdpi.set(str(self.eppi[self.vpidr]))
if self.vpidr==11:
    self.zoomb2['state'] = DISABLED
    self.zoomb1['state'] = NORMAL

def arcos(self,color):
    self.indicaciones.create_line(252,252,252,502,fill=color)
    self.indicaciones.create_line(252,252,252,2,fill=color)
    self.indicaciones.create_line(252,252,2,252,fill=color)
    self.indicaciones.create_line(252,252,503,252,fill=color)

```

```

self.indicaciones.create_arc(102,102,402,402,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(2,2,502,502,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(27,27,477,477,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(52,52,452,452,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(77,77,427,427,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(127,127,377,377,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(152,152,352,352,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(177,177,327,327,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(202,202,302,302,outline=color,start=0,extent=359.9,style="arc")
self.indicaciones.create_arc(227,227,277,277,outline=color,start=0,extent=359.9,style="arc")

```

```

def efecto2(self, inicio, fin, color):

```

```

    self.cursor.append(self.indicaciones.create_arc(102,102,402,402,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(2,2,502,502,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(27,27,477,477,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(52,52,452,452,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(77,77,427,427,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(127,127,377,377,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(152,152,352,352,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(177,177,327,327,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(202,202,302,302,outline=color,start=inicio,extent=fin,style="arc"))
    self.cursor.append(self.indicaciones.create_arc(227,227,277,277,outline=color,start=inicio,extent=fin,style="arc"))

```

adpdec.py

```

#!/usr/bin/python
# -*- coding: cp1252 -*-
from math import *
import sys
import os
from string import *
from lon_t import *

```



```

class adpdec:
    def __init__(self,modo,sac,archivo1,archivo2,archivo3,temporal):
        self.archivo1=archivo1
        self.archivo2=archivo2
        self.archivo3=archivo3
        self.archivo4=temporal
        self.modo=modo
        self.sac=sac

    def decdb(self):
        archivo=open(self.archivo1,"r+")
        lineas=archivo.readlines()
        archivo.close()
        bits=[]
        lista2=["8","9","a","b","c","d","e","f"]
        impar=["1","3","5","7","9","b","d","f"]
        par=["0","2","4","6","8","a","c","e"]
        cod=[]
        caso=0
        ntrama=1
        dproc=0
        longitud=0
        l1=[-8,-10,-12]
        elem=[-2,-4,-6]
        elem2=[-8,-10,-12]
        elem3=[-7,-9,-11]
        elem4=[-7,-3,-5]
        elem5=[-1,-3,-3]
        elem6=[-1,-3,-5]
        pos1=[6,8,10]
        pos2=[5,7,9]

```

```

pos3=[4,6,8]
pos4=[3,5,7]
treal=" "
if self.modos==0:
    self.cat="2"
elif self.modos==1:
    self.cat="1"
elif self.modos==2:
    self.cat="2"

valor="0"+self.cat
archivo4=open(self.archivo4,"w+")
x=0
if self.modos==1 or self.modos==2:
    for i in lineas:
        if len(i)>40:
            a=i[6:len(i)-2]
            if a.count("|"+valor+"|")>=1 and a.count("|"+self.sac+"|")>=1:
                digitos=list(a)
                anterior=lineas[x-1]
                archivo4.write(anterior[0:8]+" ")
                for j in digitos:
                    if j!="|":
                        archivo4.write(str(j)) # guardar los bits hexadecimales
                archivo4.write("\n")
            x=x+1
    archivo4.close()
if self.modos==0:
    for i in lineas:
        if len(i)>40:
            a=i[6:len(i)-2]
            if a.count("|"+valor+"|")>=1 and a.count("|"+self.sac+"|")>=1 and a.count("|01|")>=1:

```

```

    digitos=list(a)

    anterior=lineas[x-1]

    archivo4.write(anterior[0:8]+" ")

    for j in digitos:

        if j!="|":

            archivo4.write(str(j)) # guardar los bits hexadecimales

        archivo4.write("\n")

    x=x+1

archivo4.close()

archivo5=open(self.archivo4,"r+")

lineas2=archivo5.readlines()

archivo5.close()

x=0

for i in lineas2:

    escribir=list(i)

    escribir.remove("\n")

    for j in escribir:

        bits.append(j)

archivo2=open(self.archivo2,"w")

if self.modos==1 or self.modos==2:

    archivo3=open(self.archivo3,"w")

    for i in bits:

        if bits[x]==" ":

            treal=bits[x-8]+bits[x-7]+bits[x-6]+bits[x-5]+bits[x-4]+bits[x-3]+bits[x-2]+bits[x-1] # encuentro el tiempo del
paquete

            elif bits[x]==self.sac[0:1]: # "f"

                if x==len(bits):

                    break

                elif x+1==len(bits):

                    break

```

```

elif dproc==1:
    break
else:
    caso=0
    longitud=0
    if bits[x+1]==self.sac[1:2]: # "1"
        for c in range(0,3):
            if bits[x+elem2[c]]=="0" and bits[x+elem3[c]]==self.cat:
                longitud=self.adplong([bits[x-pos1[c]],bits[x-pos2[c]],bits[x-pos3[c]],bits[x-pos4[c]])
                if int(longitud)>=6 and int(longitud)<=100:
                    caso=c+1
                    if caso==1:
                        cond=["0","2","4","6","8","a","c","e"]
                    elif caso==2:
                        cond=["1","3","5","7","9","b","d","f"]
                    elif caso==3:
                        cond=["1","3","5","7","9","b","d","f"]
                    break

        for k in range(0,3):
            if caso==k+1:
                y=l1[k]
                for j in lista2:
                    if j==bits[x+elem[k]]:
                        if self.cat=="1":
                            for l in cond:
                                if l==bits[x+elem5[k]]:
                                    for q in cond:
                                        if q==bits[x+elem6[k]]:
                                            for p in par:
                                                if p==bits[x-1]:
                                                    if self.modo==1:

```

```

        archivo3.write(str(ntrama)+"\t"+treal+"\n")
    for n in range(0,int(longitud)):
        archivo2.write(bits[x+y])
        y=y+1
        ntrama=ntrama+1
        archivo2.write("\n")
        if self.modos==0:
            dproc=1
            break
        break
    break
elif self.cat=="2":
    for l in cond:
        if l==bits[x+elem5[k]]:
            for q in cond:
                if q==bits[x+elem6[k]]:
                    for p in par:
                        if p==bits[x-1]:
                            for n in range(0,int(longitud)):
                                archivo2.write(bits[x+y])
                                y=y+1
                                ntrama=ntrama+1
                                archivo2.write("\n")
                                if self.modos==0:
                                    dproc=1
                                    break
                                break
                            break
                    break
            break
        break
    break

    x=x+1

return ntrama

def adplong(self,lista):
    a=lon_t(lista,0)
    lon=a.hexadec()

```

```
return lon
```

adpest.py

```
#!/usr/bin/python
```

```
# -*- coding: cp1252 -*-
```

```
from math import *
```

```
import sys
```

```
import os
```

```
from string import *
```

```
class estdconf:
```

```
def __init__(self,archivo,archivo2,archivo3,archivo4,mnorte,tiem,identidad,alt,rango,acimut):
```

```
    self.archivo=archivo
```

```
    self.archivo2=archivo2
```

```
    self.archivo3=archivo3 # cron01.txt
```

```
    self.archivo4=archivo4 # cron02.txt
```

```
    self.registro=[]
```

```
    self.bitsdcal=[]
```

```
    self.bajaaltitud=[]
```

```
    self.valdalt=[] # para validar la altitud
```

```
    self.spare=[] # evaluar bits de ceros
```

```
    self.horas=0
```

```
    self.minutos=0
```

```
    self.segundos=0
```

```
    self.vueltas=0
```

```
    bv=[] # para contar los blancos validos
```

```
    self.x=0
```

```
    self.valido=0
```

```
    self.pdi=[] # las veces que aparece el blanco punto de inicio y punto final
```

```
    self.valdt=[] # en donde aparece el tiempo
```

```
    self.tiemporeal=0 # Tiempo del primer mensaje de norte
```

```
    self.mnorte=mnorte # para saber cuantos mensajes de norte hay
```

```

self.vdref=0 # valor de mensaje del primer mensaje de norte

self.tiempodb=tiem

self.altitud=[]

self.rang=[]

self.acim=[]

self.identv=[]

self.bvalidos=[] # para saber cuales son los blancos validos y cuantos son

self.iden=identidad

self.alti=alt

self.dist=rango

self.aci=acimut

self.hmnorte="" # hora del primer mensaje de norte

self.regcap = os.path.join(os.path.dirname(__file__),'registro')

self.numdbcbc=0 # para saber cuantos bits tienen baja calidad
#self.a_fspect=['1','3','5','7','9','b','d','f']

def cal_m3a(self):
    archivo=open(self.archivo,"r+")

    lineas=archivo.readlines()

    archivo.close()

    x=0

    for k in lineas: # para capturar
        if k=="I001/080\n":
            self.registro.append(lineas[x+1]) # me dice cual blanco posee baja confianza del Modo 3/A
            self.bitsdcal.append(lineas[x+3]) # se obtienen los bits de confianza

            x=x+1

    print(self.registro)

    print("numero de blancos con baja confianza de Modo 3/A ",len(self.registro))

    print(self.bitsdcal)

    x=0

```

```

for k in lineas: # para capturar
    if k=="I001/100\n":
        self.bajaaltitud.append(lineas[x+1]) # me dice cual blanco posee baja confianza de altitud
        self.valdalt.append(lineas[x+2]) # se obtienen los bits de confianza
        self.spare.append(lineas[x+3])
    x=x+1
print(self.bajaaltitud)
print("numero de blancos con baja altitud ",len(self.bajaaltitud))
print(self.valdalt)
print(self.spare)

```

```

def tdpmn(self): # metodo para obtener el primer mensaje de norte
    archivo=open(self.archivo2,"r+")
    lineas=archivo.readlines()
    archivo.close()
    x=0
    pos=0
    pmdn=[]
    for k in lineas: # para capturar
        if k=="Mensaje de Norte\n":
            pmdn.append(lineas[x-2]) # para saber en cual mensaje esta el mensaje de norte
            x=x+1
    partida=pmdn[0]
    x=0
    for i in partida: # Encontramos la I para separar el numero
        if i=="I":
            pos=x
            x=x+1
    lista=[]
    valor="0"
    for i in range(0,pos):

```



```

    lista.append(partida[i])

for i in lista:
    valor=valor+i # en valor se encuentra nuestro tiempo de referencia del primer mensaje de Norte
self.vdref=int(valor)

x=0

for k in lineas: # para capturar
    if k==str(self.vdref)+"I002/030\n":
        self.tiemporeal=lineas[x+1] # tiempo de ref del primer mensaje de norte
        x=x+1

return self.tiemporeal

def adpadb(self): # metodo para analizar los blancos
t=self.tdpmn()
archivo=open(self.archivo4,"r+") # cron02.txt
lineas=archivo.readlines()
archivo.close()
x=0
for k in lineas:
    if k==str(self.vdref)+"\n":
        punto1=lineas[x+1] # la posicion del primer mensaje de norte en el archivo de captura de datos
        x=x+1
a=punto1.split('\n')
punto11=int(a[0])

archivo2=open(self.archivo3,"r+") # cron01.txt
lineas2=archivo2.readlines()
archivo2.close()
x=0
lista2=[]
lista3=[]
for p in lineas2: # para capturar
    b=p.split('\n')

```

```

    lista2.append(int(b[0]))
x=0
for i in lista2:
    if i>punto11:
        lista3.append(lista2[x-1])
    x=x+1
antes=lista3[0]-1 # numero de blancos antes del primer mensaje de norte

pref=self.tiempodb[antes]

t2=pref.split('\n')
tiempo2=t2[0]    # tiempo de ref para el mensaje de norte

t1=t.split('\n')
tiempo1=t1[0]    # tiempo de ref de mensaje de norte

h1=int(tiempo1[0:2])
m1=int(tiempo1[3:5])
s1=int(tiempo1[6:8])

s2=int(tiempo2[6:8])
x=0
nuevot1=[]
x=antes
for i in range(antes,len(self.alti)): # para los blancos despues del mensaje de norte
    vb=self.tiempodb[i]
    t4=vb.split('\n')
    tiempo4=t4[0]

a=self.iden[i].split('\n')
self.identv.append(a[0])

```

```
b=self.alti[i].split('\n')
self.altitud.append(b[0])
```

```
c=self.dist[i].split('\n')
self.rang.append(c[0])
```

```
d=self.aci[i].split('\n')
self.acim.append(d[0])
```

```
h=h1
m=m1
s=int(tiempo4[6:8])
s=s1-(s2-s)
if h<0:
    h=24+h
if m<0:
    m=60+m
    h=h-1
if s<0:
    s=60+s
    m=m-1
if h<10:
    horas="0"+str(h)
else:
    horas=str(h)
if m<10:
    minutos="0"+str(m)
else:
    minutos=str(m)
if s<10:
    segundos="0"+str(s)
else:
```

```

    segundos=str(s)

    nuevot1.append(horas+":"+minutos+": "+segundos)

    #x=x+1

return [nuevot1,tiempo1]

def analisisdb(self,codigo,archivo1): # para analizar cada blanco aceptar o rechazar

    ldatos=self.adpadb() # tiempo

    listat=ldatos[0]

    self.hmnorte=ldatos[1]

    x=0

    lista1=[] # altitud

    lista2=[] # rango

    lista3=[] # acimut

    for i in self.identv:

        if i==codigo:

            self.pdi.append(listat[x])

            lista1.append(self.altitud[x])

            lista2.append(str(round(float(self.rang[x]),2)))

            lista3.append(str(round(float(self.acim[x]),2)))

            x=x+1

    # hemos analizado solo un blanco

    archivo1.write(codigo+"\n")

    archivo1.write(str(len(self.pdi))+ "\n")

    archivo1.write("TIEMPO\tALTITUD\tRANGO\tACIMUT\n")

    for i in range(0,len(self.pdi)):

        archivo1.write(self.pdi[i]+ " "+lista1[i]+ " "+lista2[i]+ " "+lista3[i]+ "\n")

    self.pdi=[] # se vacian las listas de tiempo,identidad,altitud,rango y acimut para un nuevo blanco

    self.altitud=[]

    self.rang=[]

    self.acim=[]

```

```

self.identv=[]

def analisisdb2(self,vueltas,codigo,tipo): # para analizar cada blanco aceptar o rechazar
    vuel=int(vueltas)

    numdblan=self.ident(self.iden) # todos los blancos despues del primer mensaje de norte
    archivo1=open(os.path.join(self.regcap,"registrodb.txt"),"w")
    for i in numdblan:
        self.analisisdb(str(i),archivo1)
    archivo1.close()
    print(numdblan)
    #print(self.bvalidos)

    archivo2=open(os.path.join(self.regcap,"registrodb.txt"),"r")
    lineas2=archivo2.readlines()
    archivo2.close()

    # filtramos primero por el numero de plots que se desean admitir (minimo 2 ó 5 plots)
    x=0
    blancostec=[] # lista de blancos que se tomaron en cuenta
    for i in numdblan:
        for j in lineas2:
            if j==str(i)+'\n':
                longitud=lineas2[x+1].split('\n')
                if int(longitud[0])>=2: # aqui limitamos cuantos plot de ese blanco queremos evaluar
                    blancostec.append(str(i))
            x=x+1
        x=0
    print("blancos que se tomaron en cuenta:")
    print(blancostec)

def elimbdup(self,tiempo,altitud,rango,acimut): # Eliminacion de blancos duplicados
    x=0

```

```

antes=""
control=0
dist=257.0
nuevot=[] # para el nuevo tiempo
nuevad=[] # para el nuevo rango
nuevaa=[] # para la nueva altitud
nuevoac=[] # para el nuevo acimut

for i in tiempo:
    #if i==antes: # igualamos el tiempo para ver que el primero con el siguiente sean iguales
    if control==1:
        h1=int(antes[0:2])*3600
        m1=int(antes[3:5])*60
        s1=int(antes[6:8])

        r1=h1+m1+s1 # todo a segundos

        h2=int(i[0:2])*3600
        m2=int(i[3:5])*60
        s2=int(i[6:8])

        r2=h2+m2+s2 # todo a segundos

    if r2>r1:
        y=0
        for t in nuevot: # revisa si ya esta en la lista nuevot si ya esta no lo agrega y pasa al siguiente
            if antes==t:
                y=y+1
        if y==0:
            nuevot.append(antes)
            nuevad.append(dist)
            nuevaa.append(valt)
            nuevoac.append(vacm)

```

```

elif r2==r1:
    if float(dist)<float(rango[x]):
        nuevot.append(antes)
        nuevad.append(dist)
        nuevaa.append(valt)
        nuevoac.append(vacm)
    elif float(rango[x])<float(dist):
        nuevot.append(antes)
        nuevad.append(rango[x])
        nuevaa.append(altitud[x])
        nuevoac.append(acimut[x])
    elif float(rango[x])==float(dist):
        nuevot.append(antes)
        nuevad.append(dist)
        nuevaa.append(valt)
        nuevoac.append(vacm)

antes=i
valt=altitud[x]
dist=rango[x]
vacm=acimut[x]
control=1 # para controlar el antes=""
x=x+1

return [nuevot,nuevaa,nuevad,nuevoac] # Elije a los mejores blancos

```

```

def evdalt(self,altitud): # evaluacion de altitud

```

```

    x=0
    antes=altitud[0]
    for i in altitud:
        if abs(int(i)-int(antes))>=23000:
            x=x+1
    antes=i

```

```
if x==len(altitud) or x>=1:
```

```
    return "plot no valido"
```

```
else:
```

```
    return "plot valido"
```

```
def evdrang(self,distancia): # evaluacion de rango
```

```
    x=0
```

```
    antes=distancia[0]
```

```
    for i in distancia:
```

```
        if abs(float(i)-float(antes))>=9.99:
```

```
            x=x+1
```

```
            antes=i
```

```
    if x==len(distancia) or x>=1:
```

```
        return "plot no valido"
```

```
    else:
```

```
        return "plot valido"
```

```
def evdacim(self,angulo): # evaluacion de acimut
```

```
    x=0
```

```
    antes=angulo[0]
```

```
    for i in angulo:
```

```
        if abs(float(i)-float(antes))>=9.99:
```

```
            x=x+1
```

```
            antes=i
```

```
    if x==len(angulo) or x>=1:
```

```
        return "plot no valido"
```

```
    else:
```

```
        return "plot valido"
```

```
def ident(self,identificacion): # para sacar cada identificacion de blanco
```

```
    lista1=[]
```

```
    antes=0
```



```

for i in identificacion:
    x=0
    lista1.sort()
    c=i.split('\n')
    antes=int(c[0])
    for j in lista1: # revisa si ya esta en la lista nuevot si ya esta no lo agrega y pasa al siguiente
        if j==antes:
            x=x+1
    if x==0:
        b=i.split('\n')
        lista1.append(int(b[0]))

return lista1 # retorna cada una de las identificaciones

def condpartida(self,tiempo,vuel,codigo,tipo): # tiempo valido,identidad valido, codigo valido
    x=0
    pdi=tiempo
    mayor=max(tiempo) # tiempo donde terminara el analisis
    control=0 # variable para controlar si el tiempo de antena se paso del ultimo tiempo del blanco
    antena=[]

    # para estandarizar las vueltas de antena

    estandar=self.hmnorte

    self.horas=int(estandar[0:2])
    self.minutos=int(estandar[3:5])
    self.segundos=int(estandar[6:8])

    apariciones=[]

    for i in range(0,vuel):
        if self.horas>23:

```

```

self.horas=0
if self.minutos>59:
    self.minutos=0
    self.horas=self.horas+1
if self.segundos>59:
    self.segundos=self.segundos-60
    self.minutos=self.minutos+1
if self.horas<10:
    h="0"+str(self.horas)
else:
    h=str(self.horas)
if self.minutos<10:
    m="0"+str(self.minutos)
else:
    m=str(self.minutos)
if self.segundos<10:
    s="0"+str(self.segundos)
else:
    s=str(self.segundos)
estandar=h+":"+m+":"+s+"\n"
antena.append(estandar)

for j in tiempo:
    x=0
    hs=int(estandar[0:2])*3600
    ms=int(estandar[3:5])*60
    ss=int(estandar[6:8])

    instante1=hs+ms+ss

    vdantes=estandar.split("\n")
    antes=vdantes[0]

```

```

ht=int(j[0:2])*3600
mt=int(j[3:5])*60
st=int(j[6:8])

instante2=ht+mt+st

if instante1>=instante2:
    for k in apariciones: # revisa si ya esta en la lista apariciones si ya esta no lo agrega y pasa al siguiente
        if k==antes:
            x=x+1
        if x==0:
            if estandar>mayor: # si el tiempo de antena sobrepasa al tiempo maximo del blanco captura en apariciones
hasta ahi
                if control==0:
                    apariciones.append(estandar)
                    control=1
                else:
                    apariciones.append(estandar)

if tipo=="1":
    self.segundos=self.segundos+6
elif tipo=="2":
    self.segundos=self.segundos+8
print(antena)
return apariciones

```

Area_de_texto.py

```
#!/usr/bin/python

from Tkinter import *

class Area_de_texto:

    def __init__(self,ventana,largo,alto,color1,color2):

        self.frame=Frame(ventana)

        self.yScroll=Scrollbar(self.frame,orient=VERTICAL,bg=color1)

        self.yScroll.grid(row=0,column=1,sticky=N+S)

        self.ta=Text(self.frame,

            yscrollcommand=self.yScroll.set,width=largo,

            height=alto,bg=color1,fg=color2)

        self.ta.grid(row=0,column=0,sticky=N+S+E+W)

        self.yScroll["command"]=self.ta.yview

        self.frame.pack(side=TOP)

    def obtener_texto(self):

        to=self.ta.get("1.0",END)

        return to

    def insertar_texto(self,texto):

        self.ta.insert("1.0",texto)

    def borrar_texto(self):

        self.ta.delete("1.0",END)

    def act_des(self,a):

        if a==0:

            self.ta['state'] = NORMAL
```

```
elif a==1:  
    self.ta['state'] = DISABLED
```

capturar.py

```
#!/usr/bin/python  
  
from math import *  
from string import *  
  
class capturar_trama:  
    def __init__(self,archivo):  
        self.archivo=archivo  
        self.paquetes=0  
  
    def mtc(self):  
        line=[]  
        ffile=open(self.archivo,"r")  
        line=ffile.read()  
        ffile.close()  
        lista1=[]  
        for k in line:  
            lista1.append(k)  
        x=0  
        for i in lista1: # para encontrar los enter  
            if i=='\n':  
                x=x+1  
        for i in range(x):  
            lista1.remove('\n') # para remover los enter  
            self.paquetes=self.paquetes+1  
        x=0  
        return [lista1,self.paquetes]
```

cat01.py

```
#!/usr/bin/python

# -*- coding: cp1252 -*-

from math import *
from string import *
import sys
import os
from octal import *
from hexabin import *

class c01:

    def __init__(self,archivo1,archivo2,archivo3,archivo4):

        self.archivo1=archivo1

        self.archivo2=archivo2

        self.archivo3=archivo3

        self.archivo4=archivo4

        self.pares=["0","2","4","6","8","a","c","e"]

self.ro=[256.00,128.00,64.00,32.00,16.00,8.00,4.00,2.00,1.00,0.5000,0.2500,0.1250,0.0625,0.03125,0.015625,0.0078125]

self.theta=[180,90,45,22.5,11.25,5.625,2.8125,1.4063,0.7031,0.3516,0.1758,0.0879,0.0439,0.0220,0.0110,0.0055]

self.altura=[102400,51200,25600,12800,6400,3200,1600,800,400,200,100,50,25]

self.npp=[32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4,2,1]

self.vedopp=[0.25,0.125,0.0625,0.03125,0.015625,0.0078125,0.0078125]

self.ttod=[256.00,128.00,64.00,32.00,16.00,8.00,4.00,2.00,1.00,1/2,1/4,1/8,1/16,1/32,1/64,1/128]

self.potrec=[64,32,16,8,4,2,1]

self.pwec=[128.00,64.00,32.00,16.00,8.00,4.00,2.00,1.00]

self.ident="Plot"

self.pista=0

self.codigo=" "

self.altitud=0

self.rango=0

self.acimut=0
```

```
self.cdc="ACC" # indicador de confianza de codigo
self.cda="ACA" # indicador de confianza de altitud
self.campos=[]
self.idd=25
self.ext=0 # para extensiones del target report descriptor
```

```
def adpcons(self):
    archivo=open(self.archivo1,"r")
    lineas=archivo.readlines()
    archivo.close()

    granlista=[]

    for i in range(len(lineas)):
        if lineas[i].count(":")==0:
            granlista.append(lineas[i])

    archivo3=open(self.archivo3,"r")
    lineas3=archivo3.readlines()
    archivo3.close()

    self.archiv3=open(self.archivo4,"w")
    blancos=0
    trama="0"
    caso=0
    self.archiv2=open(self.archivo2,"w")
    for i in granlista:
        lista=list(i[6:len(i)-1])
        for j in lista:
            a=hexabin(j)
            valor1=a.conversion()
            trama=trama+valor1
```

```

if len(trama)>32:
    caso=self.adpfspec(self.pares,i[7],i[9],i[11])
x=0
if caso==1:
    self.idd=25
    for k in range(1,8):
        if trama[k]=="1":
            self.campos.append(k)
elif caso==2:
    self.idd=33
    for k in range(1,17):
        if trama[k]=="1":
            self.campos.append(k)
elif caso==3:
    self.idd=41
    for k in range(1,25):
        if trama[k]=="1":
            self.campos.append(k)

self.archiv2.write("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++\n")

self.archiv2.write(str(blancos+1)+"\n")
valn=lineas3[blancos].split("\t")
valn1=valn[1]
tiempodc=valn1[0:8]

self.archiv2.write("TIEMPO REAL DE CAPTURA "+tiempodc+"\n")

for n in self.campos:
    if n==2:
        self.adptrd(self.idd,trama)
        self.idd=self.idd+8
    elif n==3:
        if self.ident=="Plot":
            self.adpmpcc(self.idd,trama)

```



```

        self.idd=self.idd+32
elif self.ident=="Pista":
    self.adtpn(self.idd,trama)
    self.idd=self.idd+16
elif n==4:
    if self.ident=="Plot":
        self.adpmcor(self.idd,trama)
        self.idd=self.idd+16
    elif self.ident=="Pista": # Mode-3/A Code in Octal Representation
        self.adpmppc(self.idd,trama)
        self.idd=self.idd+32
elif n==5:
    if self.ident=="Plot":
        self.adpmcbr(self.idd,trama)
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.idd=self.idd+32
elif n==6:
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+32
elif n==7:
    if self.ident=="Plot":
        self.adptod(self.idd,trama)
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.adpmcor(self.idd,trama)
        self.idd=self.idd+16
elif n==9:
    if self.ident=="Plot":
        self.idd=self.idd+16

```

```

elif self.ident=="Pista":
    self.adpmcbr(self.idd,trama)
    self.idd=self.idd+16
elif n==10:
    if self.ident=="Plot":
        self.mrds(self.idd,trama)
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+16
elif n==11:
    if self.ident=="Plot":
        self.pr(self.idd,trama)
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==12:
    if self.ident=="Plot":
        self.adpidcm3a(self.idd,trama)
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==13:
    if self.ident=="Plot":
        self.adpidcmc(self.idd,trama)
        self.idd=self.idd+32
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==14: # 13 es 14
    if self.ident=="Plot":
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.idd=self.idd+8

```

```

elif n==15: # 14 es 15
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==17: # 15 es 17
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+16
elif n==18: # 16 es 18
    if self.ident=="Plot":
        cero=0
    elif self.ident=="Pista":
        self.adpidcm3a(self.idd,trama)
        self.idd=self.idd+16
elif n==19: # 17 es 19
    if self.ident=="Plot":
        cero=0
    elif self.ident=="Pista":
        self.adpidcmc(self.idd,trama)
        self.idd=self.idd+32
blancos=blancos+1

```

```

self.archiv2.write("*****\n")

```

```

trama="0"
self.campos=[]
caso=0
self.codigo=" "
self.altitud=0
self.rango=0
self.acimut=0

```

```

self.cdc="ACC"

self.cda="ACA"

def adpfspec(self,lista,a,b,c):

    caso=0

    listat=[a,b,c]

    for i in range(0,len(listat)):

        for j in lista:

            if listat[i]==j:

                caso=i+1

                break

    return caso

def adptrd(self,part,trama):

    lista=[part+1,part+2,part+3,part+4,part+5,part+6,part+7]

    if self.ext==0:

        if part==25:

            if trama[25]=="0":

                self.ident="Plot"

                self.archiv2.write("Plot\n")

            elif trama[25]=="1":

                self.ident="Pista"

                self.archiv2.write("Pista\n")

        elif part==33:

            if trama[33]=="0":

                self.ident="Plot"

                self.archiv2.write("Plot\n")

            elif trama[33]=="1":

                self.ident="Pista"

                self.archiv2.write("Pista\n")

        elif part==41:

            if trama[41]=="0":

```

```

self.ident="Plot"

self.archiv2.write("Plot\n")

elif trama[41]=="1":

self.ident="Pista"

self.archiv2.write("Pista\n")

self.archiv2.write("I001/020 DESCRIPCION DEL INFORME DE BLANCO\n")

if trama[lista[0]]=="0":

self.archiv2.write("Actual\n")

elif trama[lista[0]]=="1":

self.archiv2.write("Simulado\n")

if trama[lista[1]]=="0" and trama[lista[2]]=="0":

self.archiv2.write("No deteccion\n")

elif trama[lista[1]]=="0" and trama[lista[2]]=="1":

self.archiv2.write("Deteccion Primaria\n")

elif trama[lista[1]]=="1" and trama[lista[2]]=="0":

self.archiv2.write("Deteccion Secundaria\n")

elif trama[lista[1]]=="1" and trama[lista[2]]=="1":

self.archiv2.write("Deteccion combinada primaria y secundaria\n")

if trama[lista[3]]=="0":

self.archiv2.write("Reporte de blanco de antena 1\n")

elif trama[lista[3]]=="1":

self.archiv2.write("Reporte de blanco de antena 2\n")

if trama[lista[4]]=="1":

self.archiv2.write("Identificacion de posicion especial\n")

if trama[lista[5]]=="1":

self.archiv2.write("De un Transponder fijo\n")

if trama[lista[6]]=="1":

self.archiv2.write("Extension de Item de Datos\n")

```

```

self.idd=self.idd+8

self.ext=1

self.adptrd(self.idd,trama)

elif self.ext==1:

if trama[part]=='1':

self.archiv2.write("Indicador de Blanco Test\n")

if trama[lista[0]]=='0' and trama[lista[1]]=='1':

self.archiv2.write("Interferencia Ilicita (codigo 7500)\n")

elif trama[lista[0]]=='1' and trama[lista[1]]=='0':

self.archiv2.write("Falla de Radio-Comunicacion (codigo 7600)\n")

elif trama[lista[0]]=='1' and trama[lista[1]]=='1':

self.archiv2.write("Emergencia (codigo 7700)\n")

if trama[lista[2]]=='1':

self.archiv2.write("Emergencia Militar\n")

if trama[lista[3]]=='1':

self.archiv2.write("Identificacion Militar\n")

if trama[lista[6]]=='1':

self.archiv2.write("Extension de Item de Datos\n")

self.idd=self.idd+8

self.ext=1

self.adptrd(self.idd,trama)

elif trama[lista[6]]=='0':

self.ext=0

def adpmppc(self,a,trama):# measured position in polar coordinates

self.archiv2.write("I001/040 POSICION MEDIDA EN COORDENADAS POLARES\n")

lista=[a,a+16,a+32]

tro=trama[lista[0]:lista[1]]

```

```

ttheta=trama[lista[1]:lista[2]]

self.rango=self.multidec(tro,self.ro)

self.acimut=self.multidec(ttheta,self.theta)

self.archiv2.write("Rango = "+str(round(self.rango,2))+ " NM Acimut = "+str(round(self.acimut,2))+ " GRADOS\n")

def adpmcor(self,a,trama):# Mode-3/A Code in Octal Representation

self.archiv2.write("I001/070 MODO 3/A EN REPRESENTACION OCTAL\n")

lista=[a+4,a+7,a+10,a+13,a+16]

lista2=[a,a+1,a+2,a+3]

if trama[lista2[0]]=='0':

self.archiv2.write("Codigo Valido\n")

elif trama[lista2[0]]=='1':

self.archiv2.write("Codigo No Valido\n")

if trama[lista2[1]]=='1':

self.archiv2.write("Codigo Basura\n")

elif trama[lista2[1]]=='0':

self.archiv2.write("Por defecto\n")

if trama[lista2[2]]=='0':

self.archiv2.write("Codigo Modo-3/A es derivado de la respuesta del transponder\n")

elif trama[lista2[2]]=='1':

self.archiv2.write("Codigo Modo-3/A es provisto por un tracker local\n")

if trama[lista2[3]]=='0':

self.archiv2.write("Bit en estado correcto\n")

elif trama[lista2[3]]=='1':

self.archiv2.write("Bit en estado Incorrecto\n")

a=octal(trama[lista[0]:lista[1]])

digito1=a.conversion()

a=octal(trama[lista[1]:lista[2]])

digito2=a.conversion()

a=octal(trama[lista[2]:lista[3]])

digito3=a.conversion()

a=octal(trama[lista[3]:lista[4]])

```

```

digito4=a.conversion()

numero=digito1+digito2+digito3+digito4

self.codigo=str(digito1)+str(digito2)+str(digito3)+str(digito4)

self.archiv2.write("Codigo Transponder = "+str(self.codigo)+"\n")

def adpmcbr(self,a,trama):# Mode-C Code in Binary Representation

self.archiv2.write("I001/090 MODO C EN REPRESENTACION BINARIA\n")

lista=[a,a+1,a+2,a+3,a+16]

if trama[lista[0]]=='0':

self.archiv2.write("Codigo Valido\n")

elif trama[lista[0]]=='1':

self.archiv2.write("Codigo No Valido\n")

if trama[lista[1]]=='0':

self.archiv2.write("Por defecto\n")

elif trama[lista[1]]=='1':

self.archiv2.write("Codigo Basura\n")

if trama[lista[2]]=='0':

self.archiv2.write("Positivo\n")

elif trama[lista[2]]=='1':

self.archiv2.write("Negativo\n")

talt=trama[lista[3]:lista[4]]

self.altitud=self.multidec(talt,self.altura)

self.archiv2.write("Altitud = "+str(self.altitud)+" ft\n")

def adpidcm3a(self,a,trama): # Modo 3/A Indicador de Confianza

self.archiv2.write("I001/080 INDICADOR DE CONFIANZA DEL MODO 3/A\n")

lista=[a,a+16]

m3a=trama[lista[0]:lista[1]]

x=16

for i in range(0,len(m3a)):

if m3a[i]=="1":

self.cdc="BCC"

```



```

        self.archiv2.write("Baja confianza bit"+str(x)+"\n")

    x=x-1

def adpidcmc(self,a,trama): # Modo C Indicador de Confianza

    self.archiv2.write("I001/100 INDICADOR DE CONFIANZA DEL MODO C\n")

    plista=[a,a+1]

    lista=[a+20,a+32]

    mc=trama[lista[0]:lista[1]]

    if trama[plista[0]]=='0':

        self.archiv2.write("Codigo Valido\n")

    elif trama[plista[0]]=='1':

        self.archiv2.write("Codigo No Valido\n")

    if trama[plista[1]]=='1':

        self.archiv2.write("Codigo Basura\n")

    x=12

    for i in range(0,len(mc)):

        if mc[i]=="1":

            self.cda="BCA"

            self.archiv2.write("Baja confianza bit "+str(x)+"\n")

            x=x-1

def adptpn(self,a,trama):# track/plot number

    self.archiv2.write("I001/161 NUMERO DE PISTA/PLOT\n")

    lista=[a,a+16]

    serie=trama[lista[0]:lista[1]]

    self.archiv2.write(str(self.multidec(serie,self.npp))+"\n")

def mrds(self,a,trama):

    self.archiv2.write("I001/120 MEDICION RADIAL DE VELOCIDAD DOPPLER\n")

    lista=[a,a+1,a+8]

    serie=trama[lista[1]:lista[2]]

    if trama[lista[0]]=="0":

        self.archiv2.write(str(self.multidec(serie,self.vedopp))+"\n")

```

```

elif trama[lista[0]]=="1":
    self.archiv2.write(str(-1*self.multidec(serie,self.vedopp))+"\n")

def pr(self,a,trama):
    self.archiv2.write("I001/131 POTENCIA RECIBIDA\n")
    lista=[a,a+1,a+8]
    serie=trama[lista[1]:lista[2]]
    if trama[lista[0]]=="0":
        self.archiv2.write(str(self.multidec(serie,self.potrec))+"\n")
    elif trama[lista[0]]=="1":
        self.archiv2.write(str(-1*self.multidec(serie,self.potrec))+"\n")

def adpttod(self,a,trama):# Truncated time of Day
    self.archiv2.write("I001/141 TIEMPO DE DIA TRUNCADO\n")
    lista=[a,a+16]
    serie=trama[lista[0]:lista[1]]

    r1=self.multidec(serie,self.ttod)/3600
    horas=int(r1)
    r2=(r1-float(horas))*60
    minutos=int(r2)
    r3=(r2-float(minutos))*60
    segundos=int(r3)

    if horas<10:
        h="0"+str(horas)
    else:
        h=str(horas)

    if minutos<10:
        m="0"+str(minutos)
    else:

```

```

m=str(minutos)

if segundos<10:
    s="0"+str(segundos)
else:
    s=str(segundos)

self.archiv2.write(h+":"+m+":"+s+"\n")

def adpwec(self,a,trama):
    if trama[lista[2]]=="0":
        self.archiv2.write("I001/030 CONDICIONES ALARMA/ERROR\n")
    lista=[a,a+7,a+8]
    wec=trama[lista[0]:lista[1]]
    rwec=self.multidec(wec,self.pwec)
    if rwec==0:
        self.archiv2.write("No hay alarma ni condiciones de error\n")
    elif rwec==1:
        self.archiv2.write("respuesta con solape\n")
    elif rwec==2:
        self.archiv2.write("refleccion\n")
    elif rwec==3:
        self.archiv2.write("respuesta de lobulo secundario\n")
    elif rwec==4:
        self.archiv2.write("plot cortado\n")
    elif rwec==5:
        self.archiv2.write("respuesta de segunda vez\n")
    elif rwec==6:
        self.archiv2.write("angeles\n")
    elif rwec==7:
        self.archiv2.write("vehiculos terrestres\n")

```

```

elif rwecc==64:
    self.archiv2.write("Posible codigo erroneo en Modo 3/A\n")
elif rwecc==65:
    self.archiv2.write("posible informacion de altitud erronea, transmitido cuando falla el chequeo de credibilidad de
Codigo C junto con el codigo Modo-C en notacion binaria\n")
elif rwecc==66:
    self.archiv2.write("posible plot fantasma MSSR\n")
elif rwecc==80:
    self.archiv2.write("plot fijo PSR\n")
elif rwecc==81:
    self.archiv2.write("plot bajo PSR\n")
elif rwecc==82:
    self.archiv2.write("plot de baja calidad PSR\n")
if trama[lista[2]]=="1":
    self.archiv2.write("Extension de Item de Datos\n")
    self.idd=self.idd+8
    self.adpwecc(self.idd,trama)

def multidec(self,lista1,lista2):
    dc=[] # lista de igual longitud
    x=0
    for i in range(0,len(lista1)):
        if lista1[i]=='1':
            dc.append(lista2[x])
        x=x+1
    return sum(dc)

```

cat01pd.py

```

#!/usr/bin/python
# -*- coding: cp1252 -*-
from math import *
from string import *

```

```

import sys

import os

from octal import *

from hexabin import *

class c01pd:

    def __init__(self,archivo1,archivo2,archivo3,archivo4):

        self.archivo1=archivo1

        self.archivo2=archivo2

        self.archivo3=archivo3

        self.archivo4=archivo4

        self.hoja=self.archivo2.split(".")

        self.hda=self.hoja[0]+".xls"

        self.pares=["0","2","4","6","8","a","c","e"]

self.ro=[256.00,128.00,64.00,32.00,16.00,8.00,4.00,2.00,1.00,0.5000,0.2500,0.1250,0.0625,0.03125,0.015625,0.0078125]

self.theta=[180,90,45,22.5,11.25,5.625,2.8125,1.4063,0.7031,0.3516,0.1758,0.0879,0.0439,0.0220,0.0110,0.0055]

self.altura=[102400,51200,25600,12800,6400,3200,1600,800,400,200,100,50,25]

self.ident="Plot"

self.pista=0

self.codigo=" "

self.altitud=0

self.rango=0

self.acimut=0

self.cdc="ACC" # indicador de confianza de codigo

self.cda="ACA" # indicador de confianza de altitud

self.campos=[]

self.idd=25

self.ext=0 # para extensiones

self.falla="Ninguna"

def adpcons(self):

```

```

archivo=open(self.archivo1,"r")
lineas=archivo.readlines()
archivo.close()
granlista=[]

for i in range(len(lineas)):
    if lineas[i].count(":")==0:
        granlista.append(lineas[i])

archivo3=open(self.archivo3,"r")
lineas3=archivo3.readlines()
archivo3.close()
self.archiv3=open(self.archivo4,"w")
blancos=0
trama="0"
caso=0
self.archiv2=open(self.archivo2,"w")
for i in granlista:
    lista=list(i[6:len(i)-1])
    for j in lista:
        a=hexabin(j)
        valor1=a.conversion()
        trama=trama+valor1
    if len(trama)>32: # 8
        caso=self.adpfspec(self.pares,i[7],i[9],i[11])
    x=0
    if caso==1:
        self.idd=25
        for k in range(1,8):
            if trama[k]=="1":
                self.campos.append(k)
    elif caso==2:

```

```

self.idd=33

for k in range(1,17):
    if trama[k]=="1":
        self.campos.append(k)

elif caso==3:
    self.idd=41
    for k in range(1,25):
        if trama[k]=="1":
            self.campos.append(k)

for n in self.campos:
    if n==2:
        self.adptrd(self.idd,trama)
        self.idd=self.idd+8
    elif n==3:
        if self.ident=="Plot":
            self.adpmppc(self.idd,trama)
            self.idd=self.idd+32
        elif self.ident=="Pista":
            self.idd=self.idd+16
    elif n==4:
        if self.ident=="Plot":
            self.adpmcor(self.idd,trama)
            self.idd=self.idd+16
        elif self.ident=="Pista": # Mode-3/A Code in Octal Representation
            self.adpmppc(self.idd,trama)
            self.idd=self.idd+32
    elif n==5:
        if self.ident=="Plot":
            self.adpmcbr(self.idd,trama)
            self.idd=self.idd+16
        elif self.ident=="Pista":
            self.idd=self.idd+32

```

```

elif n==6:
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+32
elif n==7:
    if self.ident=="Plot":
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.adpmcor(self.idd,trama)
        self.idd=self.idd+16
elif n==9:
    if self.ident=="Plot":
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.adpmcbr(self.idd,trama)
        self.idd=self.idd+16
elif n==10:
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+16
elif n==11:
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==12:
    if self.ident=="Plot":
        self.adpidcm3a(self.idd,trama)
        self.idd=self.idd+16
    elif self.ident=="Pista":

```



```

        self.idd=self.idd+8
elif n==13:
    if self.ident=="Plot":
        self.adpidcmc(self.idd,trama)
        self.idd=self.idd+32
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==14:
    if self.ident=="Plot":
        self.idd=self.idd+16
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==15:
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+8
elif n==17: # 15 es 17
    if self.ident=="Plot":
        self.idd=self.idd+8
    elif self.ident=="Pista":
        self.idd=self.idd+16
elif n==18: # 16 es 18
    if self.ident=="Plot":
        cero=0
    elif self.ident=="Pista":
        self.adpidcm3a(self.idd,trama)
        self.idd=self.idd+16
elif n==19: # 17 es 19
    if self.ident=="Plot":
        cero=0
    elif self.ident=="Pista":

```

```

        self.adpidcmc(self.idd,trama)

        self.idd=self.idd+32

valn=lineas3[blancos].split("\t")

valn1=valn[1]

tiempodc=valn1[0:8]

blancos=blancos+1

self.archiv2.write(str(blancos)+"\t"+self.codigo+"\t"+str(self.altitud)+"\t"+str(round(self.rango,2))+"\t"+str(round(self.aci
mut,2))+"\t"+tiempodc+"\t"+self.cda+"\t"+self.cdc+"\t"+self.falla+"\t"+self.ident+"\n")

        trama="0"

        self.campos=[]

        caso=0

        self.codigo=" "

        self.altitud=0

        self.rango=0

        self.acimut=0

        self.cdc="ACC"

        self.cda="ACA"

        self.falla="Ninguna"

def adpfspec(self,lista,a,b,c):

        caso=0

        listat=[a,b,c]

        for i in range(0,len(listat)):

                for j in lista:

                        if listat[i]==j:

                                caso=i+1

                                break

        return caso

def adptrd(self,part,trama):

        lista=[part+1,part+2,part+3,part+4,part+5,part+6,part+7]

```

```

if self.ext==0:
    if part==25:
        if trama[25]=="0":
            self.ident="Plot"
        elif trama[25]=="1":
            self.ident="Pista"
    elif part==33:
        if trama[33]=="0":
            self.ident="Plot"
        elif trama[33]=="1":
            self.ident="Pista"
    elif part==41:
        if trama[41]=="0":
            self.ident="Plot"
        elif trama[41]=="1":
            self.ident="Pista"
    if trama[lista[6]]=='1':
        self.idd=self.idd+8
        self.ext=1
        self.adptrd(self.idd,trama)
elif self.ext==1:
    if trama[lista[0]]=='0' and trama[lista[1]]=='1':
        self.falla="Interferencia (codigo 7500)"
    elif trama[lista[0]]=='1' and trama[lista[1]]=='0':
        self.falla="Falla de Radio-Comunicacion (codigo 7600)"
    elif trama[lista[0]]=='1' and trama[lista[1]]=='1':
        self.falla="Emergencia (codigo 7700)"
    if trama[lista[6]]=='1':
        self.idd=self.idd+8
        self.ext=1
        self.adptrd(self.idd,trama)
elif trama[lista[6]]=='0':

```

```

self.ext=0

def adpmppc(self,a,trama):# measured position in polar coordinates
    lista=[a,a+16,a+32]
    tro=trama[lista[0]:lista[1]]
    ttheta=trama[lista[1]:lista[2]]
    self.rango=self.multidec(tro,self.ro)
    self.acimut=self.multidec(ttheta,self.theta)

def adpmcor(self,a,trama):# Mode-3/A Code in Octal Representation
    lista=[a+4,a+7,a+10,a+13,a+16]
    lista2=[a,a+1,a+2,a+3]
    a=octal(trama[lista[0]:lista[1]])
    digito1=a.conversion()
    a=octal(trama[lista[1]:lista[2]])
    digito2=a.conversion()
    a=octal(trama[lista[2]:lista[3]])
    digito3=a.conversion()
    a=octal(trama[lista[3]:lista[4]])
    digito4=a.conversion()
    numero=digito1+digito2+digito3+digito4
    self.codigo=str(digito1)+str(digito2)+str(digito3)+str(digito4)

def adpmcbr(self,a,trama):# Mode-C Code in Binary Representation
    lista=[a,a+1,a+2,a+3,a+16]
    talt=trama[lista[3]:lista[4]]
    if trama[lista[2]]=='0':
        self.altitud=self.multidec(talt,self.altura)
    elif trama[lista[2]]=='1':
        self.altitud=-(self.multidec(talt,self.altura))

def adpidcm3a(self,a,trama): # Modo 3/A Indicador de Confianza

```

```

lista=[a,a+16]

m3a=trama[lista[0]:lista[1]]

var=0

for i in range(0,len(m3a)):

    if m3a[i]=="1":

        self.cdc="BCC"

def adpidcmc(self,a,trama): # Modo C Indicador de Confianza

    lista=[a+20,a+32] # revisar

    var=0

    mc=trama[lista[0]:lista[1]]

    for i in range(0,len(mc)):

        if mc[i]=="1":

            self.cda="BCA"

def multidec(self,lista1,lista2):

    dc=[] # lista de igual longitud

    x=0

    for i in range(0,len(lista1)):

        if lista1[i]=='1':

            dc.append(lista2[x])

        x=x+1

    return sum(dc)

```

cat02.py

```

#!/usr/bin/python

# -*- coding: cp1252 -*-

from math import *

from string import *

import sys

import os

from octal import *

from hexabin import *

```

```

class c02:

    def __init__(self,archivo1,archivo2):

        self.archivo1=archivo1

        self.archivo2=archivo2

        self.pares=["0","2","4","6","8","a","c","e"]

self.lista1=[256.00,128.00,64.00,32.00,16.00,8.00,4.00,2.00,1.00,0.5000,0.2500,0.1250,0.0625,0.03125,0.015625,0.00781
25]

self.ttod=[65536.00,32768.00,16384.00,8192.00,4096.00,2048.00,1024.00,512.00,256.00,128.00,64.00,32.00,16.00,8.00,
4.00,2.00,1.00,0.5000,0.2500,0.1250,0.0625,0.03125,0.015625,0.0078125]

    self.campos=[]

    self.idd=25

    self.conteo=0

def adpcons(self):

    archivo=open(self.archivo1,"r")

    lineas=archivo.readlines()

    archivo.close()

    blancos=0

    trama="0"

    caso=0

    self.archiv2=open(self.archivo2,"w")

    for i in lineas:

        lista=list(i[6:len(i)-1])

        for j in lista:

            a=hexabin(j)

            valor1=a.conversion()

            trama=trama+valor1

        if len(trama)>8:

            caso=self.adpfspec(self.pares,i[7],i[9],i[11])

    x=0

```

```

if caso==1:
    self.idd=25
    x=x+1
    for k in range(1,8):
        if trama[k]=="1":
            self.campos.append(k)
            x=x+1
elif caso==2:
    self.idd=33
    x=x+1
    for k in range(1,17):
        if trama[k]=="1":
            self.campos.append(k)
            x=x+1
elif caso==3:
    self.idd=41
    x=x+1
    for k in range(1,25):
        if trama[k]=="1":
            self.campos.append(k)
            x=x+1

self.archiv2.write("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n")

self.archiv2.write(str(blancos+1)+"\n")

for n in self.campos:
    if n==2:
        self.tipodm(self.idd,trama)
        self.idd=self.idd+8
    elif n==3:
        self.idd=self.idd+8
    elif n==4:
        self.adptod(self.idd,trama)

```

```

        self.idd=self.idd+24
elif n==5:
    self.adpvra(self.idd,trama)
    self.idd=self.idd+16
elif n==6:
    self.idd=self.idd+8
elif n==7:
    self.adpmpe(self.idd,trama)
    self.idd=self.idd+8

blancos=blancos+1

self.archiv2.write("*****\n")

trama="0"
self.campos=[]
caso=0

def adpfspec(self,lista,a,b,c):
    caso=0
    listat=[a,b,c]
    for i in range(0,len(listat)):
        for j in lista:
            if listat[i]==j:
                caso=i+1
                break
    return caso

def adptod(self,a,trama):# Time of Day
    self.archiv2.write("I002/030 TIEMPO DE DIA\n")
    lista=[a,a+25]
    tod=trama[lista[0]:lista[1]]

```



```

r1=self.multidec(tod,self.ttod)/3600

horas=int(r1)

r2=(r1-float(horas))*60

minutos=int(r2)

r3=(r2-float(minutos))*60

segundos=int(r3)

if horas<10:
    h="0"+str(horas)
else:
    h=str(horas)

if minutos<10:
    m="0"+str(minutos)
else:
    m=str(minutos)

if segundos<10:
    s="0"+str(segundos)
else:
    s=str(segundos)

self.archiv2.write(h+":"+m+":"+s+"\n")

def tipodm(self,a,trama): # Message Type
    self.archiv2.write("I002/000 TIPO DE MENSAJE\n")
    lista=[a,a+8]
    if trama[lista[0]:lista[1]]=='00000001':
        self.archiv2.write("Mensaje de Norte\n")
    elif trama[lista[0]:lista[1]]=='00000010':
        self.archiv2.write("Mensaje de Sector\n")
    elif trama[lista[0]:lista[1]]=='00000011':

```

```

        self.archiv2.write("Mensaje de Sur\n")
elif trama[lista[0]:lista[1]]=='00001000':
    self.archiv2.write("Zona de filtrado activacion de blind\n")
elif trama[lista[0]:lista[1]]=='00001001':
    self.archiv2.write("Paro de Zona de filtrado\n")

def adpvra(self,a,trama):# Antenna Rotation Speed
    self.archiv2.write("I002/041\n")
    lista=[a,a+16]
    serie=trama[lista[0]:lista[1]]
    self.archiv2.write(str(self.multidec(serie,self.lista1))+ "\n")

def adpmpc(self,a,trama):# Station Processing Mode
    self.archiv2.write("I002/060\n")
    lista=[a,a+4,a+5,a+6,a+7]
    if trama[lista[0]:lista[1]]=='0000':
        self.archiv2.write("Estado correcto de bits 0000\n")
    else:
        self.archiv2.write("Error de bits\n")

    if trama[lista[1]]=='0':
        self.archiv2.write("Estado correcto de bit 0\n")
    else:
        self.archiv2.write("Error de bit\n")

    if trama[lista[2]]=='0':
        self.archiv2.write("Sin Alarma de Reloj Externa\n")
    elif trama[lista[2]]=='1':
        self.archiv2.write("Presencia de Alarma de Reloj Externa\n")

    if trama[lista[3]]=='0':
        self.archiv2.write("Sin Alarma de Tiempo Norte\n")

```

```
elif trama[lista[3]]=='1':  
    self.archiv2.write("Presencia de Alarma de Tiempo Norte\n")
```

```
if trama[lista[4]]=='1':  
    self.archiv2.write("Extension de Item de datos\n")
```

```
def multidec(self,lista1,lista2):  
    dc=[] # lista de igual longitud  
    x=0  
    for i in range(0,len(lista1)):  
        if lista1[i]=='1':  
            dc.append(lista2[x])  
            x=x+1  
    return sum(dc)
```

cat02pd.py

```
#!/usr/bin/python  
# -*- coding: cp1252 -*-  
from math import *  
from string import *  
import sys  
import os  
from octal import *  
from hexabin import *
```

```
class c02pd:  
    def __init__(self,archivo1,archivo2):  
        self.archivo1=archivo1  
        self.archivo2=archivo2  
        self.pares=["0","2","4","6","8","a","c","e"]
```

```
self.ttod=[65536.00,32768.00,16384.00,8192.00,4096.00,2048.00,1024.00,512.00,256.00,128.00,64.00,32.00,16.00,8.00,  
4.00,2.00,1.00,0.5000,0.2500,0.1250,0.0625,0.03125,0.015625,0.0078125]
```

```

self.campos=[]

self.idd=25

self.conteo=0

def adpcons(self):
    archivo=open(self.archivo1,"r")
    lineas=archivo.readlines()
    archivo.close()
    blancos=0
    trama="0"
    caso=0
    self.archiv2=open(self.archivo2,"w")
    for i in lineas:
        lista=list(i[6:len(i)-1])
        for j in lista:
            a=hexabin(j)
            valor1=a.conversion()
            trama=trama+valor1
        if len(trama)>8:
            caso=self.adpfspec(self.pares,i[7],i[9],i[11])
            x=0
        if caso==1:
            self.idd=25
            x=x+1
            for k in range(1,9):
                if trama[k]=="1":
                    self.campos.append(x)
                    x=x+1
        elif caso==2:
            self.idd=33
            x=x+1
            for k in range(1,17):

```

```

        if trama[k]=="1":
            self.campos.append(x)
            x=x+1
elif caso==3:
    self.idd=41
    x=x+1
    for k in range(1,25):
        if trama[k]=="1":
            self.campos.append(x)
            x=x+1

blancos=blancos+1
self.conteo=blancos
for n in self.campos:
    if n==2:
        self.idd=self.idd+8
    elif n==3:
        self.adptod(self.idd,trama,str(self.conteo))
        self.idd=self.idd+24

trama="0"
self.campos=[]
caso=0

def adpfspec(self,lista,a,b,c):
    caso=0
    listat=[a,b,c]
    for i in range(0,len(listat)):
        for j in lista:
            if listat[i]==j:
                caso=i+1
                break

```

```

return caso

def adptod(self,a,trama,numero):# Time of Day
    lista=[a,a+25]
    tod=trama[lista[0]:lista[1]]

    r1=self.multidec(tod,self.ttod)/3600

    horas=int(r1)
    r2=(r1-float(horas))*60
    minutos=int(r2)
    r3=(r2-float(minutos))*60
    segundos=int(r3)

    if horas<10:
        h="0"+str(horas)
    else:
        h=str(horas)

    if minutos<10:
        m="0"+str(minutos)
    else:
        m=str(minutos)

    if segundos<10:
        s="0"+str(segundos)
    else:
        s=str(segundos)

    self.archiv2.write(numero+"\t"+h+": "+m+": "+s+"\n")

def multidec(self,lista1,lista2):
    dc=[] # lista de igual longitud

```

```

x=0
for i in range(0,len(lista1)):
    if lista1[i]=='1':
        dc.append(lista2[x])
    x=x+1
return sum(dc)

```

estdpd.py

```

#!/usr/bin/env python
# Este archivo usa el encoding: utf-8

from math import *
import sys
import os
from string import *

class estc01:
    def __init__(self,archivo,var,archivo2,tipo,aldd,archivo3,archivo4):
        self.archivo=archivo
        self.archivo2=archivo2
        self.archivo3=archivo3
        self.archivo4=archivo4
        self.codigo=var
        self.br=[]
        self.diferentes=[]
        self.tipodp=tipo
        self.hora=" "
        self.pdtfijo=0 # variable que controla si hay un transponder fijo
        self.listadt=[] # lista para saber el codigo de los transponders fijos
        self.blancos=[]
        self.resultados=[]
        self.cpd=0.0
        self.aldd=aldd

```

```

self.rangomax=[] # para el grafico de PD vrs Rango Maximo
self.rangomin=[] # para el grafico de PD vrs Rango Maximo
self.altmax=[] # para el grafico de PD vrs Altitud Maxima
self.altmin=[] # para el grafico de PD vrs Altitud Maxima
self.probdd=[] # para cada una de las PD de los blancos
self.datost=0 # numero total de blancos
self.ntotaldresp=[] # numero total de respuestas de todos los blancos
self.nrespblanc=[] # respuestas de falsos blancos

def codtd(self):
    try:
        archivob=open(self.archivo,"r+")
        lineas=archivob.readlines()
        archivob.close()

        ok=0

        lista=[]
        xx=lineas[0]
        a=xx.split("\t")
        self.diferentes.append(a[1])

        for n in lineas: # contar el numero de codigos transponder diferentes
            lista=n.split("\t")

            for m in self.diferentes:
                if lista[1]==m:
                    ok=1

            if ok==0:
                self.diferentes.append(lista[1])

            ok=0

        for n in self.diferentes:
            if n==" ":
                self.diferentes.remove(" ")

            elif n=="0":
                self.diferentes.remove("0")

```



```

self.diferentes.sort()

return ["transponder diferentes = "+str(len(self.diferentes))+"\n",self.diferentes]

except IOError:

    print("ERROR!!! no se encuentra el archivo "+self.archivo+"\nRealice el procesamiento para analisis")

return [" ",self.diferentes]

def est1(self):

    self.altaconfd()

    archivob=open(self.archivo4,"r+")

    lineas=archivob.readlines()

    archivob.close()

    ok=0

    r3=0

    noincluir=0 # para no incluir blancos no validos

    lista=[]

    xx=lineas[0]

    a=xx.split("\t")

    self.diferentes.append(a[1])

    for n in lineas: # contar el numero de codigos transponder diferentes

        lista=n.split("\t")

        for m in self.diferentes:

            if lista[1]==m:

                ok=1

        if ok==0:

            self.diferentes.append(lista[1])

        ok=0

    h=1

    for n in self.diferentes:

        if n==" ":

            self.diferentes.remove(" ")

```

```

elif n=="0":
    self.diferentes.remove("0")
elif n=="0000":
    self.diferentes.remove("0000")
    self.nrespblanc.append(h)
    h=h+1

self.diferentes.sort()
self.datost=len(self.diferentes)
codtdif=self.codtransdif()
for i in codtdif:
    self.adtranspf(i)
# se averigua cuales se han descartado para no incluirlos en el calculo de PD individual
lctd=self.codtransdif()
for i in codtdif:
    for j in self.diferentes:
        if i==j:
            lctd.remove(i)

# remover el transponder fijo
for i in self.listadtf:
    lctd.remove(i)
x=0
y=0
ok=0
escrito1=0
escrito2=0
escritos=[]
if self.tipodp==1:
    for i in lctd:
        if self.codigo==i:
            noincluir=1

```

```

if noincluir==0:
    archivom=open(self.archivo2,"r+")
    lineasn=archivom.readlines()
    archivom.close()

    for i in lineasn:
        h=i.split("\t")
        hmn=h[1]
        x=0
        hm2=int(hmn[0:2])
        mm2=int(hmn[3:5])
        sm2=int(hmn[6:8])
        h2=hm2*3600
        m2=mm2*60
        r2=h2+m2+sm2
        for i in lineas:
            tiem=i.split("\t")
            tb=tiem[5]
            ha=int(tb[0:2])
            ma=int(tb[3:5])
            sa=int(tb[6:8])
            h1=ha*3600
            m1=ma*60
            r1=h1+m1+sa
            if r2==r1:
                r3=x
                break
            x=x+1
        x=0
    for i in range(r3,len(lineas)):
        lista=lineas[i].split("\t")
        if lista[1]==self.codigo:

```

```

        self.br.append(lineas[i])
if len(self.br)!=0:
    unal=self.edbd(self.br)
    y=round(float(self.edbd2(unal)),0)
esta=0
for i in self.diferentes:
    if self.codigo==i:
        esta=1
        self.resultados.append(str(y)+"\n")
        break
for i in self.listadtf:
    if self.codigo==i:
        esta=2
        self.pdtfijo=1
        self.resultados.append(self.codigo)
        break
if esta==1:
    vueltas=y
    datosv=round(float(self.adalt()),0)
    self.blancos=[]
    self.br=[]
    if datosv>vueltas:
        intro=round(float(self.adrya()),0)
        self.resultados.append(str(intro[0])+"\n")
        self.resultados.append(str(round((float(intro[0])/float(vueltas))*100,2))+ " % "+" \n")
        self.cpd=round((float(intro[0])/float(vueltas))*100,2)
    elif datosv==0:
        if vueltas==1:
            self.resultados.append(self.codigo)
            self.resultados.append("no se")
            self.resultados.append("considera ")
            self.cpd=0

```

```

        self.pdtfijo=3
elif datosv!=0:
    self.resultados.append(str(datosv)+"\n")
    self.resultados.append(str(round((float(datosv)/float(vueltas))*100,2))+ " %"+"%\n")
    self.cpd=round((float(datosv)/float(vueltas))*100,2)
elif esta==2:
    self.resultados.append("se considera que es")
    self.resultados.append("un transponder fijo\n")
    esta=0
elif noincluir==1:
    self.resultados.append(self.codigo)
    self.resultados.append("no se")
    self.resultados.append("considera ")
    self.pdtfijo=2
    noincluir=0
elif self.tipodp==2:
    r3=0
    res=0
    pdt=[]
    archivom=open(self.archivo2,"r+")
    lineasn=archivom.readlines()
    archivom.close()

    for i in lineasn:
        h=i.split("\t")
        hmn=h[1]
        x=0
        hm2=int(hmn[0:2])
        mm2=int(hmn[3:5])
        sm2=int(hmn[6:8])
        h2=hm2*3600
        m2=mm2*60

```

```

r2=h2+m2+sm2
for i in lineas:
    tiem=i.split("\t")
    tb=tiem[5]
    ha=int(tb[0:2])
    ma=int(tb[3:5])
    sa=int(tb[6:8])
    h1=ha*3600
    m1=ma*60
    r1=h1+m1+sa
    if r2==r1:
        r3=x
        break
    x=x+1
x=0
for j in self.diferentes:
    for i in range(r3,len(lineas)):
        lista=lineas[i].split("\t")
        if lista[1]==j:
            self.br.append(lineas[i])
unal=self.edbd(self.br)
y=round(float(self.edbd2(unal)),0)
vueltas=y
datosv=round(float(self.adalt()),0)
self.blancos=[]
self.br=[]

if datosv==0:
    if vueltas==1:
        res=0
        self.datost=self.datost-1

```

```

        # falsos blancos que aparecen solo una vez
        self.nrespblanc.append(vueltas)
elif datosv>=1:
    if vueltas>1:
        res=(float(datosv)/float(vueltas))*100
        self.ntotaldresp.append(float(vueltas))
    if res<=100:
        pdt.append(round(res,2))
        self.resultados.append(j+" es "+str(round(res,2))+ " %"+"\\n")
        self.probdd.append(round(res,2))
        self.rangomax.append(self.adrgraf())
        self.rangomin.append(self.adrgraf2())
        self.altmax.append(self.adagraf())
        self.altmin.append(self.adagraf2())
self.resultados.append(str(round(float(sum(pdt))/float(self.datost),2))+ " %"+"\\n")
self.cpd=round(float(sum(pdt))/float(self.datost),2)

return [self.resultados,self.cpd,self.rangomax,self.probdd,self.altmax,self.rangomin,self.altmin,self.pdtfijo]

def anfb(self): # numero total de respuestas y el numero de respuestas de falsos blancos
    self.est1()
    return [sum(self.ntotaldresp),sum(self.nrespblanc)]

def adalt(self):
    tipo=0
    deteccion=0
    for i in self.blancos:
        alti=i.split("\\t")
        if int(alti[2])>=int(self.aldd) and int(alti[2])<=50000:
            deteccion=deteccion+1
    return deteccion

```

```

def edbd(self,lista):
    archivo2=open(self.archivo3,"w")
    # eliminacion de blancos duplicados en tiempo
    tiempo=[]
    blancos=[]
    mlista=lista[0].split("\t")
    tiempo.append(mlista[5])
    blancos.append(self.br[0])
    for j in self.br:
        lista1=j.split("\t")
        for i in range(len(tiempo)):
            if tiempo[i]==lista1[5]:
                escrito1=1

        if escrito1==0:
            tiempo.append(lista1[5])
            blancos.append(j)
            archivo2.write(j)
            escrito1=0
    archivo2.close()
    return blancos

def edbd2(self,lista):
    # eliminacion de blancos duplicados por baja confianza, altitud, rango y acimut
    y=0
    lista2=[]
    lista3=[]
    lista4=[]
    lista5=[]
    x=0
    ok=0

```



```

ok1=0

for j in lista:
    if x==len(lista)-1:
        break
    else:
        va=j.split("\t")
        t1=va[5]
        pe=float(va[3])
        lb=lista[x+1]
        vb=lb.split("\t")
        t2=vb[5]
        pe2=float(vb[3])
        if pe!=pe2 and t1!=t2:
            for i in lista2:
                if i==pe:
                    ok=1
                    break
            if ok==0:
                lista2.append(pe)
                lista3.append(j)
            for i in lista2:
                if i==pe2:
                    ok1=1
                    break
            if ok1==0:
                lista2.append(pe2)
                lista3.append(lb)
            ok=0
            ok1=0

x=x+1

# Analisis de confianza

```

```

for i in lista3:
    pdl=i.split("\t")
    cda=pdl[6]
    cdc=pdl[7]
    if cda=="ACA" and cdc=="ACC":
        lista4.append(i)
    else:
        y=y+1
# Analisis de rango y acimut logicos
resultado=self.adrya(lista4)
z=resultado[0]
y=y+z
lista5=resultado[1]

# Analisis de altitud logica
for i in lista5:
    pdl=i.split("\t")
    altitud=int(pdl[2])
    if altitud>0 and altitud<=100000:
        self.blancos.append(i)
    y=y+1
if y==0:
    y=1
return y

def adrya(self,lineas): # basarse en el historial
    lista=[]
    x=0
    y=0
    plots=0
    cambiar=0
    for i in range(len(lineas)):

```

```

if i==0:
    v1=lineas[x].split("\t")
    ar=float(v1[3])
    aa=float(v1[4])
else:
    v2=lineas[x].split("\t")
    r1=float(v2[3])
    a1=float(v2[4])

if r1>=ar-4.0 and r1<=ar+4.0:
    if a1>=258.00 and a1<=260.00:
        if a1>=0.0 and a1<2.0:
            lista.append(lineas[x])
        else:
            if a1>=aa-1.5 and a1<=aa+1.5:
                lista.append(lineas[x])
if r1>=ar-4.0 and r1<=ar+4.0:
    if a1>=258.00 and a1<=260.00:
        if a1>=0.0 and a1<2.0:
            lista.append(lineas[x])
            cambiar=1
        else:
            if a1>=aa-1.5 and a1<=aa+1.5:
                lista.append(lineas[x])
                cambiar=1
else:
    y=y+1
if cambiar==1:
    plots=plots+1
if plots==2:
    ar=r1
    aa=a1

```

```

    plots=0
    cambiar=0
elif plots==1:
    ar=ar
    aa=aa
elif plots==0:
    ar=r1
    aa=a1
x=x+1

self.nrespblanc.append(y)

final=[]
for i in range(len(lista)/2):
    final.append(lista[i])

return [y,final]

def adnrpr(self,codigo): # Analisis de numero de respuestas por rango
    rango=[]
    dif=[]
    num=[]
    archivo=open(self.archivo,"r+")
    lineas=archivo.readlines()
    archivo.close()
    ok=0
    for i in lineas:
        v1=i.split("\t")
        if codigo==v1[1]:
            rango.append(round(float(v1[3]),2))
    dif.append(round(rango[0],2))
# contar el numero de rangos diferentes

```

```

for n in rango:
    for m in dif:
        if n==m:
            ok=1
        if ok==0:
            dif.append(round(n,2))
        ok=0
for i in dif:
    num.append(rango.count(i)) # conteo del numero de respuestas

return [dif,num]

```

```

def adrgraf(self): # Analisis de rango para el grafico

```

```

    lista=[]
    maximo=0.0
    archivo=open(self.archivo3,"r")
    lineas=archivo.readlines()
    archivo.close()
    for i in lineas:
        v1=i.split("\t")
        ar=float(v1[3])
        lista.append(ar)
    if len(lista)!=0:
        maximo=max(lista)
    return maximo

```

```

def adagraf(self): # Analisis de altitud para el grafico

```

```

    lista=[]
    maximo=0.0
    archivo=open(self.archivo3,"r")
    lineas=archivo.readlines()
    archivo.close()

```

```

for i in lineas:
    v1=i.split("\t")
    ar=float(v1[2])
    lista.append(ar)
if len(lista)!=0:
    maximo=max(lista)
return maximo

```

```

def adagraf2(self): # Analisis de altitud para el grafico

```

```

    lista=[]
    archivo=open(self.archivo3,"r")
    lineas=archivo.readlines()
    archivo.close()
    minimo=0.0
    for i in lineas:
        v1=i.split("\t")
        ar=float(v1[2])
        if ar>0:
            lista.append(ar)
    if len(lista)!=0:
        minimo=min(lista)
    return minimo

```

```

def adrgraf2(self): # Analisis de rango minimo para el grafico

```

```

    lista=[]
    minimo=0.0
    archivo=open(self.archivo3,"r")
    lineas=archivo.readlines()
    archivo.close()
    for i in lineas:
        v1=i.split("\t")
        ar=float(v1[3])

```

```

    lista.append(ar)

if len(lista)!=0:
    minimo=min(lista)

return minimo

def altaconfd(self): # escoge todas las respuestas de alta confianza de altitud y de codigo

    archivo=open(self.archivo,"r+")

    lineas=archivo.readlines()

    archivo.close()

    archivo2=open(self.archivo4,"w")

    for i in lineas:

        blancos=i.split("\t")

        if int(blancos[2])>0 and int(blancos[2])<=50000:

            if blancos[6]=="ACA" and blancos[7]=="ACC":

                archivo2.write(i)

    archivo2.close()

def adtranspf(self,codigo): # analiza si el codigo es un transponder fijo

    archivo=open(self.archivo,"r+")

    lineasda=archivo.readlines()

    archivo.close()

    lineas=[]

    for i in lineasda:

        blancos=i.split("\t")

        if blancos[1]==codigo:

            lineas.append(i)

    y=0

    lista2=[]

    lista3=[]

    longitudt=0

    x=0

```

```

ok=0
ok1=0
for j in lineas:
    if x==len(lineas)-1:
        break
    else:
        va=j.split("\t")
        t1=va[5]
        pe=float(va[3])
        lb=lineas[x+1]
        vb=lb.split("\t")
        t2=vb[5]
        pe2=float(vb[3])
        if pe!=pe2 and t1!=t2:
            for i in lista2:
                if i==pe:
                    ok=1
                    break
            if ok==0:
                lista2.append(pe)
                lista3.append(j)
            for i in lista2:
                if i==pe2:
                    ok1=1
                    break
            if ok1==0:
                lista2.append(pe2)
                lista3.append(lb)
            ok=0
            ok1=0
        x=x+1
x=0

```



```

y=0
longitudt=len(lista3) # longitud del transponder fijo debe ser de 2
for i in range(len(lista3)):
    if i==0:
        v1=lista3[y].split("\t")
        ala=int(v1[2])
        ar=round(float(v1[3]),1)
        aa=round(float(v1[4]),0)
        hb=v1[5]
        hb1=int(hb[0:2])
        mb1=int(hb[3:5])
        sb1=int(hb[6:8])
        h1=hb1*3600
        m1=mb1*60
        rh1=h1+m1+sb1
    else:
        v2=lista3[y].split("\t")
        al1=int(v2[2])
        r1=round(float(v2[3]),1)
        a1=round(float(v2[4]),0)
        ha=v2[5]
        ht=int(ha[0:2])
        mt=int(ha[3:5])
        st=int(ha[6:8])
        h2=ht*3600
        m2=mt*60
        rh2=h2+m2+st
    if rh2>rh1:
        if r1==ar:
            if a1==aa:
                if al1==ala:
                    if longitudt==2:

```

```

        self.listadtf.append(codigo)

        break

    y=y+1
return x

def codtransdif(self):
    archivob=open(self.archivo,"r+")
    lineas=archivob.readlines()
    archivob.close()
    diferentes=[]
    ok=0
    r3=0
    lista=[]
    xx=lineas[0]
    a=xx.split("\t")
    diferentes.append(a[1])
    for n in lineas: # contar el numero de codigos transponder diferentes
        lista=n.split("\t")
        for m in diferentes:
            if lista[1]==m:
                ok=1
        if ok==0:
            diferentes.append(lista[1])
        ok=0
    for n in diferentes:
        if n==" ":
            diferentes.remove(" ")
        elif n=="0":
            diferentes.remove("0")
    diferentes.sort()
    return diferentes

```

```

def estadb(self): # Para las estadísticas

    archivo=open(self.archivo,"r+")

    lineas=archivo.readlines()

    archivo.close()

    altitud=[]

    codigo=[]

    rango=[]

    confalt=[]

    confcod=[]

    codemer=[]

    plotpista=[]

    caso1=0

    caso2=0

    caso3=0

    caso4=0

    for i in lineas:

        blancos=i.split("\t")

        if blancos[6]=="ACA" and blancos[7]=="ACC":

            caso1=caso1+1

        elif blancos[6]=="ACA" and blancos[7]=="BCC":

            caso2=caso2+1

        elif blancos[6]=="BCA" and blancos[7]=="ACC":

            caso3=caso3+1

        elif blancos[6]=="BCA" and blancos[7]=="BCC":

            caso4=caso4+1

        codigo.append(blancos[1])

        altitud.append(int(blancos[2]))

        rango.append(float(blancos[3]))

        confalt.append(blancos[6])

        confcod.append(blancos[7])

        codemer.append(blancos[8])

```

```

plotpista.append(blancos[9])

altcero=0
altmenos=0
altmin=[]
m=0
for i in range(len(altitud)):
    if altitud[i]>0 and altitud[i]<=50000:
        altmin.append(altitud[i])
    elif altitud[i]==0:
        altcero=altcero+1
    elif altitud[i]<0:
        altmenos=altmenos+1
    elif altitud[i]>50000:
        m=m+1
maxalt=max(altitud)
minalt=min(altmin)
maxrang=max(rango)
minrang=min(rango)

indbcc=0
indacc=0
for i in confcod:
    if i=="ACC":
        indacc=indacc+1
    elif i=="BCC":
        indbcc=indbcc+1

indbca=0
indaca=0
for i in confalt:
    if i=="ACA":

```

```

        indaca=indaca+1
    elif i=="BCA":
        indbca=indbca+1

sincod=0
for i in codigo:
    if i==" ":
        sincod=sincod+1

emer=0
for i in codemer:
    if i!="Ninguna":
        emer=emer+1

plot=0
pista=0
for i in plotpista:
    if i=="Plot\n":
        plot=plot+1
    elif i=="Pista\n":
        pista=pista+1

return
[maxalt,minalt,maxrang,minrang,m,altcero,altmenos,indbcc,indacc,indbca,indaca,sincod,emer,plot,pista,caso1,caso2,caso
3,caso4]

```

hexabin.py

```

#!/usr/bin/python
from math import *
from string import *

class hexabin:
    def __init__(self,numero):
        self.num=numero

```

```

self.nhex=['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f']
self.nbin=['0000','0001','0010','0011','0100','0101','0110','0111','1000','1001','1010','1011','1100','1101','1110','1111']
self.binario=0

```

```

def conversion(self):
    x=0
    for j in self.nhex:
        if self.num==j:#para la longitud del fspec
            self.binario=self.nbin[x]
            x=x+1
    return self.binario

```

lon_t.py

```

#!/usr/bin/python
from math import *
from string import *
from hexabin import *

```

```

class lon_t:
    def __init__(self,lista,indice):
        self.lista=lista
        self.inicio=indice

    def hexadec(self):
        n1=hexabin(self.lista[self.inicio])
        valor1=n1.conversion()

        n2=hexabin(self.lista[self.inicio+1])
        valor2=n2.conversion()

        n3=hexabin(self.lista[self.inicio+2])
        valor3=n3.conversion()

```

```
n4=hexabin(self.lista[self.inicio+3])
```

```
valor4=n4.conversion()
```

```
longitud=valor1+valor2+valor3+valor4
```

```
lista2=[32768,16384,8192,4096,2048,1024,512,256,128,64,32,16,8,4,2,1]
```

```
x=0
```

```
lista3=[]
```

```
for i in range(len(longitud)):
```

```
    if longitud[i]=='1':
```

```
        lista3.append(lista2[x])
```

```
        x=x+1
```

```
    return sum(lista3)*2
```

mostrarb.py

```
#!/usr/bin/python
```

```
# -*- coding: cp1252 -*-
```

```
from math import *
```

```
from string import *
```

```
class mostrar_blanco:
```

```
    def __init__(self, tipo, lista1, lista2, lista3, activar, escala, codigo, altitud, fallas, mapa, confda, confdc, pista):
```

```
        self.lista1=lista1 # cursor
```

```
        self.lista2=lista2 # indicaciones
```

```
        self.lista3=lista3 # lista cpdb
```

```
        self.activar=1
```

```
        self.ro=0
```

```
        self.theta=0
```

```
        self.escala=escala
```

```
        self.div=0
```

```
        self.codigo=codigo
```

```
self.altitud=altitud
self.fallas=fallas
self.mapa=mapa
self.confda=confda
self.confdc=confdc
self.tipo=tipo
self.pista=pista
```

```
def mbep(self):
    if self.escala==250:
        self.div=1
    elif self.escala==225:
        self.div=0.9
    elif self.escala==200:
        self.div=0.8
    elif self.escala==175:
        self.div=0.7
    elif self.escala==150:
        self.div=0.6
    elif self.escala==125:
        self.div=0.5
    elif self.escala==100:
        self.div=0.4
    elif self.escala==75:
        self.div=0.3
    elif self.escala==50:
        self.div=0.2
    elif self.escala==25:
        self.div=0.1
    elif self.escala==12.5:
        self.div=0.05
    elif self.escala==6.25:
```



```

self.div=0.025

if self.activar==1:

    self.ro=float(self.lista3[0])/self.div

    self.theta=-float(self.lista3[1])+90.00

    xx=self.ro*cos(self.theta*pi/180)+266

    yy=253-self.ro*sin(self.theta*pi/180)

    if self.fallas=="Interferencia Ilicita (codigo 7500)":

        if self.pista=="SI":

            self.lista2.create_text(xx-21,yy,text="I",font=("Arial",10),fill="red")

            pista11=self.lista2.create_text(xx,yy,text="I A "+self.codigo,font=("Arial",10),fill="red")

            self.lista1.append(pista11)

        elif self.pista=="NO":

            self.lista1.append(self.lista2.create_text(xx,yy,text="I A "+self.codigo,font=("Arial",10),fill="red"))

            self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+" ft",font=("Arial",10),fill="red"))

            self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+" NM",font=("Arial",10),fill="red"))

        elif self.fallas=="Falla de Radio-Comunicacion (codigo 7600)":

            if self.pista=="SI":

                self.lista2.create_text(xx-21,yy,text="R",font=("Arial",10),fill="red")

                pista12=self.lista2.create_text(xx,yy,text="R A "+self.codigo,font=("Arial",10),fill="red")

                self.lista1.append(pista12)

            elif self.pista=="NO":

                self.lista1.append(self.lista2.create_text(xx,yy,text="R A "+self.codigo,font=("Arial",10),fill="red"))

                self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+" ft",font=("Arial",10),fill="red"))

                self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+" NM",font=("Arial",10),fill="red"))

            elif self.fallas=="Emergencia (codigo 7700)":

                if self.pista=="SI":

                    self.lista2.create_text(xx-21,yy,text="E",font=("Arial",10),fill="red")

                    pista13=self.lista2.create_text(xx,yy,text="E A "+self.codigo,font=("Arial",10),fill="red")

                    self.lista1.append(pista13)

                elif self.pista=="NO":

                    self.lista1.append(self.lista2.create_text(xx,yy,text="E A "+self.codigo,font=("Arial",10),fill="red"))

```

```

self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+" ft",font=("Arial",10),fill="red"))

self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[x],font=("Arial",10),fill="red"))

self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+" NM",font=("Arial",10),fill="red"))

else:

if self.mapa=="DESCRIPTIVO" and self.tipo==0:

    if self.confdc=="BCC":

        if self.pista=="SI":

            self.lista2.create_text(xx-21,yy,text="*",font=("Arial",10),fill="red")

            pista1=self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="red")

            self.lista1.append(pista1)

            if self.confda=="BCA":

                self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="red"))

                elif self.confda=="ACA":

                    self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="black"))

                    self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+"
NM",font=("Arial",10),fill="brown"))

            elif self.pista=="NO":

                self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="red"))

                if self.confda=="BCA":

                    self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="red"))

                    elif self.confda=="ACA":

                        self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="black"))

                        self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+"
NM",font=("Arial",10),fill="brown"))

            elif self.confdc=="ACC":

                if self.pista=="SI":

                    self.lista2.create_text(xx-21,yy,text="*",font=("Arial",10),fill="dark green")

                    pista2=self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="dark green")

                    self.lista1.append(pista2)

```

```

        if self.confda=="BCA":

            self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="red"))

            elif self.confda=="ACA":

                self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="black"))

                self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+"
NM",font=("Arial",10),fill="brown"))

        elif self.pista=="NO":

            self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="dark
green"))

            if self.confda=="BCA":

                self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="red"))

                elif self.confda=="ACA":

                    self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",width=200,font=("Arial",10),fill="black"))

                    self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+"
NM",font=("Arial",10),fill="brown"))

        elif self.mapa=="SATELITE" and self.tipo==1:

            if self.confdc=="BCC":

                if self.pista=="SI":

                    self.lista2.create_text(xx-21,yy,text="*",font=("Arial",10),fill="red")

                    self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="red"))

                if self.pista=="NO":

                    self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="red"))

            elif self.confdc=="ACC":

                if self.pista=="SI":

                    self.lista2.create_text(xx-21,yy,text="*",font=("Arial",10),fill="orange")

                    self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="orange"))

                elif self.pista=="NO":

                    self.lista1.append(self.lista2.create_text(xx,yy,text="* A "+self.codigo,font=("Arial",10),fill="orange"))

```

```

        if self.confda=="BCA":
            self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+" ft",font=("Arial",10),fill="red"))
        elif self.confda=="ACA":
            self.lista1.append(self.lista2.create_text(xx,yy-14,text="C "+self.altitud+"
ft",font=("Arial",10),fill="yellow"))
            self.lista1.append(self.lista2.create_text(xx,yy+14,text=self.lista3[0]+" NM",font=("Arial",10),fill="cyan"))

```

octal.py

```

#!/usr/bin/python
from math import *
from string import *

class octal:
    def __init__(self,numero):
        self.num=numero
        self.nbin=['000','001','010','011','100','101','110','111']
        self.noct=['0','1','2','3','4','5','6','7']
        self.noctal=0

    def conversion(self):
        x=0
        for j in self.nbin:
            if self.num==j:
                self.noctal=self.noct[x]
            x=x+1
        return self.noctal

```

trama_asterix.py

```

#!/usr/bin/python
from math import *
from string import *

class trama_asterix:

```

```

def __init__(self,ndblanco,archivo):

    self.numero=ndblanco # string

    self.archivo=archivo

    self.lista2=[]

    self.lista1=[]

def proc01(self):

    listat=[]

    ffile=open(self.archivo,"r")

    lineas=ffile.readlines()

    ffile.close()

    yi=0

    yf=0

    x=0

    for i in lineas:

        if i==self.numero+"\n":

            yi=x

        if i==self.numero+'-----\n':

            yf=x

            x=x+1

    for i in range(yi,yf): # obtener un blanco especifico

        self.lista1.append(lineas[i])

    x=0

    for i in self.lista1:

        if i==self.numero+"I001/010\n":

            self.lista2.append("numero de blanco = "+self.numero+"\n")

            self.lista2.append("I001/010 IDENTIFICACION DE LA FUENTE DE DATOS\n")

            self.lista2.append(self.lista1[x+1])

            self.lista2.append(self.lista1[x+2])

        elif i==self.numero+"I001/020\n":

            self.lista2.append("I001/020 DESCRIPCION DEL INFORME DE BLANCO\n")

            self.lista2.append(self.lista1[x+1])

```

```

self.lista2.append(self.lista1[x+2])

self.lista2.append(self.lista1[x+3])

self.lista2.append(self.lista1[x+4])

elif i==self.numero+"I001/040\n":

    self.lista2.append("I001/040 POSICION MEDIDA EN COORDENADAS POLARES\n")

    rango1=list(self.lista1[x+1])

    rango2=self.bn(rango1)

    self.lista2.append("Rango="+rango2+" NM\n")

    acimut1=list(self.lista1[x+2])

    acimut=self.bn(acimut1)

    self.lista2.append("Acimut="+acimut+" grados\n")

elif i==self.numero+"I001/070\n":

    self.lista2.append("I001/070 MODO 3/A EN REPRESENTACION OCTAL\n")

    self.lista2.append(self.lista1[x+1])

    self.lista2.append(self.lista1[x+3])

    self.lista2.append("Codigo Transponder="+self.lista1[x+4])

elif i==self.numero+"I001/090\n":

    self.lista2.append("I001/090 MODO C EN REPRESENTACION BINARIA\n")

    self.lista2.append(self.lista1[x+1])

    self.lista2.append(self.lista1[x+3])

    altitud1=list(self.lista1[x+4])

    altitud=self.bn(altitud1)

    self.lista2.append("Altitud del blanco="+altitud+" pies \n")

elif i==self.numero+"I001/130\n":

    self.lista2.append("I001/130 CARACTERISTICAS DEL PLOT\n")

    self.lista2.append(self.lista1[x+1])

elif i==self.numero+"I001/141\n":

    self.lista2.append("I001/141 TIEMPO TRUNCADO DEL DIA\n")

    self.lista2.append("Tiempo de captura del blanco en UTC HH:MM:SS "+self.lista1[x+1]+" \n")

x=x+1

return self.lista2

```

```

def proc02(self):

    listat=[]

    ffile=open(self.archivo,"r")

    lineas=ffile.readlines()

    ffile.close()

    yi=0

    yf=0

    x=0

    for i in lineas:

        if i==self.numero+"\n":

            yi=x

        if i==self.numero+'-----\n':

            yf=x

            x=x+1

    for i in range(yi,yf): # obtener un blanco especifico

        self.lista1.append(lineas[i])

    x=0

    for i in self.lista1:

        if i==self.numero+"I002/010\n":

            self.lista2.append("numero de mensaje = "+self.numero+"\n")

            self.lista2.append("I002/010 IDENTIFICACION DE LA FUENTE DE DATOS\n")

            self.lista2.append(self.lista1[x+1])

            self.lista2.append(self.lista1[x+2])

        elif i==self.numero+"I002/000\n":

            self.lista2.append("I002/000 TIPO DE MENSAJE\n")

            self.lista2.append(self.lista1[x+1])

            self.lista2.append(self.lista1[x+2])

        elif i==self.numero+"I002/030\n":

            self.lista2.append("I002/030 TIEMPO DE DIA\n")

            self.lista2.append(self.lista1[x+1])

        elif i==self.numero+"I002/041\n":

            self.lista2.append("I002/041 PERIODO DE ROTACION DE ANTENA\n")

```

```
        self.lista2.append(self.lista1[x+1])
    x=x+1
return self.lista2
```

```
def bn(self,lista):
    lista.remove('\n')
    x=0
    y=" "
    for i in range(len(lista)):
        if i==0:
            y=lista[i]
        else:
            y=y+lista[i]
        x=x+1
    return y
```


ANEXO C – PASOS PARA INSTALAR SOFTWARE DE LA APLICACION

Debido a que ADPADEC utiliza Python 2.6, con Tkinter, Tshark y Mergecap a continuación se detalla la instalación de cada uno de estos software.

NOTA: Se asume que la computadora donde se van a instalar estos software, esta conectada a internet y se esta usando Linux Ubuntu.

INSTALACION DE TSHARK

El usuario puede escribir en la terminal lo siguiente:

```
sudo apt-get install tshark
```

si lo anterior no funciona escribir tshark en el terminal y dar enter, para que se de información en el terminal, del comando para la instalación desde internet.

INSTALACION DE MERGECAP

El usuario puede escribir en la terminal lo siguiente:

```
sudo apt-get install -d wireshark-common
```

si lo anterior no funciona escribir mergecap en el terminal y dar enter, para que se de información en el terminal, del comando para la instalación desde internet.

INSTALACION DE PYTHON 2.6

En Linux Ubuntu Python 2.6 ya viene instalado. No es prudente cambiar la versión predeterminada de Python, es decir, lo que se obtiene cuando se escribe "python" en un terminal. Sin embargo, se puede tener varias versiones de python instaladas. El truco está en asegurarse de que el programa llamado "python" en el camino es la versión del sistema suministrado. Para ejecutar la instalación de Python 2.6, a continuación, se debe escribir python2.6 en el shell para ponerlo en marcha. Descargar el paquete y descomprimirlo y luego ejecutar:

```
./configure make sudo make install ls -l /usr/local/bin
```

El usuario debe ver un Python y un archivo de python2.6, ambos creados en el día que ejecutó make install, eliminar el archivo de python. Luego, cuando python se pone en marcha el sistema estándar de la versión de Python /usr/bin se llevará a cabo, y cuando python2.6 está dirigido a obtener su nuevo y brillante 2.6rc2 python. Python muestra la versión cuando se inicia un intérprete interactivo.

INSTALACION DE TKINTER PARA PYTHON 2.6

Escribir en el terminal: apt-get install python-tk. si lo anterior no funciona escribir python-tk en el terminal y dar enter, para que se de información en el terminal, del comando para la instalación desde internet.