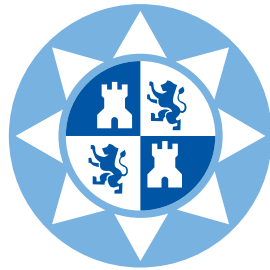


ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Universidad
Politécnica
de Cartagena

Trabajo de Fin de Grado

**Aplicación de ArcGIS al Desarrollo de
Herramientas para la Medida del Canal de
Radiocomunicaciones**



AUTOR: Carlos San Emeterio Villalaín
DIRECTORES: Leandro Juan Lácer y
Concepción García Pardo
Junio 2012



Autor	Carlos San Emeterio Villalaín
E-mail del Autor	carlos.s.emeterio@gmail.com
Director(es)	Leandro Juan Llácer y Concepción García Pardo
E-mail del Director	Leandro.juan@upct.es conchi.gpardo@upct.es
Codirector(es)	-----
Título del PFC	Aplicación de ArcGIS al Desarrollo de Herramientas para la Medida del Canal de Radiocomunicaciones
Descriptor(es)	MIMO, interiores, integración, ENA
Resumen	<p>Hoy en día ya se ha podido demostrar con éxito la funcionalidad de los sistemas multi-antena MIMO, los cuales aparecieron para intentar optimizar un espectro radio el cual cada vez se está quedando más escaso.</p> <p>Para desarrollar dispositivos MIMO es de vital importancia conocer el comportamiento del canal de propagación MIMO. Una forma de hacerlo es mediante la realización de medidas experimentales del canal y la posterior extracción de parámetros de interés.</p> <p>La misión principal de este trabajo es el estudio de las funciones de ArcGIS que permiten el desarrollo de herramientas software integradas en ArcGIS para la realización de medidas del canal de radiocomunicaciones para interiores.</p> <p>Una vez realizado el estudio inicial, el objetivo será enumerar y describir todas las funcionalidades desarrolladas para conseguir el fin de nuestro software.</p>
Titulación	Grado en Ingeniería Telemática
Intensificación	Sistemas y Redes de Telecomunicación
Departamento	Tecnologías de la Información y las Comunicaciones (TIC)
Fecha de Presentación	Junio - 2012

Índice

Capítulo I	2
1.1 OBJETIVOS	2
1.2 HERRAMIENTAS UTILIZADAS	4
1.2.1 Software	4
1.3 CONTENIDO.....	4
Capítulo II	5
2.1 LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG) (1).....	6
2.1.1 ¿Qué es un Sistema de Información Geográfica?.....	6
2.1.2 Tipos de datos en un SIG	6
2.2 DESCRIPCIÓN BREVE DE ARCGIS	9
2.2.1 ¿Qué es ArcGIS?	9
2.2.3 ArcMap	11
2.3 ANALIZADOR DE REDES ENA	15
2.4 OTROS ELEMENTOS DEL SISTEMA DE MEDIDAS	17
Capítulo III	19
3.1 REQUISITOS PREVIOS	19
3.2 ASPECTOS Y FUNCIONES FUNDAMENTALES EN ARCGIS.....	21
3.2.1 Introducción.....	21
3.2.2 Emplazamientos	21
3.2.2.1 Crear Emplazamiento	21
3.2.2.2 Copiar Emplazamiento	23
3.2.2.3 Consultar Emplazamiento	24
3.2.2.4 Eliminar Emplazamiento	25
3.2.3 Rutas	26
3.2.3.1 Crear Ruta	26
3.2.4 Anexo: Códigos	28
3.2.4.1 Carga del PropertySet y Workspace del Elemento a Tratar.....	28
3.2.4.2 Listar Todos los Emplazamientos Existentes	29
3.2.5 Conclusiones	29
3.3 PROCESO DE CAMPAÑA DE MEDIDAS.....	30
3.3.1 Introducción.....	30
3.3.2 Explicación del Proceso.....	30
3.3.2.1 Creación de la Campaña de Medidas	30
3.3.2.2 Selección de Ruta	31

3.3.2.3	Configuración.....	31
3.3.2.3.1	Configuración de Parámetros	32
3.3.2.3.2	Otros Parámetros.....	35
3.3.2.3.3	Calibración.....	36
3.3.2.4	Realización de las Medidas.....	47
Capítulo IV.....		51
4.1	CONCLUSIONES	51
4.2	LÍNEAS FUTURAS DE TRABAJO	52
Capítulo V.....		54

Agradecimientos

Quisiera agradecer a mis dos directores: Don Leandro Juan Llácer y Doña Concepción García Pardo toda la inestimable ayuda aportada durante estos duros meses de trabajo.

También quisiera dar las gracias a la UPCT en general, y la ETSIT en particular, la oportunidad que se me ha brindado de poder trabajar con soldadura en el laboratorio correspondiente prácticamente sin límite.

Capítulo I

Introducción y Objetivos



1.1 OBJETIVOS

Los Sistemas de Información Geográfica (SIG) son herramientas potentes en el mundo de la gestión de la información espacial y pueden resultar muy útiles para aplicaciones donde se necesite la referenciación geográfica.

Uno de los campos donde puede resultar esencial dicha referenciación es en la caracterización del canal de radiocomunicaciones mediante la realización de medidas radioeléctricas, ya que para efectuar las medidas se utilizan antena/s transmisora/s y receptoras que se van posicionando en el entorno según convenga.

Por este motivo, el objetivo de este trabajo es proporcionar una posible metodología para desarrollar un software que integre las características de los SIG (en nuestro caso ArcGIS), con la realización de medidas con un analizador de redes ENA:

Para ello, en este trabajo describimos una serie de funcionalidades programadas con las librerías de ArcGIS que consideramos fundamentales a la hora de poder establecer la asociación de las medidas con la referenciación geográfica, para posteriormente proponer un posible proceso de realización de las mismas, indicando al lector qué se debe hacer en cada paso, e incluyendo las partes más significativas del código fuente.

De esta forma, dejaremos construido un posible esqueleto que el desarrollador puede tomar como referencia a la hora de desarrollar su versión del software de integración.

1.2 HERRAMIENTAS UTILIZADAS

1.2.1 Software

- Los programas de los que partiremos para la realización de las funcionalidades de este trabajo son:
 - Sistema de Información Geográfica ESRI ArcGIS 9.1 [1], del cual se ha usado la aplicación ArcMap, sobre la que está basada nuestro trabajo.
 - Microsoft Visual Basic 6 [2], para el desarrollo y la programación de las funciones.
 - Sumatra PDF Reader [3], para leer toda la documentación proporcionada.

1.3 CONTENIDO

Este trabajo de final de grado está dividido en 5 capítulos fundamentales.

En el **capítulo segundo**, proporcionamos una introducción a los Sistemas de Información Geográfica (SIG), y después describimos las características del analizador de redes.

El **capítulo tercero** representa el núcleo de este trabajo. Comentamos en primer lugar qué funciones en ArcGIS consideramos necesarias para realización del posible software, y una vez establecida esa base explicamos un posible proceso para la realización de las medidas, comentando cada paso.

En el **capítulo cuarto**, se hace una breve exposición de las conclusiones derivadas de este trabajo y de cuáles podrían ser las líneas futuras de desarrollo en este campo.

Por último, en el **capítulo quinto**, se recogen las principales referencias bibliográficas empleadas en el desarrollo de este trabajo.

Capítulo II

Introducción a los SIG y al Analizador de Redes ENA



2.1 LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG) (1)

2.1.1 ¿Qué es un Sistema de Información Geográfica?

Un Sistema de Información Geográfica (SIG, o GIS en su acrónimo inglés), es una colección organizada de hardware, software, datos geográficos y personal, diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión [5].

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

El Sistema de Información Geográfica separa la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Las principales funciones de un Sistema de Información Geográfica son [6]:

- Entrada de datos.
- Output.
- Manipulación de los datos.
- Análisis

(1): Una parte considerable del texto y la totalidad de las figuras incluidas en este capítulo están sacadas del PFC *Desarrollo e implementación de una aplicación para la gestión de una red de sensores conectada remotamente vía el sistema GSM-GPRS*, Concepción García Pardo, 2007. [4]

2.1.2 Tipos de datos en un SIG

Los Sistemas de Información Geográfica manejan fundamentalmente dos tipos de información: la información espacial o geográfica y la información descriptiva. A su vez, la información geográfica se puede almacenar de dos maneras: mediante formato raster o formato vectorial.

La información espacial o geográfica, describe la localización y la forma de diversas características geográficas (orografía, vías de comunicación, ríos, etc.) Estos datos se pueden almacenar en dos formatos: raster o vectorial:

- El formato raster o de retícula se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas iguales donde cada una de ellas representa un único valor. Cuanto mayor sean las dimensiones de las celdas (la resolución), más precisión y detalle se va a perder en la representación del espacio geográfico.
- En el formato de datos vectorial, el objetivo se centra en la precisión de localización de los elementos sobre el espacio. Para modelar digitalmente las entidades del mundo real se utilizan tres objetos espaciales: el punto, la línea y el polígono.

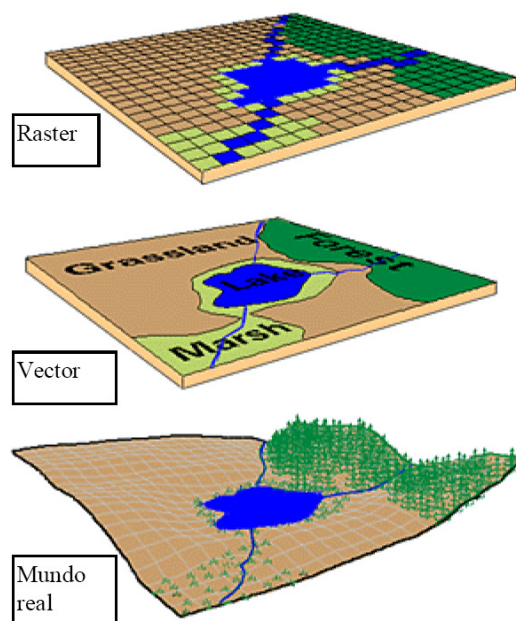


Figura 2.1. La geografía real a modelar (abajo); la información geográfica en formato vectorial (centro), y en formato raster (arriba).

La información descriptiva es aquella que suministra información geográfica (tipo de vía de comunicación, tipo de río, tipo de montaña...), o dicho de otra forma, la describe. Estos datos se almacenan mediante tablas de atributos, las cuales relacionan la información geográfica con la descriptiva mediante un identificador.

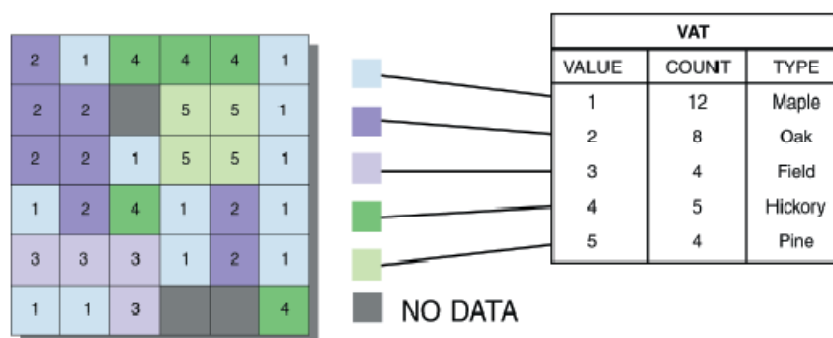


Figura 2.2. Tabla de atributos (información descriptiva) asociada a un Raster [7]

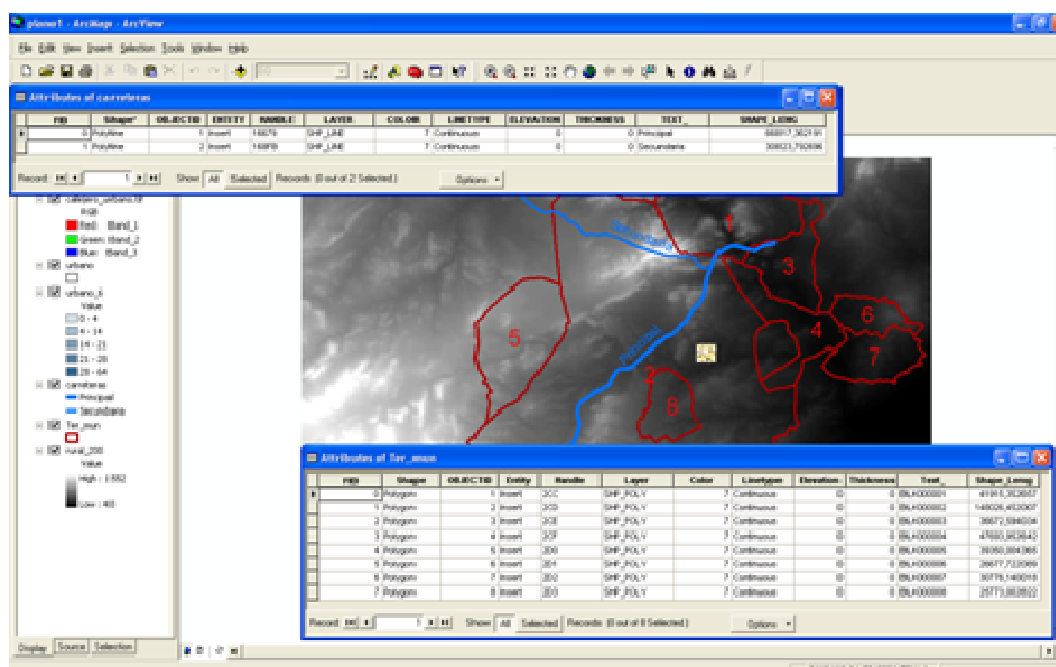


Figura 2.3. Tablas de atributos asociadas a la característica vectorial “Ter_mun” o “carreteras”.

En un SIG, podemos trabajar con diversas capas simultáneamente, describiendo cada una de ellas información geográfica asociada a tablas de atributos:

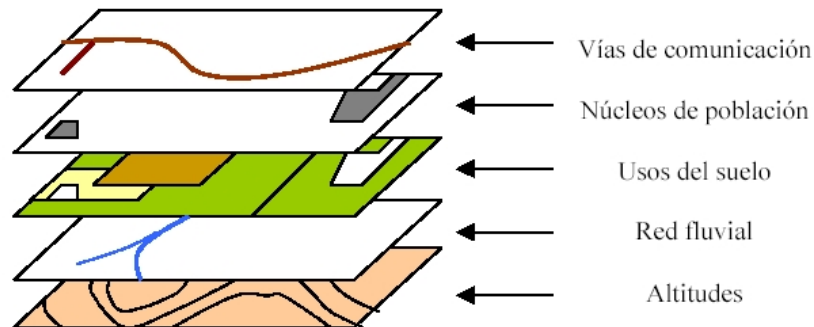


Figura 2.4. Diversas capas que representan diferentes características geográficas [8]

2.2 DESCRIPCIÓN BREVE DE ARCGIS

2.2.1 ¿Qué es ArcGIS?

ArcGis es un producto de ESRI que proporciona un marco general para implementar Sistemas de Información Geográfica para uno o varios usuarios en ordenadores, servidores, Internet, etc.

ArcGIS es una colección integrada de software GIS para construir un Sistema de Información Geográfica complejo. Se compone de cuatro partes: [9]

- *ArcGIS Desktop*: una suite integrada para aplicaciones GIS profesionales.
- *ArcGIS Engine*: componentes integrados (embebidos) para aplicaciones personalizadas.
- *Server GIS*: como su propio nombre indica, es un servidor de GIS.
- *Mobile GIS*: aplicaciones anteriores a adaptadas a TabletPC.

ArcGIS está basado en ArcObjects™, una librería modular común que comparten todos los componentes software del GIS. Incluye una gran gama de componentes programables, desde objetos como geometrías individuales, hasta

mapas enteros que interactúan con documentos ArcMap. Los desarrolladores de GIS tienen a su disposición la librería ArcObjects para crear plataformas alternativas de software GIS, basándose en los cuatro productos anteriormente comentados.

En las figuras 2.5 y 2.6 se pueden ver más detalladamente todos estos productos.



Figura 2.5. ArcObjects es una librería común para los cuatro componentes ArcGIS [10]

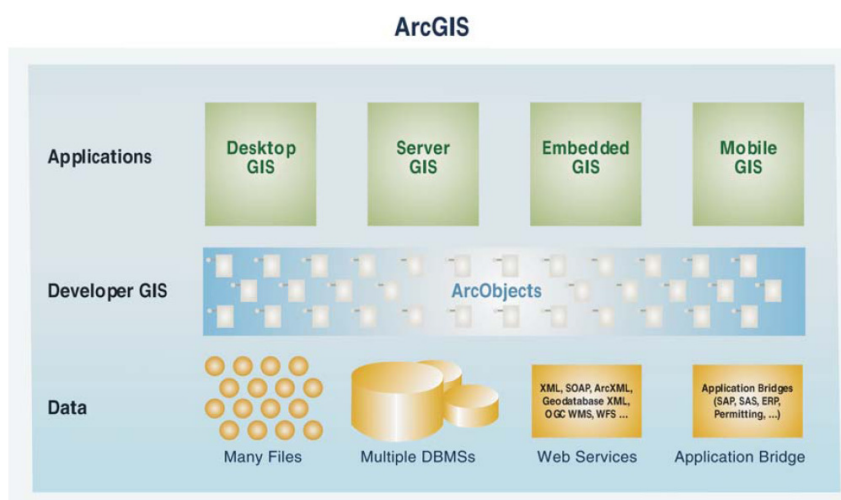


Figura 2.6. ArcObjects es empleado por desarrolladores GIS en conjunto con los otros cuatro productos [11]

En el desarrollo de este proyecto se ha trabajado exclusivamente con ArcMap, la cual es una parte del módulo ArcGIS Desktop, por lo cual vamos a describir únicamente ArcMap. Si el lector desea conocer más a fondo ArcGIS y sus productos puede consultar [11].

2.2.3 ArcMap

ArcMap es la aplicación central de ArcGis Desktop para trabajos basados en mapas, incluyendo cartografía, análisis de mapas, y edición. La interfaz gráfica de este software se muestra a continuación:

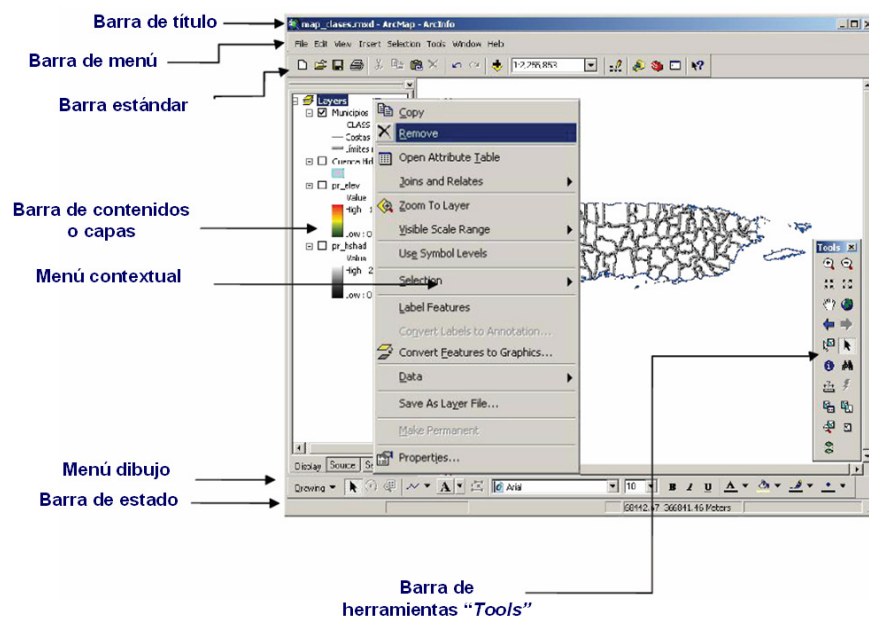


Figura 2.7. Interfaz gráfica de ArcMap.

El software ofrece dos tipos de vistas de mapas: *Data View* y *Layout View*. La primera de ellas muestra, hace cambios, introduce datos, hace búsquedas geográficas en las tablas de atributos. Por su parte, la vista *Layout View* es la interfaz gráfica que sirve para producir mapas y gráficas. En las figuras 2.13 y 2.14 se muestran respectivamente estas dos interfaces.

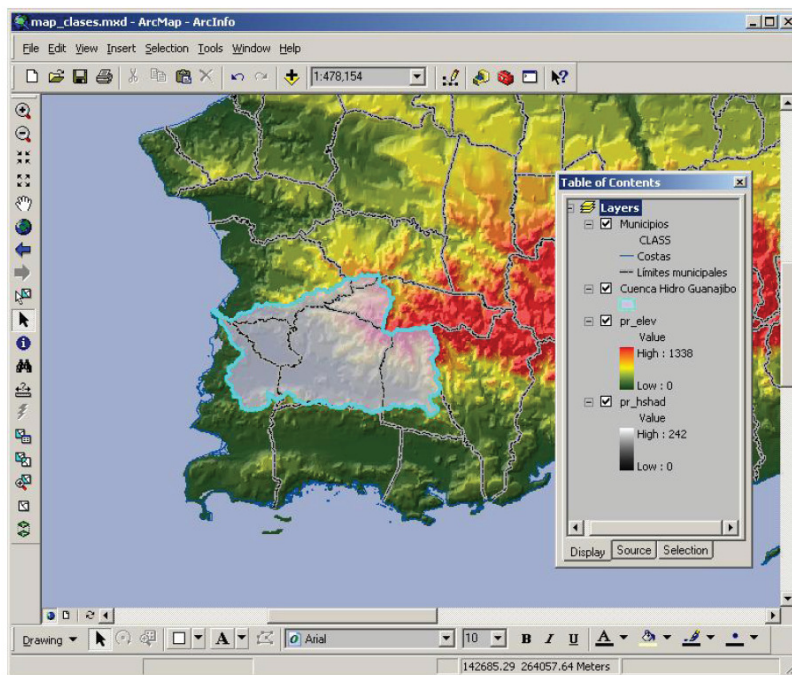


Figura 2.8. Interfaz Data View

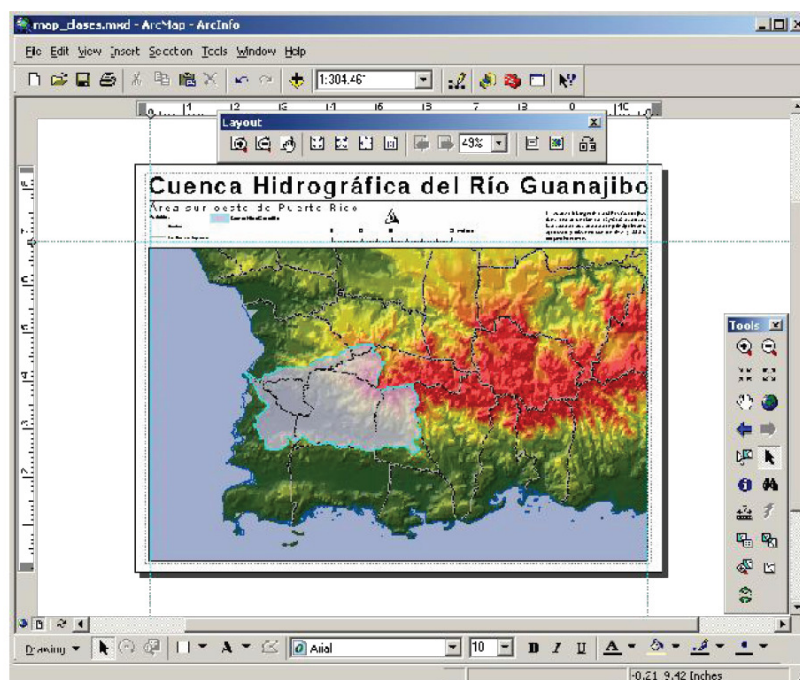


Figura 2.9. Interfaz Layout View

ArcMap presenta varios tipos de información a la vez acerca de un área. Es decir, un mapa puede contener capas y al mismo tiempo, en cada capa se emplean símbolos para dibujar, los cuales se denominan *features*.

ArcMap proporciona varias formas de interactuar con mapas:

- *Exploración*: visualizar y explorar mapas así como la relación entre las diferentes *features* en el mapa.
- *Análisis*: crear nueva información añadiendo capas al mapa.
- *Presentación de resultados*: permite imprimir mapas, añadirlos a documentos, etc. Es decir, manipulación de mapas para su visualización en distintos formatos.
- *Personalización (Custom)*: Se pueden crear mapas que posean exactamente las herramientas que el usuario necesita, con el objetivo de hacer más rápido el trabajo. Se puede personalizar la interfaz de ArcMap añadiendo herramientas a las barras ya existentes o barras de herramientas propias. Con este objetivo, ArcMap incluye un editor de Visual Basic® for Applications, aunque también se pueden programar con un editor de Visual Basic independiente y luego añadirlas a ArcMap.

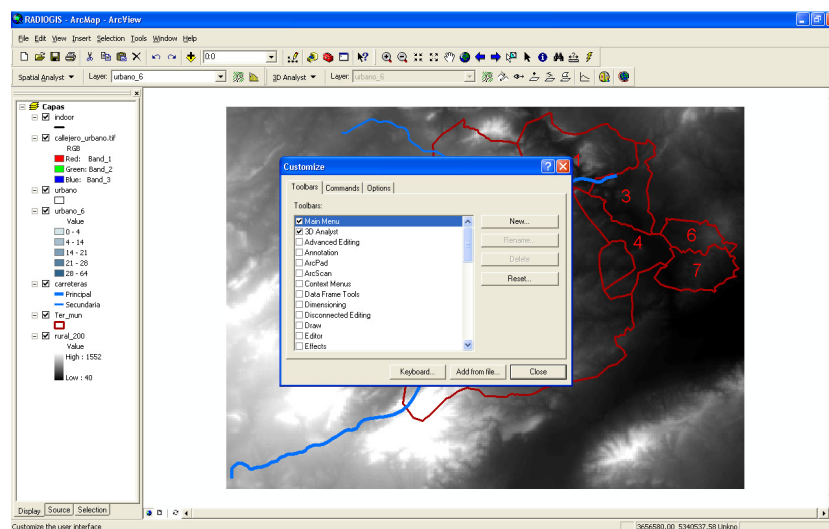


Figura 2.10. Adición de nuevas herramientas a ArcMap.

- *Programación*: Se pueden construir interfaces completamente nuevos para interactuar con mapas y crear nuevos tipos de *features*. Para ello se emplean lenguajes de programación COM (Microsoft Component Object Model).

Para más información acerca del funcionamiento de ArcMap diríjase al manual de ArcGIS *Getting Started With ArcGis* [10] y para conocer más acerca de cómo personalizar el entorno ArcMap con herramientas propias puede consultar los manuales *Exploring ArcObjects: Vol.1-Application and Cartography* [11] y *Exploring ArcObjects: Vol.2 – Graphics Data Management* [12].

2.3 ANALIZADOR DE REDES ENA

Existen en el mercado una gran multitud de analizadores de redes de diferentes características. En nuestro caso, vamos a utilizar el popular ENA E5071B, de Agilent Technologies, el cual proporciona facilidad de manejo y buenos resultados.

El analizador de redes (ENA) E5071B, fabricado por Agilent Technologies [14], es un instrumento que sirve para caracterizar con precisión los dispositivos de RF en un amplio rango de frecuencias seleccionable por el usuario y en distintos formatos.

El ENA satisface la necesidad de los ingenieros para caracterizar el comportamiento de componentes pasivos y activos con velocidad, exactitud y en un amplio espectro de frecuencias. Lo utilizaremos para medir la magnitud y la fase de la señal recibida. A partir de dichos datos y utilizando programas auxiliares podremos generar la matriz H (matriz que contiene todos los impulsos de los transmisores hacia los receptores) que caracteriza al sistema MIMO.



Figura 2.11. El analizador E5071B, en sus versiones de 2,3 y 4 puertos.

La serie E5071B lleva utiliza como sistema operativo Microsoft Windows 2000 Professional, el cual, además de proporcionar una seguridad y estabilidad superiores, también está adaptado a entornos de red, lo cual nos permite que estos instrumentos se conecten a la red como cualquier otro hardware de oficina, industrial, etc.

Para aumentar la funcionalidad del ENA, utilizamos un sistema multipuerto (E5091A), el cual nos añade varios puertos adicionales para casos en los que se necesita un número de antenas entre 2 y 4.



Figura 2.12. Multipuerto E5091A.

Finalmente el conjunto de ENA+Multipuerto tiene este aspecto:





Figura 2.13. ENA 5071B y Multipuerto E5091A [15].

En este montaje, el puerto 1 del ENA emite la señal de transmisión, la cual es conmutada en el conmutador entre las antenas transmisoras que haya. Las antenas receptoras van conectadas directamente al multipuerto.

2.4 OTROS ELEMENTOS DEL SISTEMA DE MEDIDAS

A continuación proporcionamos una tabla con una posible lista de elementos que conformarían un teórico sistema de medidas MIMO. Los modelos de cada dispositivo pueden variar según las necesidades de cada caso.

Nombre	Descripción	Fotografía
Conmutador Agilent 87406B	Conmuta la señal transmisora entre todas las antenas que actúen de transmisoras.	
Controlador 11713A	Dirige el conmutador dando las órdenes para que éste conmute entre las distintas salidas disponibles.	
Ordenador Portátil	Sincroniza y da órdenes a todos los elementos del sistema. Aquí reside el software utilizado.	
Cable GPIB-USB	Asocia el controlador con el ordenador portátil.	

<p>Cable RJ45 Directo</p>	<p>Conecta el ENA al ordenador portátil.</p>	
<p>Transmisor y Receptor de fibra óptica ORTEL 3540</p>	<p>Transmite y recibe señales de RF en grandes rangos de frecuencia.</p>	
<p>Cable de fibra óptica</p>	<p>Interconecta el ENA con el equipo de transmisión.</p>	

Por último, incluimos un posible esquema de montaje a la hora de realizar una campaña de medidas:

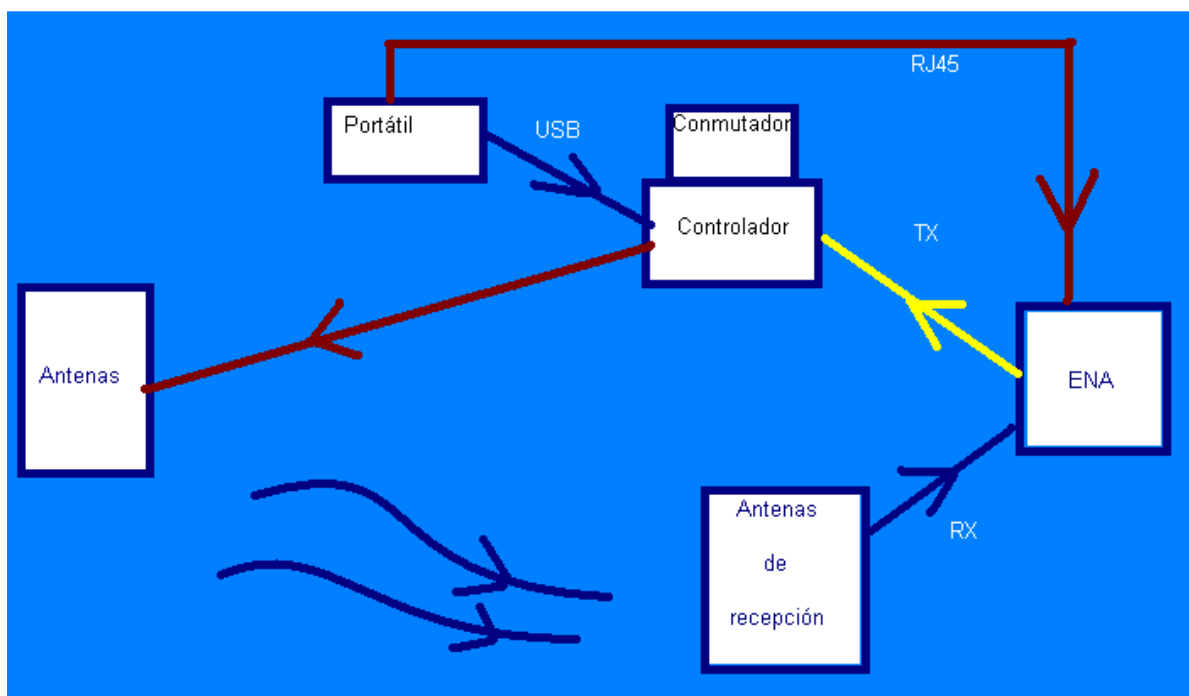


Figura 2.14 Esquema de Montaje

Capítulo III

Propuesta para Proceso para Software de Medidas en ArcGIS



3.1 REQUISITOS PREVIOS

Antes de proceder al desarrollo de nuestras funciones, debemos tener en cuenta un par de pasos que se deben llevar a cabo para poder tener todas las librerías y software en general a nuestra disposición.

Estos pasos son los siguientes:

- Instalar **Visual Basic** (en nuestro caso 6.0), o en general cualquier IDE que soporte la programación en VB.
- Instalar **ArcGIS**; es imprescindible su instalación porque de esta forma obtendremos toda la serie de librerías que nos proporciona ESRI, las cuales son vitales para la realización de este trabajo.
- Una vez instalados ambos software, hay que crear un proyecto en Visual Basic, para posteriormente añadir las librerías de ESRI al mismo. Esto se hace pulsando en la pestaña *Proyecto*, y en la opción de *Referencias*. De todas las posibles librerías que se muestran, hay que seleccionar o bien todas las de ESRI, o bien únicamente aquellas que se vayan a considerar necesarias.

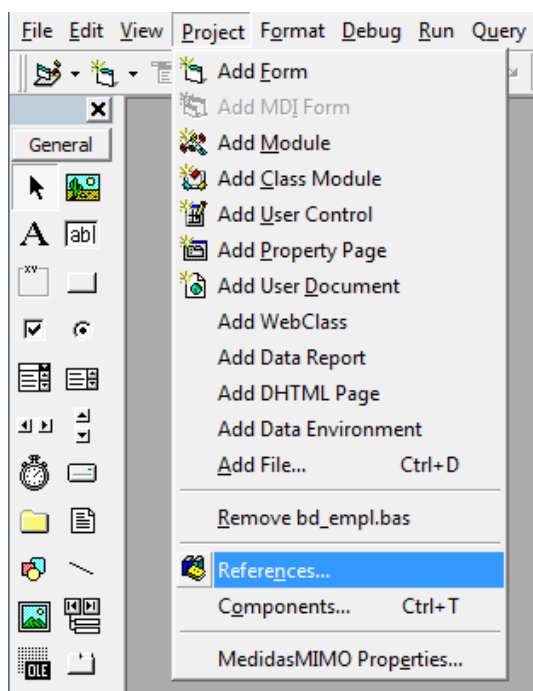


Figura 3.1. Referencias

3.2 ASPECTOS Y FUNCIONES FUNDAMENTALES EN ARCGIS

3.2.1 Introducción

El objetivo principal de este trabajo es el proponer un posible método para generar un software que integre la realización de campañas de medidas con el analizador de redes (ENA) con las posibilidades de referenciación geográfica del ArcGIS.

Dicha referenciación consiste principalmente en la creación en ArcGIS de emplazamientos y rutas, para posteriormente asociar las medidas que se vayan realizando con dichos elementos geográficos. Por lo tanto, antes proceder a desarrollar el método de producción de medidas, es necesario tener declarado una parte del software que sea capaz de gestionar tanto los emplazamientos como las rutas, a nivel de creación, copia, etc. Por ello, en este apartado vamos a dar una orientación sobre cómo se puede llevar a cabo la gestión de esta base de datos, resaltando las partes más significativas de los códigos fuente.

Es necesario destacar que las funciones que vienen a continuación están realizadas casi íntegramente con librerías proporcionadas con ArcGIS, lo cual hace indispensable cumplir el apartado 3.1.

3.2.2 Emplazamientos

3.2.2.1 Crear Emplazamiento

Crear un emplazamiento es sencillo en el hecho de que sólo tiene 3 parámetros de entrada: Coordenada X, coordenada Y y nombre. Estos 3 parámetros podemos obtenerlos fácilmente creando un formulario y recogiendo los mediante 3 *TextBox*, o bien para el caso de las coordenadas, se puede desarrollar una pequeña herramienta que nos permita seleccionar con el ratón directamente sobre el plano el punto que queramos. Las coordenadas X e Y estarán definidas en metros.

Una vez que el usuario ordene la creación del emplazamiento, por ejemplo mediante un *Button*, el código más esencial a tener en cuenta es el siguiente:

```
-----
'Añadir código apartado 3.2.4.1'
-----
```

```
Dim pSpatRef As ISpatialReference
Set pSpatRef = New UnknownCoordinateSystem

Dim pFeatcls As IFeatureClass
Set pFeatcls = pAccessFeatWS.CreateFeatureClass(FormTest.nombre.Text, pFlds,
Nothing, Nothing, esriFTSimple, "Shape", "")

Dim emplaza As IFeature
Set emplaza = New Feature

Dim emp As IPoint
Set emp = New Point

Dim emplaz As String
Dim x_coord As Double
Dim y_coord As Double

emplaz = FormTest.nombre.Text
x_coord = FormTest.x.Text
y_coord = FormTest.y.Text

emp.x = x_coord
emp.y = y_coord

Set emplaza = pFeatcls.CreateFeature

Set emplaza.Shape = emp
emplaza.value(2) = x_coord
emplaza.value(3) = y_coord
emplaza.value(4) = emplaz
emplaza.Store
```

Podemos observar cómo de esta forma creamos una *Feature* (característica) en la base de datos del ArcGIS, y para definirla creamos un punto con las coordenadas X e Y obtenidas de nuestro formulario (*FormTest*), el cual pasa a ser la *Shape* (forma) de nuestra *Feature*. Adicionalmente, para culminar la construcción, almacenamos como valores (*value*) las dos coordenadas y el nombre del punto.

Finalmente con la directriz *Store*, culminamos el proceso.

3.2.2.2 Copiar Emplazamiento

El proceso de copiar un emplazamiento existente se limita básicamente a elegir el emplazamiento a copiar, obtener las características de su *Feature* asociado, y almacenar estos datos en otro *Feature* nuevo, que será la copia a producir.

En este caso la opción más atractiva es tener un formulario que nos liste mediante un *ListBox* los emplazamientos existentes, y que nos permita elegir el que queramos. También tendríamos un *TextBox* para especificar el nombre de la copia.

Una vez que el usuario mandara la orden de Copiar, el cuerpo básico del código sería el siguiente:

```
-----  
'Añadir código apartado 3.2.4.1'  
-----
```

```
Dim pfeatcls2 As IFeatureClass  
Dim pfeatcls As IFeatureClass  
Set pfeatcls = pAccessFeatWS.OpenFeatureClass(FormTest2.Lista_emp.Text)  
  
Set dataset = pfeatcls  
dataset.Copy FormTest2.nuevo.Text, paccessws2  
  
Set pfeatcls = pAccessFeatWS2.OpenFeatureClass(FormTest2.nuevo.Text)  
  
Dim feat As IFeature  
Set feat = pfeatcls.GetFeature(0)  
  
feat.value(4) = FormTest2.nuevo.Text  
feat.Store
```

Obsérvese cómo lo que hacemos aquí es abrir la *Feature* con el nombre que hemos seleccionado de la *ListBox* ('lista_emp'), le sacamos sus datos y se los asociamos a nuestra *Feature*, añadiéndole como parámetro adicional el nombre de la copia.

3.2.2.3 Consultar Emplazamiento

En este caso el proceso es simple: Crear un formulario que recoja los emplazamientos que ya existen, nos permita elegir uno, y al aceptar nos devuelva otra ventana con sus datos. Un posible código para mostrar en un *ListBox* todos los emplazamientos existentes se puede encontrar en el *Anexo* de este trabajo.

El código fundamental de esta opción sería el siguiente:

```
-----  
'Añadir código apartado 3.2.4.1'  
-----
```

```
Dim feat As IFeature  
Set feat = New Feature  
  
Set feat = pfeatcls.GetFeature(0)  
coord_x = feat.value(2)  
coord_y = feat.value(3)  
nombre = feat.value(4)  
  
Dim coords As IPoint  
Set coords = New Point  
  
coords.x = coord_x  
coords.y = coord_y  
  
dibuja.AddLabel nombre, coords  
datos_emp.Caption = "Emplazamiento - " + nombre  
datos_emp.nom.Text = nombre  
datos_emp.ubicacion.Text = App.path + "\datos\emplazamientos\  
datos_emp.x.Text = Round(coord_x, 0)  
datos_emp.y.Text = Round(coord_y, 0)  
abrir_empl.Hide  
  
datos_emp.Show
```

Lo que estamos haciendo aquí es obtener los datos de la *Feature* correspondiente al emplazamiento que hemos elegido (representado por la variable 'empl'). Una vez obtenidos los datos, los representamos en un formulario aparte ('datos_emp'). Estos datos incluyen las coordenadas, el nombre, y el lugar donde está almacenado.

3.2.2.4 Eliminar Emplazamiento

Un método sencillo para llevar a cabo esta opción sería mostrar de nuevo la lista de emplazamientos creados al igual que en la opción *Consultar*, salvo que en este caso en vez de mostrar los datos vamos a eliminar el emplazamiento del sistema.

El esqueleto del código sería tal que así:

```
nombre = emp_elim.Lista_emp.Text
emp_elim.Lista_emp.RemoveItem (emp_elim.Lista_emp.ListIndex)

Dim dataset As IDataset

-----
'Añadir código apartado 3.2.4.1'
-----

Dim pfeatcls As IFeatureClass
Set pfeatcls = pAccessFeatWS.OpenFeatureClass(nombre)

Set dataset = pfeatcls
dataset.Delete
```

Realmente la parte más significativa del código son las dos últimas líneas, que especifican que el set de datos asociado al emplazamiento sea eliminado, de esta forma se borrará cualquier dato de este emplazamiento en nuestra base de datos.

3.2.3 Rutas

Para nosotros, una ruta consiste en una agrupación de emplazamientos que recorreremos de forma secuencial a la hora de realizar las medidas. La idea es que las medidas que se vayan haciendo se asocien con cada punto receptor de la ruta.

Es muy importante destacar que en nuestra ruta habrá que distinguir entre emplazamiento que actúa como transmisor, y receptores. Así como solo podrá haber 1 emplazamiento transmisor, podrá haber un número indefinido de receptores.

3.2.3.1 Crear Ruta

Este aspecto quizá sea el más delicado de todas las funcionalidades de ArcGIS que estamos nombrando en este capítulo. Esto se debe a que debemos crear un método de tal forma que nos aseguremos muy bien de que sólo hay 1 transmisor, y de que ese transmisor sea el primero que se guarde en la ruta.

Una posible solución es crear un formulario que muestre todos los emplazamientos, y crear dos *ListBox* modificables, uno para transmisor y otro para receptor, pero siempre asegurándose de que el *ListBox* del transmisor solo pueda tener una entrada.

El núcleo del código de esta función sería el siguiente:

```

Set pfeatcls2 = pAccessFeatWS.OpenFeatureClass(FORMTEST3.listaT.list(0))
Set pfeatlayer = New FeatureLayer
Set pfeatlayer.FeatureClass = pfeatcls2
pfeatlayer.name = pfeatcls2.AliasName

Set pfirsttable = pfeatlayer
inputarray.Add pfeatlayer

For i = 0 To FORMTEST3.ListaR.ListCount - 1

    Set pfeatcls2 = pAccessFeatWS.OpenFeatureClass(FORMTEST3.ListaR.list(i))
    Set pfeatlayer = New FeatureLayer
    Set pfeatlayer.FeatureClass = pfeatcls2
    pfeatlayer.name = pfeatcls2.AliasName

    Set pfirsttable = pfeatlayer
  
```



```
        inputarray.Add pfeatlayer

Next

Dim pfeatclassname As IFeatureClassName
Set pfeatclassname = New FeatureClassName

With pfeatclassname

    .FeatureType = esriFTSimple
    .ShapeFieldName = "Shape"
    .ShapeType = pfeatcls2.ShapeType

End With

Dim pPropSet2 As IPropertySet
Set pPropSet2 = New PropertySet
pPropSet2.SetProperty "Database", path

Dim pworkspace2 As IWorkspace
Set pworkspace2 = pAccessFact.Open(pPropSet2, 0)

Dim pworkspace2name As IWorkspaceName
Set pworkspace2name = New WorkspaceName

With pworkspace2name

    .ConnectionProperties = pPropSet2
    .WorkspaceFactoryProgID = "esriCore.ShapefileWorkspacefactory.1"

End With

Dim pdatasetname As IDatasetName
Set pdatasetname = pfeatclassname
pdatasetname.name = FORMTEST3.nombre.Text
Set pdatasetname.WorkspaceName = pworkspace2name

Dim pbgp As IBasicGeoprocessor
Set pbgp = New BasicGeoprocessor

Dim poutputfeatclass As IFeatureClass
Set poutputfeatclass = pbgp.Merge(inputarray, inputarray.Element(0),
pfeatclassname)
```

Obsérvese cómo primero tratamos al emplazamiento que actúa como transmisor (el contenido en el *ListBox* 'ListaT'), para después tratar a los receptores mediante un bucle. De esta forma nos aseguramos de que el primer emplazamiento en la ruta sea el transmisor, evitándonos posibles conflictos.

Una vez que hemos gestionado todos los emplazamientos, creamos el *Dataset* que va a contener esta ruta, asociándole el nombre que hayamos elegido en el formulario previo.

3.2.4 Anexo: Códigos

En este apartado incluimos códigos que resultan interesantes, bien por ser comunes a todas las opciones, o bien porque pueden resultar necesario en alguna de ellas.

3.2.4.1 Carga del PropertySet y Workspace del Elemento a Tratar

Este código se deberá añadir con la modificación necesaria en 'path' en aquellos lugares donde esté escrito la siguiente advertencia:

```
-----  
'Añadir código apartado 3.2.4.1'  
-----
```

```
Dim pPropSet As IPropertySet  
Set pPropSet = New PropertySet  
pPropSet.SetProperty "Database", path
```

```
'path representa el directorio raíz donde estén almacenados todos los  
emplazamientos'
```

```
Dim pAccessFact As IWorkspaceFactory  
Set pAccessFact = New ShapefileWorkspaceFactory
```

```
Dim paccessws As IWorkspace  
Set paccessws = pAccessFact.Open(pPropSet, 0)
```

```
Dim pAccessFeatWS As IFeatureWorkspace  
Set pAccessFeatWS = paccessws
```

3.2.4.2 Listar Todos los Emplazamientos Existentes

```

Dim path As String
    path = App.path + "\datos\emplazamientos\" & "Ejemplo"

    Dim pPropSet As IPropertySet
    Set pPropSet = New PropertySet
    pPropSet.SetProperty "Database", path

    Dim pAccessFact As IWorkspaceFactory
    Set pAccessFact = New ShapefileWorkspaceFactory

    Dim paccessws As IWorkspace
    Set paccessws = pAccessFact.Open(pPropSet, 0)

    Dim enume As IEnumDataset
    Set enume = paccessws.Datasets(esriDTFeatureClass)

    Dim dta As IDataset
    enume.Reset

    Set dta = enume.Next

    Do While Not dta Is Nothing

        abrir_empl.listaEMP.AddItem dta.name
        Set dta = enume.Next
        'Suponiendo que tuvieramos un formulario llamado 'abrir_empl', y
        dentro de él un ListBox llamado 'listaEMP'
    Loop

```

3.2.5 Conclusiones

De esta forma, ya tendríamos programadas las funcionalidades necesarias para poder pasar al proceso de toma de medidas. Si el desarrollador lo considera interesante, también se podrían añadir funciones opcionales como la consulta y copia de rutas, etc.

El objetivo ahora es asociar esta base de datos de emplazamientos y rutas que hemos generado con el proceso de medición.

3.3 PROCESO DE CAMPAÑA DE MEDIDAS

3.3.1 Introducción

Una vez que hemos comentado qué características básicas obtenidas con las librerías de ArcGIS necesitamos para poder realizar nuestro trabajo, nos adentramos a explicar el proceso de realización de medidas en sí.

A la hora de realizar medidas, el analizador de redes ENA debe ser previamente configurado mediante una serie de parámetros a indicar de forma secuencial y en cierto orden, lo cual hace lógico que enfoquemos esta parte del trabajo desde ese mismo punto secuencial. De esta forma simplificaremos el proceso y eliminaremos en buena medida la posibilidad de errores por culpa de mala configuración.

3.3.2 Explicación del Proceso

A continuación vamos a pasar a proporcionar una orientación sobre cómo enfocar mediante software el proceso de realización de medidas con un ENA. El desarrollador es libre de establecer la jerarquía y enfoque que se desee, pero en nuestro caso como ya hemos comentado partiremos del orden real de configuración de un ENA (especificación de frecuencias y nº de antenas, calibración, etc.)

3.3.2.1 Creación de la Campaña de Medidas

Para iniciarnos en el proceso, es interesante considerar a la campaña de medidas como un ente en sí, el cual guardaremos en una carpeta independiente para cada campaña. En dicha carpeta se guardarán las medidas que realicemos, se copiará la ruta a elegir como posteriormente veremos, y por último se incluirá un fichero '*info.txt*' a rellenar por el usuario que servirá para describir la campaña en cuestión.

Por lo tanto, una buena forma de iniciar el proceso es crear un formulario que recoja el nombre de la campaña (el cual será realmente el nombre de la carpeta que incluye todo lo arriba comentado), un *TextBox* para introducir el texto que irá guardado en *info.txt*, y por último de manera opcional, un *ListBox* que recoja los nombres de las campañas de medidas que ya existan.

3.3.2.2 Selección de Ruta

Una vez creada la campaña de medidas, el siguiente paso es ni más ni menos que especificar qué ruta se va a tomar como base. El caso ideal es tener en ArcMap la topología del lugar cargada, y en ella tener representados los puntos que forman la ruta.

El posible formulario en este caso se reduce a un *ListBox* que muestre las rutas disponibles. Cuando el usuario elige una, y pulsa *Siguiente* (*Button* que consideramos presente en todos los formularios, al igual que *Atrás*), el software copiará los contenidos de la ruta a la carpeta de la campaña de medidas. El código para el copiado de ruta se puede consultar en el Anexo de este capítulo.

3.3.2.3 Configuración

Llegamos a una de las partes más delicadas y la más compleja de nuestro hipotético software: La configuración del ENA, la cual nosotros la dividiremos en 3 fases:

- Configuración de Parámetros
- Otros Parámetros
- Calibración del ENA

En este caso es muy importante seguir el orden jerárquico de la configuración, ya que por ejemplo si calibramos el ENA antes de establecer sus parámetros, esta calibración no nos servirá de nada. Debido a esto, nos ha resultado interesante crear un formulario con 3 *Buttons*, uno por cada fase, pero con la particularidad de que inicialmente sólo esté disponible el botón de Configuración de Parámetros. Al realizar esta parte, se activará el botón de Otros Parámetros, y finalmente el de Calibración.

De esta forma evitamos completamente la posibilidad de fallos en el proceso.

Para activar o desactivar un *Button*, el código a utilizar sería el siguiente:

```
ButtonXX.Enable = false '(o true, según convenga)'
```

3.3.2.3.1 Configuración de Parámetros

En este apartado vamos a configurar los parámetros radioeléctricos más importantes del ENA. En este caso nos vamos a apoyar de un posible formulario para que resulte más sencilla la explicación:

Figura 3.2. Parámetros(2)

A continuación vamos a explicar cada parámetro brevemente, e incluiremos el comando en Visual Basic que manda la orden de configuración al ENA. Es vital destacar que estos comandos que incluimos son íntegramente realizados con las librerías de Agilent para nuestro ENA E5071B, por lo que es vital tenerlas instaladas en el PC para evitar fallos de dependencias.

- Antenas Tx: Establece el número de antenas transmisoras (1-4).
- Antenas Rx: Establece el número de antenas receptoras (1-4).
 - Algoritmo para calcular el N^o de ventanas en el ENA:

```
Dim aTx, aRx As Long
antTx$ = Parametros.TextBox1.Text
antRx$ = Parametros.TextBox2.Text
aTx = antTx$
aRx = antRx$
ventH = aRx \ 2 + (aRx Mod 2)
ventV = aTx
ventT = ventH * ventV
If (aRx Mod 2) = 0 Then
    traz = True
Else
    traz = False End If
```

(2): Este formulario es una adaptación del formulario propuesto por Rafael Antonio Lucas Ávila en su PFC Software para la automatización de medidas de un Sistema MIMO de banda ancha (2005)[16]

- Frecuencia Inicial: Frecuencia desde la que se empieza a medir (en Hz).
- Frecuencia Final: Frecuencia en la que se para de medir (en Hz).

- Algoritmo para indicar estos datos para cada traza en el ENA:

```
For R = 1 To ventT
    RR$ = R
    y = SendCommand_ENA(":SENS"+RR$+":FREQ:STAR"+StartVal$)
    y = SendCommand_ENA("SENS"+RR$+":FREQ:STOP"+StopVal$)
Next R
```

- Número de Puntos: Establece la precisión de la medida.

- Algoritmo para indicar este dato para cada traza en el ENA:

```
For R = 1 To ventT
    RR$ = R
    y = SendCommand_ENA("SENS" + RR$ + ":SWE:POIN " + Nop$)
Next R
```

- IFBW: Frecuencia intermedia de ancho de banda.

- Algoritmo para indicar este dato para cada traza en el ENA:

```
For R = 1 To ventT
    RR$ = R
    y = SendCommand_ENA("SENS" + RR$ + ":BAND " + FIBW$)
Next R
```

➤ Average.

- Algoritmo para indicar este dato para cada traza en el ENA:

```

For R = 1 To ventT
    RR$ = R
    If AV = 0 Then
        y = SendCommand_ENA("SENS" + RR$ + ":AVER OFF")
    Else
        y = SendCommand_ENA("SENS" + RR$ + ":AVER ON")
        y = SendCommand_ENA("SENS" + RR$ + ":AVER:COUN"+Aver$)
        y = SendCommand_ENA("SENS" + RR$ + ":AVER:CLE")
    End If
Next R

```

➤ Formato:

- Algoritmo para indicar este dato para cada traza en el ENA:

```

elecform = Trim(Left(Parametros.ComboBox1.Text, 2))
'Hay 16 posibles formatos, consultar manual [17] de Agilent sobre
cómo obtener el formato en cuestión, y asociarlo a la variable
Formato$'

For R = 1 To ventT
    RR$ = R
    y = SendCommand_ENA(":CALC" + RR$ + ":FORM " + Formato$)
Next R

```

Cabe destacar que antes de enviar ningún comando al ENA, es necesario establecer la comunicación con el mismo, y para ello hay que hacer lo siguiente:

```

Call StartIt
Call get_hostnameENA 'Obtiene la IP del ENA mediante el formulario de
                    Otros Parámetros'
y = OpenSocket_ENA(HostnameENA$, ScpiPort)

```

Y una vez que terminamos con las órdenes, se debe cerrar la conexión, de esta manera:

```

Call CloseConnection_ENA
Call EndIt

```


3.3.2.3.2 Otros Parámetros

Una vez configurados los aspectos radioeléctricos del ENA, se nos activaría la opción de *Otros Parámetros*. En esta sección configuramos aspectos más generales del ENA, como indicar su dirección IP, su dirección GPIB, el directorio destino de los datos etc.

De nuevo, nos apoyamos en un formulario para simplificar la explicación:

Figura 3.3. Otros(2)

➤ Dirección IP ENA:

```
HostnameENA$ = Otros.Text1.Text
```

➤ Dirección GPIB:

```
HostnameGPIB$ = Otros.Text2.Text
```

➤ Dirección destino de los datos:

- En este caso, para simplificar el proceso lo que hacemos es crear una carpeta en el ENA con exactamente el mismo nombre que le hayamos dado nosotros a nuestra campaña de medidas en el PC. De esta forma evitamos incoherencias.

```
y = SendCommand_ENA(":MMEM:MDIR " + "" + directorio$ + "")
```

➤ Número de Repeticiones:

```
repetic$ = Otros.TextBox5.Text
```

```
'Lo usamos una vez dentro de la rutina de medidas'
```

➤ Auto Scale:

```
For i = 1 To 8
  For j = 1 To 2
    ii$ = i
    jj$ = j
    y = SendCommand_ENA(":DISP:WIND" + ii$ + ":TRAC" + jj$ +
":Y:AUTO")
  Next j
Next i
```

➤ Preset:

```
y = OpenSocket_ENA(HostnameENA$, ScpiPort)
y = SendCommand_ENA(":SYST:PRES")
```

3.3.2.3.3 Calibración

La calibración remota, al menos desde el punto de vista de cómo lo enfocamos, resulta con bastante diferencia la sección más difícil de programar de nuestro trabajo.

La razón es la dificultad de realizar un bucle que contemple las 16 posibles combinaciones de antenas de las que disponemos, y por ende las 16 diferentes combinaciones de trazas.

Como el lector sabrá, es necesario a la hora de realizar una campaña de medidas calibrar todos los cables de antena con todos los puertos posibles, para eliminar en la medición el ruido introducido por los mismos. Este proceso resulta verdaderamente engorroso y confuso en casos de tener un número alto de antenas, por lo que la idea es desarrollar una función que vaya indicando al usuario mediante mensajes en pantalla qué cable debe conectar a qué puerto para realizar la calibración.

De esta forma, al llevar de la mano al usuario, se elimina buena parte de la confusión y se hace bastante más sencillo saber qué combinaciones se han calibrado y cuáles no.

Para intentar explicar esta compleja función de la forma más amena posible, en vez de plasmar el código fuente entero de una tacada, vamos a ir explicándolo de forma segmentada, ya que debido a su dificultad resulta un código bastante complejo de entender echándole meramente un vistazo.

1. Establecimiento de comunicación

```

Call calcVentanas
Call StartIt
Call get_hostnameENA
  y1 = OpenSocket_ENA(HostnameENA$, ScpiPort)
  y = SendCommand_ENA(":MMEM:MDIR " + "" + directorio$ + "")
  y = SendCommand_ENA(":SENS:MULT1:DISP ON")

Call opc_ENA

  y = SendCommand_ENA(":SENS:MULT1:STAT ON")
Call opc_ENA
  'sirve para pasar el control al puerto del aparato que hay debajo del
ENA
  A = SendCommand_ENA(":DISP:SPL?")  'Para ver el tipo de configuración q
tiene
  A = RecvAscii_ENA(recvBuf, 20)
Call opc_ENA

  llam = True
  R = 1
  While llam  'Para pasar lo q ha recibido antes a un formato legible
    aux4 = Right(Left(recvBuf, R + 1), 1)
    If aux4 <> " " Then
      aux2 = Left(recvBuf, R)
      aux3 = Right(aux2, 1)
      aux = aux + aux3
    Else
      llam = False
    End If
    R = R + 1
  Wend

```

En este caso lo que hacemos es establecer la comunicación con el ENA, crear la carpeta donde se van a guardar las muestras dentro del ENA, y hacer una llamada al aparato para comprobar qué configuración tiene. Finalmente pasamos esa

configuración a formato legible. La llamada 'opc_ENA' viene en el manual de Agilent_[17] y sirve para recibir posibles datos de respuesta del ENA.

2. Obtención de situación de trazas

Una vez obtenida la configuración de las trazas en el ENA, hacemos un Case para calcular el número de "ventanas" que existen representadas en el ENA, lo cual es un aspecto básico a la hora de hacer las calibraciones.

```
Select Case aux
Case "D1"
    ventV = 1
    ventH = 1
Case "D12"
    ventV = 1
    ventH = 2
Case "D1_2"
    ventV = 2
    ventH = 1
Case "D123"
    ventV = 1
    ventH = 3
Case "D1_2_3"
    ventV = 3
    ventH = 1
Case "D12_34"
    ventV = 2
    ventH = 2
Case "D1234"
    ventV = 1
    ventH = 4
Case "D1_2_3_4"
    ventV = 4
    ventH = 1
Case "D123_456"
    ventV = 2
    ventH = 3
Case "D12_34_56"
    ventV = 3
    ventH = 2
Case "D1234_5678"
    ventV = 2
    ventH = 4
```

```

Case "D12_34_56_78"
    ventV = 4
    ventH = 2
Case "D123_456_789"
    ventV = 3
    ventH = 3
End Select
ventT = ventV * ventH

```

3. Cambio de todas las trazas a tiempo continuo

```

RM$ = 0
RMA$ = 0
repet = repet + 1
LL$ = repet
    k = 1
    L = 0
    M = 0
    tr = 0
    If k = 1 Then
        Call controlador1
    End If

    ventT2$ = ventT
    B = SendCommand_ENA(":CALC" + ventT2$ + ":PAR:COUN?") 'Comprobar el
numero de trazas que hay en la última ventana
    B = RecvAscii_ENA(recvBuf, 20)
    Call opc_ENA
    aux2 = Right(Left(recvBuf, 2), 1)
    If aux2 = 1 Then
        aux4 = (ventH * 2) - 1
    Else
        aux4 = ventH * 2
    End If

    For kk = 1 To ventV

For RA = 1 To ventH
        RM$ = RM$ + 1
        B = SendCommand_ENA(":INIT" + RM$ + ":CONT ON")
        Call opc_ENA
Next RA

Next kk

```

```
Call opc_ENA
```

4. Manejo del Controlador

```
k = 1
```

```
L = 0
```

```
M = 1
```

```
tr = 0
```

```
RB = 0
```

```
RC = -1
```

```
For kk = 1 To ventT
```

```
tr = tr + 1
```

```
If tr = ventH + 1 Then
```

```
tr = 1
```

```
k = k + 1
```

```
End If
```

```
If ventH > 1 Then
```

```
If kk Mod 2 <> 0 Then
```

```
RB = RB + 1
```

```
End If
```

```
Else:
```

```
If ventV > 1 Then
```

```
RB = RB + 1
```

```
End If
```

```
End If
```

```
If RB = 1 Then
```

```
Call controlador1
```

```
End If
```

```
If RB = 2 Then
```

```
Call controlador2
```

```
End If
```

```
If RB = 3 Then
```

```
Call controlador3
```

```
End If
```

```
If RB = 4 Then
```

```
Call controlador4
```

```
End If
```

```
If RB > 4 Then
```

```
MsgBox "Error"
```

```

End If

kka$ = kk
B = SendCommand_ENA(":CALC" + kka$ + ":PAR:COUN?")
B = RecvAscii_ENA(recvBuf, 20)
Call opc_ENA
aux2 = Right(Left(recvBuf, 2), 1)
P$ = kk

```

En esta parte calculamos cómo debemos de manejar el controlador. Para ello tenemos 4 posibles funciones ('controlador1-4'), que se comunican con el mismo mediante el puerto GPIB_[18] e indican cómo debe actuar en cada caso.

Adjuntamos como ejemplo 'controlador1':

```

Sub controlador1()

Dim Dev As New AgtGPIB
Dim idn As String

Dev.connect ("GPIB0::28")

Dev.Write.command "*RST"
Dev.Write.command "*IDN?"

Dev.Write.command "B0 A1 B2 B3 B4 B5 B6 B7 B8 B9 B0"

Call opc_ENA
End Sub

```

5. Cálculo de Combinación a Calibrar, y Aviso al Usuario

Esta representa una parte importante de la función. Después de un duro proceso hemos podido obtener la traza que debemos calibrar, y para que sea más evidente la calibración pasamos la traza a formato logarítmico. Lo que hacemos ahora es calcular qué antena transmisora (variable 'M') debe calibrarse con cuál puerto del receptor (R1+, R1-, R2+ o R2-, variable 'Rv'). Finalmente avisamos mediante un *MsgBox* al usuario de qué cable debe conectar, y esperamos a que pulse *Aceptar* para iniciarse la calibración en sí (Paso 6).

```

For MM = 1 To aux2
    L = L + 1
    If L = aux4 + 1 Then
        L = 1
    End If

    P$ = kk
    t$ = MM

    Npan$ = MM + 1

    If RC = 3 And ventV = 2 And ventH = 1 And kk Mod 2 = 0 Then

        RC = 1
        M = M + 1

    Else:

        If RC = 2 And (ventV + ventH) <= 4 And kk Mod 2 <> 0 Then
            M = M + 1
            RC = 1

        Else:

            If RC = 3 And ventV >= 1 And kk Mod 2 = 0 Then
                RC = 2

            Else:

                RC = RC + 2

```



```
End If
End If
End If

lab$ = L
Mev$ = k

Y1 = SendCommand_ENA(":DISP:WIND" + P$ + ":ACT")
Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR" + t$ + ":SEL")
Y1 = SendCommand_ENA(":CALC" + P$ + ":FORM MLOG")
```

```
Select Case RC
```

```
Case 1
```

```
Rv = "R1+"
```

```
Case 2
```

```
Rv = "R2+"
```

```
Case 3
```

```
Rv = "R1-"
```

```
Case 4
```

```
Rv = "R2-"
```

```
End Select
```

```
MsgBox "Conecte el transmisor " & M & " con el receptor " & Rv, vbOKOnly, ""
```

6. Calibración

Una vez que el usuario acepta, seleccionamos el puerto del receptor a utilizar, y efectuamos al fin la orden de calibración.

```
Select Case RC

    Case 1

        Y1 = SendCommand_ENA(":SENS" + P$ +
":MULT1:TSET9:PORT3 R1")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR1:DEF S21")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR1:SEL")

    Case 2

        'Y1 = SendCommand_ENA(":SENS" + P$ + ":MULT1:PORT3 R2")
        Y1 = SendCommand_ENA(":SENS" + P$ +
":MULT1:TSET9:PORT3 R2")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR1:DEF S21")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR1:SEL")

    Case 3

        Y1 = SendCommand_ENA(":SENS" + P$ +
":MULT1:TSET9:PORT4 R1")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR2:DEF S31")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR2:SEL")

    Case 4

        Y1 =SendCommand_ENA(":SENS" + P$ + ":MULT1:TSET9:PORT4
R2")

        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR2:DEF S31")
        Y1 = SendCommand_ENA(":CALC" + P$ + ":PAR2:SEL")

End Select

Y1 = SendCommand_ENA(":SENS" + P$ + ":CORR:COLL:CKIT 1")
Y1 = SendCommand_ENA(":SENS" + P$ + ":CORR:COLL:METH:THRU " +
Npan$ + ",1")
```

```
Y1 = SendCommand_ENA(":SENS" + P$ + ":CORR:COLL:THRU " + Npan$ +  
",1")  
  
    Call opc_ENA  
Y1 = SendCommand_ENA(":SENS" + P$ + ":CORR:COLL:SAVE")  
  
    If RC = 4 Then RC = -1  
  
Next MM  
  
    If ventV = 2 And ventT = 4 And kk Mod 2 <> 0 And kk > 1 Then  
M = M + 1  
End If  
  
    If ventT = 8 And ventV = 4 And kk Mod 2 = 0 Then  
RC = -1  
M = M + 1  
End If  
  
    If ventT = 6 And kk Mod 2 = 0 Then  
RC = -1  
M = M + 1  
End If  
  
    If ventH = 1 Then  
RC = -1  
M = M + 1  
End If  
  
    If RC = 4 Then RC = -1  
Next kk
```

7. Final del Bucle de Calibración

Como paso final, una vez acabado el bucle de calibración, devolvemos a todas las trazas su formato original, y guardamos el fichero de calibración dentro del ENA, en caso de que el usuario quiera salvarlo en una memoria USB para uso futuro.

```

MsgBox "Calibración completada", vbOKOnly, ""

format2$ = format1

For R = 1 To vent1
    Npan$ = R
    For MM = 1 To aux2
        t$ = MM
        Y1 = SendCommand_ENA(":DISP:WIND" + Npan$ + ":ACT")
        Y1 = SendCommand_ENA(":CALC" + Npan$ + ":PAR" + t$ + ":SEL")
        Y1 = SendCommand_ENA(":CALC" + Npan$ + ":FORM " + format2$)
    Next MM
Next R

Y1 = SendCommand_ENA(":MMEM:STOR " + "" + directorio$ + "/" +
"calibrado.sta" + "")
Call CloseConnection_ENA
Call EndIt

```

3.3.2.4 Realización de las Medidas

Una vez que hemos configurado el ENA en base a nuestras necesidades y tenemos todos los componentes bien calibrados, pasamos a la parte final, que sería la realización de la campaña en sí.

La misión en este caso es obtener los puntos receptores de la ruta que hayamos elegido, y realizar las mediciones por cada punto. Una buena forma de llevar a cabo esta tarea es ir informando al usuario sobre desde qué punto de la ruta se van a realizar las medidas. De esta manera el usuario llevará fácilmente la cuenta sobre qué medidas se han realizado y cuáles no, y una vez terminada la campaña todas las medidas efectuadas estarán asociadas a cada punto receptor de la ruta, el cual es el principal objetivo de este trabajo.

Partiendo de esta base, la manera más interesante y sencilla de realizar esta tarea es crear un bucle que vaya pasando por todos los puntos receptores de una ruta, y por cada iteración del bucle, alertar al usuario, y una vez que el usuario quite la alerta hacer una llamada a la rutina de medidas.

Una vía sencilla de hacer este bucle es de la siguiente manera:

```
'Obtenemos previamente el acceso a la Feature como vimos en el apartado
3.2.4.1'

For i = 1 To h
Set feat = pfeatcls.GetFeature(i)

coord_x = feat.value(2)
coord_y = feat.value(3)
nombre = feat.value(4)
MsgBox "Medición desde el punto: " & i & Chr(13) & "Nombre: " & nombre &
Chr(13) & "Posición: " & coord_x & " , " & coord_y, vbExclamation
posicion = nombre
rutina_de_medidas (i)
MsgBox "Medidas en la posición n°" & i & " completadas", vbOKOnly

Next
```

Como se puede observar, lo único que hacemos en este bucle es obtener los datos de cada punto receptor, mostrar el aviso al usuario mediante un *MsgBox*, y hacer una llamada a la rutina de medidas que ahora comentaremos. Finalmente una vez realizadas las medidas se avisa al usuario de que se ha efectuado el subproceso correctamente.

Por último queda comentar el código fuente de la rutina de medidas. La mayor parte del código es muy similar al código de calibración. De hecho, los pasos 1-4 arriba comentados son prácticamente iguales en ambos casos. Centrémonos, pues, en la sección final del código, que es donde realmente se hacen las medidas.

```

k = 1
L = 0
M = 0
tr = 0

For kk = 1 To ventT

    tr = tr + 1
    If tr = ventH + 1 Then
        tr = 1
        k = k + 1
    End If

    kka$ = kk
    B = SendCommand_ENA(":CALC" + kka$ + ":PAR:COUN?")
    B = RecvAscii_ENA(recvBuf, 20)
    Call opc_ENA
    aux2 = Right(Left(recvBuf, 2), 1)

    For MM = 1 To aux2
        L = L + 1
        If L = aux4 + 1 Then
            L = 1
        End If

        AV$ = kk
        BV$ = MM

        num$ = i
        lab$ = L
        Mev$ = k
    
```

```

pos$ = posicion
V$ = R

If Crear = False Then
y = SendCommand_ENA(":MME:MDIR " + "" + directorio$ + "/" +
pos$ + "") ' (1) '
Crear = True

End If

Y1 = SendCommand_ENA(":DISP:WIND" + AV$ + ":ACT") ' (2) '
Y1 = SendCommand_ENA(":CALC" + AV$ + ":PAR" + BV$ + ":SEL") ' (3) '

yl = SendCommand_ENA(":MME:STOR:FDAT " + "" + directorio$ +
"/" + pos$ + "/" + "datos" + lab$ + Mev$ + "_" + V$ + "_" + num$ + "") ' (4) '
Call opc_ENA

Next MM
Next kk

Call espera
Next R
Call CloseConnection_ENA
Call EndIt

```

Lo que estamos haciendo aquí es un bucle dirigido por la variable R la cual va a ir recorriendo todas las trazas, y va a enviar una orden de medir por cada traza. Para simplificar la jerarquía, lo que hacemos es crear una carpeta en el ENA por cada punto receptor de la ruta, siendo el nombre de la carpeta el del punto en cuestión (línea de código (1)).

Posteriormente realizamos la medida (líneas (2), (3), (4)), y guardamos cada archivo con la siguiente notación:

- DATOSXY_Z_A
 - X: Receptor.
 - Y: Transmisor.
 - Z: Nº de Repetición.
 - A: Posición del punto con respecto a la secuencia.

Una vez realizadas las medidas en todos los puntos receptores, habremos acabado la campaña y en general, todo el proceso. De esta forma, en el ENA tendremos una carpeta raíz con el nombre de la campaña. Dentro de esta carpeta tendremos el archivo *calibrado.sta*, con la calibración previamente realizada, y subcarpetas con nombres de cada punto receptor de la ruta. Cada subcarpeta contendrá las medidas de cada punto, con la notación arriba descrita.

De esta forma quedaría finalizada la orientación hacia el desarrollo del software, y por ende, del trabajo.

Capítulo IV

Conclusiones y líneas futuras de trabajo



4.1 CONCLUSIONES

Después de estos meses de trabajo, hemos conseguido desarrollar una orientación para que el usuario sea capaz de realizar un software que integre la realización de campañas de medidas con las numerosas ventajas que nos proporciona la referenciación geográfica, utilizando ArcGIS. Existen numerosas maneras de afrontar este desarrollo, pero creemos que la nuestra es de las más amenas y lineales posibles.

La creación de un software de estas características puede ser muy útil para simplificar, mejorar y gestionar la toma de medidas, y como tal aconsejamos que se lleve a cabo el desarrollo del mismo.

Se ha decidido no incluir el 100% del código fuente propuesto para dejar un cierto margen de investigación al usuario, y también para dejar abierto el camino para que el desarrollador pueda expandir y mejorar las características aquí expuestas.

En conclusión, esperamos que esta guía sea útil al lector de la misma, y sirva como esqueleto para la realización de proyectos más completos y ambiciosos.

4.2 LÍNEAS FUTURAS DE TRABAJO

La guía está por supuesto sujeta a mejora, tanto en la forma de explicar las características como en el número de funcionalidades. A continuación destacamos las funcionalidades más interesantes que se podrían incluir:

- *Posibilidad de enviar de forma automática la carpeta de medidas desde el ENA hasta el ordenador portátil, ya que al realizar las medidas, la carpeta se queda almacenada en el ENA, teniendo que usar un disco extraíble para obtener los datos. Fue un aspecto que se investigó, pero no se llegó a dar con la manera de poder enviar carpetas enteras (ya que archivos sueltos sí se puede).*
- *Opción de poder cargar el fichero **calibrado.sta** directamente desde el portátil, ya que en el software no se ofrece la opción para ello, teniendo que usarlo directamente desde el ENA.*

- *Poder enlazar el software con el analizador de las medidas, ya que en esta guía no explicar de ninguna forma cómo estudiar las medidas una vez realizadas.*

Capítulo V

Bibliografía



- [1] http://science.nature.nps.gov/nrgis/training/intro_arc9.aspx
- [2] <http://www.vb6.us/>
- [3] blog.kowalczyk.info/software/sumatrapdf/
- [4] Proyecto de final de carrera: *Desarrollo e implementación de una aplicación para la gestión de una red de sensores conectada remotamente vía el sistema GSM-GPRS*, Concepción Pardo García, UPCT, 2007
- [5] <http://www.esri.com/what-is-gis/index.html>
- [6] http://www.archive.org/stream/ManualEnEspaolArcgis9.1/Arcgis91-FundamentosDeArcgisTutorialDeLecturas_djvu.txt
- [7] *Using ArcGis Spatial Analyst*, Jill McCoy, Kevin Johnston, Environmental Systems Research Institute (Redlands, Calif.), Environmental Systems Research Institute, 2001
- [8] <http://www.esri.es/es/formacion/que-es-un-sig-/>
- [9] *What Is Arcgis 9.1*, ESRI Press, ESRI, 2005
- [10] *Getting started with ArcGIS*, Bob Booth, Andy Mitchell, Environmental Systems Research Institute (Redlands, Calif.), Environmental Systems Research Institute, 2001
- [11] *Exploring ArcObjects: Applications and Cartography*, ESRI, 2001
- [12] *Exploring ArcObjects Vol. II - Graphics Data Management*, Paperback, ESRI, 2001

- [13] *RADIOGIS: Herramienta para la realización de prácticas relacionadas con sistemas de radiocomunicaciones*. Rubén Ibernón Fernández, José M. Molina García-Pardo, José Víctor Rodríguez Rodríguez y Leandro Juan Llácer. <http://www.upct.es/sicomo/indradiogis.php>
- [14] www.agilent.es
- [15] Proyecto de final de carrera: *Caracterización Experimental del Canal Radio a 2.4Ghz para Sistemas MIMO en Entornos Interiores*. Iván Castillo Olmo, 2006.
- [16] Proyecto de final de carrera: *Software para la automatización de medidas de un sistema MIMO de banda ancha*, Rafael Antonio Lucas Ávila, 2005.
- [17] *Agilent E5070B/E5071B ENA Series RF Network Analyzers VBA Programmer's Guide*, Agilent Technologies, 2007
- [18] <http://www.ni.com/white-paper/3419/es>