ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Proyecto Fin de Carrera

Desarrollo e Implementación de nuevas funcionalidades en una herramienta de gestión y cálculo de cobertura radioeléctrica basada en Sistemas de Información Geográfica

AUTOR: Paula Jiménez Pérez
DIRECTOR: Leandro Juan Llácer
Enero / 2006

Autor	Paula Jiménez Pérez
E-mail del Autor	paula_jimenez@telefonica.net
Director	Leandro Juan Llácer
E-mail del Director	leandro.juan@upct.es
Título del PFC	Desarrollo e Implementación de nuevas funcionalidades en una herramienta de gestión y cálculo de cobertura radioelétrica basada en Sistemas de Información Geográfica

Resúmen

Las aplicaciones basadas en Sistemas de Información Geográfica permiten, entre otras funcionalidades, almacenar las coberturas radioeléctricas como si de un mapa más se tratase, además de facilitar la creación, almacenamiento y gestión de todos los parámetros que intervienen en la planificación de sistemas de radiocomunicación como el GSM, UMTS, LMDS, MMDS, etc. RAGIS es una aplicación informática para la gestión y cálculo de cobertura radioeléctrica de sistemas de radiocomunicación, que ha sido desarrollada por el Grupo de Investigación de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena. Hasta ahora, esta herramienta permitía entre algunas de sus funcionalidades, el cálculo de la estación mejor servidora de un sistema de coberturas para cada punto de un mapa y el cálculo de porcentaje de territorio que una cobertura o sistema de potencia cubría por encima de un umbral de potencia establecido.

Así pues, el objetivo de este proyecto es el de programar en Visual Basic y ArcObjects una nueva funcionalidad de Cálculo de Servidores, para poder conocer todas las estaciones que prestan servicio a cada punto de un mapa, y la de añadir funcionalidad a la opción de cálculo de porcentaje sobre umbral, obteniendo además de la representación de los porcentajes en el mapa, un archivo en formato Excel, con diversas tablas y gráficos, a modo de informe de resultados.

Titulación	Ingeniero de Telecomunicación
Intensificación	Sistemas y Redes de Telecomunicación
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Enero - 2006

CAPÍTULO 1.- INTRODUCCIÓN - 5 -

1.1.- Fundamentos y Objetivo del proyecto - 5 -

1.2.- Herramientas utilizadas - 7 -

1.3.- Contenido y estructura del proyecto - 7 -

CAPÍTULO 2.- INTRODUCCIÓN A RAGIS - 9 -

2.1.- Introducción a los sistemas Celulares - 9 -

Gestión de la movilidad: Sistemas móviles celulares - 12 -

Parámetros de un enlace radioeléctrico: Balance de Potencias - 15 -

2.2.- RAGIS: Herramienta de gestión y de cálculo de cobertura radioeléctrica basada en SIG - 16 -

Algunas funcionalidades de RAGIS - 17 -

CAPÍTULO 3.- DESARROLLO DE NUEVAS FUNCIONES PARA RAGIS - 22 -

3.1.- El entorno de desarrollo de ArcGIS: ArcObjects - 22 -

Objetos Raster - 24 -

Operaciones con datos Raster: Spatial Analyst - 25 -

3.2.- Coberturas en RAGIS - 25 -

3.3.- Cálculo de mejores servidores - 28 -

3.4.- Cálculo de porcentajes de territorio con cobertura por encima de un umbral y generación de informe de resultados - 37 -

CAPÍTULO 4.- RESULTADOS SOBRE SIMULACIÓN DE COBERTURA RADIOELÉCTRICA GSM - 54 -

4.1.- Simulación en entorno Rural - 56 -

4.2.- Simulación en entorno Urbano - 67 -

CAPÍTULO 5.- CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO - 73 -

CAPÍTULO 6.- REFERENCIAS - 75 -

APÉNDICE A: MODELOS DE PROPAGACIÓN - 76 -

A.1.- UIT-526 - 76 -

A.2.- COST 231 - 77 -

APÉNDICE B: SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG) - 80 -

APÉNDICE C: MANEJO DE LA HERRAMIENTA - 83 -

C.1.- Mejores Servidores - 83 -

C.2.- % Sobre Umbral - 85 -

APÉNDICE D: CÓDIGO PROGRAMADO - 90 -

D.1.- Procedimientos y Funciones - 90 -

D.2.- Código Fuente - 93 -

CAPÍTULO 1.- INTRODUCCIÓN

1.1.- Fundamentos y Objetivo del proyecto

Hoy en día las redes celulares prestan un servicio imprescindible tanto en el ámbito público como en el privado. Desde las redes GSM, GPRS y UMTS con que usuarios particulares y profesionales se comunican incluso en llamadas internacionales, hasta las redes TETRA, con que las empresas facilitan la intercomunicación de sus profesionales. Con la aparición de nuevas tecnologías y servicios la demanda se dispara y surge la necesidad de planificar y mantener al día redes de forma rápida y eficiente. Las herramientas de planificación facilitan las labores de ingeniería reduciendo el tiempo de trabajo para aquellas tareas menos agradables como la búsqueda y validación de los emplazamientos de las estaciones base de la red celular que se completan de forma casi automática en cuestión de horas. A las herramientas *software* modernas se suma la disponibilidad de datos cartográficos digitales abundantes, baratos, y con unos niveles muy altos de resolución. A esta categoría de herramientas de planificación de redes pertenece el software RAGIS.

RAGIS es una herramienta de gestión y análisis de sistemas de radiocomunicaciones basada en el Sistema de Información Geográfica (SIG) (ver Anexo B al final) ArcView 8.2 de ESRI (*Environmental Systems Research Institute*) y desarrollada por el Grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena. RAGIS aprovecha todas las facilidades de tratamiento de la información de los SIG, y es tanto un software de gestión de bases de datos de sistemas de radiocomunicaciones como una potente herramienta para el cálculo radioeléctrico, incorporando diversos modelos de propagación para entornos abiertos, urbanos e interiores.

Una de las funciones importantes que presenta esta herramienta es la realización de diferentes tipos de cálculos radioeléctricos. De esta forma, la herramienta permite realizar cálculos de coberturas de potencia definiendo todos aquellos parámetros que son necesarios para dicho cálculo. La herramienta también permite el cálculo de sistemas de potencia y mejores servidores para los que el usuario debe determinar, entre otros, cuales son los emplazamientos que quiere que formen parte de los sistemas. Una

vez concluidos los cálculos, el SIG presenta por pantalla los valores máximos de potencia en cada una de las celdas así como las estaciones que están proporcionando mayor cobertura. Asimismo, la herramienta permite realizar el cálculo de porcentajes de cobertura sobre una zona determinada. En este caso el usuario debe seleccionar un sistema de potencia o cobertura radioeléctrica y una zona del mapa donde quiere realizar los cálculos. Una vez realizados se presenta por pantalla un mapa y un gráfico donde se representan los porcentajes de cobertura totales en la zona seleccionada.

Centrándose en estos dos últimos puntos, el presente proyecto se plantea principalmente dos objetivos:

El primer objetivo es el derrarrollo e implamentación en la herramienta RAGIS de una nueva funcionalidad de calculo de estaciones servidoras de un sistema de potencia. Esta función partirá de un sistema de potencia cargado en un mapa y mostrará qué emplazamientos de los que forman el sistema prestan servicio en cada punto de dicho mapa. Por otra parte, los datos generados a partir de esta nueva funcion (mapas de servidores), serán integrados en las funciones de bases de datos que RAGIS incorpora en el menú *Bases de datos*.

El segundo objetivo es el de la mejora del cálculo de porcentajes de cobertura (funcionalidad llamada *% Sobre Umbral* en RAGIS), añadiendo la generación de un archivo con formato *.xls* (libro de Excel) a modo de informe de resultados. En este informe se mostraran distintas tablas y gráficos, no sólo con información sobre la cobertura recibida sobre la zona total de cálculo seleccionada, sino también con la correspondiente a cada uno de los territorios (llamados *features* en ArcView) de la zona de cálculo seleccionados por separado. Además, en el caso de haber elegido para el cálculo una cobertura correspondiente a un sistema cuyos mejores servidores hayan sido calculados con anterioridad, se añadirá al informe información adicional referida al porcentaje de cobertura en cada territorio por cada estación trabajando como mejor servidora.

Dichas funciones se programarán en lenguaje Visual Basic para aplicaciones (VBA) y ArcObjects y se integrarán en los menús de RAGIS, más concretamente en el de cálculo de sistemas.

1.2.- Herramientas utilizadas

Para la realización de este proyecto se han usado las herramientas hardware y software siguientes:

- Ordenador portátil con Procesador Intel Pentium IV a 3,06 GHz y 1 GB de memoria RAM
- Borland C++ 5.0
- ArcGIS Desktop 8.2 con la extensión *Spatial Analyst*
- Fichero *mxt* con la herramienta RAGIS

Vale incluir aquí un comentario respecto de este software. Bajo el nombre ArcGIS Desktop son comercializados los sistemas ArcInfo, ArcEditor y ArcView, que comparten un mismo núcleo y un número de funciones que varía de la versión más completa (ArcInfo) hasta la más simple (ArcView, la utilizada para este proyecto). Pero cada una de ellas está compuesta por tres aplicaciones diferentes: ArcMap (permite visualizar, consultar, editar y realizar análisis sobre nuestros datos), ArcCatalog (semejante a Windows Explorer para datos espaciales) y ArcToolbox (conjunto de herramientas de conversión de datos). En este texto, siempre que nombremos ArcGIS o ArcView se estará refiriendo a las aplicaciones ArcMap y ArcCatalog. Por otro lado, se hace necesaria la instalación del programa Borland C++ para el reconocimiento de las funciones de cálculo de pérdidas por propagación integradas en la herramienta RAGIS y programadas en lenguaje C.

1.3.- Contenido y estructura del proyecto

Este proyecto, y tras un primer capitulo de introducción, queda estructurado de la siguiente forma:

En primer lugar, en el capítulo 2 se realiza una pequeña introducción a la herramienta de cálculo y gestión de coberturas radioléctricas RAGIS sobre la que se ha llevado a cabo este proyecto. Dicho capítulo se divide en dos subapartados. El primer subapartado es una introducción teórica de los sistemas celulares, en los que se basa la

herramienta RAGIS y en el segundo subapartado se exponen las funciones más importantes de que dispone dicha herramienta.

En el capitulo 3, se explican con detalle las funciones programadas para la realización del proyecto. Primero, se hace una introducción teórica del tipo de lenguaje y objetos utilizados en la programación, y de cómo RAGIS maneja los archivos de coberturas, ya que éstos son los que se utilizan a lo largo de todo el proyecto. Una vez hecho esto, en los subapartados posteriores, se explican con detalle los puntos más importantes del código de las funciones programadas, a través de diagramas de flujo y de las partes del código que las realizan.

En el capitulo 4 se realiza el cálculo de un sistema de potencia formado por tres emplazamientos. Con dicho sistema se calculan los mejores servidores y el % sobre umbral en dos tipos de entorno distintos a modo de prueba de las funciones programadas. En este capítulo se muestran los parámetros del sistema introducidos, correspondientes a estaciones GSM, y los resultados obtenidos en RAGIS para un entorno rural y para un entorno urbano.

En el capítulo 5 se recogen las conclusiones derivadas de este proyecto.

En el capítulo 6 se recogen las referencias bibliográficas que se han usado para la realización del proyecto.

El anexo A es una descripción de los modelos de propagación utilizados para el capítulo 4 de simulación de cobertura radioeléctrica.

El anexo B es una introducción a los Sistemas de Información Geográfica (SIG).

El anexo C se presenta como una ayuda al usuario para el manejo de las nuevas funcionalidades.

El anexo D recoge todo el código fuente en lenguaje Visual Basic programado, así como una descripción de cada una de los métodos y funciones programadas.

CAPÍTULO 2.- INTRODUCCIÓN A RAGIS

2.1.- Introducción a los sistemas Celulares

El uso de un recurso tan limitado como es el espectro radioeléctrico ha sido siempre uno de los principales problemas a solventar por parte de cualquier tecnología inalámbrica. El concepto de *sistema celular* fue un gran avance en la resolución del problema de la congestión espectral y de la capacidad del usuario. Éste ofrecía una gran capacidad en una localización limitada del espectro sin grandes cambios tecnológicos.

La idea de un sistema celular consiste en un sistema basado en varios niveles de celdas: un transmisor de gran potencia (celda grande) con muchos transmisores de baja potencia (celdas pequeñas) (Figura 1), cada una proporcionando cobertura a sólo una pequeña porción del área de servicio. A cada estación base se le asigna una porción del número total de canales disponibles en el sistema completo, y a las estaciones base cercanas se les asignan diferentes grupos de canales de forma que los canales disponibles son asignados en un número relativamente pequeño de estaciones base vecinas. A las estaciones base vecinas se les asigna diferentes grupos de canales de forma que las interferencias entre las estaciones base (y entre los usuarios móviles bajo su control) se reducen. Espaciando sistemáticamente las estaciones base y sus grupos de canales a través de un mercado, los canales disponibles se distribuyen a través de una región y pueden ser reutilizadas tantas veces como sea necesario, siempre que la interferencia entre estaciones con el mismo canal se mantenga por debajo de unos niveles aceptables.

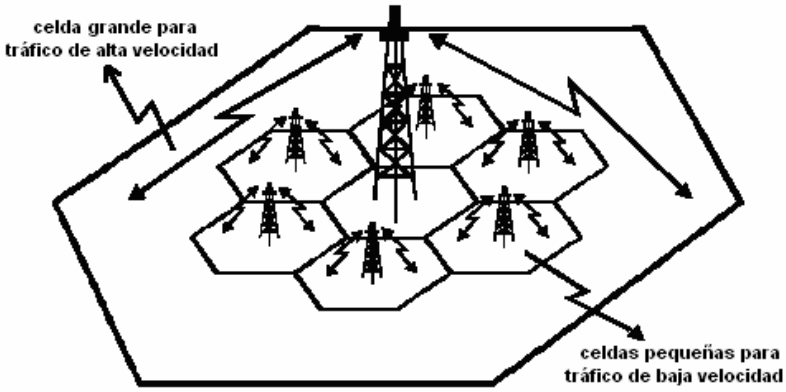


Figura 1: Sistema celular. Las celdas más pequeñas (*microceldas*) pueden usarse para complementar o mejorar el servicio de las celdas mayores (*macroceldas*).

Conforme crece la demanda de servicios, se debe incrementar el número de estaciones base, proporcionando una capacidad de radio adicional sin incremento del espectro de radio. Este principio es el fundamento de todos los sistemas modernos de comunicaciones inalámbricos.

Reutilización de Frecuencias

Los sistemas de radio celulares se basan en la colocación inteligente así como de la reutilización de los canales a través de una región de cobertura. Al proceso de diseño de seleccionar y colocar grupos de canales en todas las estaciones base dentro de un sistema, se le llama *reutilización de frecuencias* o *planificación de frecuencias*.

La Figura 2 ilustra el concepto de reutilización de frecuencias, donde las celdas con el mismo número utilizan el mismo grupo de canales. La separación que debe de haber entre estas celdas viene dada por el parámetro *D* (*distancia cocanal* o *de reuso*). La forma hexagonal de la celda mostrada en la figura es conceptual y es un modelo simple de la cobertura de radio para cada estación base, pero ha sido universalmente adoptado dado que el hexágono permite un análisis fácil y manejable de un sistema celular. La cobertura real de una celda se conoce como *huella* (*footprint*) y se determina de los *modelos de campo* o de los *modelos de predicción de la propagación*.

A las *N* celdas que usan un conjunto completo de frecuencias disponibles se les llama *cluster*. Valores típicos de *N* (tamaño del *cluster*) son 4, 7 y 12. Si el tamaño del *cluster* se reduce mientras que el tamaño de la celda permanece constante, se requerirán más *clusters* para cubrir un área dada y por tanto se logra una mayor capacidad. Cuanto mayor sea *N*, mayor va a ser la distancia entre estaciones base con el mismo grupo de canales, menor será su interferencia, pero la capacidad del sistema será menor también. Desde un punto de vista del diseñador, es deseable usar el valor más pequeño de *N* posible, para maximizar la capacidad dentro de un área de cobertura.

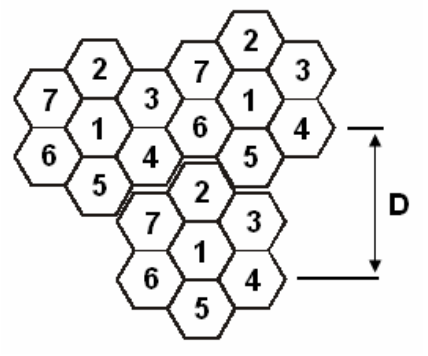


Figura 2: Reutilización de frecuencias. Las celdas del mismo número utilizan las mismas frecuencias. Éstas celdas están separadas una distancia *D*. Dichas celdas se reúnen formando los *clusters* (*N*=7). Las celdas adyacentes utilizan diferentes conjuntos de frecuencias.

División de Celdas (Cell-splitting)

El *splitting* es el proceso de subdividir una celda congestionada en celdas más pequeñas (Figura 3), cada una con su propia estación base y la correspondiente reducción en la altura de la antena y de la potencia de transmisión.

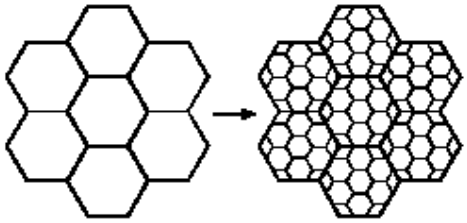


Figura 3: Cell-splitting. División de una celda en celdas menores para aumento de la capacidad.

El *splitting* incrementa la capacidad de un sistema celular dado que incrementa el número de veces que se reutilizan los canales. Definiendo nuevas celdas que tengan un radio más pequeño que las celdas originales e instalando estas pequeñas celdas entre las celdas existentes, se incrementa la capacidad debido al incremento de canales por unidad de área.

Interferencias

La *interferencia* es el principal factor que limita el desarrollo de los sistemas celulares. Las fuentes de interferencias incluyen a otras estaciones móviles dentro de la misma celda, o cualquier sistema no celular que de forma inadvertida introduce energía dentro de la banda de frecuencia del sistema celular. Las interferencias son más fuertes en las áreas urbanas, debido al mayor ruido de radio frecuencia y al gran número de estaciones base y móviles. Las interferencias son las responsables de formar un cuello de botella en la capacidad y de la mayoría de las llamadas entrecortadas. Los dos tipos principales de interferencias generadas por sistemas son las *interferencias co-canal*, entre celdas que utilizan las mismas frecuencias, y las *interferencias entre canales adyacentes*, procedentes de señales que son adyacentes en frecuencia a la señal deseada.

Gestión de la movilidad: Sistemas móviles celulares

La movilidad de los usuarios en un sistema celular es la fuente de mayores diferencias con la telefonía fija, en particular con las llamadas recibidas. Una red puede encaminar una llamada hacia un usuario fijo simplemente sabiendo su dirección de red (p. ej. su número de teléfono), dado que el conmutador local, al cual se conecta directamente la línea del abonado, no cambia. Sin embargo en un sistema celular la celda en la que se debe establecer el contacto con el usuario cambia cuando éste se mueve.

Un *sistema de radiocomunicaciones móvil*, según el Reglamento de Radiocomunicaciones, se define como un servicio de radiocomunicaciones entre estaciones móviles y estaciones terrestres (fijas) o entre estaciones móviles únicamente.

En la Figura 4 se representa el esquema básico de la estructura de un sistema de radiocomunicaciones móvil celular, en el que uno de los extremos (receptor o transmisor) se encuentra en movimiento.

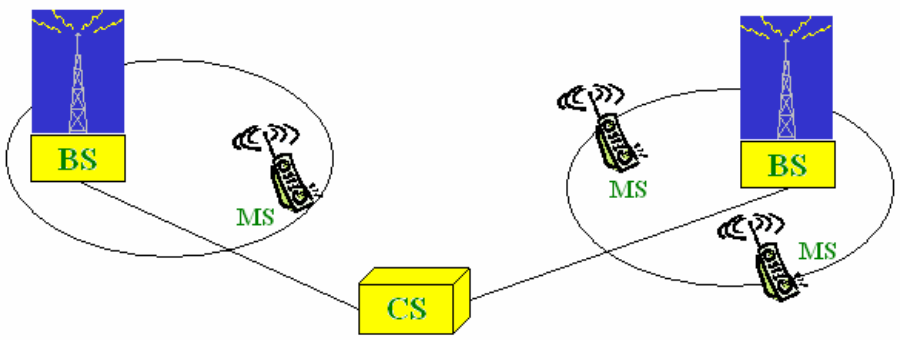


Figura 4: Sistema móvil celular. Esquema básico de una red móvil celular.

Los elementos que componen dicha estructura y sus acrónimos (que se utilizaran a lo largo de este proyecto) son:

- Estación Base (*BS*): estaciones fijas radiotransmisoras que son controladas por un centro de control (*CS*).

- Estación de Control (*CS*): estaciones fijas que realizan funciones de control sobre las *BS*. Entre estas funciones se encuentran las de señalización, localización, identificación, conexión entre sistemas, ...

- Estación Móvil (*MS*): estación con movilidad.

Con el fin de dar soporte a la movilidad del terminal, los sistemas móviles cuentan con una serie de funciones específicas, denominadas *funciones de movilidad*, que no son necesarias en los sistemas fijos. Entre ellas destacan la actualización de la posición, las funciones asociadas al establecimiento de llamada (que llamaremos genéricamente localización) y el *handover* o función de traspaso entre celdas. La importancia de estas funciones radica en que, junto a las limitaciones en los servicios ofrecidos que impone el empleo de una interfaz radio, determinan la necesidad de una infraestructura diferente a, o una ampliación de las capacidades desarrolladas en, los sistemas que dan soporte a las comunicaciones fijas.

Función de Traspaso (Handover)

Cuando una llamada está en progreso, la movilidad del usuario puede inducir a la necesidad de cambiar de celda servidora, en particular cuando la calidad de la transmisión cae por debajo de un umbral. Con un sistema basado en celdas grandes, la probabilidad de que ocurra esto es pequeña y la pérdida de una llamada podría ser aceptable. Sin embargo, si queremos lograr grandes capacidades tenemos que reducir el tamaño de la celda, con lo que el mantenimiento de las llamadas es una tarea esencial para evitar un alto grado de insatisfacción en los abonados.

Al proceso de la transferencia automática de una comunicación (de voz o datos) en progreso de una celda a otra para evitar los efectos adversos de los movimientos del usuario se le llama *handover*.

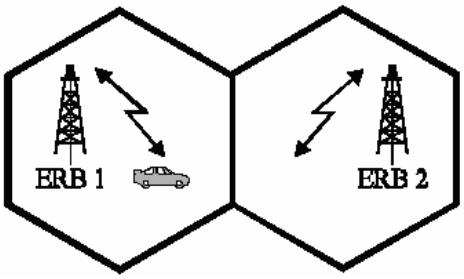


Figura 5: Handover. Cuando un terminal móvil se mueve a través de celdas diferentes la llamada en curso es traspasada automáticamente a un nuevo canal perteneciente a una nueva estación base.

Función de Seguimiento (Roaming)

En los sistemas de telecomunicaciones accedidos a través de un enlace fijo, la elección de qué red proporciona el servicio está hecha desde el principio. Cuando se introduce la movilidad, todo cambia. Diferentes servidores pueden proporcionar servicio a un usuario dado dependiendo de dónde esté. Cuando cooperan diferentes operadores de red, pueden usar esta posibilidad para ofrecer a sus abonados un área de cobertura mucho mayor que cualquiera de ellos pudiera ofrecer por sí mismo. A esto es a lo que se llama *roaming*.

Parámetros de un enlace radioeléctrico: Balance de Potencias

Para la planificación de un sistema celular (sistemas de telefonía móvil p. ej.) es importante el calculo del área de cobertura de cada una de las estaciones base que forman la red y el terminal móvil. Para el cálculo de estas áreas se recurre a un análisis del balance de potencia que existe entre la estación y el terminal. El radio teórico de la celda vendrá dado por la distancia máxima para la cual la potencia recibida tanto en el terminal como en la estación basa son mayores que un cierto umbral llamado sensibilidad.

Los parámetros que intervienen en un balance de potencias entre un transmisor y un receptor (*BS* y *MS*) se representan en el siguiente esquema:

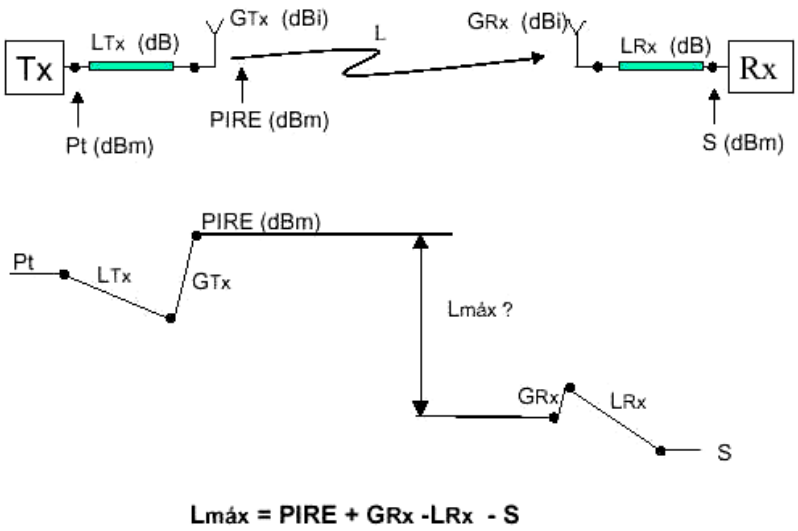


Figura 6: Balance de potencias. Todas las potencias y pérdidas que intervienen en el enlace entre el Tx y el Rx.

Estos son:

Pt: Potencia transmitida por el transmisor

LTx: Pérdidas en el transmisor (debidas a cables, combinadores, …)

GTx: Ganancia de la antena transmisora

PIRE: Potencia Isótropa Radiada Equivalente, que se define como:

$$PIRE\ (dBm) = Pt\ (dBm) – LTx\ (dB) + Gt\ (dBi)$$

L: Pérdidas de propagación. Dependen del tipo de medio (espacio libre, urbano, rural, ….)

GRx: Ganancia de la antena receptora

LRx: Pérdidas en el receptor (debidas a cables, combinadores, …)

S: Potencia umbral o Sensibilidad.

La sensibilidad del receptor es la potencia mínima que debe de haber en el receptor para que la recepción se produzca correctamente, es decir:

$$Pr \text{ (Potencia Recibida en el Rx)} = PIRE - L + GRx - LRx \geq S$$

De está ecuación se obtiene que, en un enlace, las *pérdidas máximas admisibles* se obtienen como:

$$Lmax = PIRE + GRx - LRx - S$$

Con todos los parámetros medidos en unidades logarítmicas.

2.2.- RAGIS: Herramienta de gestión y de cálculo de cobertura radioeléctrica basada en SIG

En este apartado se hace una introducción a la aplicación RAGIS, una herramienta de gestión y análisis de sistemas de radiocomunicaciones basada en el Sistema de Información Geográfica (SIG) ARCVIEW 8.2 y desarrollada por el Grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena. RAGIS aprovecha todas las facilidades de tratamiento de la información de los SIG, y es tanto un software de gestión de bases de datos de sistemas de radiocomunicaciones como una potente herramienta para el cálculo radioeléctrico, incorporando diversos modelos de propagación para entornos abiertos, urbanos e interiores. En los siguientes apartados se comentan algunas de las funciones más interesantes de dicha herramienta y sobre las que se ha trabajado para la realización de este proyecto.

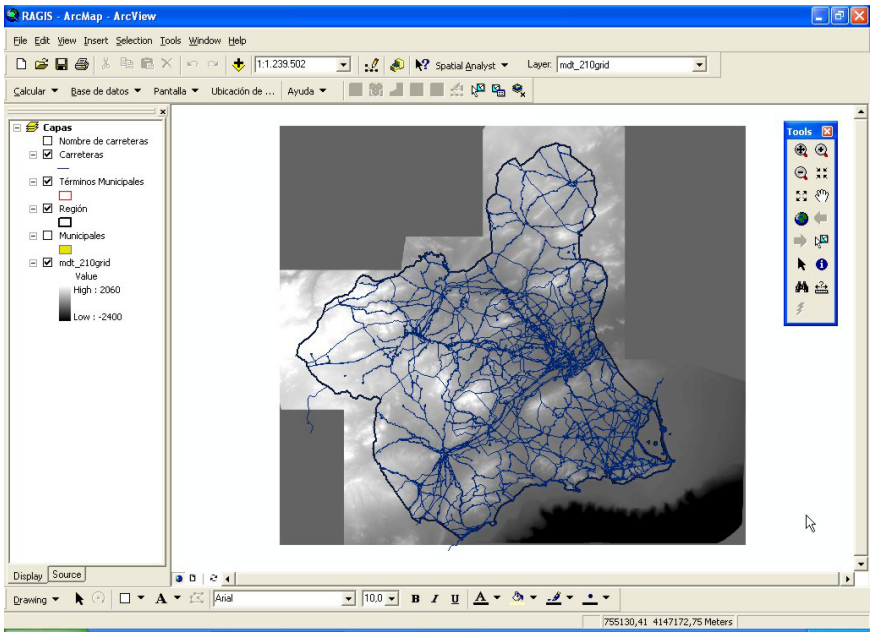


Figura 7: RAGIS. Aspecto general de RAGIS en Arcview 8.2.

Algunas funcionalidades de RAGIS

Gestión de Bases de Datos

RAGIS se comporta como un importante gestor de bases de datos de coberturas radioeléctricas, sistemas de potencia, mapas de densidad de potencia y campo eléctrico, cálculos de porcentajes de cobertura, bases de datos de emplazamientos y campañas de medidas. En todas las bases de datos la herramienta introduce las opciones '*Consultar*', '*Eliminar*' y '*Copiar*' además de algunas otras particulares para cada una de las bases de datos.

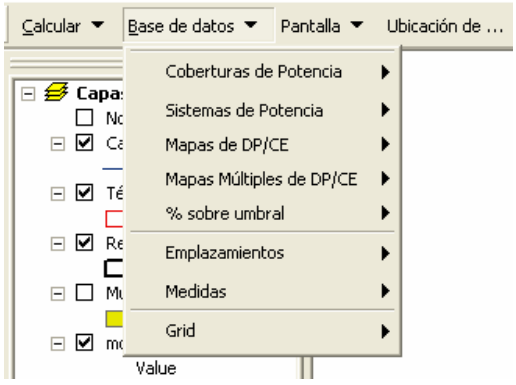


Figura 8: Menú de Bases de datos de RAGIS. Todas las opciones disponibles en el menú de Bases de datos de RAGIS.

La **base de datos de coberturas radioeléctricas** contiene registros con las distintas coberturas radioeléctricas calculadas y guardadas con anterioridad. Cada cobertura consiste en una estructura de tipo *raster* (ver Anexo B al final) donde se almacenan los valores de potencia para cada una de las celdas de la zona donde se ha calculado la cobertura. Esta estructura de tipo *raster* tiene a su vez asociada una estructura de tipo vectorial cuya *tabla de atributos* contiene información relativa a los parámetros que se han utilizado durante la simulación (PIRE, Sensibilidad del receptor, frecuencia, diagrama de radiación de las antenas, orientación del diagrama, modelo de propagación, etc.).

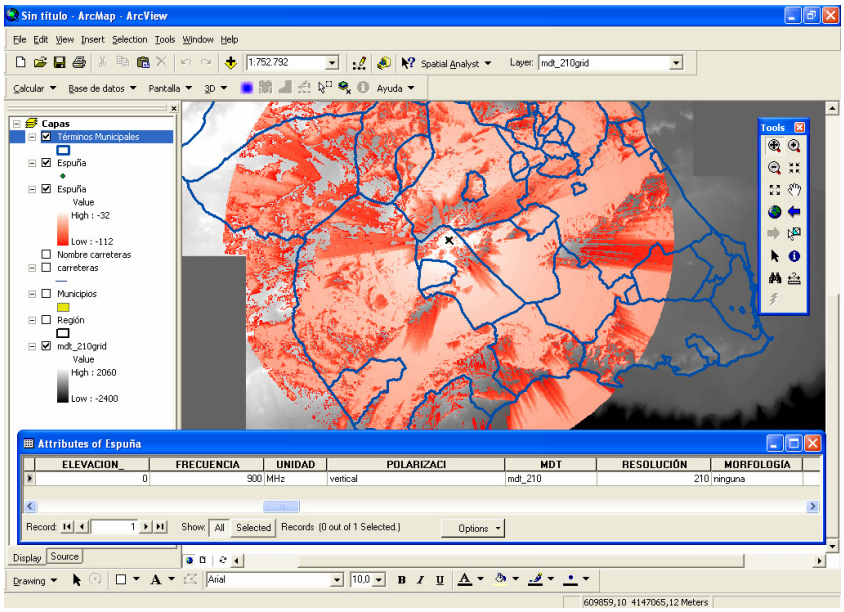


Figura 9: Cobertura radioeléctrica. Mapa de cobertura radioeléctrica de una estación base y su tabla de atributos asociada.

La **base de datos de sistemas de potencia** y **mejores servidores** tiene una estructura semejante a la base de datos de coberturas. Por un lado disponemos de un *raster* donde se almacenan los valores de potencia (sistemas de potencia) para las distintas celdas y por otro lado esta estructura tiene asociada una tabla de atributos donde se incluyen todas aquellas coberturas que forman parte del sistema calculado. En el caso de que, al hacer el cálculo del sistema, también se haya hecho uso de las opciones de '*Calcular mejor Servidor*' o '*Calcular servidores (1º, 2º, 3º, ...)*' (ambas explicadas en capítulos siguientes), esta opción permitirá la visualización de los *rasters* con dichos datos asociados al sistema.

En el caso de la **base de datos de emplazamientos,** el usuario, además de emplazamientos individuales (para cada emplazamiento se genera una *capa vectorial* con su *tabla de atributos* asociada que contiene información relativa a la posición geográfica, etc.), puede definir *grupos de emplazamientos* aprovechando las estructuras de datos que ofrece Arcview 8.2.

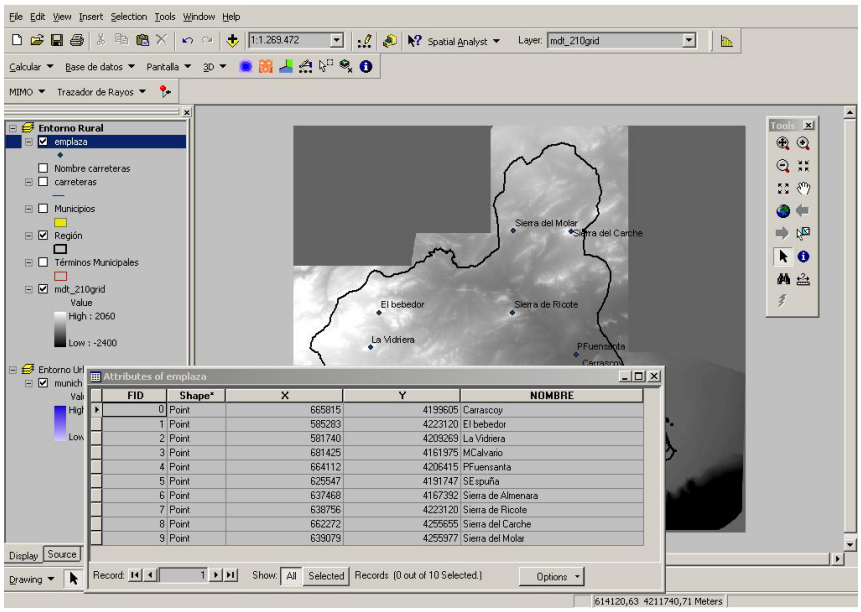


Figura 10: Base de datos de emplazamientos. Capa vectorial de un emplazamiento con su tabla de atributos.

Con la **base de datos de porcentajes** se dispone de un registro de los cálculos que el usuario realice para determinar el grado (porcentaje) de cobertura que se dispone en una determinada zona (cualquier *capa vectorial* como un término municipal, autopista, carretera nacional, etc). Para ello en la estructura de tipo *raster* se almacena información referente a si hay o no cobertura en cada una de las celdas y se le asocian diversas gráficas obtenidas a partir de los resultados calculados.

Por último, cabe indicar que RAGIS también dispone de una **base de datos de medidas**, cuyos registros se guardan en formato de ficheros con extensión *.dbf* importables y editables desde la herramienta. Cargando dichas campañas de medidas el usuario puede visualizar en pantalla los distintos puntos donde éstas se han realizado y, por tanto, seguir el trayecto recorrido durante la elaboración de la campaña.

Cálculos radioeléctricos

Otra de las funciones importantes que presenta RAGIS es la realización de diferentes tipos de cálculos radioeléctricos. Se han implementado, en lenguaje C++, además del *modelo de espacio libre* que puede ser utilizado tanto para entorno abierto como urbano, otros específicos para entorno urbano como los *modelos de Okumura-Hata*, *Walfish-Bertoni* y el *COST-231* mientras que para entorno abierto se dispone de los modelos *UITR-526* y *Meeks*. En entorno *indoor*, se ha implementado un modelo basado en *trazado de rayos*.

De esta forma, la herramienta permite realizar **cálculos de coberturas de potencia** definiendo todos aquellos parámetros que son necesarios para dicho cálculo. Así el usuario debe ir recorriendo los distintos menús e introducir los valores de PIRE, sensibilidad, frecuencia, zona de cálculo, etc. Una vez realizados dichos cálculos el GIS muestra en pantalla los resultados en forma de capa de tipo *raster* donde para cada celda tenemos el valor de potencia asociado. RAGIS también permite el **cálculo de sistemas de potencia** (Figura 11) para los que el usuario debe determinar, entre otros, cuáles son los emplazamientos que quiere que formen parte de los sistemas.

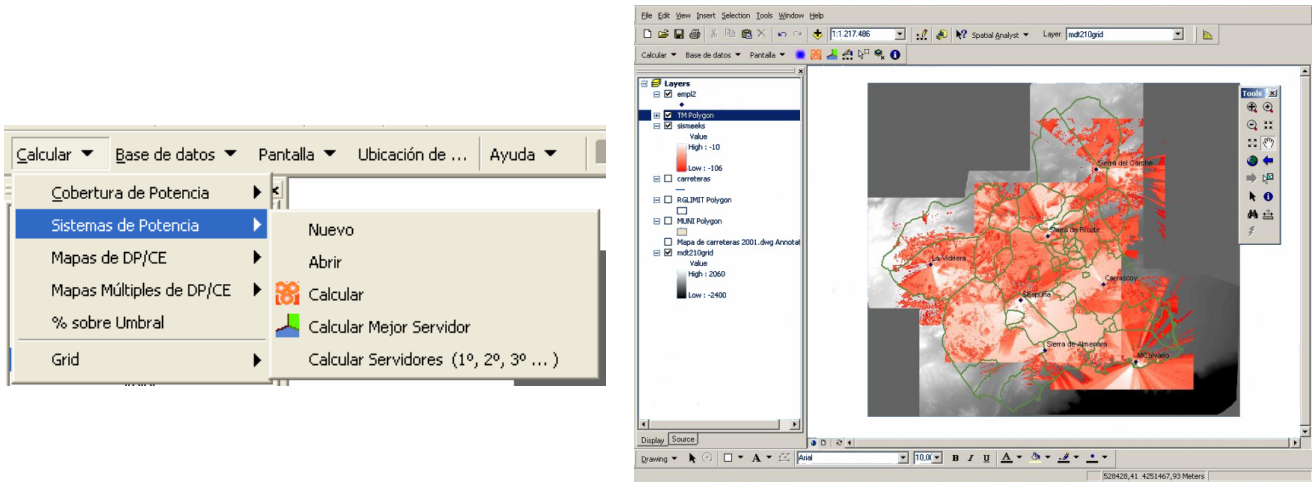


Figura 11: Sistema radioeléctrico. Opciones del menú de cálculo de sistemas y representación de un mapa calculado.

Dentro de este submenú existen las opciones de *'Calcular mejor servidor'* y *'Calcular mejores servidores (1º, 2º, 3º ...)'* que realizan mapas con las bases servidoras que cubren en cada punto de la zona de cobertura. En el primer caso, se representa la estación base cuya potencia es mayor en cada punto (mejor servidor) y en el segundo

caso se representan todas las estaciones base capaces de dar cobertura a la zona de cálculo del sistema. Todo esto es explicado con más detalle en capítulos posteriores.

Asimismo, la herramienta permite realizar el **cálculo de porcentajes de cobertura** sobre una zona determinada. En este caso el usuario debe seleccionar un sistema de potencia o cobertura radioeléctrica y una zona del mapa (representada por una capa vectorial) donde quiera realizar los cálculos de porcentaje y un valor de potencia a modo de umbral sobre el que se calculará el porcentaje de cobertura que lo supera sobre el área total seleccionada. RAGIS genera un mapa con las diferentes zonas de cobertura o no cobertura de cada uno de los emplazamientos que forman el sistema.

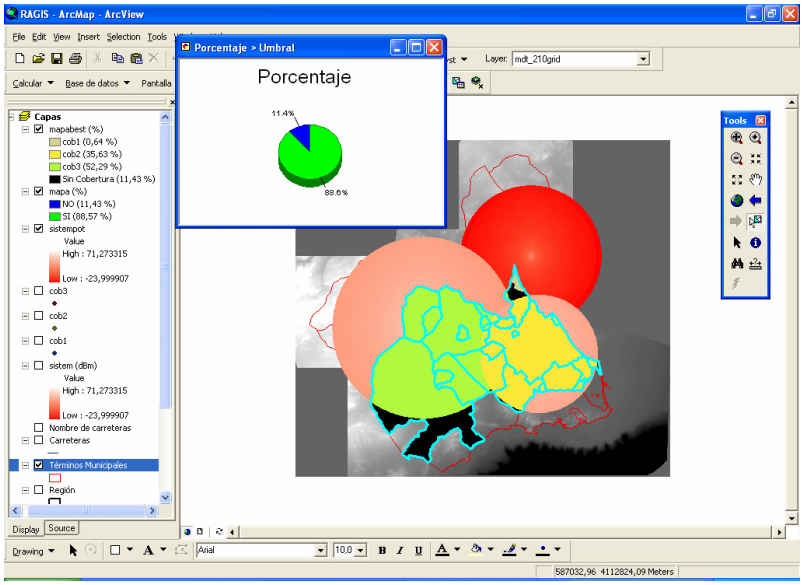


Figura 12: % sobre umbral. Cada color representa a un emplazamiento distinto. Los porcentajes de cobertura se detallan en la ventana de la izquierda.

Algunas de estas funciones se han añadido o mejorado para la realización de este proyecto. Las nuevas funciones añadidas, así como las mejoras en algunas de las ya existentes, se explican de forma detallada en el próximo capítulo.

CAPÍTULO 3.- DESARROLLO DE NUEVAS FUNCIONES PARA RAGIS

Como ya se ha comentado en el primer capítulo, el objetivo de este proyecto era el de añadir nuevas funcionalidades a la herramienta RAGIS. Más concretamente, se ha añadido una nueva función (*Calcular Servidores 1º, 2º, 3º, ...*) y se ha mejorado otra que ya estaba programada (*% sobre Umbral*).

Toda la programación llevada a cabo, se ha basado en el manejo de las coberturas y sistemas radioeléctricos que RAGIS es capaz de calcular y almacenar. Como ya se indicó en el capítulo de introducción a RAGIS, este tipo de estructuras consisten en un tipo de datos denominado *raster*. Por ello, se hace a continuación una pequeña introducción del tipo de programación y datos que utiliza el software ArcGIS, así como la forma en que la herramienta RAGIS maneja las coberturas y sistemas de potencia.

3.1.- El entorno de desarrollo de ArcGIS: ArcObjects

En este caso, para la programación se ha utilizado ArcObjects y Visual Basic para aplicaciones (VBA). ESRI ArcObjects es la plataforma de desarrollo para la familia de aplicaciones ArcGIS como ArcMap, ArcCatalog y ArcScene. ArcObjects es un espacio de trabajo que permite la creación de componentes específicos a partir de un *modelo de objetos*.

La tecnología ArcObjects cumple con las especificaciones COM (*Component Object Model*), y su empleo permite desarrollar nuevas herramientas y funciones, o crear flujos de trabajo para ArcGIS Desktop. También es posible, a través de desarrollos más avanzados, generar aplicaciones independientes que cumplan una funcionalidad concreta, así como añadir clases de elementos personalizadas para extender el modelo de datos de ArcGIS.

Todas las personalizaciones realizadas directamente con ArcObjects, se llevan a cabo a través de Visual Basic para aplicaciones (VBA) o lenguajes de programación que cumplen con las especificaciones COM, como Visual Basic, Visual C++ o Delphi. Además de las personalizaciones básicas que es posible realizar sin ningún desarrollo, en el interfaz de usuario de ArcCatalog y ArcMap, la tecnología ArcObjects permite personalizaciones más avanzadas que pueden agruparse en dos categorías:

Personalización mediante VBA: El empleo de ArcObjects a través de VBA embebido en ArcGIS, permite añadir menús y herramientas personalizadas, así como flujos de trabajo al entorno de trabajo de ArcGIS. La combinación ArcObjects/VBA es una buena opción cuando se quiere desarrollar aplicaciones que se ejecutan en el entorno de ArcGIS Desktop (como es el caso de RAGIS).

Empleo directo de ArcObjects: A través de lenguajes COM como Visual Basic, Visual C++ o Delphi. Este desarrollo permite a programadores ampliar el modelo de datos de la geodatabase con elementos personalizados, y crear módulos de software reutilizables.

Para la realización de este proyecto, se ha utilizado el editor de Visual Basic para trabajar sobre el código ya programado de la herramienta RAGIS e integrar en dicho código las nuevas funciones o variaciones de lo ya programado con anterioridad por otros compañeros.

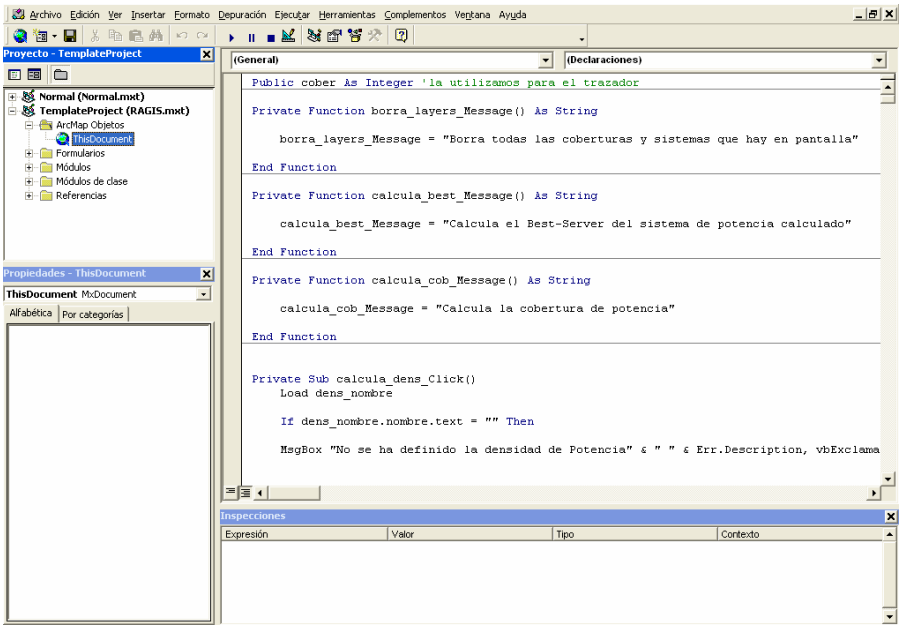


Figura 13: Editor de Visual Basic. Editor de VBA integrado en ArcView, permite la creación de macros para el desarrollo de aplicaciones.

Objetos Raster

Los objetos de datos raster analizan y visualizan datos raster. Éstos proporcionan control, acceso a pixels, herramientas de visualización y están diseñados para trabajar con cualquier grupo de datos raster (raster *dataset* a partir de ahora).

Los tres objetos que pueden acceder directamente a datos raster almacenados en disco son la coclase *RasterDataset*, *Raster*, y *RasterBand*. Entender el funcionamiento de estos tres objetos es fundamental para trabajar con datos raster.

Un *RasterDataset* representa un raster *dataset* almacenado en disco. Se pueden manejar datos raster desde el disco o bien crear objetos *Raster* y *RasterBand* a partir de estos datos.

El objeto *Raster* es una representación virtual de datos raster basados en datos raster grabados en disco. El *Raster* puede ser modificado sin que los cambios se reflejen en los datos guardados en disco, es decir con un raster de este tipo no se está accediendo al disco duro y los datos no son persistentes.

Con un *RasterBand* se pueden leer y escribir datos de píxel, leer estadísticas o tablas de una banda de un raster *dataset* grabado en disco. Los *Raster* y *RasterDataset* están formados por bandas raster.

Las relaciones espaciales de los datos raster se analizan más fácilmente si éstos se visualizan de una forma apropiada. Los objetos *RasterLayer* y *RasterRenderers* controlan las técnicas de visualización que se pueden aplicar a los datos raster.

El objeto *RasterRenderer* controla las técnicas de visualización que pueden ser aplicadas a un raster, mientras que una *RasterLayer* contiene una referencia a una fuente de datos raster para proporcionar instrucciones de cómo pintar esos datos en pantalla.

Otros objetos con otras funcionalidades específicas son el *RasterWorkspaceFactory* y *RasterWorkspace* que permiten abrir los raster *dataset*.

Resumiendo, los objetos *RasterWorkspaceFactory* y *RasterWorkspace* se usan para acceder a un raster almacenado en el disco, representado mediante el objeto *RasterDataset*, y para el análisis de éstos a su vez se crean objetos *Raster* y *RasterBand*. Una vez realizado los análisis y cálculos deseados se visualizan los rasters resultantes mediante los objetos *RasterLayer* y *RasterRenderers*.

Operaciones con datos Raster: Spatial Analyst

Una de las ventajas de los sistemas de información geográfica (SIG) es la capacidad de aplicar operadores espaciales a datos SIG para la generación de nueva información (Figura 14). De los tres principales tipos de datos SIG – raster, vector y tin – la estructura raster es la que proporciona el mejor entorno de modelado y operadores para análisis espacial. *ArcGIS Spatial Analyst* es el conjunto de herramientas de análisis y modelado espacial, que integradas con el núcleo de ArcGIS permiten realizar todo tipo de análisis utilizando información raster.

Para utilizar las funcionalidades de Spatial Analyst a través de código se utiliza el lenguaje llamado *Map Algebra*, el cuál a través de ArcObjects e instrucciones específicas permite la generación de datos de salida mediante el análisis de datos de entrada. En el caso de este proyecto, se estará siempre tratando de datos raster.

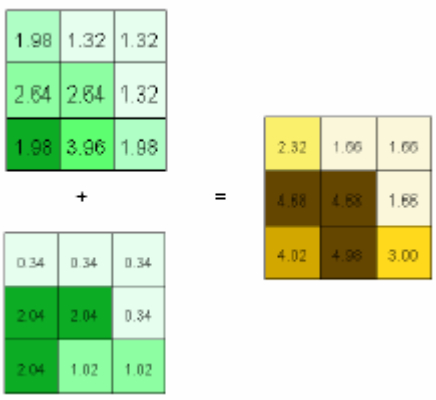


Figura 14: Map Algebra. Con las instrucciones y objetos *Map Algebra* se pueden obtener datos de salida a través de diversos datos de entrada. Para ello se pueden aplicar operadores aritméticos (suma, resta, …), booleanos (and, or, …), relacionales (mayor que, menor que, ….) u otras funciones (con, setnull, …)

3.2.- Coberturas en RAGIS

La herramienta RAGIS permite la creación y almacenamiento de coberturas, sistemas y mapas de Densidad de Potencia\Campo Eléctrico. Todas estas estructuras se crean en formato matricial o *raste*r (llamados *grid* en el software ArcView).

Un *raster* es una estructura de tipo matricial compuesta por celdas del mismo tamaño con un valor determinado almacenado en cada celda. Los valores almacenados en un raster pueden ser de tres tipos: tipo entero (*integer*), reales (*float*) o ausencia de valor (*NO DATA*).

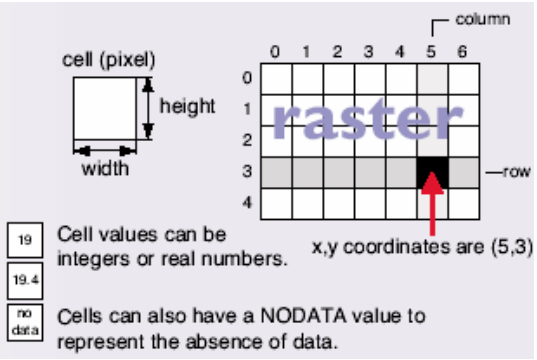
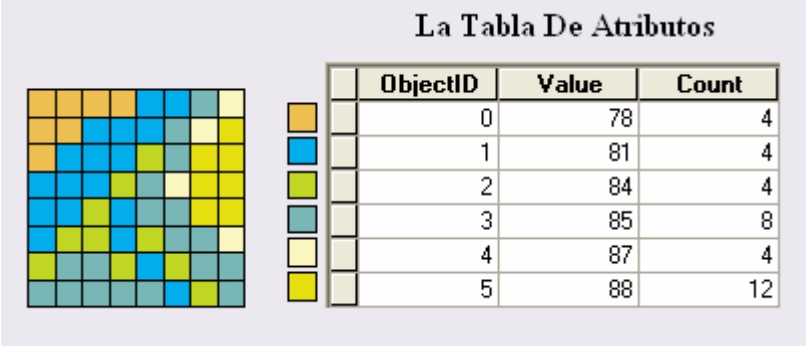


Figura 15: Raster. Estructura de un raster.

Además estos valores quedan reflejados de forma ordenada en la *tabla de atributos* del raster, donde además a cada grupo de valores iguales se le asigna un identificador. Por defecto, la tabla de atributos de un raster tiene tres columnas, la primera con el identificador de objeto (*ObjectID*), la segunda (*Value*) recoge todos los valores diferentes almacenados en el raster y la tercera es un contador de cada uno de estos valores (*Count*) (Figura 16).



ObjectID	Value	Count
0	78	4
1	81	4
2	84	4
3	85	8
4	87	4
5	88	12

Figura 16: Tabla de atributos de un raster. Representación de un raster a través de su tabla de atributos en donde los valores de éste son almacenados en forma de lista ordenada.

Una **cobertura** en RAGIS (Figura 17) se compone de varios rasters que son creados a lo largo del proceso del cálculo de la cobertura y un archivo tipo *shapefile* (archivo vectorial estándar con extensión **.SHP*), en este caso un punto, que representa la localización del emplazamiento, y cuya tabla de atributos asociada contiene todos los datos de éste (coordenadas, PIRE, tipo de antena, ganancia, pérdidas, altura, etc.)

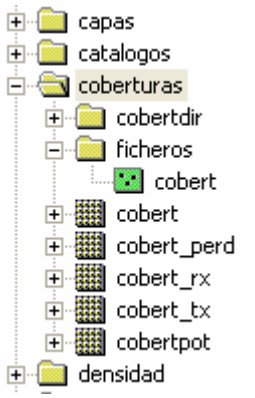


Figura 17: ficheros que componen una cobertura. En ArcCatalog se pueden visualizar todos los datos espaciales como si de simples "archivos" se tratasen. En este caso, la cobertura "cobert" engloba varios rasters con valores de potencia y un shapefile.

En el caso de los **sistemas**, la forma de almacenamiento es análoga a la de las coberturas a diferencia de que éstos, además de los rasters correspondientes al cálculo de potencia, tienen asociados dos rasters que, en lugar de valores de potencia, contienen información sobre el servicio de los emplazamientos que forman el sistema (Figura 18).

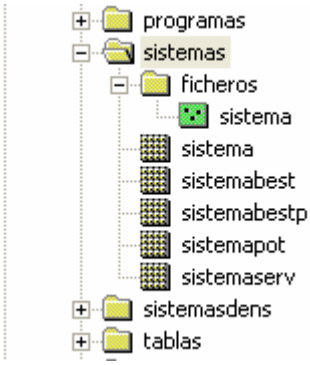


Figura 18: ficheros que componen un sistema. El sistema "sistema" además de los rasters de potencia, tiene asociados los rasters "sistemabest" y "sistemaserv" ya que se han usado las funciones de cálculo de mejores servidores. El *shapefile* contiene información de los emplazamientos que forman el sistema.

El cálculo de las coberturas y sistemas se realiza sobre un Modelo Digital del Terreno o *mdt*. En este caso el *mdt* utilizado corresponde a una representación de un entorno rural y un entorno urbano. Dicho *mdt* está compuesto por dos capas raster, '*urbano_6'*, de una resolución de 6 metros (celda de tamaño 6m x 6m), con la representación de un entorno urbano (alturas de edificios), y la capa *'rural_200'*, de resolución 210 metros, con una representación de un entorno rural (altura del terreno). Además de estos rasters, en el mapa hay dos capas con información vectorial: *Urbano* (delimitación de edificios) y *Términos Municipales* (delimitación de términos municipales de territorio rural).

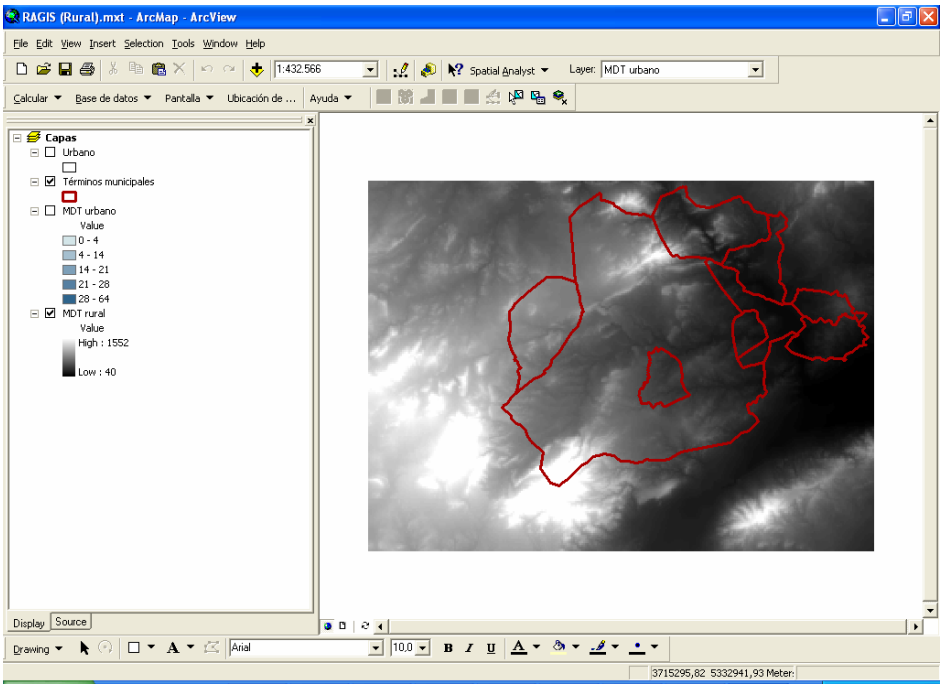


Figura 19: MDT. Capas raster de entorno rural y entorno urbano. Se dispone también de dos capas vectoriales, *Términos municipales* (con delimitación de términos municipales) para el entorno rural y *Urbano* (delimitación de edificios) para el entorno urbano.

3.3.- Cálculo de mejores servidores

La función "*Calcular Servidores (1º, 2º, 3º, ...)*" es un complemento a la opción de cálculo de mejor servidor ("*Calcular Mejor Servidor*") que ya incluía RAGIS y por tanto se ha integrado en el menú de *Sistemas de Potencia*.

Su funcionalidad consiste en la representación de una capa raster que contiene, en cada punto, una lista de las estaciones servidoras que cubren dicho punto. En realidad lo que contiene el raster en cada celda es un valor que representa una posición de una lista que es la que contiene la información de las estaciones servidoras. A continuación se muestra un diagrama de flujo (Figura 20) en donde quedan representados los puntos más importantes y su orden dentro de todo el código programado. Estos puntos se explicarán más adelante, así como las partes del código que hacen referencia a ellos.

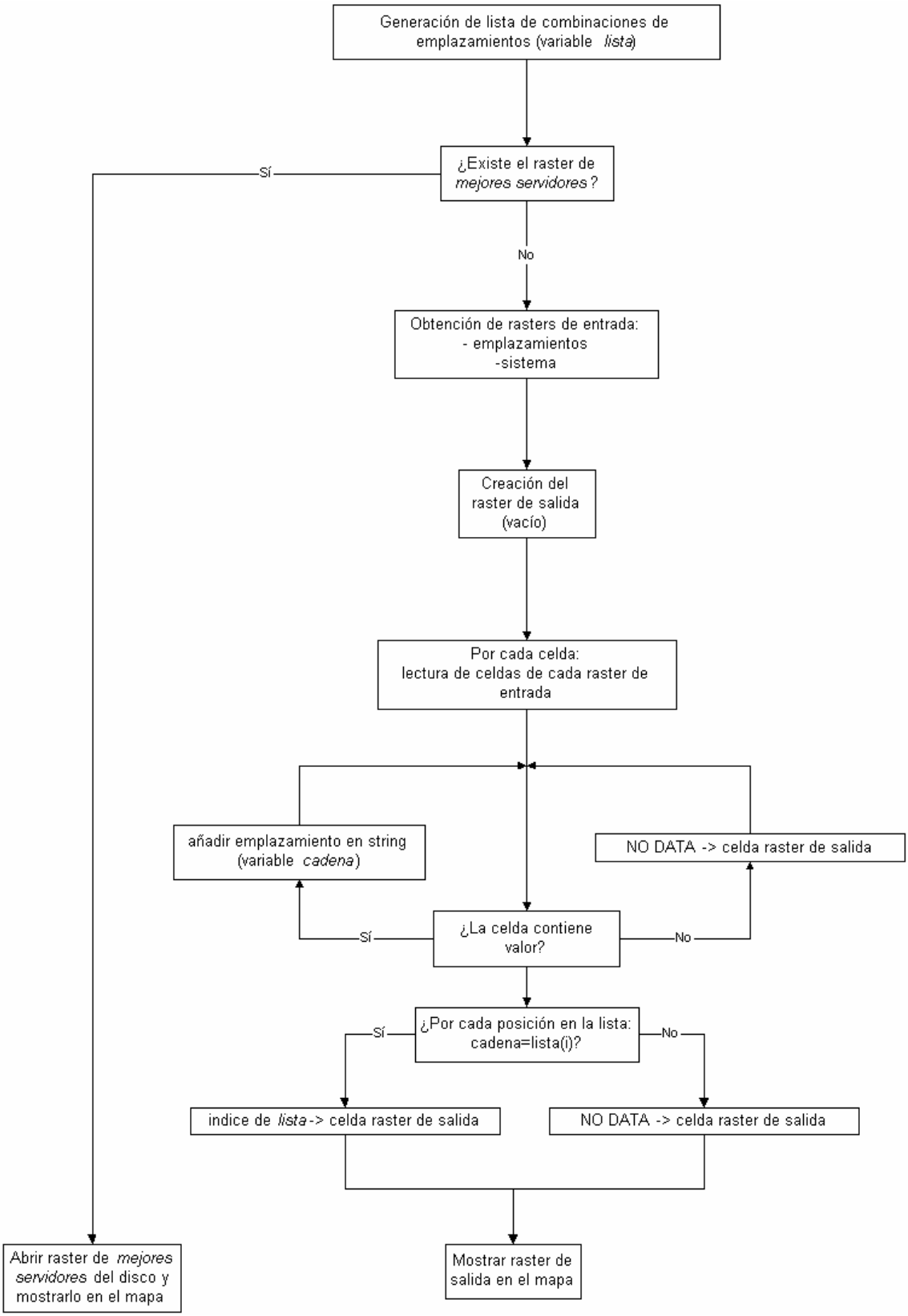


Figura 20: Diagrama de la función mejores servidores. Esquema que muestra los puntos más importantes de la programación de la función de mejores servidores.

La parte del código que realiza el cálculo de los mejores servidores de un sistema se engloba en un procedimiento llamado `M_servidores()` dentro del módulo `sistem`.

En primer lugar, se procede a la creación de una lista con todas las posibles combinaciones que pudiera haber entre los emplazamientos que forman el sistema. Dichas combinaciones se referirán a todas las posibilidades de servicio que puede ofrecer dicho sistema. Veamos ésto con un ejemplo:

Supongamos que nuestro sistema está formado por dos emplazamientos llamados *cobert1* y *cobert2*, la lista de posibles combinaciones entre estos dos emplazamientos sería:

```
cobert1          [1] cobert1
cobert2    ---->  [2] cobert2
                  [3] cobert1, cobert2
```

Es decir, en un determinado punto de la zona de cobertura se podría obtener servicio de cualquiera de los emplazamientos o de los dos a la vez. Como ya se ha comentado en otros apartados, un raster sólo puede almacenar un valor por celda, por tanto, en este caso lo que se almacena en el raster de resultados es el valor del índice de la lista que corresponda para cada punto. En el caso del ejemplo, y para un determinado punto de la zona de cobertura, se almacenaría un 1 si sólo *cobert1* cubriera dicho punto, un 2 si fuera *cobert2* la estación servidora o un 3 si las dos estaciones prestaran servicio.

Llegados a este punto cabe comentar que el algoritmo de cálculo no considera el nivel de servicio de las estaciones, es decir, no se considera el orden en el que se presta dicho servicio. Esto se debe principalmente a la limitación que supone el poder almacenar únicamente un valor por celda en un raster. En el caso de haber considerado el orden de servicio (ordenando los emplazamientos por magnitud de potencia), el número de posibles combinaciones para cada sistema (tamaño de la lista) aumentaría considerablemente llegando el momento en que la representación en el mapa (con un color para cada posible combinación) no sería clara. Por tanto, a la hora de la programación de esta función se ha optado por no considerar dicho orden de servicio.

En el código, dicha lista es en realidad una variable de tipo array (un vector), la variable llamada `lista`, cuyo contenido son strings (cadenas de texto). Cada posición de `lista` corresponde a una posible combinación de emplazamientos. El algoritmo que construye dicha lista se encuentra en la función llamada `Combinaciones()` que recibe

un array con el nombre de los emplazamientos (llamado `cob` en el código) que forman el sistema (de *n* elementos) y devuelve la lista de combinaciones en un array de cadenas de texto (de 2^n-*1* elementos):

```
Dim lista()
ReDim lista(1 To (2 ^ ((UBound(cob) + 1)) - 1))

lista = Combinaciones(cob)    variable con una lista de las posibles
combinaciones de las coberturas que forman el sistema
```

Las cadenas que componen `lista` están ordenadas de menor a mayor longitud (las de menor longitud se encuentran en las primeras posiciones) para una mayor claridad a la hora de la representación de la leyenda del mapa de resultados. De esto se encarga la función `Ordenar()` llamada dentro de la función `Combinaciones()`.

Una vez obtenida dicha lista, nos centramos en la obtención de los datos de entrada que se van a utilizar para el cálculo de los servidores. Dichos datos consisten en los rasters correspondientes a las coberturas de los emplazamientos que forman el sistema y el raster del sistema sobre el que se desee hacer el cálculo.

En este caso, como el algoritmo de cálculo consiste en una comparación celda a celda de todos los rasters de las coberturas que forman el sistema, se hace necesario que estos rasters sean del mismo tamaño, es decir, que tengan el mismo número de filas y columnas. Por defecto, las coberturas calculadas en RAGIS se guardan cada una con un tamaño diferente dependiendo de su extensión espacial. Una manera de conseguir que todos los rasters con los que se quiere trabajar tengan el mismo tamaño es utilizando la propiedad `SetExtent` del objeto `Envelope`. Con estos objetos se puede 'extender' el espacio de análisis, y por tanto el de los datos de salida que se obtendrán después de los cálculos necesarios, para que los límites de éste sean los apropiados y engloben a todos los rasters de entrada (Figura 21).

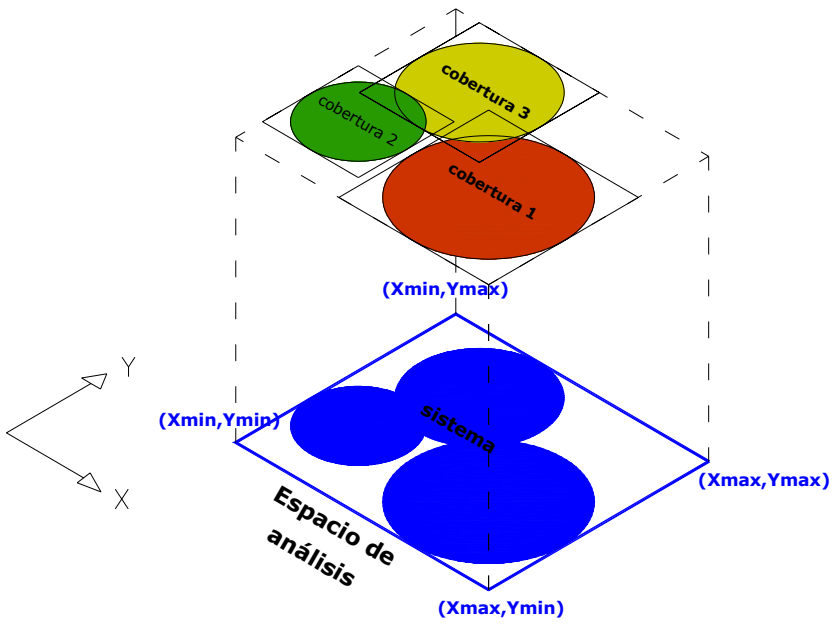


Figura 21: Envelope y SetExtent. Mediante el método `SetExtent` invocado sobre el objeto `Envelope`, se pueden especificar los límites del espacio de análisis y por tanto del raster de salida.

Para la obtención de los límites del espacio de análisis, se comparan los extremos de cada uno de los rasters de entrada, para obtener las coordenadas mínimas y máximas de dichos límites:

```
Dim RasProps As IrasterProps

Dim xmin
Dim xmax
Dim ymin
Dim ymax
Dim resolucion


For i = 0 To UBound(cob)   para cada una de las coberturas se obtienen sus límites y se
comparan para obtener los máximos y mínimos

   Set pInputRasterDS(i) = pRWScob.OpenRasterDataset(cob(i))
   Set pInputRaster1(i) = pInputRasterDS(i).CreateDefaultRaster
   Set RasProps = pInputRaster1(i)

   If i = 0 Then

      xmin = RasProps.Extent.xmin
      xmax = RasProps.Extent.xmax
      ymin = RasProps.Extent.ymin
      ymax = RasProps.Extent.ymax
      resolucion = CDbl(RasProps.MeanCellSize.x)

   Else

      If (resolucion > RasProps.MeanCellSize.x) Then resolucion = RasProps.MeanCellSize.x

         If (xmin > RasProps.Extent.xmin) Then xmin = RasProps.Extent.xmin
         If (xmax < RasProps.Extent.xmax) Then xmax = RasProps.Extent.xmax
         If (ymin > RasProps.Extent.ymin) Then ymin = RasProps.Extent.ymin
```

```
        If (ymax < RasProps.Extent.ymax) Then ymax = RasProps.Extent.ymax

      End If

Next
```

Una vez hecho esto, con el objeto `Envelope` se especifican los límites del espacio de análisis de rasters:

```
Dim pExt As IEnvelope
Set pExt = New Envelope

Set pExt = New Envelope

pExt.xmin = xmin
pExt.xmax = xmax
pExt.ymin = ymin
pExt.ymax = ymax

pEnv.SetExtent esriRasterEnvValue, pExt
```

Llegados a este punto, el siguiente paso es el de calcular cuáles de los emplazamientos que forman el sistema son los que cubren cada punto de la zona tratada. Para ello, el procedimiento es el de ir leyendo cada una de las celdas de los rasters de todas las coberturas que forman el sistema. Cuando alguna de estas celdas contiene un valor numérico (es decir un valor distinto a NO DATA), el nombre del emplazamiento es añadido a una cadena. Para este punto, se utiliza el objeto `IPixelBlock` y su método `GetVal` que permite la lectura y escritura de un bloque de celdas de un raster. Los valores de dichas celdas se almacenan en un array bidimensional llamado `SafeArray`. Este método también es utilizado para escribir los valores en el raster de resultados o de salida.

Cada vez que el valor de una celda de un raster de cobertura sea distinto a nulo, el nombre de dicha cobertura es añadido a una cadena de texto con el mismo formato que tendría en la lista de combinaciones anteriormente formada:

```
For k = 0 To pInputRasProps(0).Width - 1     n°de filas

   For j = 0 To pInputRasProps(0).Height - 1   n°de columnas

      If sistemaBlock.GetVal(0, k, j) <> -9999 Then   valor de la celda (k,j)
del raster del sistema

         cadena = ""

         For i = 0 To UBound(cob)
            If (pInputBlock(i).GetVal(0, k, j) <> -9998) Then  valor de la
celda (k,j) del raster de la cobertura i del sistema

               If cadena = "" Then    string donde se van almacenando los
```

```
nombres de los emplazamientos que dan cobertura a la celda (k,j)
                 cadena = cadena + "  " + cob(i)
              Else
                 cadena = cadena + ", " + cob(i)
              End If

           End If

        Next
```

Después, dicha cadena (la variable string llamada `cadena` en el código) es comparada con toda la lista de combinaciones obtenida anteriormente y el índice de la posición de la lista que coincide con la cadena es el valor que se introduce en el raster de salida:

```
        a = 0     contador de coincidencias entre la variable 'cadena' y
'lista'
        For i = 1 To UBound(lista)

           If cadena = lista(i) Then

              servidores_v(k, j) = i  en el vector safearray del raster de
salida se introduce el índice que representa la cadena con los emplazamientos
de cobertura en el punto (k,j)
              a = a + 1
              Exit For

           End If

        Next

        If a = 0 Then: servidores_v(k, j) = servidoresRasProps.NoDataValue

     Else

        servidores_v(k, j) = servidoresRasProps.NoDataValue

     End If

   Next
Next

servidoresRawPixel.Write pOrigen, servidoresBlock   se escribe el safearray
con los índices en el raster de salida
```

Por último, una vez que se ha calculado el raster de salida con los servidores se procede a su representación en el mapa. Para esto se crea una capa, *RasterLayer*, a partir de este raster y se define de tal manera que el raster quede representado con colores discretos. En este punto también se define la forma en la que se representa la leyenda:

```
Dim pRLayer As IrasterLayer
Set pRLayer = New RasterLayer          layer del raster de servidores
pRLayer.CreateFromRaster servidores

Dim pcolor As IRgbColor
Set pcolor = New RgbColor

Dim pUvren As IRasterUniqueValueRenderer   colores discretos para cada valor
del raster
Set pUvren = New RasterUniqueValueRenderer

Dim pren As IRasterRenderer
Set pren = pUvren

Set pren.raster = pRLayer.raster
pren.Update

Dim psym As IColorSymbol

For j = 0 To pUvren.ClassCount(0) - 1

   pUvren.Label(0, j) = ""
   pUvren.Heading(0) = "Servidores"
   pUvren.Label(0, j) = lista(pUvren.Value(0, j, 0))   leyenda
   pcolor.Red = Int(255 * Rnd)
   pcolor.Green = Int(255 * Rnd)
   pcolor.Blue = Int(255 * Rnd)
   Set psym = New ColorSymbol
   psym.Color = pcolor

   pUvren.Symbol(0, j) = psym
Next

pren.Update
Set pRLayer.Renderer = pren

pRLayer.name = nombre

pmap.AddLayer pRLayer
```

En la figura siguiente se muestran los resultados obtenidos sobre un sistema formado por cuatro emplazamientos.

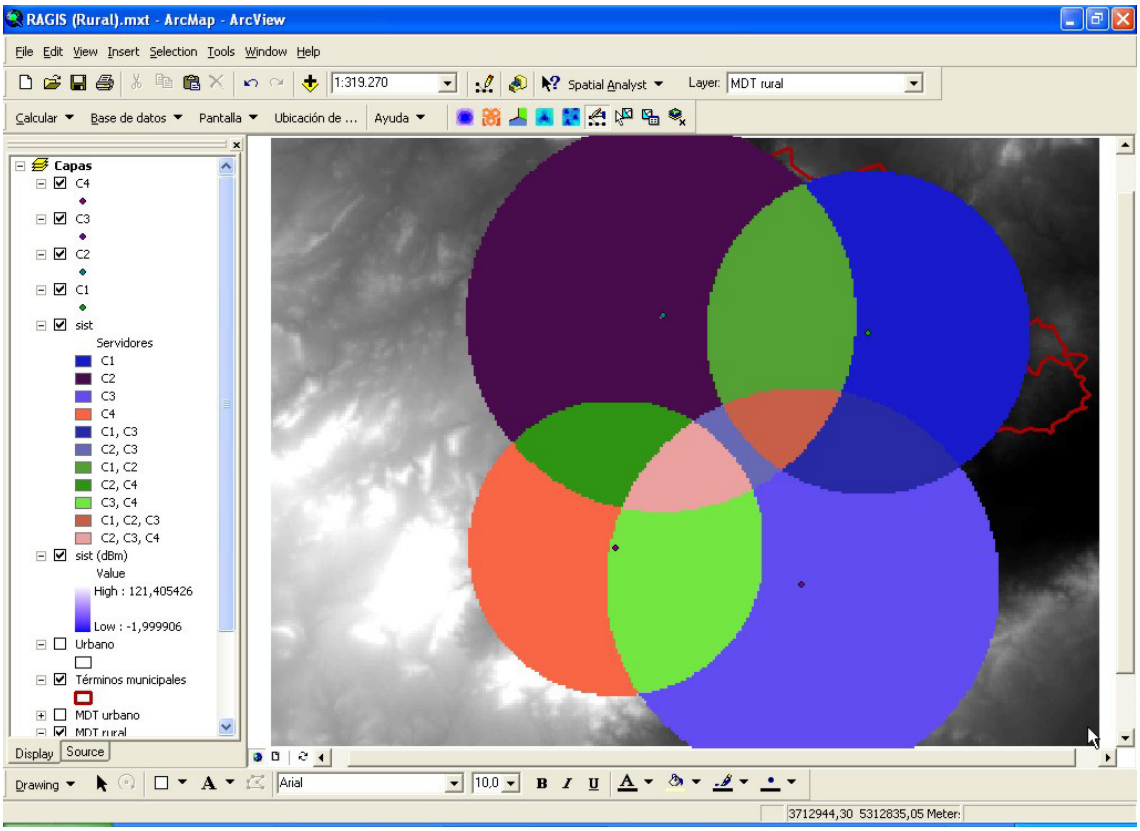


Figura 22: Mejores Servidores. El código de colores indica qué emplazamientos son los que tienen cobertura en cada punto del mapa.

Como se puede comprobar, la lista de todas las posibles combinaciones entre los emplazamientos que forman el sistema se muestra a la izquierda y cada una de dichas combinaciones se representa con un color distinto. Así, las zonas de color rosa corresponden a los puntos cubiertos por las estaciones base *C2*, *C3* y *C4*; las de color azul marino las cubiertas sólo por *C1* y así respectivamente.

Al poder calcular este tipo de raster para cada sistema, se ha añadido también un botón en la ventana de consulta de sistemas de potencia guardados en disco (en el menú *Base de datos*). Al igual que pasaba con el raster de mejor servidor, si los mejores servidores han sido calculados para un determinado sistema, aparecerá activado el botón correspondiente para la visualización de dicho raster en el mapa. De esto se encarga el procedimiento programado `MostrarMejoresServidores()`.

3.4.- Cálculo de porcentajes de territorio con cobertura por encima de un umbral y generación de informe de resultados

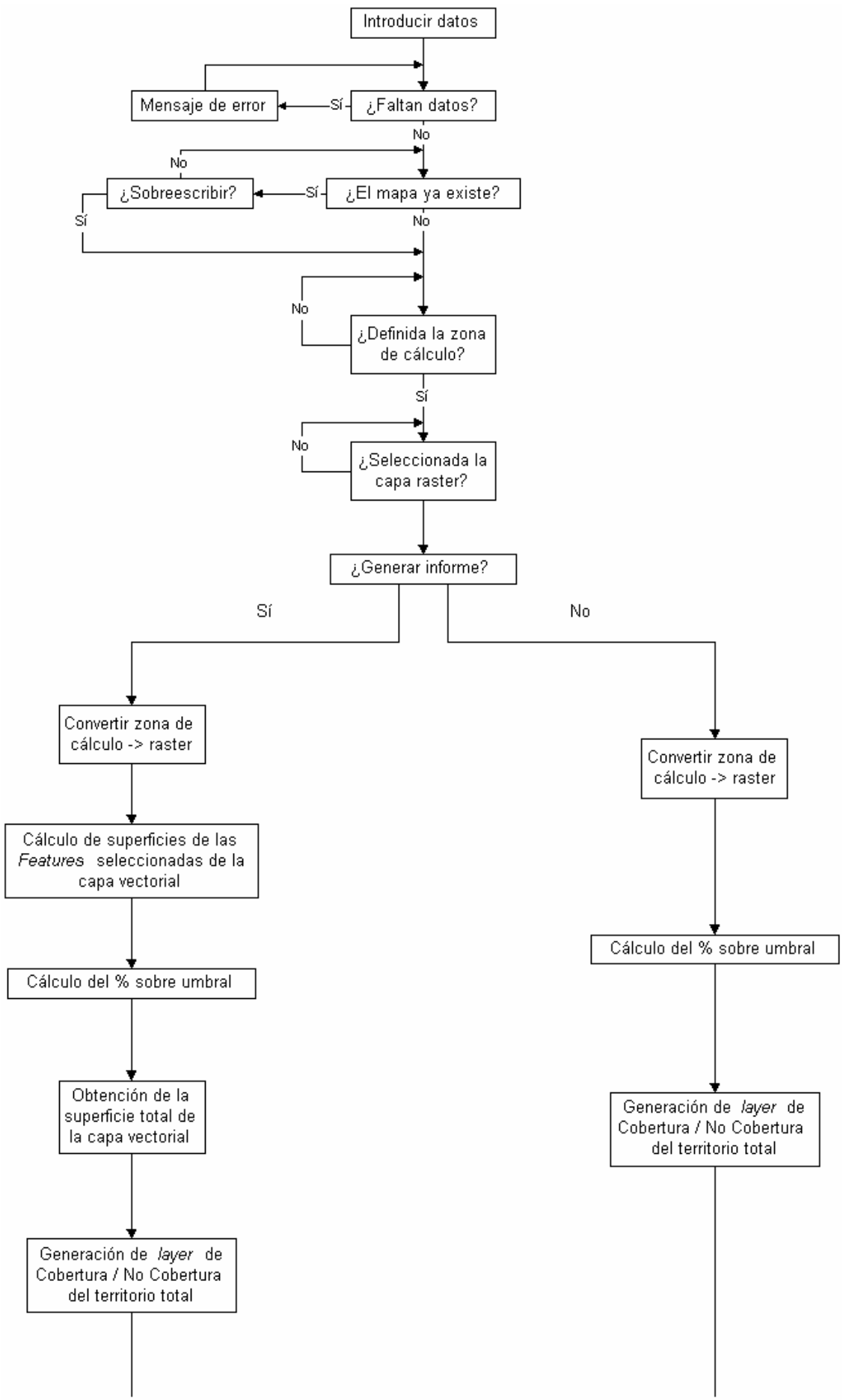
Otra de las funciones programadas es la del cálculo del tanto por ciento de territorio cuya potencia recibida sobrepasa un valor umbral establecido por el usuario. A diferencia de la anterior, RAGIS ya contaba con tal funcionalidad dentro del menú *Calcular* y con este proyecto lo que se ha hecho ha sido ampliarla.

La función de *% sobre Umbral* calcula el porcentaje de territorio (una capa vectorial) cuya potencia recibida queda por encima de un valor umbral establecido por el usuario. Para calcular dicho porcentaje se utiliza la potencia de una cobertura, sistema o cualquier capa raster guardada en disco, y se realiza sobre un área del mapa. Dicha área del mapa puede ser definida de varias formas mediante información vectorial: en forma de *Rectángulo,* que englobe una pequeña área; en forma de *Polígono*, igual que el rectángulo pero con forma arbitraria; *Polilínea*, la zona a tratar se toma como una línea, y en forma de *Capa vectorial*, en la que el cálculo se realizaría sobre una de las capas vectoriales que el usuario tuviera cargadas en el mapa en ese momento.

Por otro lado, la capa raster que se utiliza para el cálculo de la potencia puede ser una cobertura de potencia, sistema de potencia, mapa individual de DP/CE, mapa múltiple de DP/CE u otro tipo de capa raster que se encuentre guardada en disco. Por último, el usuario ha de introducir el valor de potencia que se consideraría como umbral.

La funcionalidad que se ha añadido a esta opción es la de la generación de un informe en formato Excel donde se recoge diversa información sobre los porcentajes de territorios de cobertura calculados. Como la generación de dicho informe implica la creación de datos rasters adicionales en los archivos temporales de RAGIS, se ha optado por permitir al usuario la opción de no generar dicho archivo. De esta manera, si el usuario decide no generar el informe, el programa actuará de la misma forma que hacía hasta ahora, utilizando las funciones programadas anteriormente a este proyecto. Si por el contrario, el usuario decide crear el archivo, el programa utilizará las nuevas funciones programadas y que se explicarán a continuación.

Al igual que en la función anterior se muestra a continuación el diagrama de flujo que sigue el programa a modo de representación en forma esquemática de los pasos más importantes y su orden de ejecución dentro del código. Cada uno de estos puntos es explicado con más detalle a lo largo de este apartado.


Introducir datos
¿Faltan datos?
Sí
Mensaje de error
No
¿El mapa ya existe?
Sí
¿Sobreescribir?
No
Sí
No
¿Definida la zona de cálculo?
No
Sí
¿Seleccionada la capa raster?
No
¿Generar informe?
Sí
No
Convertir zona de cálculo -> raster
Cálculo de superficies de las *Features* seleccionadas de la capa vectorial
Cálculo del % sobre umbral
Obtención de la superficie total de la capa vectorial
Generación de *layer* de Cobertura / No Cobertura del territorio total
Convertir zona de cálculo -> raster
Cálculo del % sobre umbral
Generación de *layer* de Cobertura / No Cobertura del territorio total

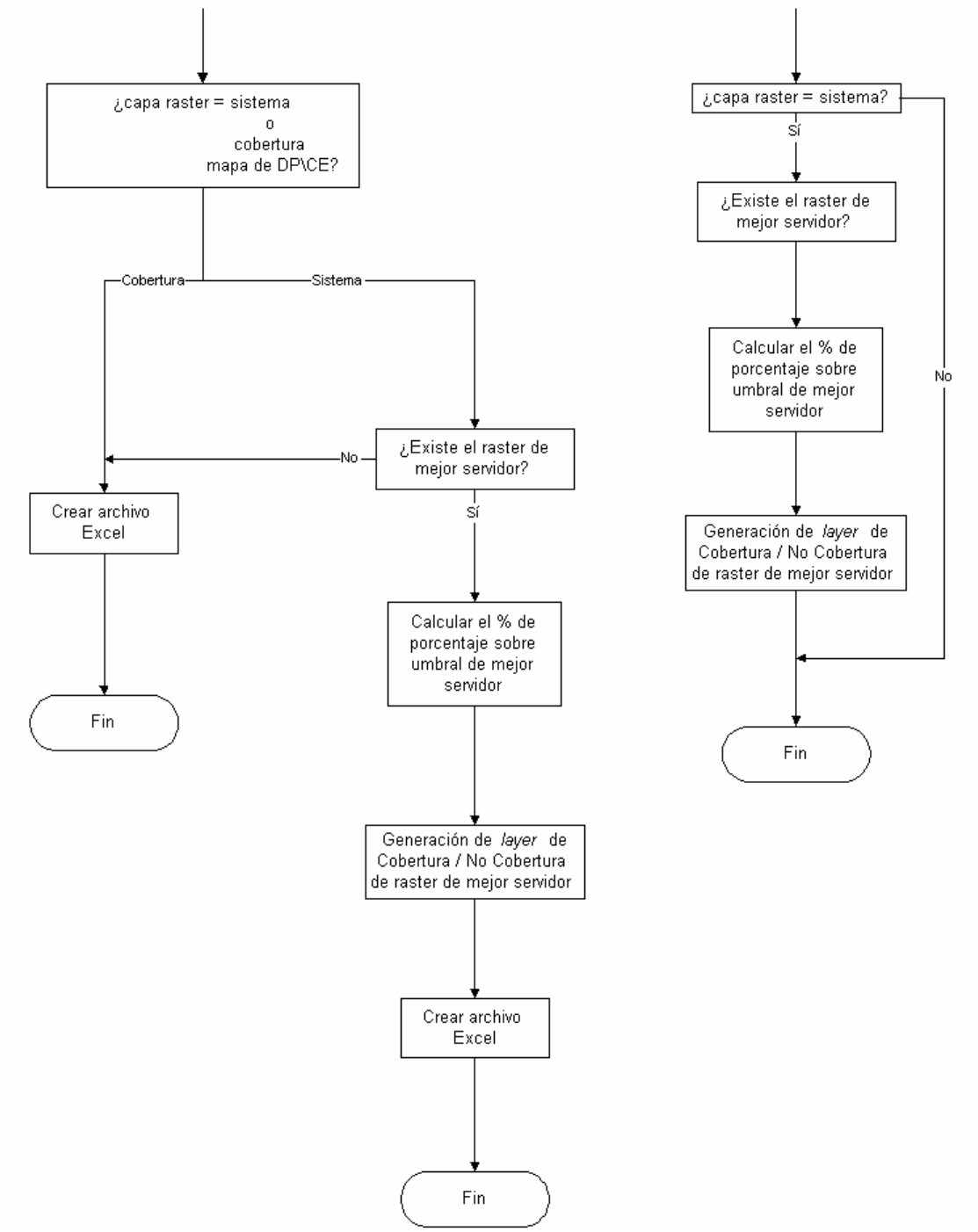


Figura 23: Diagrama de la función % sobre umbral. Esquema que muestra los puntos más importantes de la programación de la función de % sobre umbral.

Al seleccionar la opción *% sobre Umbral* en el menú, aparece la ventana de introducción de datos (Figura 24). En ella, el usuario debe de introducir el nombre del mapa que se va a calcular, la opción de generación o no de informe de resultados (que de aquí en adelante se considerará siempre afirmativa), el tipo de zona de cálculo, el tipo y nombre de la capa raster y el valor de potencia que se tendrá como umbral.

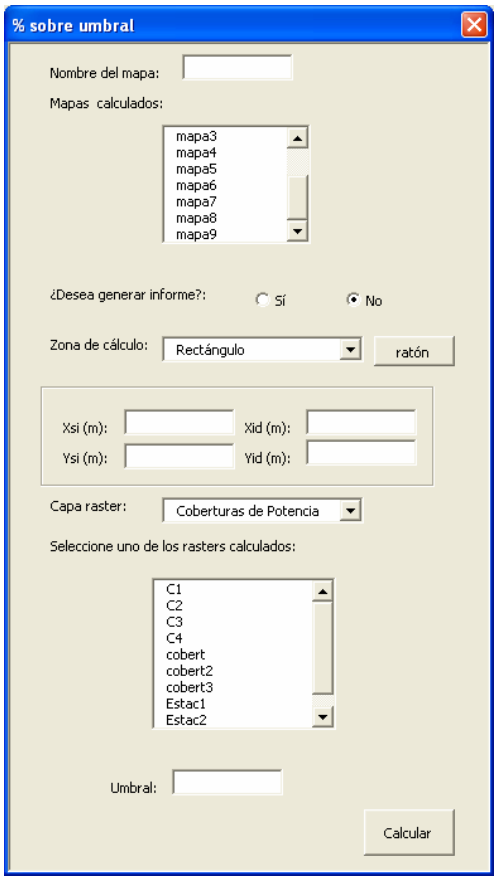


Figura 24: % sobre umbral. Ventana donde el usuario introduce los datos para el cálculo del % sobre umbral.

Como ya se ha comentado, las zonas de cálculo sobre las que se calcula el porcentaje de territorio que tiene cobertura son capas de información vectorial y pueden ser de diversas clases. El usuario en la ventana de introducción de datos, y mediante un menú despegable, puede elegir como zona de cálculo:

Rectángulo: La zona de cálculo queda definida mediante un rectángulo que puede introducirse mediante el ratón o directamente introduciendo las coordenadas de sus límites.

Polígono: La zona de cálculo queda definida como un polígono que puede ser irregular. El usuario utiliza el ratón para ello.

Polilínea: La zona de cálculo queda definida por una línea. Esta opción es útil en el caso de querer saber el porcentaje de tramo de por ejemplo una carretera que es cubierto por una estación base.

Capa Vectorial: La zona de cálculo es una capa vectorial que se encuentra cargada en el mapa. En este caso, aparece un segundo menú despegable para seleccionar

la capa deseada y una vez hecho esto, con el ratón, el usuario debe seleccionar las *features* (Ver Anexo B al final) que le interesen de dicha capa.

En este mismo punto, si se ha seleccionado con anterioridad la opción de generar un informe de resultados, aparecerá activado un tercer menú despegable. El contenido del informe de resultados consiste en tablas y gráficos con distintos porcentajes de territorio calculados sobre cada una de las *features* seleccionadas de la capa vectorial a su vez también seleccionada por el usuario. Para la realización de dichas tablas se utiliza un identificador para cada *feature*. Dicho identificador es el contenido de la columna de la tabla de atributos de dicha capa vectorial que escoja el usuario a través de dicho menú despegable.

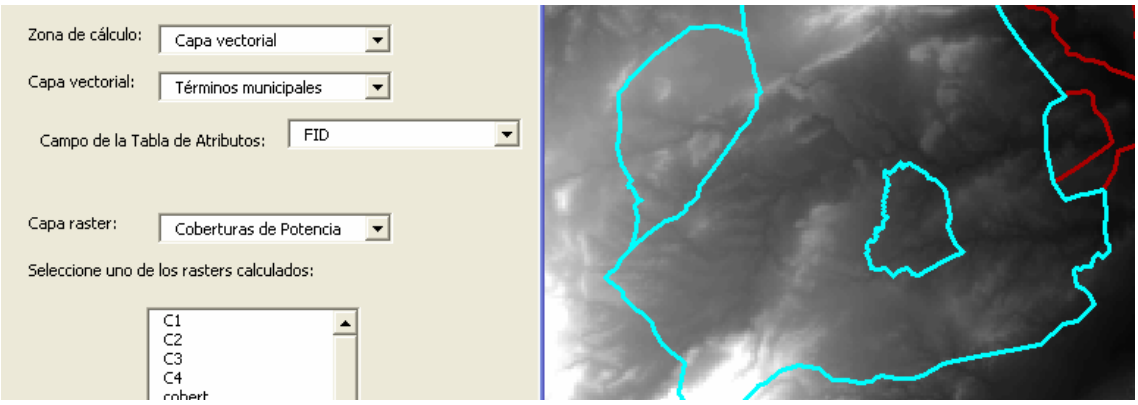


Figura 25: Selección de capa vectorial y *features*. En este caso, se selecciona una capa vectorial del mapa y después las *features* que interesen para la zona de cálculo. Si además se ha elegido la opción de generación de informe de resultados, se ha de seleccionar un campo de la tabla de atributos de la capa vectorial, para la posterior identificación de cada *feature* en dicho informe. En este caso, el identificador sería el contenido del campo FID de la tabla de atributos de la capa vectorial *Términos Municipales* para cada *feature* (municipios en este caso) seleccionada (en azul en el mapa).

Una vez que se ha comprobado que el usuario ha introducido todos los datos necesarios para el programa, se pasa la información vectorial seleccionada (zona de cálculo) a información raster, para poder operar posteriormente con los rasters seleccionados por el usuario (coberturas o sistemas de potencias) y los creados a partir de las capas vectoriales. En el caso de zonas de cálculo de tipo Rectángulo, Polílinea y Polígono, esto se hace mediante el procedimiento `Vect_poliARaster()`. Para ello, este procedimiento utiliza el método `ToRasterDataSet` que crea un raster a partir de la capa vectorial seleccionada. Lo que hace este procedimiento es crear un primer raster a partir de la capa vectorial seleccionada con ceros como valores de celda. Para evitar esto, se cambian dichos valores a uno y se borra de disco el primer raster:

```
se crea un primer raster a partir de la capa vectorial seleccionada.
Éste tiene  ceros como valores de celda
Set prds = pConvOp.ToRasterDataset(pfeatcls, "GRID", pRWS, "1")


Dim palgbop As IMapAlgebraOp
Set palgbop = New RasterMapAlgebraOp


Dim pEnv As IRasterAnalysisEnvironment
Set pEnv = palgbop


Set pEnv.OutWorkspace = pRWS

Dim pRaster As IRaster
Dim prds2 As IRasterDataset
Set pRaster = prds.CreateDefaultRaster

Call palgbop.BindRaster(pRaster, "raster")


se pasan todos los valores del raster a uno
Dim out As IRaster
Set out = palgbop.Execute("con( [raster] <> 1, 1, 1)")


Set prds2 = pConvOp.ToRasterDataset(out, "GRID", pRWS, nombre)

Call palgbop.UnbindRaster("raster")
Set pRaster = Nothing
Set out = Nothing

se borra el primer raster creado
Call borra.grid("c:\arcgis\RAGIS\temp\", "1")
```

Si la zona elegida es de tipo Capa Vectorial, no sólo se crea un raster de la capa vectorial completa (llamado *territorio total* a partir de ahora) sino que se crean tantos rasters como *features* de la capa vectorial se hayan seleccionado. Esto la hace el procedimiento `VectARaster()`. Para crear el raster del territorio total durante su ejecución se llama a la función `FeatTotal()`:

```
Array con los campos de la tabla de atributos seleccionados de cada
una de las features seleccionadas
ReDim capas(pFSel.SelectionSet.count - 1)

Dim pScratchFC As IFeatureClass
Set pScratchFC = FeatTotal(pFlayer.FeatureClass, pFSel, campo, capas)
```

Esta función toma cada una de las *features* seleccionadas y las "une" para formar una capa vectorial de todas ellas. A su vez, va introduciendo el identificador correspondiente a cada *feature* en un array de cadenas llamado `capas`:

```
    Dim pInFeat As IFeature
    Dim pOutBuff As IFeatureBuffer

    Set pOutBuff = pScratchFC.CreateFeatureBuffer
    Set pInFeat = pInFCur.NextFeature

    Dim l As Long
    Dim i As Integer
    Do While Not pInFeat Is Nothing
        For l = 0 To pOutBuff.Fields.FieldCount - 1
             pOutBuff.Value(l) = pInFeat.Value(l)
             If pOutBuff.Fields.field(l).name = campo Then
                  la variable 'cadena' se va llenando con los campos de cada
                  feature
                  cadena(i) = pInFeat.Value(l)
                  i = i + 1
             End If
        Next l
        cada una de las features se van añadiendo a un buffer de features para
        unirse y convertirse en la feature del territorio total
        pOutFCur.InsertFeature pOutBuff
        Set pInFeat = pInFCur.NextFeature
    Loop

    pOutFCur.Flush

    Set FeatTotal = pScratchFC
```

Una vez obtenida la *feature* del territorio total es transformada en un raster:

```
    Set prds = pConvOp.ToRasterDataset(pScratchFC, "GRID", pRWS, nombre)

    Dim pRLayer As IRasterLayer
    Set pRLayer = New RasterLayer
    pRLayer.CreateFromDataset prds

    pScratchDS.Delete
```

A continuación de esto, y en todos los casos, se llama al procedimiento `umbral()`:

```
If municipio.figura.text = "Capa vectorial" Then

        convertimos la feature seleccionada en raster
        porcentaje.VectARaster municipio.nombre.text, name, municipio.campos.text, capas

        realizamos los cálculos
        porcentaje.umbral municipio.nombre.text, municipio.zona.text, municipio.zona2.text,
        name, municipio.umbral.text, municipio.campos.text, capas

ElseIf municipio.figura.text = "Rectángulo" Or municipio.figura.text = "Polígono" Or
municipio.figura.text = "Polilínea" Then

        municipio.campos.text = ""
        municipio.zona = ""
        ReDim capas(0)
        capas(0) = municipio.figura.text

        convertimos la feature seleccionada en raster
        porcentaje.Vect_poliARaster municipio.nombre.text, name
```

```
        realizamos los cálculos
        porcentaje.umbral municipio.nombre.text, municipio.figura.text, municipio.zona2.text,
        name, municipio.umbral.text, municipio.campos.text, capas

End If
```

Este procedimiento se encarga del cálculo del porcentaje de territorio con cobertura. Para ello, primero se calculan las superficies de cada una de las *features* seleccionadas. De esto se encarga el método `ObtenerSuperficies1()` que acepta el raster de territorio total creado anteriormente y devuelve cada una de las superficies en el array llamado `superficies`. Dichas superficies se usaran más adelante.

```
Antes de nada, obtenemos las superficies de cada uno de las features para
después calcular los porcentajes de cobertura
Dim superficies()
ReDim superficies(UBound(capas))

Call ObtenerSuperficies1(pinraster2, superficies)
```

Llegados a este punto, se lleva a cabo el cálculo del % sobre umbral. Para la realización de los cálculos con raster se utiliza la extensión *Spatial Analyst* de ArcGis. Para ello se utiliza el método `Execute` que recibe como parámetro de entrada una cadena con la instrucción a llevar a cabo.

```
    Bindamos los rasters y las variables
    Call palgbop.BindRaster(pinraster1, "cober")
    Call palgbop.BindRaster(pinraster2, "muni")

    Dim out As IRaster
    Dim out2 As IRaster
    Dim out3 As IRaster
    Dim out4 As IRaster
    Dim out5 As IRaster

    out  -> ponemos a 1 los puntos donde la potencia sea mayor que el umbral y a 0 el resto
    out2 -> donde out no está definida (NO DATA) ponemos un 0
    out3 -> ponemos a 1 todos los valores del raster del territorio total
    out4 -> multiplicamos out2 y out3 (cobertura->1, no cobertura->0)
    out5 -> multiplicamos out2 y muni (territorio con cobertura)

    On Error GoTo Error

    Set out = palgbop.Execute("con( [cober] >= " & potencia & ", 1, 0)")

    Call palgbop.BindRaster(out, "out")
    Set out2 = palgbop.Execute("con (isnull([out]), 0, [out])")

    Set out3 = palgbop.Execute("con([muni] <> 1, 1, [muni])")

    Call palgbop.BindRaster(out2, "out2")
    Call palgbop.BindRaster(out3, "out3")

    Set out4 = palgbop.Execute("[out2] * [out3]")
```

```
Set out5 = palgbop.Execute("[out2] * [muni]")

Call palgbop.BindRaster(out4, "out4")
Call palgbop.BindRaster(out5, "out5")
```

Los rasters con los que se realizan los cálculos son el de la capa raster (`[cober]` en el código) y el de la capa vectorial obtenido anteriormente (`[muni]`). El raster llamado `[cober]` en el código pertenece a la capa raster seleccionada, cuyos valores de celda representan niveles de potencia recibida o NO DATA en los puntos donde no se reciba potencia. Las celdas del raster `[muni]` contienen valores distintos para cada *feature* diferente y NO DATA en aquellos puntos que no pertenezcan al territorio seleccionado. En la siguiente figura se muestra una representación gráfica de estos rasters:

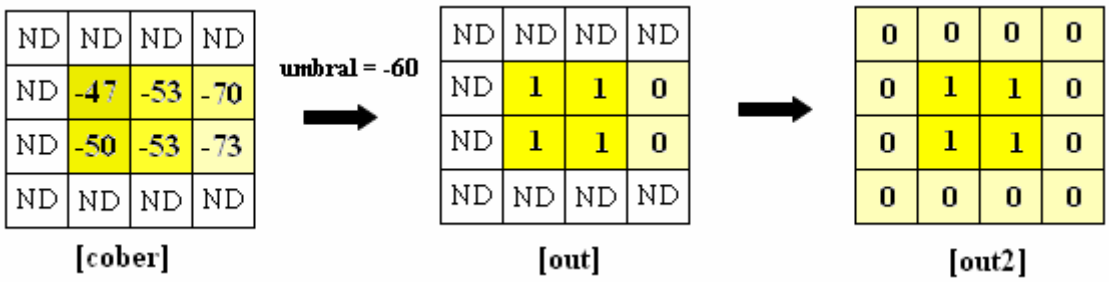
ND	ND	ND	ND
ND	-47	-53	-70
ND	-50	-53	-73
ND	ND	ND	ND

[cober]

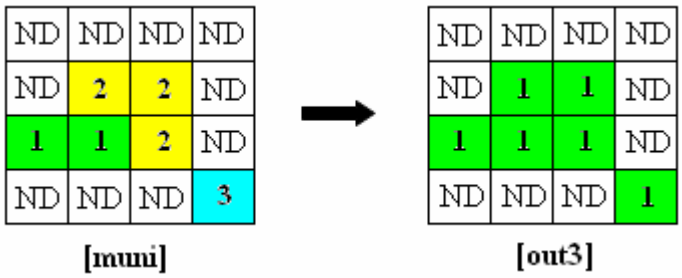
ND	ND	ND	ND
ND	2	2	ND
1	1	2	ND
ND	ND	ND	3

[muni]

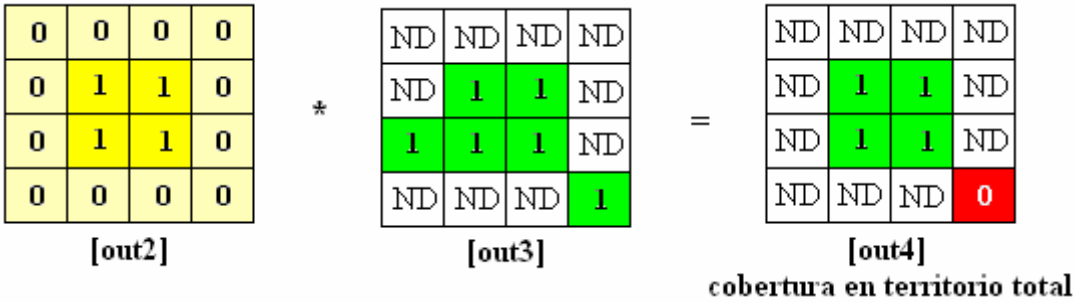
Con la primera instrucción, se cambia el valor de las celdas del raster `[cober]` cuyo valor de potencia sea mayor que el umbral a uno. Con la siguiente instrucción, se pasan las celdas con NO DATA a ceros:



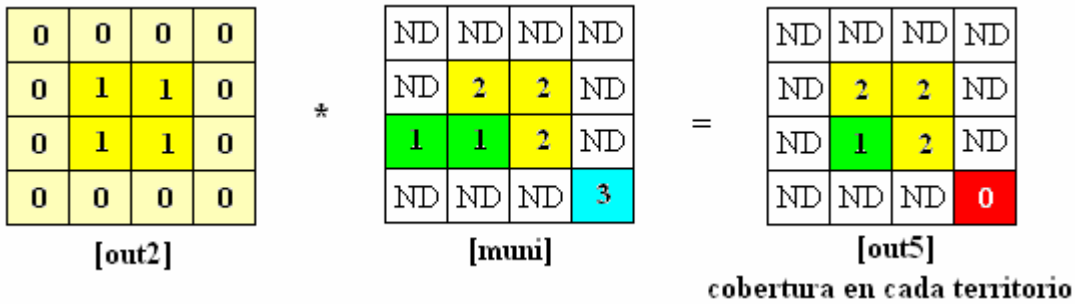
La siguiente instrucción pasa todos los valores del raster `[muni]` a uno:



Para obtener el raster de cobertura del territorio total se multiplican los rasters obtenidos por estas dos últimas instrucciones:



Y para obtener el raster con las coberturas en cada una de las *features*, se realiza la siguiente multiplicación:



Después de esto, se ha obtenido el raster `[out4]` formado por unos, para representar los puntos en donde la potencia recibida es mayor que la introducida como umbral, y por ceros, para los puntos en donde no hay cobertura porque dicha potencia queda por debajo del umbral. Este raster se representa en la pantalla con dos colores, verde para las zonas de cobertura, y rojo para las zonas de no cobertura.

El siguiente paso es comprobar si la capa raster seleccionada es una cobertura o un sistema de coberturas. Si es una cobertura o mapa de DP/CE, se llama al procedimiento `CrearExcel()` al cuál se le pasa como parámetro el raster calculado

anteriormente.

Si es un sistema de coberturas, se comprueba si existe o no el raster de mejor servidor. Si no existe, se llama al método `CrearExcel()` como en el caso anterior. Si dicho raster está calculado para el sistema que estamos tratando, se realizan con él una serie de cálculos análogos a los anteriores y por último se llama al procedimiento `CrearExcel()`. Todo esto queda reflejado en la siguiente parte de código fuente:

```
If tipo = 1 Then    Si es una cobertura, simplemente creamos el excel con lo
ya calculado

        creamos un archivo excel con los datos
        Call CrearExcel(tipo, nombre, c_r, lista, c_v, Format(potencia), campo, capas, out5,
        superficies)

ElseIf tipo = 2 Then   si es un sistema, calculamos el porcentaje de los mejores servidores

        Dim bestserver As New FileSystemObject

        If Not bestserver.FileExists(rutas.sistemas + lista + "bestp.aux") Then
             MsgBox "Para más información, realizar el cálculo del mejor servidor"
             creamos el archivo Excell
             Call CrearExcel(1, nombre, c_r, lista, c_v, Format(potencia), campo, capas,
             out5, superficies)
             Exit Sub
        End If

        Dim rdsbest As IRasterDataset
        Set rdsbest = pWS2.OpenRasterDataset(lista + "bestp")

        Dim pinraster3 As IRaster
        Set pinraster3 = rdsbest.CreateDefaultRaster

        Bindamos los rasters y las variables
        Call palgbop.BindRaster(pinraster1, "sis")
        Call palgbop.BindRaster(pinraster3, "best")

        Dim s_out As IRaster
        Dim s_out1 As IRaster
        Dim s_out2 As IRaster
        Dim s_out3 As IRaster
        Dim s_out4 As IRaster

        s_out->ponemos a 1 donde se supera la sensibilidad y NO DATA donde no
        s_out1->multiplicamos s_out por el mejor servidor
        s_out2->donde s_out1 sea null lo ponemos a 9998 (valor alto aleatorio)
        s_out3->ponemos todos los valores de muni a 1
        s_out4->multiplicamos s_out2 por s_out3

        Set s_out = palgbop.Execute("con( [sis] >= " & potencia & ", 1, setnull([sis]))")
        Call palgbop.BindRaster(s_out, "s_out")

        Set s_out1 = palgbop.Execute("[s_out] * [best]")
        Call palgbop.BindRaster(s_out1, "s_out1")

        Set s_out2 = palgbop.Execute("con (isnull([s_out1]), 9998, [s_out1])")
        Call palgbop.BindRaster(s_out2, "s_out2")

        Set s_out3 = palgbop.Execute("con([muni] <> 1, 1, [muni])")
        Call palgbop.BindRaster(s_out3, "s_out3")

        Set s_out4 = palgbop.Execute("[s_out2] * [s_out3]")
```

```
Call palgbop.BindRaster(s_out4, "s_out4")

Call palgbop.UnbindRaster("s_out")
Call palgbop.UnbindRaster("s_out1")
Call palgbop.UnbindRaster("s_out3")

Set s_out = Nothing
Set s_out1 = Nothing
Set s_out3 = Nothing

Dim s_out5 As IRaster
Dim s_out6 As IRaster

Dim muns() As IRaster
ReDim muns(UBound(capas) + 1)

For i = 0 To UBound(capas)

     Set s_out5 = palgbop.Execute("con( [muni] <> " + Format(i + 1) + ",
     setnull([muni]), 1)")
     Call palgbop.BindRaster(s_out5, "s_out5")

     Set s_out6 = palgbop.Execute("[s_out2] * [s_out5]")
     Call palgbop.BindRaster(s_out6, "s_out6")

     Set muns(i) = s_out6

     Call palgbop.UnbindRaster("s_out5")
     Call palgbop.UnbindRaster("s_out6")

     Set s_out5 = Nothing
     Set s_out6 = Nothing

Next

Call palgbop.UnbindRaster("s_out2")
Set s_out2 = Nothing

Set muns(UBound(capas) + 1) = s_out4
```

En este caso, los rasters de los que se parte para realizar los cálculos son el raster del sistema (`[sis]`) y el de mejor servidor que debe de haberse calculado con anterioridad y estar almacenado en disco (`[best]`). A continuación se muestra una representación gráfica de la forma y valores de dichos rasters:

-47	ND	ND	-70
ND	-47	-53	-70
ND	-50	-53	-73
-50	ND	ND	ND

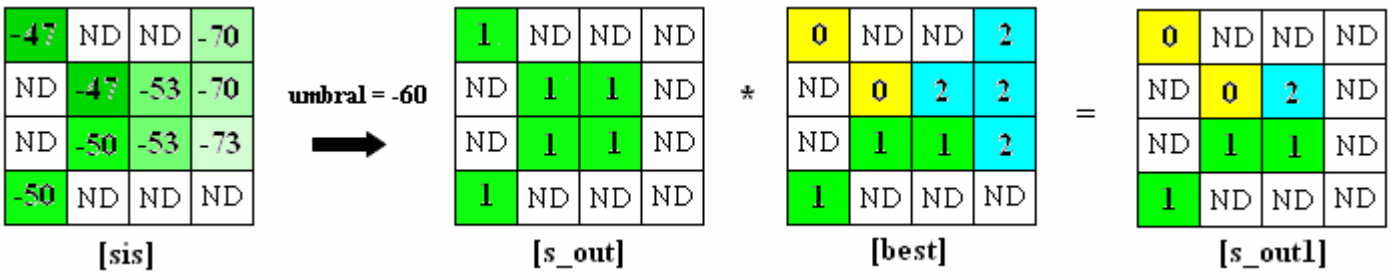
[sis]

0	ND	ND	2
ND	0	2	2
ND	1	1	2
1	ND	ND	ND

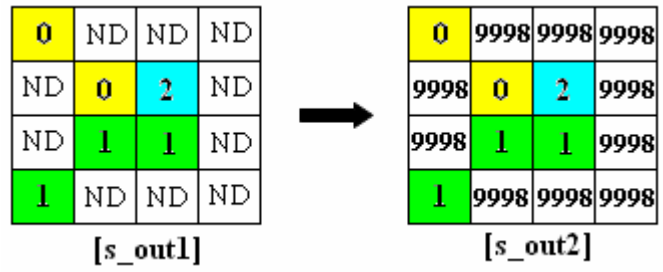
[best]

La primera instrucción pasa los valores del raster `[sis]` que superen el umbral a unos y los demás a NO DATA. Después se multiplica el raster obtenido por el raster de

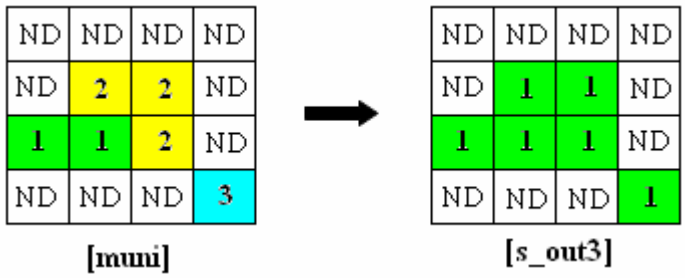
mejores servidores `[best]`:



En la siguiente instrucción se cambia el valor nulo de las celdas (NO DATA) a un valor alto aleatorio (9998):



La siguiente instrucción pasa todos los valores del raster `[muni]` a uno:



La última instrucción multiplica los rasters `[s_out2]` y `[s_out3]` para obtener el tanto por ciento de cobertura como mejor servidor de cada uno de los emplazamientos que forman el sistema elegido sobre el territorio total:

0	9998	9998	9998
9998	0	2	9998
9998	1	1	9998
1	9998	9998	9998

[s_out2]

*

ND	ND	ND	ND
ND	1	1	ND
1	1	1	ND
ND	ND	ND	1

[s_out3]

=

ND	ND	ND	ND
ND	0	2	ND
9998	1	1	ND
ND	ND	ND	9998

[s_out4]

emplazamientos como mejor servidor

Lo siguiente es conseguir un array de rasters (llamado `muns` en el código) cuyo contenido es un raster por cada *feature* seleccionada en donde se encuentren el porcentaje de cobertura de cada emplazamiento actuando como mejor servidor pero para cada territorio por separado. En la última posición de este array se almacena el raster del territorio total obtenido por los cálculos anteriores. Al igual que antes, se muestran los cálculos de manera gráfica para una mejor compresión. Así para la *feature* i-ésima, el raster que se creará tendrá la siguiente forma:

1

ND	ND	ND	ND
ND	i	i	ND
1	1	i	ND
ND	ND	ND	3

[muni]

ND	ND	ND	ND
ND	1	1	ND
ND	ND	1	ND
ND	ND	ND	ND

[s_out5]

2

ND	ND	ND	ND
ND	1	1	ND
ND	ND	1	ND
ND	ND	ND	ND

[s_out5]

*

0	9998	9998	9998
9998	0	2	9998
9998	1	1	9998
1	9998	9998	9998

[s_out2]

=

ND	ND	ND	ND
ND	0	2	ND
ND	ND	1	ND
ND	ND	ND	ND

[s_out6]

Este raster será almacenado en el array `muns`, así como los demás calculados para cada *feature*. Una vez calculados todos los rasters correspondientes a cada *feature*, se introduce en la última posición de `muns`, el raster correspondiente a los cálculos para todo el territorio total. Por tanto el tamaño de dicho array dependerá de la cantidad de *features* que el usuario haya seleccionado en la ventana de introducción de datos.

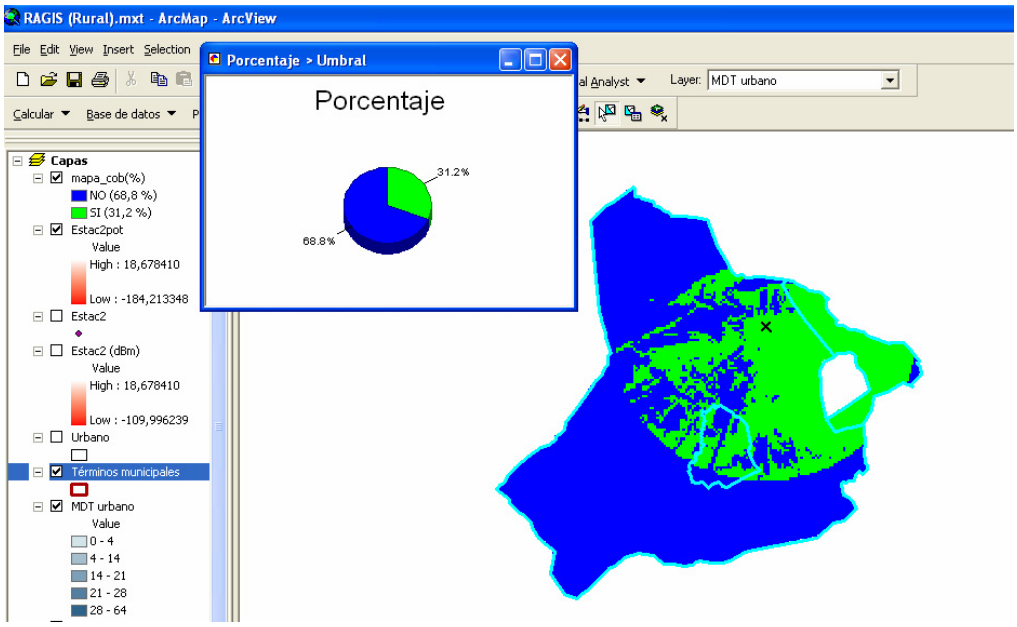


Figura 26: % sobre umbral para una cobertura (aspecto del mapa). En este caso se ha elegido la capa raster *Estac2* y 3 *features* de la capa vectorial *Términos municipales* del mapa. Tanto en el gráfico como en la leyenda se muestra el porcentaje de territorio con y sin cobertura.

El siguiente paso en la ejecución del programa es la de la creación de un informe en formato *.xls* de Excel con diversa información del mapa de cobertura creado, así como de los porcentajes de territorio de las *features* seleccionadas cuya potencia recibida es superior a la potencia umbral establecida por el usuario. Como ya se ha comentado anteriormente, el procedimiento que se encarga de esto se llama `CrearExcel()` y distingue entre dos casos:

1.- En el caso de la que capa raster seleccionada sea una cobertura de potencia, mapa de DP/CE o un sistema en el que no se haya calculado el raster de mejor servidor, se crea el archivo con una hoja en la que se muestran, junto con los datos del mapa calculado, el porcentaje de territorio, para cada *feature* seleccionada, que recibe y que no recibe la potencia sobre el umbral establecido. Como ya se ha comentado para cada *feature* existe un identificador que corresponde al contenido del campo de la tabla de atributos seleccionado por el usuario en la ventana de introducción de datos. Además de los porcentajes de las *features,* se muestra también el porcentaje de territorio total con y sin cobertura por encima del umbral.

Cada una de estas tablas va acompañada por su correspondiente gráfico.

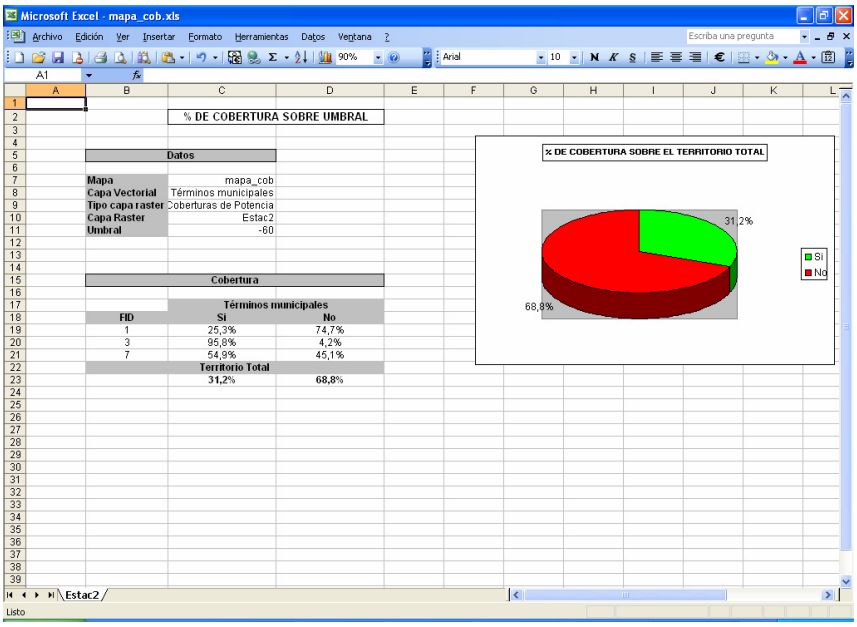


Figura 27: % sobre umbral para una cobertura (aspecto del informe de resultados). Se muestran los datos del mapa calculado, los porcentajes de territorio con cobertura de las tres *features* seleccionadas (en este caso con el campo FID como identificador), el porcentaje del territorio total, así como los gráficos correspondientes a cada tabla.

2.- En el caso de un sistema de potencia en el que se haya calculado anteriormente el raster del mejor servidor, además de la primera hoja con lo comentado antes para las coberturas, se añadirá al archivo una hoja por cada emplazamiento que forme parte del sistema. En cada una de estas hojas se representaran los tantos por ciento de territorio de cada *feature* y del territorio total en donde el emplazamiento en cuestión (uno en una hoja diferente) actúa como mejor servidor y tiene una cobertura por encima del umbral. Al igual que en la primera hoja, cada tabla irá acompañada por un gráfico, bien de barras para las *features* o bien de pastel para el territorio total. Además en este caso, aparece una tabla resumen y un gráfico con los porcentajes totales de mejor servidor en la primera hoja del libro.

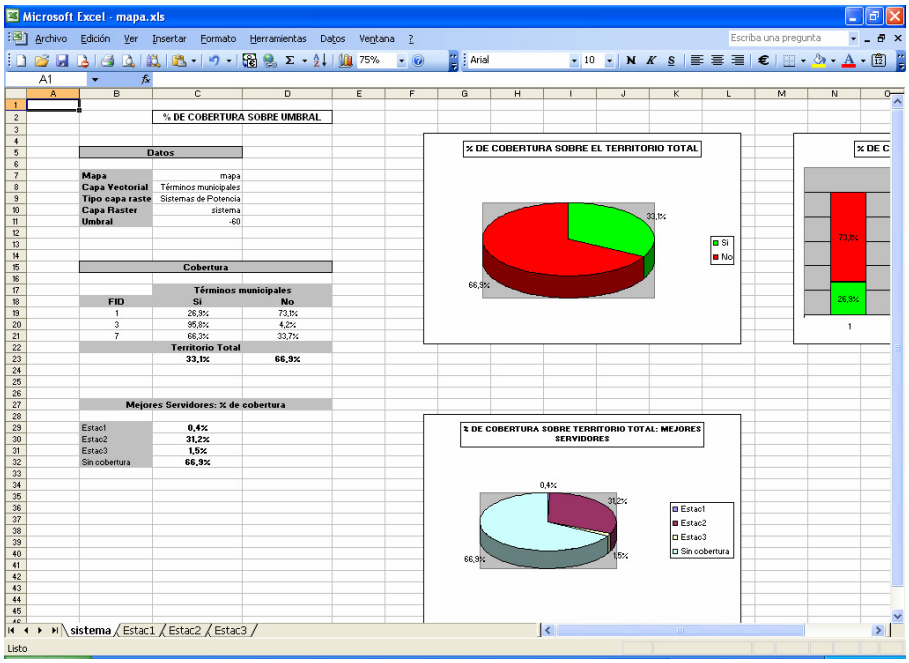


Figura 28: % sobre umbral de un sistema de coberturas (Hoja 1 del informe de resultados). En este caso, el sistema está formado por los emplazamientos *Estac1*, *Estac2* y *Estac3*. En la imagen se pueden apreciar la primera hoja del libro Excel con los datos correspondientes al sistema total y tres hojas correspondientes a cada uno de los emplazamientos.

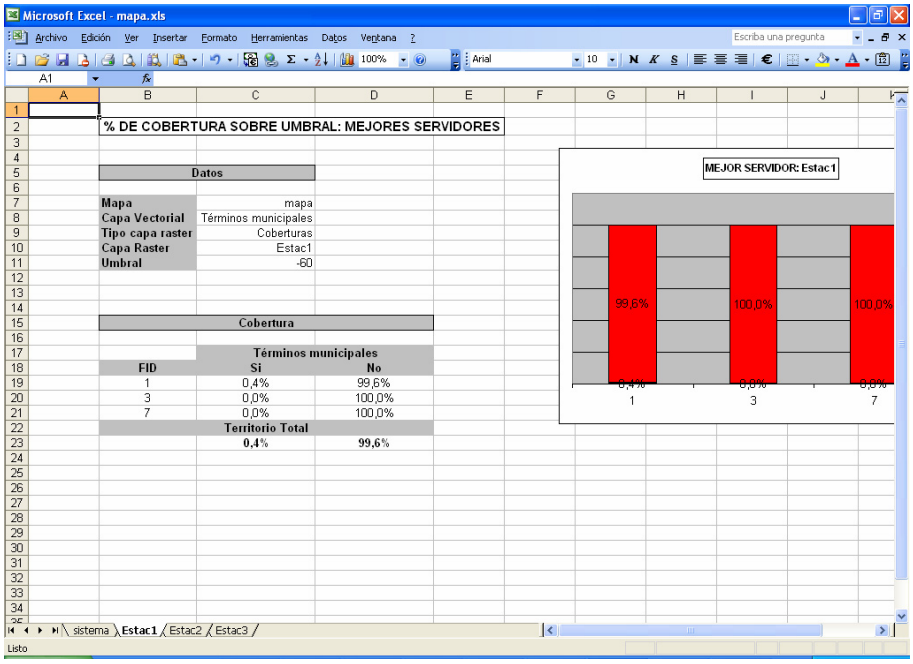


Figura 29: % sobre umbral de un sistema de coberturas (Hoja de emplazamiento del informe de resultados). En esta hoja, se muestran los porcentajes de territorio de cada *feature* y de territorio total que son cubiertas por el emplazamiento correspondiente (*Estac1* en este caso) como mejor servidor.

Dicho archivo es creado en la carpeta correspondiente al mapa calculado contenida a su vez en la carpeta *porcentajes* del sistema de directorios de *arcgis/RAGIS*.

CAPÍTULO 4.- RESULTADOS SOBRE SIMULACIÓN DE COBERTURA RADIOELÉCTRICA GSM

A continuación se realiza sobre RAGIS una simulación de un radieoenlace GSM y se muestran los resultados obtenidos tras la ejecución de las nuevas funciones programadas. Puesto que cualquier cobertura radioeléctrica de comunicación bidireccional viene delimitada por el peor enlace, realizamos primero el balance de potencias de los enlaces descendente, de Estación Base a Móvil, y ascendente, de Móvil a Estación Base. Para ello se han tomado, de catálogos de material y equipos GSM, los siguientes parámetros para la estación base y el móvil:

Transmisión	Unidades	Estación Base	Terminal Móvil
Banda de Frecuencias	MHz	960	915
Potencia de Tx	W	25	2
Tipo de Combinador		1:2	
Pérdidas combinador	dB	3	0
Tipo de duplexor			
Pérdidas duplexor	dB		
Tipo de cable de antena		1/2"	
Atenuación del cable de antena	dB/100m	3,9	
Longitud del cable de antena	m	35	0
Tipo Sistema Radiante (antena)	Katherein	738 192	dipolo
Ganancia de la antena	dBi	14	2
Otras pérdidas (uso término móvil, ...)	dB	0	4

Figura 30: Parámetros en transmisión en los equipos GSM.

Recepción	Unidades	Estación Base	Terminal Móvil
Banda de Frecuencias	MHz	915	960
Sensibilidad	dBm	-107	-104
Tipo de divisor de potencia			
Pérdidas divisor de potencia	dB	3	0
Tipo de duplexor			
Pérdidas duplexor	dB		
Tipo de cable antena		1/2"	
Atenuación del cable de antena	dB/100m	3,9	0
Longitud del cable de antena	m	35	0
Tipo Sistema radiante (antena)	Katherein	738 192	dipolo
Ganancia de la antena	dBi	14	2
Ganacia extra por diversidad	dB	3	0
Otras pérdidas (uso término móvil, ...)	dB	0	4

Figura 31: Parámetros en recepción en los equipos GSM.

Para la realización del balance de potencias y por tanto la comprobación del peor enlace, tomamos los parámetros anteriores y obtenemos las potencias, pérdidas y ganancias en recepción y transmisión:

	Transmisión	Uds	Cálculo Realizado
Pt	Potencia Tx	dBm	$10*\log(\frac{Pt(W)}{0{,}001})$
Ltx	Pérdidas en transmisión	dB	$L_{combinador}+\frac{At_{cable}/100m*long.cable}{100}+L_{otros}$
Gtx	Ganancia en transmisión	dBi	G_{antena_tx}
PIRE	Potencia isótropa radiada equivalente	dBm	$P_t(dBm)-L_{tx}(dB)+G_{tx}(dBi)$
	Recepción		
Grx	Ganancia en recepción	dBi	$G_{antena_rx}+G_{diversidad}$
Lrx	Pérdidas en recepción	dBm	$\frac{At_{cable}/100m*long.cable}{100}+L_{divisor}+L_{otros}$
S	Sensibilidad	dBm	S_{rx}

L máx	**Pérdidas máximas admisibles**	dB	$PIRE+Grx-Lrx-S$

Figura 32: Parámetros del balance de potencias.

Realizando dichos cálculos se obtienen los siguientes valores:

	Transmisión	Uds	Base a Móvil (descendente)	Móvil a Base (ascendente)
Pt	Potencia Tx	dBm	43,98	33,01
Ltx	Pérdidas en transmisión	dB	4,37	4,00
Gtx	Ganancia en transmisión	dBi	14,00	2,00
PIRE	Potencia isótropa radiada equivalente	dBm	53,61	31,01
	Recepción			
Grx	Ganancia en recepción	dBi	2,00	17,00
Lrx	Pérdidas en recepción	dBm	4,00	4,37
S	Sensibilidad	dBm	-104,00	-107,00

L máx	**Pérdidas máximas admisibles**	dB	155,61	150,65

Figura 33: Resultados del balance de potencias. Tras el balance de potencias, se obtienen las pérdidas máximas admisibles para los enlaces ascendente y descendente.

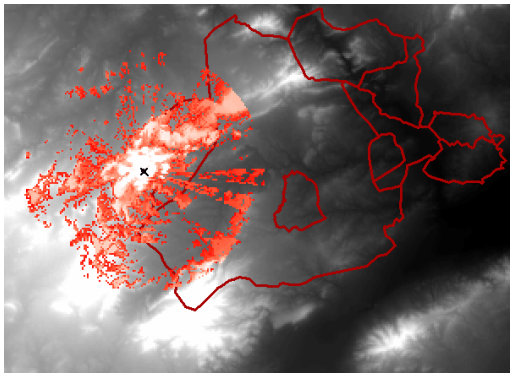
Tras los cálculos se ha obtenido que el peor enlace, es decir aquél que admite unas pérdidas máximas admisibles menores, es el ascendente (de móvil a base). Por tanto la cobertura de un sistema de potencia cuyos equipos (estaciones base y móviles) tengan estas caracteríasticas estará limitada por este enlace.

Simulamos por tanto el enlace ascendente de un sistema de potencia formado por tres emplazamientos GSM en RAGIS. Los parámetros de transmisión (móvil) y recepción (estaciones base) se van a tomar de la Figura 33. Se van a realizar dos simulaciones, una en entorno rural en la que se utilizará el método *UITR-526* como método de propagación y una segunda simulación en entorno urbano, con el método *COST 231*.

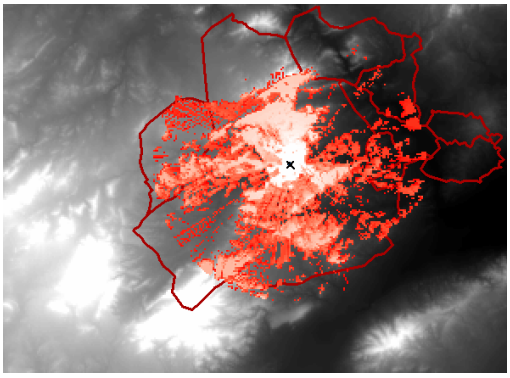
4.1.- Simulación en entorno Rural

La primera simulación de radioenlace se realiza sobre un mapa digital de terreno con la altimetría de un entorno rural formado por ocho téminos municipales. En este caso, y como ya se ha comentado anteriormente, los parámetros de los equipos que se han introducido corresponden a los de la Figura 33. Como el peor enlace es el ascendente, se tendrá en cuanta que transmite el terminal móvil y recibe la estación base. Primero se calculan las tres coberturas que van a formar el sistema:

Estac1:



Estac2:



Estac3:

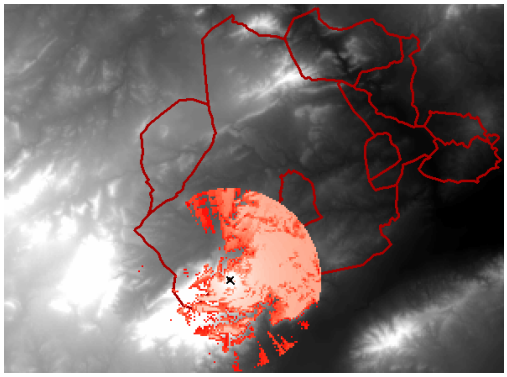


Figura 34: Coberturas de potencia. Coberturas de los emplazamientos que forman el sistema de potencia en entorno rural.

Los parámetros de dichos emplazamientos son:

PARÁMETROS DE LAS COBERTURAS	
Transmisión	
PIRE:	31 dBm
Recepción	
Grx:	17
Lrx:	4
Sensibilidad:	-107
Parámetros comunes	
H estacion base:	25 m
H móvil:	2 m
Frecuencia:	900 MHz
Polarización:	Vertical
MDT:	rural_200
Resolución:	210 metros
Capa morfológica:	Ninguna
Factor k:	1,33
Método de cálculo:	UITR-526

El modelo de entorno rural utilizado en este caso es el UITR-526 (ver Anexo A al final) y el sistema obtenido es:

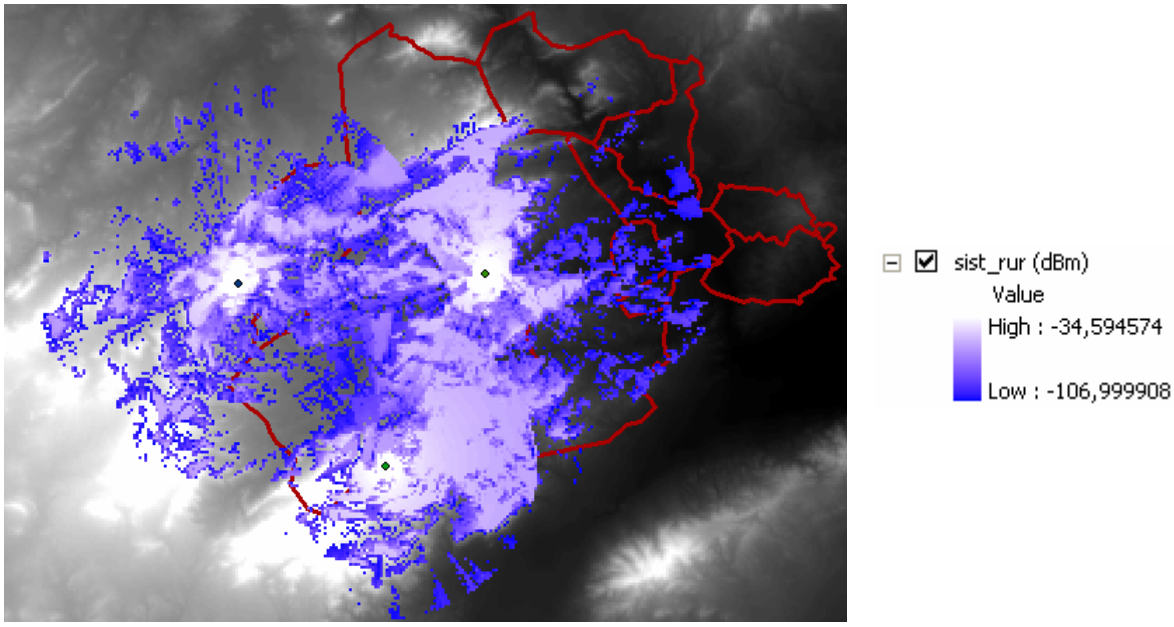


Figura 35: Sistema de potencia. Representación del sistema calculado y niveles de potencia.

Una vez calculados los niveles de potencia del sistema, realizamos el cálculo del mejor servidor con la opción *Calcular Mejor Servidor*. Los resultados obtenidos son:

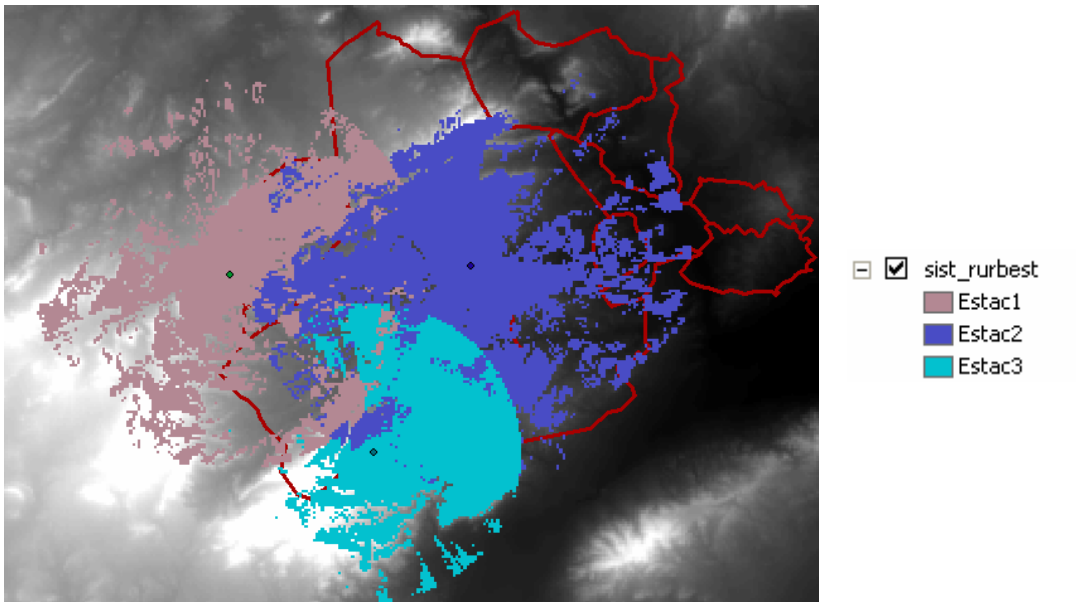


Figura 36: Mejores Servidores. El código de colores representa el mejor servidor del sistema en cada punto del mapa.

Donde en cada punto del mapa se puede comprobar la estación que actúa como mejor servidor según el código de colores de la leyenda de la derecha.

Una vez hecho esto, sería interesante determinar para cada punto, no sólo cuál de las estaciones base que forman el sistema es de la que se recibe más potencia, sino, todas las que cubren un determinado punto. Para esto, ejecutamos la función de *Calcular Servidores (1º, 2º, 3º ...)* en el mismo submenú que la función anterior:

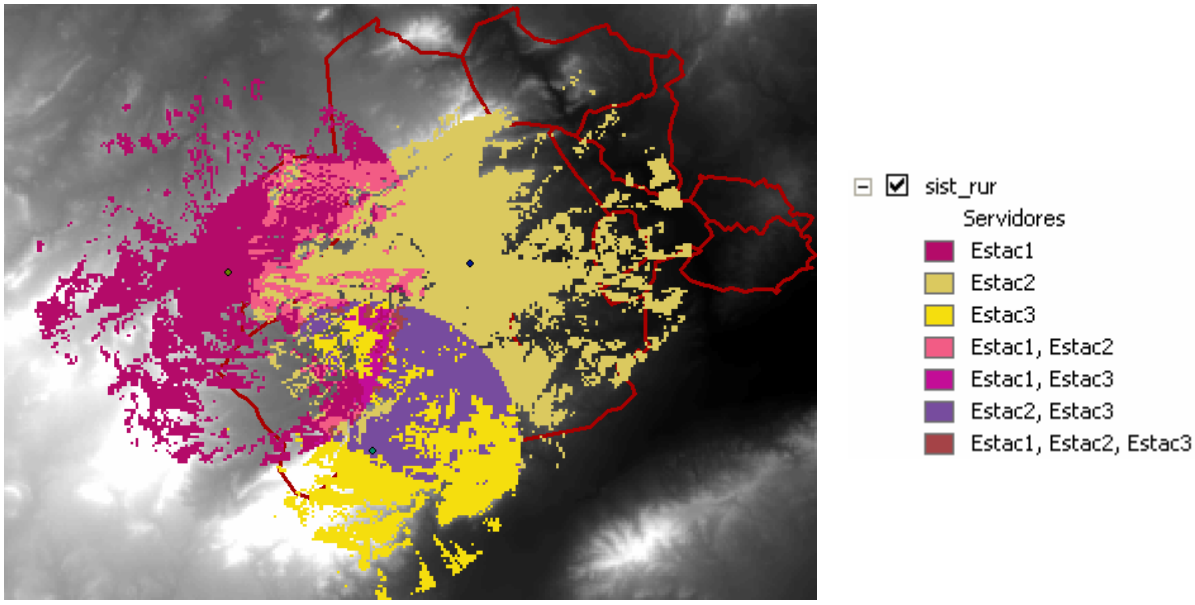


Figura 37: Servidores. El código de colores representa ahora todas las estaciones que cubren cada punto del mapa.

Al comparar la Figura 36 y Figura 37, se puede comprobar la información que añade esta última con respecto a la primera. Aunque la mayor parte del territorio sea cubierta por *Estc2* como mejor estación servidora, sabemos que parte de ese territorio también es cubierto por *Estc3*. Así si *Estc2* fallara o una estación móvil no se pudiera conectar a ella, podría intentar hacerlo a *Estc3*. En la Figura 37 también se pueden ver los puntos donde se recibe potencia de las tres estaciones.

Como ya se ha comentado en otros apartados, esta función no tiene en cuenta niveles de potencia, es decir, no se muestran las estaciones por orden servicio. A la hora de la programación se ha optado por hacerlo de esta manera para mayor claridad en la representación de los resultados y siempre teniendo en cuenta que dicha función es un complemento de la de mejor servidor, donde ya es mostrada la estación de la que se recibe mayor potencia.

Calculamos ahora el tanto por ciento de territorio de una parte del mapa que quedaría por encima de un valor umbral de potencia recibida. Como los niveles de potencia recibida del sistema se encuentran comprendidos entre -35 y -107 dBm, tomamos el valor umbral de potencia como -80 dBm. Para ello seleccionamos la opción *% sobre Umbral* dentro del menú *Calcular*. Introducimos los datos, donde la zona de cálculo va a consistir en la capa vectorial *Términos Municipales* que es una representación de entorno rural, seleccionamos cuatro *features* de esa capa, y seleccionamos como identificador el campo FID de la tabla de atributos de dicha capa:

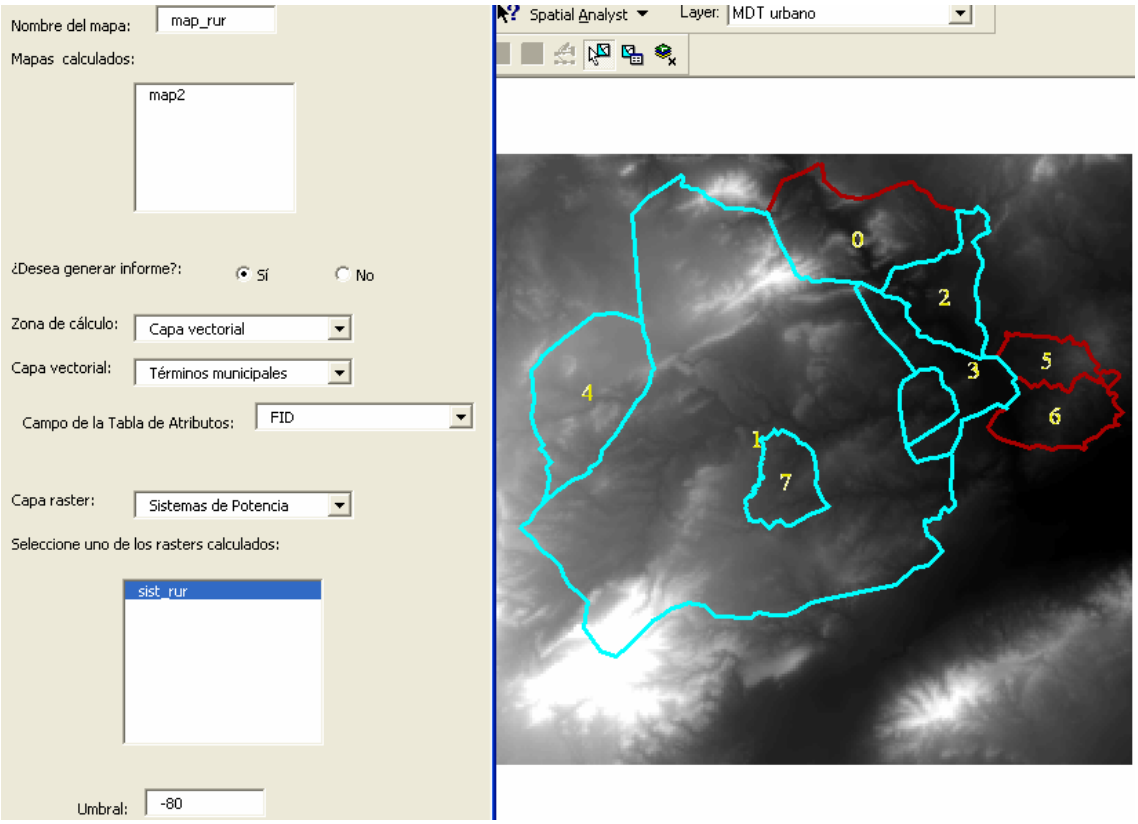


Figura 38: % sobre umbral. El mapa se calcula sobre la capa vectorial *Términos Municipales* (entorno rural), con cuatro *features* de esta capa seleccionadas. Cada *feature* se muestra etiquetada con el contenido del campo FID de la tabla de atributos de la capa para mayor facilidad de identificación.

Después de esto, el aspecto que muestra RAGIS es el siguiente:

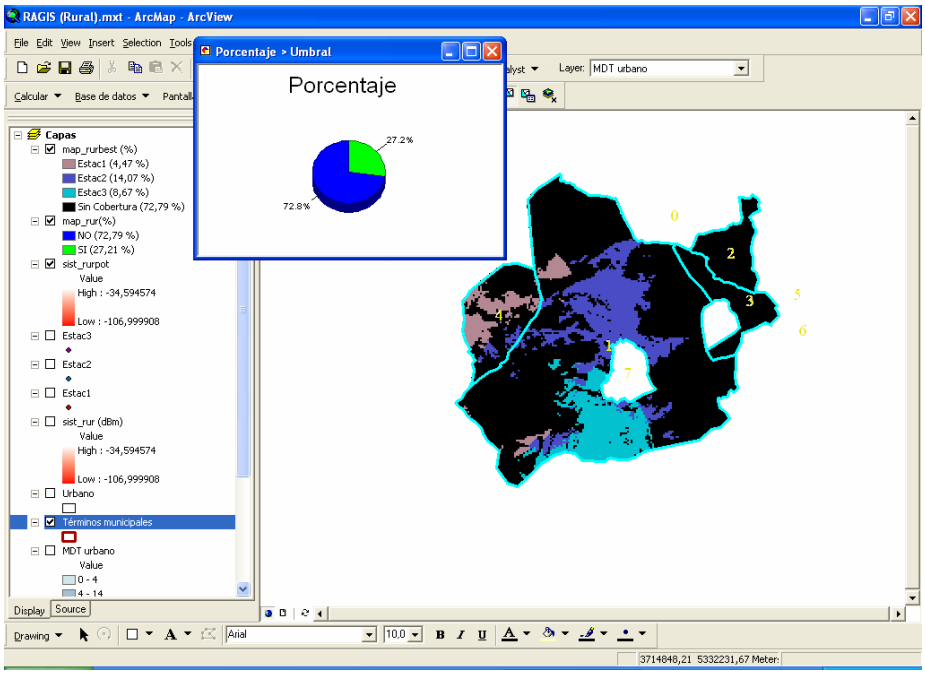


Figura 39: Mapa de resultados de % sobre umbral.

Representándolo con más detalle:

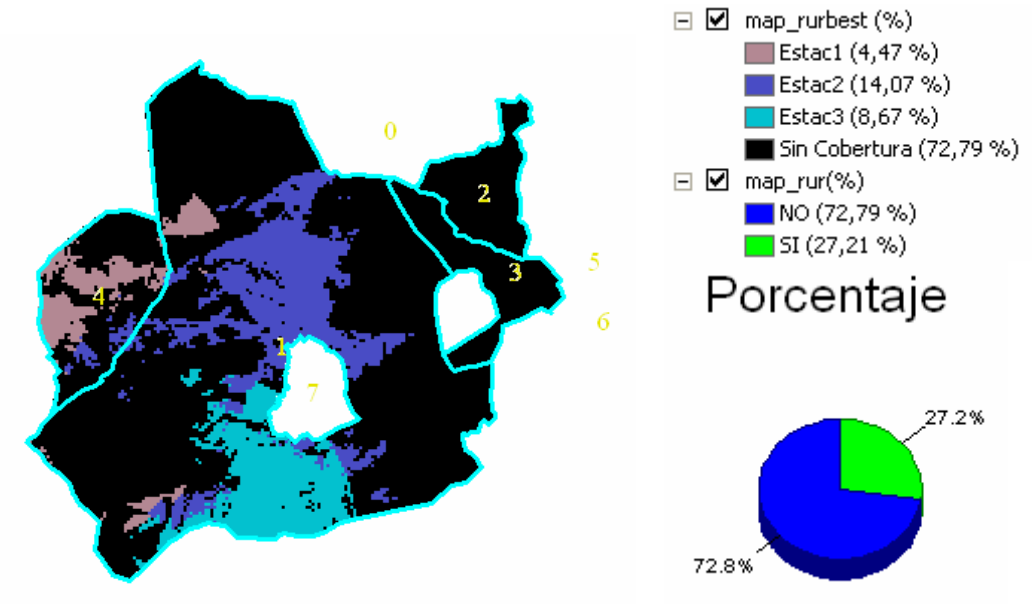


Figura 40: Detalle de resultados de % sobre umbral.

En las leyendas se pueden comprobar varios resultados:

1.- que el 27,21 % del territorio total seleccionado (las *features* 1, 2, 3 y 4) recibe una potencia del sistema por encima de -80 dBm (umbral establecido), mientras que el resto (72,79 %) queda por debajo de éste y

2.- ese 27,21 % se reparte entre las estaciones de la siguiente manera: en el 4,47 % del territorio con cobertura, la E.B. *Estc1* actúa como mejor servidor (zona rosa), *Esct2* es el mejor servidor en el 14,07 % del territorio con cobertura (zona azul marino) y la E.B. *Estc3* sólo en el 8,67 % de este territorio (zona azul claro).

En el archivo Excel generado, se representan éstos y otros datos, ya que los porcentajes anteriores sólo se refieren al territorio total (los cuatro municipios juntos). Sin embargo, puede resultar interesante conocer estos porcentajes para uno solo de los municipios. Esto se puede conocer mediante el informe de resultados. En una primera tabla se muestran los datos del mapa de porcentaje calculado:

Datos	
Mapa	map_rur
Capa Vectorial	Términos municipales
Tipo capa raster	Sistemas de Potencia
Capa Raster	sist_rur
Umbral	-80

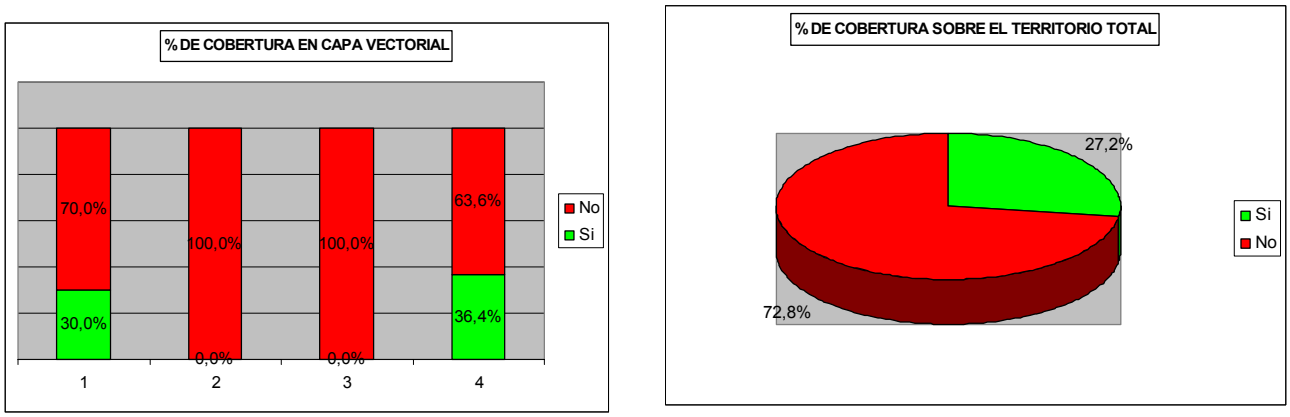
Figura 41: Tabla de datos.

A continuación de ésta, se muestra una tabla con los porcentajes de cobertura del sistema de potencia total (considerando todos los emplazamientos) de cada uno de los municipios seleccionados por separado. Como ya se ha comentado cada uno de ellos es identificado por el campo de la tabla de atributos que el usuario eligió en la ventana de introducción de datos:

Cobertura		
	Términos municipales	
FID	**Si**	**No**
1	30,0%	70,0%
2	0,0%	100,0%
3	0,0%	100,0%
4	36,4%	63,6%
Territorio Total		
	27,2%	**72,8%**

Figura 42: Tabla de % de cobertura del sistema de potencia.

En ella, se puede comprobar que el municipio con mayor cobertura por encima del umbral es el municipio con FID = 4 con un 36,4% de cobertura frente a municipios con nada de cobertura por encima de los -80 dBm. Al final de la tabla se muestra el porcentaje calculado sobre el territorio total, al igual que se mostraba en el mapa. Estos resultados también pueden verse en forma de gráficos:



(a) **(b)**

Figura 43: Gráficos de % de cobertura del sistema. (a) Gráfico de barras con la cobertura de cada término municipal. **(b)** Gráfico de pastel con la cobertura de todo el territorio total.

El gráfico de barras muestra cada uno de los municipios y el pastel representa la cobertura en todo el territorio.

En la primera hoja del documento, aparece una tercera tabla en donde se representan los porcentajes de cobertura que corresponden a cada una de las estaciones base como mejor servidor. Esta información ya la hemos visto reflejada en las leyendas del mapa:

Mejores Servidores: % de cobertura	
Estac1	**4,5%**
Estac2	**14,1%**
Estac3	**8,7%**
Sin cobertura	**72,8%**

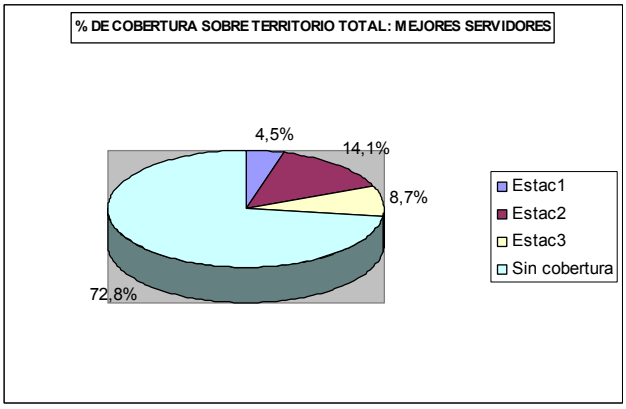


Figura 44: Tabla y gráfico de % de cobertura como mejor servidor. En éstos se muestra el porcentaje de territorio que cubren cada uno de los emplazamientos como mejor servidor.

Como en este caso, el raster de mejor servidor sí estaba calculado y guardado en disco para el sistema utilizado como capa raster, se ha podido utilizar dicho raster para la generación de más información adicional. En este caso, se crea una hoja por cada estación base que componga el sistema de potencia y se utiliza el raster de cobertura de dichas estaciones base para calcular los porcentajes de territorio cubiertos por dichos emplazamientos como mejores servidores:

Estc1:

Datos	
Mapa	map_rur
Capa Vectorial	Términos municipales
Tipo capa raster	Coberturas
Capa Raster	Estac1
Umbral	-80

Cobertura

	Términos municipales	
FID	**Si**	**No**
1	1,7%	98,3%
2	0,0%	100,0%
3	0,0%	100,0%
4	30,9%	69,1%
Territorio Total		
	4,5%	**95,5%**

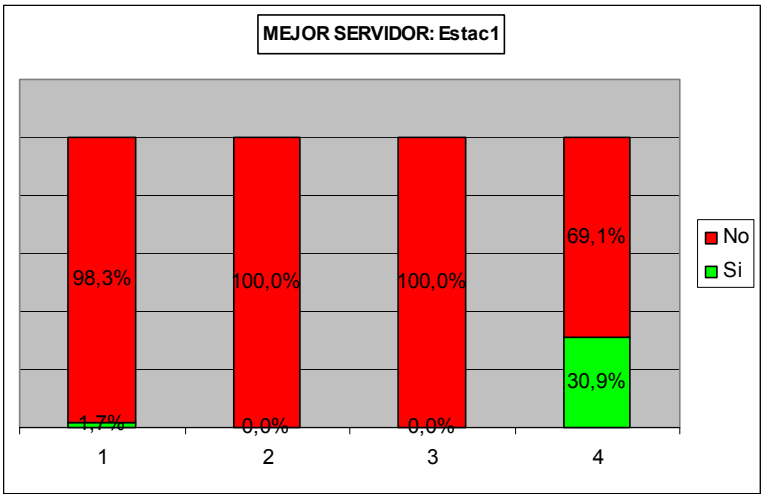


Figura 45: Hoja del emplazamiento *Est1*.

Estc2:

Datos	
Mapa	map_rur
Capa Vectorial	Términos municipales
Tipo capa raster	Coberturas
Capa Raster	Estac2
Umbral	-80

Cobertura

	Términos municipales	
FID	**Si**	**No**
1	17,2%	82,8%
2	0,0%	100,0%
3	0,0%	100,0%
4	5,5%	94,5%
	Territorio Total	
	14,1%	**85,9%**

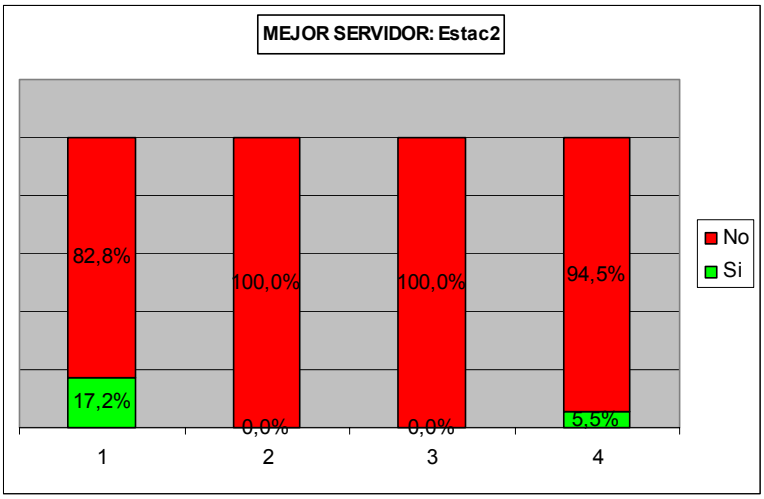


Figura 46: Hoja del emplazamiento *Est2*.

Estc3:

Datos	
Mapa	map_rur
Capa Vectorial	Términos municipales
Tipo capa raster	Coberturas
Capa Raster	Estac3
Umbral	-80

Cobertura

	Términos municipales	
FID	**Si**	**No**
1	11,0%	89,0%
2	0,0%	100,0%
3	0,0%	100,0%
4	0,0%	100,0%
Territorio Total		
	8,7%	**91,3%**

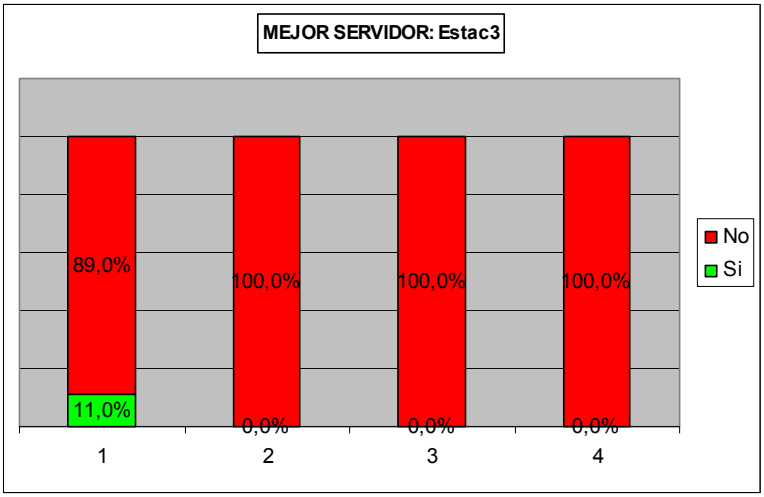


Figura 47: Hoja del emplazamiento *Est3*.

En cada figura se pueden ver las tablas y los gráficos correspondientes a cada una de las estaciones base. Se puede comprobar que en este caso, la estación base que ofrece servicio como mejor servidora en el mayor porcentaje de territorio es la E.B. *Estc2* con un 14,1% de porcentaje de cobertura del territorio total. A su vez, ésta cubre en un 17,2% al municipio con FID = 1, un 5,5% del municipio con FID = 4 y no ofrece cobertura como mejor servidor ni al municipio FID = 2 ni a aquél con FID = 3.

4.2.- Simulación en entorno Urbano

Realizamos ahora el mismo sistema, pero en entorno urbano. En este caso, el modelo de propagación utilizado es el COST-231, modelo semiempírico también conocido como Walfisch-Ikegami y que es útil para ambientes urbanos densos (ver Anexo A al final). El mdt utilizado ahora tiene una resolución de 6 metros. En este caso, los parámetros de las coberturas son los siguientes:

PARÁMETROS DE LAS COBERTURAS	
Transmisión	
PIRE:	31 dBm
Recepción	
Grx:	17
Lrx:	4
Sensibilidad:	-107
Parámetros comunes	
H estacion base:	30 m
H móvil:	2 m
Frecuencia:	900 MHz
Polarización:	Vertical
MDT:	urbano_6
Resolución:	6 metros
Capa morfológica:	Ninguna
Factor k:	1,33
Método de cálculo:	COST-231

Las intensidades de potencia de recepción y emisión son las mismas que en el caso anterior. De la misma forma que antes, mostramos los resultados obtenidos en el cálculo de las coberturas, tres emplazamientos, y del sistema:

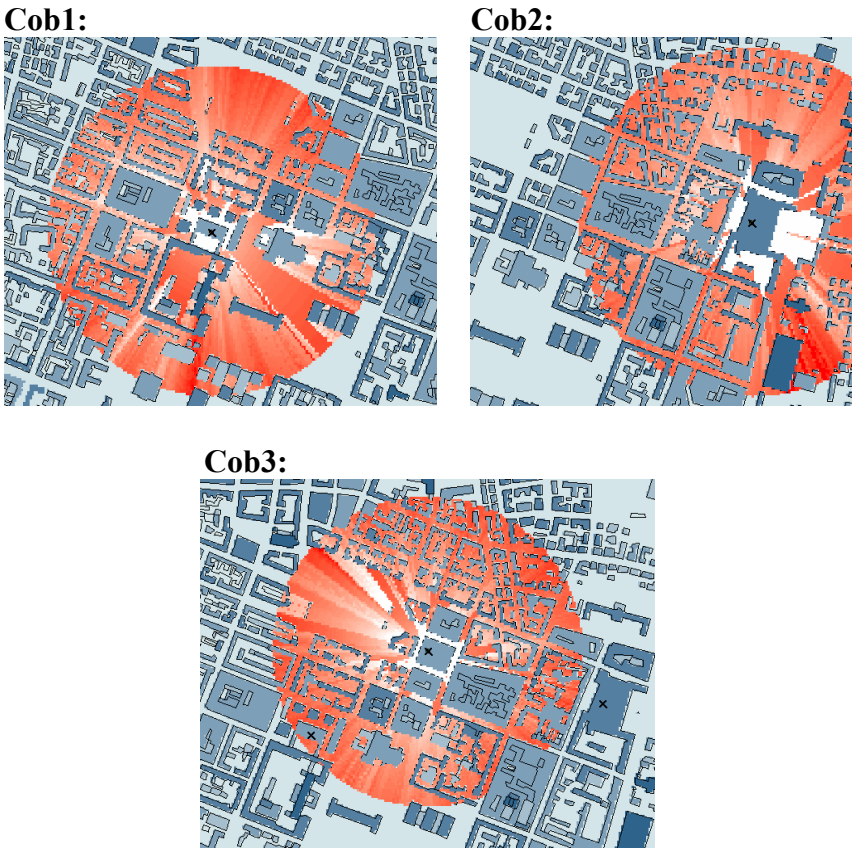


Figura 48: Coberturas de potencia. Coberturas de los emplazamientos que forman el sistema de potencia en entorno urbano.

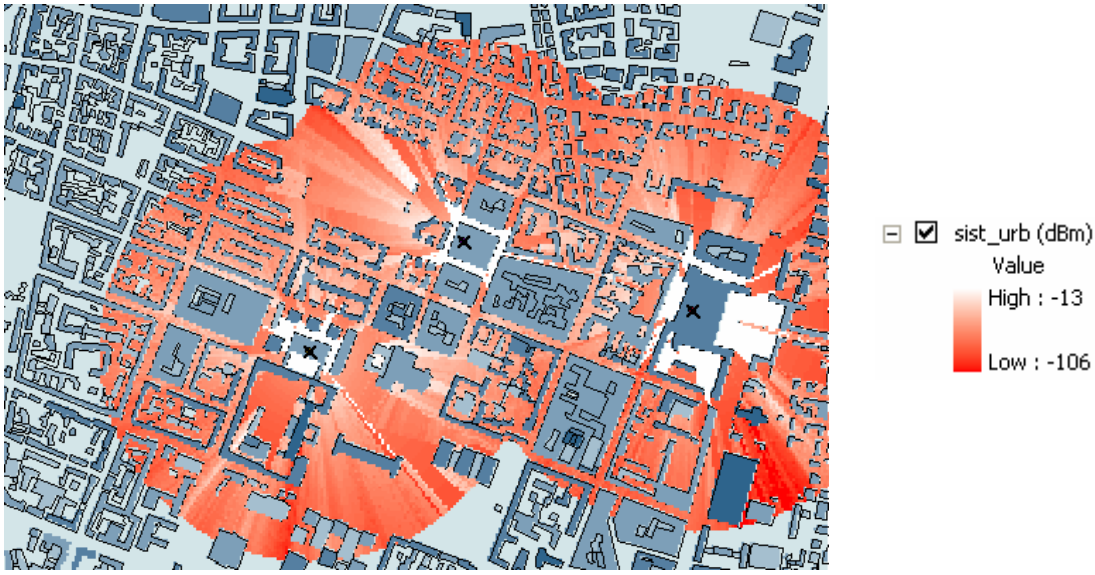


Figura 49: Sistema de potencia. Representación del sistema calculado y niveles de potencia.

En este caso, los resultados obtenidos del cálculo de los mejores servidores son:

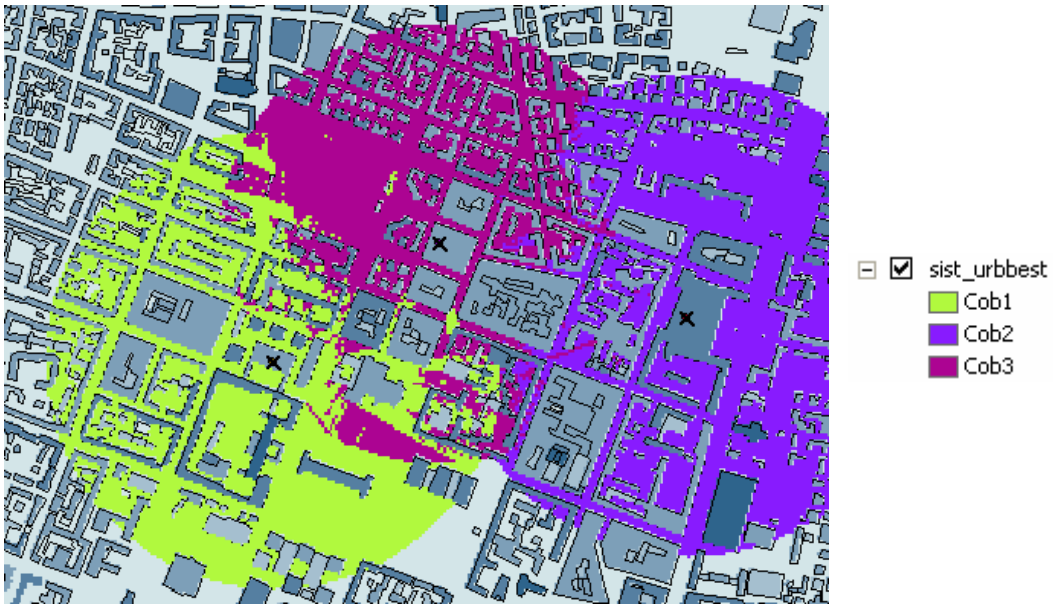


Figura 50: Mejores Servidores. El código de colores representa el mejor servidor del sistema en cada punto del mapa.

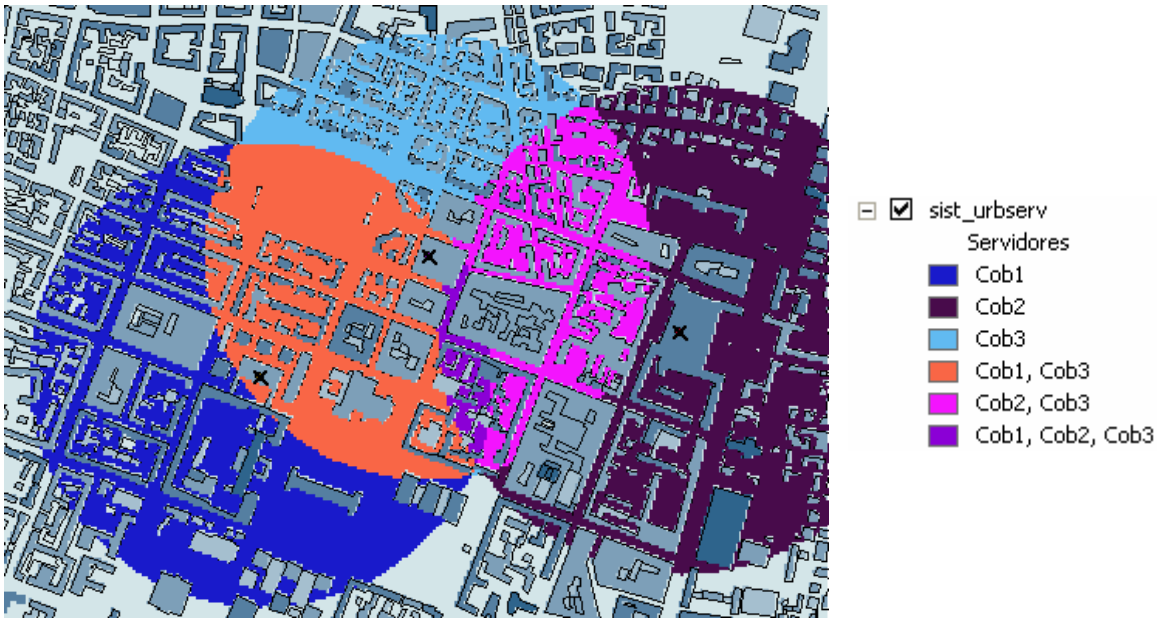
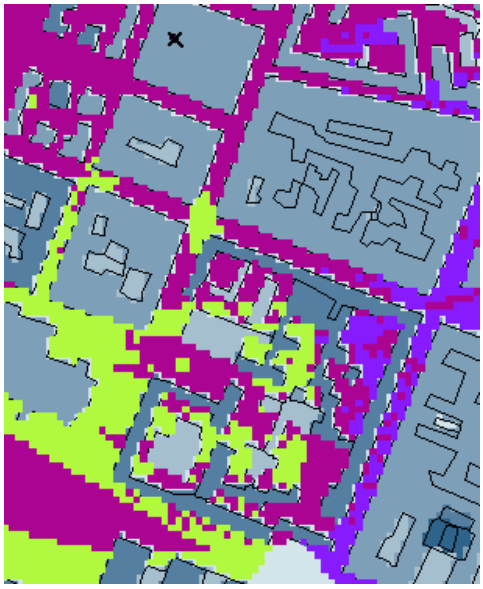


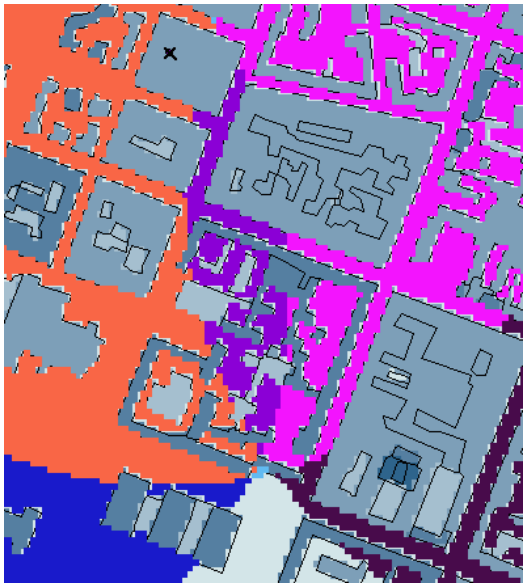
Figura 51: Servidores. El código de colores representa ahora todas las estaciones que cubren cada punto del mapa.

Al igual que en la simulación anterior, se puede comprobar la información adicional que añade este mapa con respecto al del mejor servidor. En este caso, parte de la zona cubierta por *Cob3* como mejor servidora, también es cubierta por las estaciones

Cob1 (zonas en rojo) y *Cob 2* (zonas en rosa). Támbién es útil conocer la zona (en el centro del mapa en color lila) cubierta por las tres estaciones:



(a)



(b)

Figura 52: Zona central del mapa de entorno urbano. Comparación del mapa de mejor servidor **(a)** con el de servidores **(b)**.

Con el primer mapa se pueden conocer los emplazamientos de los que se recibe una mayor potencia, sin embargo observando la misma zona en el segundo mapa se comprueba que esa zona está cubierta por cualquiera de las tres estaciones que forman el sistema.

Para realizar el cálculo del % sobre umbral se toma un valor de potencia umbral de -90 dBm. En el caso de un entorno urbano es interesante conocer el porcentaje de cobertura que se recibe del sistema en zonas muy concretas del mapa (como calle, plazas, manzanas…). Esta simulación en concreto se va a realizar sobre una calle mediante la elección de la zona de cálculo de tipo Polilínea. Para ello, se selecciona como zona de cálculo la opción Polílinea de la lista desplegable y con ayuda del ratón se señala la calle sobre la que se quiere realizar el mapa de porcentaje. Después de esto, los resultados obtenidos son:

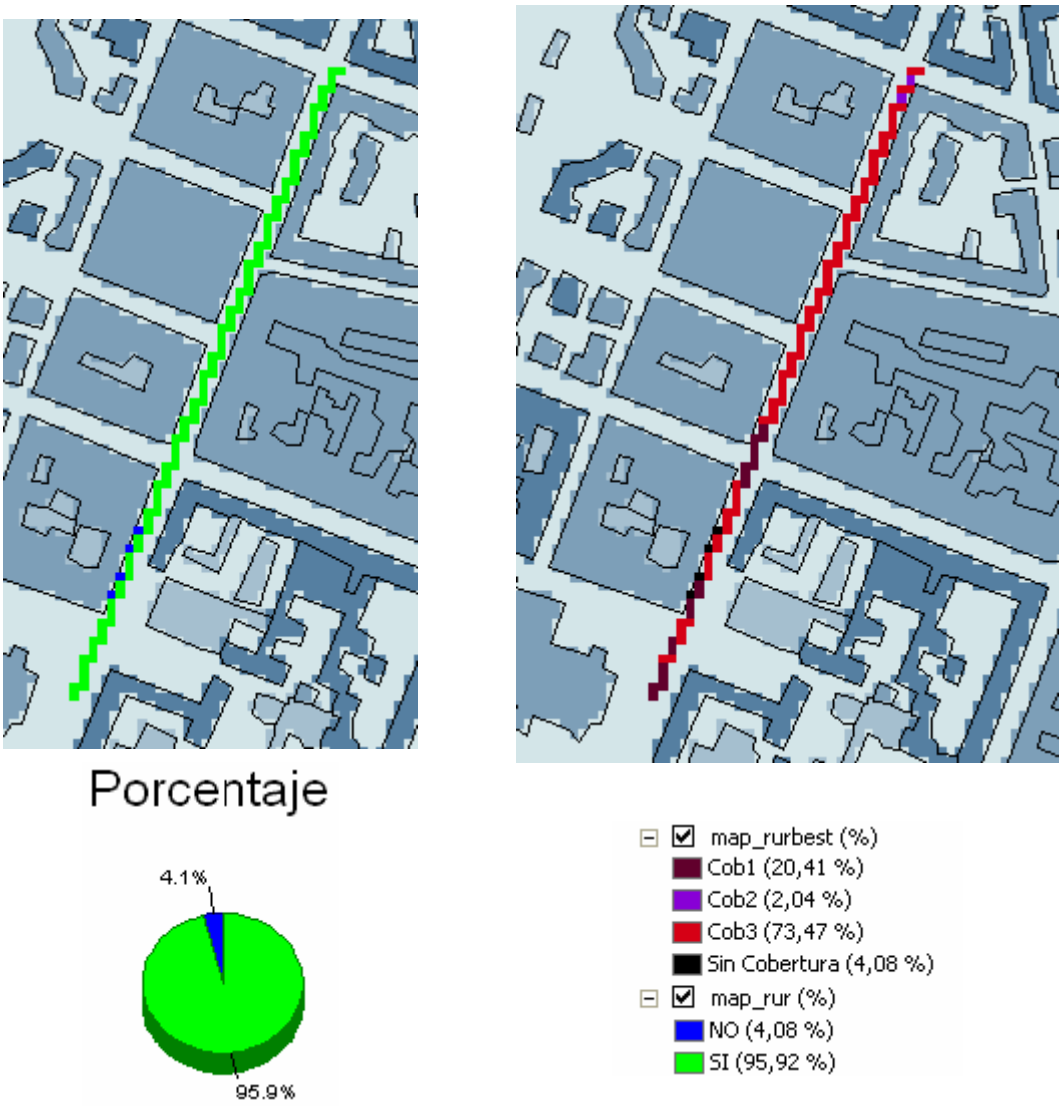


Figura 53: % sobre umbral. Resultados obtenidos en el caso de entorno urbano.

Tras el cálculo se comprueba que un 95,92% de la calle es cubierta por el sistema y que de ese porcentaje, el 73,47% es cubierto por la estación *Cob3* como mejor servidora. Sólo un 4,08% de la calle queda sin cobertura por encima de -90dBm.

Al igual que antes, tras este cálculo, se crea el correspondiente archivo Excel en la carpeta del mapa. En este caso, al tratarse la zona de cálculo de una polílinea (un único raster creado) no se muestra en este archivo ningún tipo de información adicional a la mostrada en el mapa.

CAPÍTULO 5.- CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

En este capítulo se hace una exposición de las distintas conclusiones obtenidas tras la realización de este proyecto.

Como primera conclusión, cabe destacar la introducción en el manejo del Sistema de Información Geográfica ArcGis para la realización de todo el proyecto. Este punto comprende diversos aspectos como el manejo y representación de información espacial (rasters) y vectorial (capas, *features* y sus tablas de atributos correspondientes) y utilización del lenguaje de desarrollo ArcObjects. En este punto también cabe mencionar el manejo del lenguaje Visual Basic para aplicaciones, así como el uso de objetos propios de la aplicación Excel.

Por otra parte, se puede concluir que a lo largo de este proyecto se han cumplido los dos propósitos que se planteaban a su inicio:

En primer lugar, se ha realizado la programación de una nueva función de cálculo de servidores que realiza una representación de todos los emplazamientos de un sistema de potencia que cubren un determinado punto de un mapa. Esta utilidad se presenta como un complemento de la función *Calcular Mejor Servidor* que ya disponía RAGIS. Así par un determinado punto o zona del mapa se puede saber, no sólo cuál es la estación que se recibe con mayor potencia, sino también cuales de todas las que forman parte del sistema ofrecen servicio.

En segundo lugar se ha mejorado la función *% Sobre Umbral* dentro del menú de sistemas de RAGIS. Se ha mejorado la funcionalidad de la ventana de introducción de datos y se ha añadido la opción de generación de un informe de resultados. Este informe consiste en un archivo en formato *.xls*, o libro de la aplicación Excel. Éste, además de añadir información adicional a la que mostraba RAGIS inicialmente, introduce la opción de poder imprimir los datos y resultados de las simulaciones llevadas a cabo para el cálculo de un mapa.

En cuanto a las simulaciones llevadas a cabo y los resultados obtenidos, se puede decir que ambas funciones añaden información complementaria a las que ya disponía la herramienta RAGIS. En el caso de los *Mejores Servidores*, es útil saber para

un determinado punto cuáles son todas las estaciones servidoras, para, por ejemplo, un caso de fallo de la mejor servidora. La utilidad de la parte programada de la función de *% Sobre Umbral* sobre la información aportada hasta ahora radica en la separación del territorio total en las diversas *features* que el usuario ha ido seleccionando por separado. Así, no sólo se puede conocer el porcentaje de cobertura sobre el territorio total de una región rural, sino también cuál es el porcentaje de cobertura sobre cada uno de los términos municipales que el usuario ha seleccionado por separado. Por otra parte, en el caso de un sistema de potencia que disponga de mapa de mejores servidores calculado, se podrá determinar qué porcentaje de cobertura aporta como mejor servidora cada una de las estaciones que componen dicho sistema, tanto para el territorio total (información aportada hasta ahora), como para cada término municipal por separado.

En cuanto a futuras líneas de trabajo con las que se podría mejorar y ampliar las funciones que aquí se han implementado se pueden indicar las siguientes:

Como ya se ha comentado en capítulos anteriores, la representación de los mejores servidores no contempla el orden del servicio que cada uno de éstos presta. Esto se debe a que en ese caso, la representación en el mapa mediante un código de colores no sería clara para el usuario por el gran número de posibles combinaciones que podría llegar a tener un determinado sistema. Sin embargo, sería interesante obtener este orden, para poder conocer cuál es el segundo mejor servidor, el tercer mejor servidor… Esto se podría llevar a cabo añadiendo una ventana con la representación del orden, de tal manera que al pulsar sobre un punto del mapa, ésta apareciera con la información del servicio de las estaciones ordenado para dicho punto.

Por otra parte, al informe de resultados de la función de *% Sobre Umbral* se le puede añadir más información, extraída bien de los rasters que se calculan a través de la función ya programada o bien a través del cálculo de nuevos rasters. También sería interesante un acceso directo al informe desde el menú de RAGIS.

CAPÍTULO 6.- REFERENCIAS

[1] *Exploring ArcObjects, Vol. 1 y Vol. 2,* ESRI

[2] *Modeling Our World. The ESRI Guide to Geodatabase Design,* ESRI

[3] *Using ArcGis Spatial Analyst,* ESRI

[4] Fco. Javier Ceballos Sierra, *Visual Basic. Curso de programación*, 2ª ed., ra-ma

[5] *Microsoft Visual Basic 6.0. Manual del programador*, Mc Graw Hill

[6] John Walkenbach, *Programación en Excel 2000 con VBA*, Anaya Multimedia

[7] ESRI Support Center: http://www.support.esri.com

[8] Canal Visual Basic: http://www.canalvisualbasic.net

[9] Introducción a los sistemas celulares: http://ceres.ugr.es/~alumnos/c_avila/gsm22.htm

[10] Telefonía celular: http://www.inatel.br/docentes/dayan/index.html

[11] RAGIS: herramienta de gestión y de cálculo de cobertura radioeléctrica basada en sistemas de información geográfica: http://www.ahciet.net/comun/portales/1000/10002/10007/10556/docs/09.pdf

APÉNDICE A: MODELOS DE PROPAGACIÓN

En este apéndice se hace una pequeña introducción de los dos modelos de propagación utilizados a la hora de la simulación de una cobertura radioeléctricae en distintos entornos.

A.1.- UIT-526

El método UITR-526 es válido para entornos rurales en los que existen múltiples obstáculos (Figura 54):

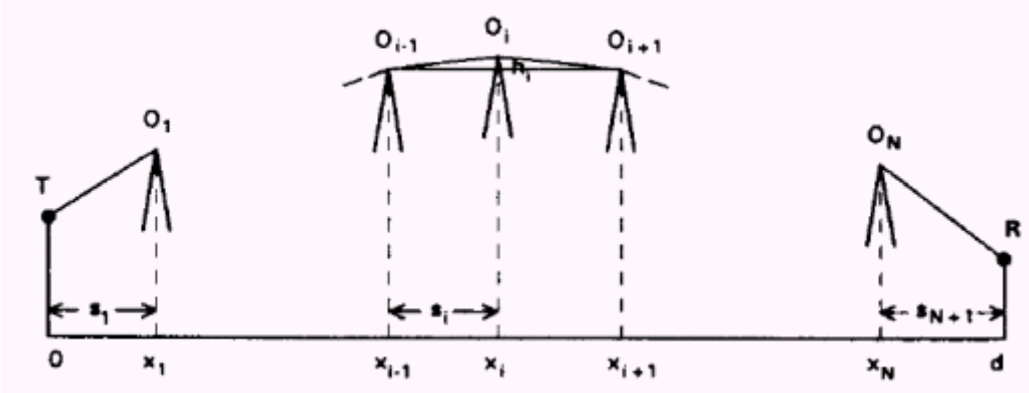


Figura 54: Múltiples obstáculos.

donde:

- N: número de obstáculos
- h_i: altura del obstáculo O_i respecto a la recta que une O_{i-1} con O_{i+1}
- x_i: abscisa del obstáculo O_i
- $s_i = x_i - x_{i-1}$: distancia entre los obstáculos O_{i-1} y O_{i+1}
- $s_1 = x_1$: distancia entre el primer obstáculo y el transmisor
- $s_{n+1} = d - x_n$: distancia del último obstáculo al transmisor (d es la distancia entre emisor y receptor)

La atenuación por difracción viene dada, para múltiples obstáculos, por:

$$L_D(dB) = \sum_{i=1}^{n} L_D(v_i) + \sum_{i=1}^{j} L_{SD}(v_i') - 10 \cdot \log(C_n)$$

donde:

$$v_i = 2{,}58 \cdot 10^{-3} \cdot h_i \cdot \sqrt{f \cdot \frac{s_i + s_{i+1}}{s_i \cdot s_{i+1}}}$$

L_{SD} son las pérdidas debidas a obstáculos en el subvano O_{i-1}, O_i, O_{i+1} que queden por debajo de la línea O_{i-1}, O_{i+1} con despejamiento insuficiente. Únicamente se tendrá en cuenta un único obstáculo, el que tenga el parámetro v menos negativo (el peor caso). C_n es un factor de corrección que viene dado por:

$$C_n = \left[\frac{s_2 \cdot \ldots \cdot s_N \cdot (s_1 + s_2 + \ldots + s_{N+1})}{(s_1 + s_2) \cdot \ldots \cdot (s_N + s_{N+1})} \right]$$

A.2.- COST 231

Para la predicción más precisa de la pérdida básica de propagación en un medio urbano se ha propuesto el método COST 231, que tiene en cuenta el efecto de las estructuras urbanas en cuyo entorno está situado el móvil.

Este método es aplicable en situaciones de propagación para las cuales el rayo directo entre el transmisor y el receptor está obstruido por los edificios, como se muestra en la Figura 55.

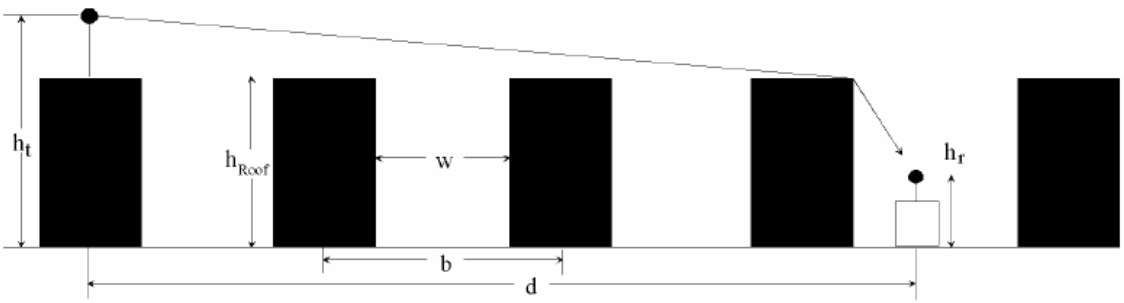


Figura 55: Edificios en entorno urbano.

Los parámetros que intervienen en la predicción son los siguientes:

- h_B: Altura sobre el suelo de la antena de la estación base (m)
- h_r: Altura sobre el suelo de la antena del móvil (m)
- h_{Roof}: Altura media de los edificios (m) ($h_{Roof} > h_m$)
- w: Anchura de la calle donde se encuentra el móvil (m)
- b: Distancia entre centros de edificios (m)

- d: Distancia base-móvil (Km)
- α: Ángulo de inclinación del rayo (º)
- $\Delta h_B = h_B - h_{Roof}$: Altura de la antena de la estación base sobre la altura media de los edificios circundantes (m)
- $\Delta h_{Roof} = h_{Roof} - h_m$: Altura media de los edificios sobre la altura de la antena del móvil (m)

De acuerdo con este método, la pérdida básica de propagación es:

$$L_b = L_{bf} + L_{rts} + L_{msd}$$

donde:

L_{bf} es la pérdida en condiciones de espacio libre:

$$L_{bf} = 32{,}45 + 20 \cdot \log_{10}(f(MHz)) + 20 \cdot \log_{10}(d(Km))$$

L_{rts} es la pérdida debida a la difracción "tejado-calle" entre el tejado de los edificios y el móvil, que se calcula mediante la ecuación:

$$L_{rts} = -16{,}9 - 10 \cdot \log_{10}(w) + 10 \cdot \log_{10}(f) + 20 \cdot \log_{10} \Delta h_{Roof}$$

Si $L_{rts} \leq 0$ se toma $L_{rts} = 0$

L_{msd} es una estimación de la "difracción multiobstáculo" que experimenta el rayo entre la antena transmisora y el edificio próximo al receptor, debido a los edificios interpuestos entre ambos. Su valor es:

$$L_{msd} = L_{bsh} + k_a + k_d \cdot \log_{10} d + k_f \cdot \log_{10}(f) - 9 \cdot \log_{10} b$$

Los parámetros que intervienen en esta expresión se calculan como sigue:

$$L_{bsh} = -18 \cdot \log(1 + \Delta h_B)$$

Si $\Delta h_B < 0$, $L_{bsh} = 0$

$$k_a = \begin{cases} 54 & \text{para} \quad \Delta h_B \geq 0 \\ 54 - 0{,}8 \cdot \Delta h_B & \text{para} \quad \Delta h_B < 0 \ \text{ y } \ \mathrm{d} \geq 0{,}5 \\ 54 - 0{,}8 \cdot \Delta h_B \cdot d/0{,}5 & \text{para} \quad \Delta h_B < 0 \ \text{ y } \ \mathrm{d} < 0{,}5 \end{cases}$$

$$k_d = \begin{cases} 18 & \text{para} \quad \Delta h_B \geq 0 \\ 18 - 15 \cdot \Delta h_B / h_{Roof} & \text{para} \quad \Delta h_B < 0 \end{cases}$$

$$k_d = \begin{cases} -4 + 0{,}7(f/925 - 1) & \text{- para ciudades de tamaño medio y zonas suburbanas} \\ & \text{con densidad de vegetación moderada} \\ 18 - 15 \cdot \Delta h_B / h_{Roof} & \text{- para grandes centros metropolitanos} \end{cases}$$

La gama de valores de los parámetros dentro de la cual es aplicable el modelo es la siguiente:

- $800 \leq f \leq 2000$ MHz
- $4 \leq h_B \leq 50$ m
- $1 \leq h_r \leq 3$ m
- $0{,}02 \leq d \leq 5$ Km

APÉNDICE B: SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Un *Sistema de Información Geográfica* (*SIG* o GIS, en su acrónimo inglés) se puede contemplar como un conjunto de mapas de la misma porción del territorio, donde un lugar concreto (definido por unas coordenadas) tiene la misma localización en todos los mapas incluidos en el sistema de información. La información se encuentra georreferenciada, es decir, las coordenadas de todos los elementos gráficos son conocidas en un sistema de referencia conocido. De este modo, es posible realizar análisis de sus características espaciales y temáticas. También puede definirse un *SIG* como una colección organizada de software y hardware diseñado para capturar, almacenar, actualizar, manipular y representar eficientemente cualquier información georreferenciada.

Con ayuda de los SIG se pueden gestionar dos tipos básicos de información geográfica:

- Información espacial, que describe la localización y la forma de diversas características geográficas (p. ej. vías de comunicación, etc.). La información espacial puede almacenarse en formato *raster* (mediante una matriz de celdas) o *vectorial* (mediante un conjunto ordenado de coordenadas de un sistema de referencia conocido).

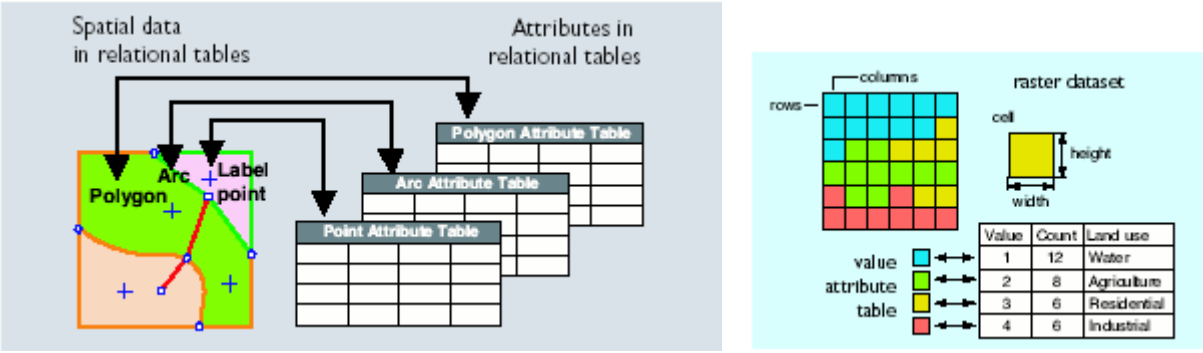
- Información descriptiva, que proporciona información adicional asociada a cada una de esas características geográficas (tipo de vía de comunicación: autopista, autovía, etc.). La información descriptiva en un *SIG* se almacena a través de *tablas de atributos*. Estas tablas contienen un identificador por medio del cual la información geográfica (*vectorial* o *raster*) queda relacionada con la información descriptiva: modelo georrelacional. (Figura 56)

Como se ha comentado, la información espacial puede almacenarse en dos formatos diferentes:

Formato vectorial: los elementos geográficos se guardan como una lista ordenada de coordenadas definidas en un sistema de referencia conocido. De esta forma podemos almacenar *puntos* (una sola pareja de coordenadas), *líneas* (lista ordenada de

puntos) y *polígonos* (líneas cerradas a las que se les asocia un área). Ejemplos típicos de información almacenada en formato vectorial pueden ser: ríos (líneas), edificios (polígonos), picos acotados (puntos), carreteras (líneas), núcleos de población (puntos), provincias (polígonos), etc.

Formato matricial o *raster*: un *raster* consiste en una matriz de celdas definida por una resolución determinada en la que se guarda información que varía de un modo continuo a lo largo del terreno. Para modelar esta variación continua, se guarda el valor asociado a una retícula regular de puntos, a cada uno de los cuales corresponde una celda de la matriz. Evidentemente cuanto mayor número de celdas tengamos (mayor resolución) más se aproximará el modelo a la realidad y más grande y costoso de almacenar será la matriz asociada. Ejemplos típicos de información que se suele almacenar en formato matricial encontramos: altura del terreno, vegetación, temperaturas, etc.



(a) **(b)**

Figura 56: Información espacial y tabla de atributos. La información espacial, vectorial **(a)** y raster **(b)**, y la descriptiva se relacionan mediante las tablas de atributos.

El SIG permite separar toda esta información en diferentes *capas* temáticas, también llamadas *layers,* que a su vez estan compuestas por las llamadas *features*. Estas capas se almacenan independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al usuario la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no se podría obtener de otra forma. Esta función es muy útil a la hora de generación de mapas en los que el usuario puede seleccionar aquella información que desea que aparezca.

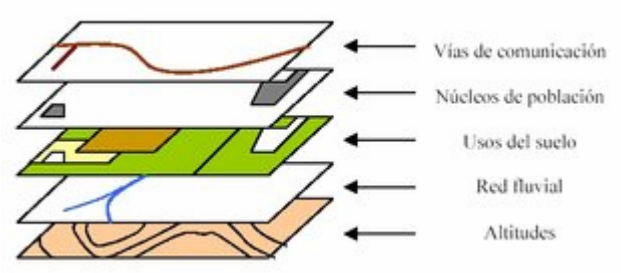


Figura 57: Capas. La información en un SIG puede dividirse en *capas* que estan compuestas de *features*. Cada una de estas capas tiene asociada una tabla de atributos con información descriptiva.

APÉNDICE C: MANEJO DE LA HERRAMIENTA

Este anexo es una descripción de los pasos a seguir para el manejo y la utilización de las funciones programadas para la herramienta RAGIS.

C.1.- Mejores Servidores

Para el cálculo de los mejores servidores, los pasos a seguir son:

1.- Cargar en el mapa un sistema de potencia. Para ello, se realiza el cálculo de uno nuevo o bien se carga uno guardado con anterioridad mediante el menú: *Calcular → Sistemas de Potencia → Abrir*

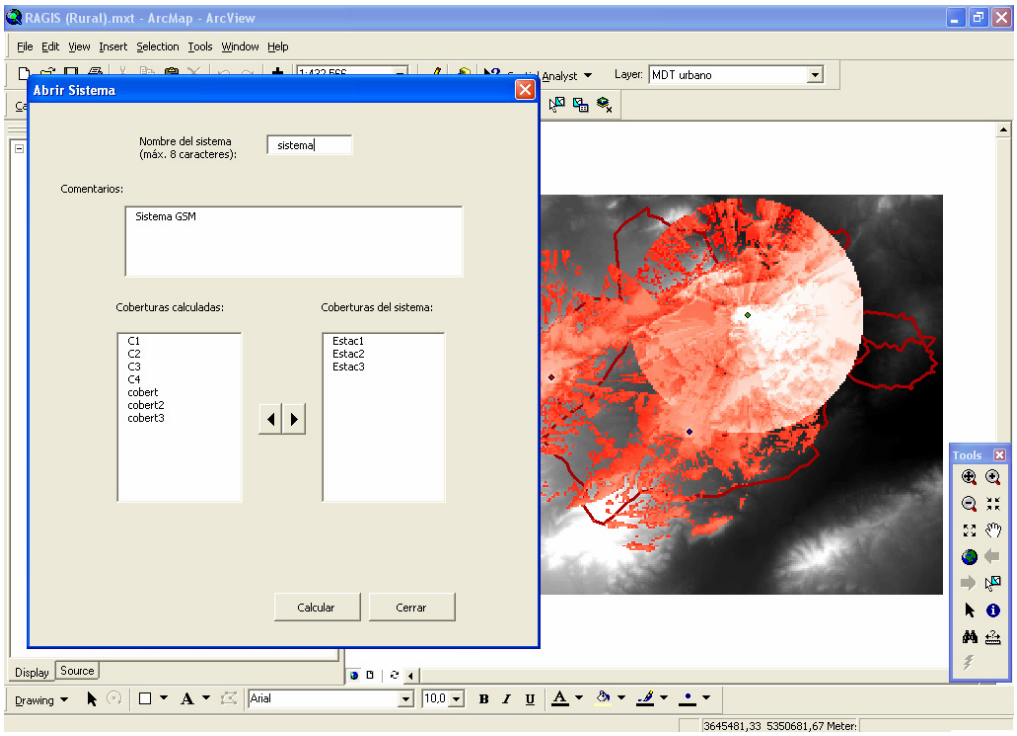


Figura 58: Menú Abrir Sistema de Potencia. Carga un sistema de potencia anteriormente calculado y guardado en el disco duro.

2.- Seleccionar la opción: *Calcular → Sistemas de Potencia → Calcular Servidores (1º, 2º, 3º, ...)*

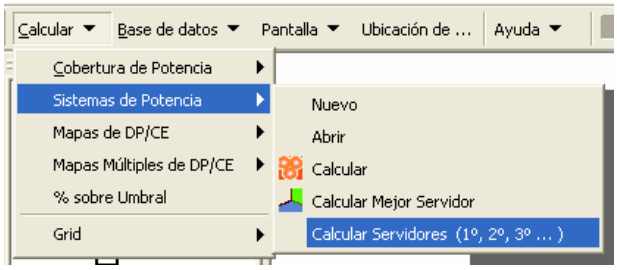


Figura 59: Menú de cálculo de sistemas de potencia. De esta forma es como queda integrada la nueva función.

En el caso de un sistema de potencia nuevo, el programa calculará la capa de mejores servidores. Si dicha capa ya se ha calculado anteriormente, sólo se cargará por pantalla.

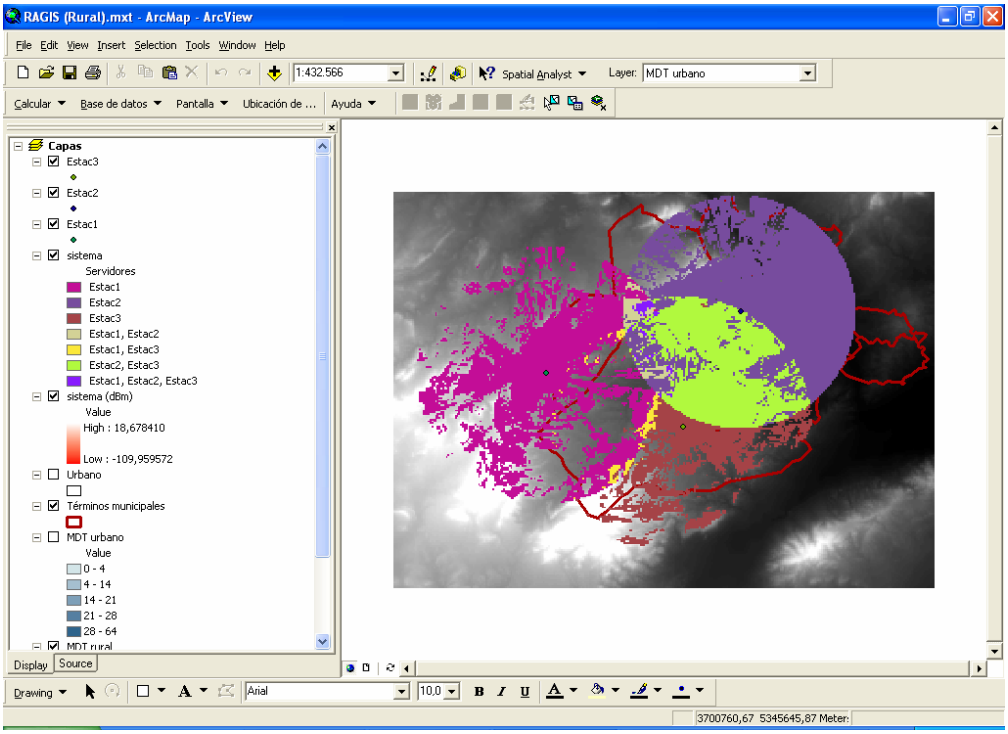


Figura 60: Capa Mejores Servidores. Cada uno de los colores muestra un tipo de servicio para el sistema seleccionado.

La leyenda muestra un color para cada tipo de servicio.

C.2.- % Sobre Umbral

Para esta función, los pasos a seguir por el usuario son:

1.- Seleccionar la opción: *Calcular → % Sobre Umbral*

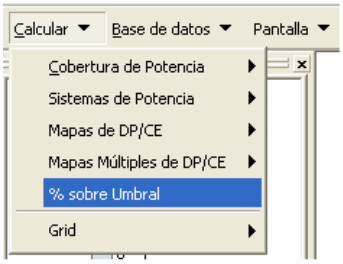


Figura 61: Menú de cálculo de % sobre umbral. La opción del cálculo de % sobre umbral se encuentra en el menú Calcular.

Tras esto, aparece la ventana de introducción de datos:

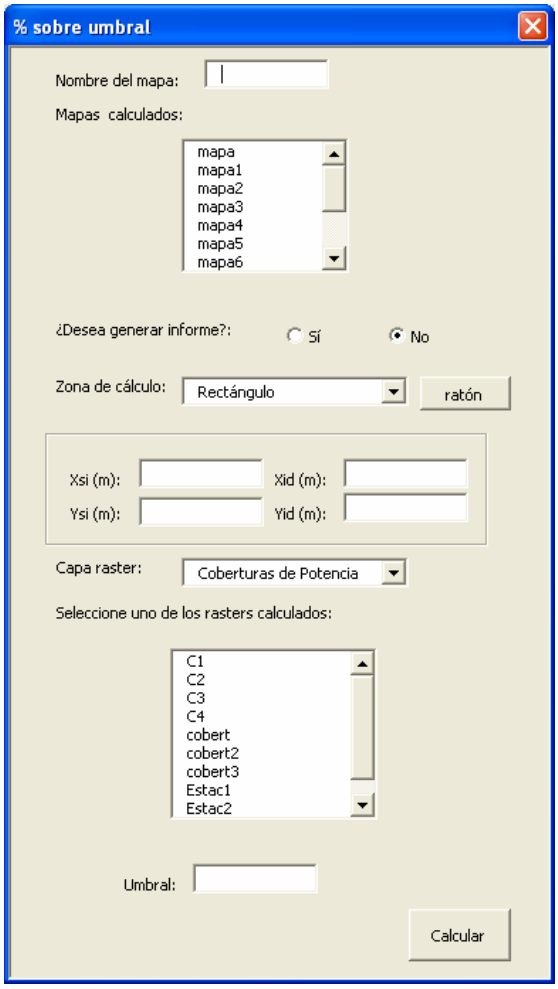


Figura 62: Ventana de introducción de datos.

2.- Introducir los siguientes datos:

2.1.- - Nombre del mapa a calcular

- Seleccionar si se desea generar informe de resultados (archivo Excel)

- Tipo de zona de calculo sobre la que se realizarán los cálculos: Rectángulo, Polígono, Polilínea o Capa Vectorial. En este caso, dependiendo de qué tipo de zona se elija, aparecen o no otros submenús:

2.2.- Si se selecciona la opción de Rectángulo, Polígono o Polilínea, el usuario debe de dibujar con el ratón la zona en el mapa, para ello, se pulsa el botón Ratón de la ventana y con el botón *dibujar con ratón* de la barra de RAGIS, se dibuja la forma deseada en el mapa:

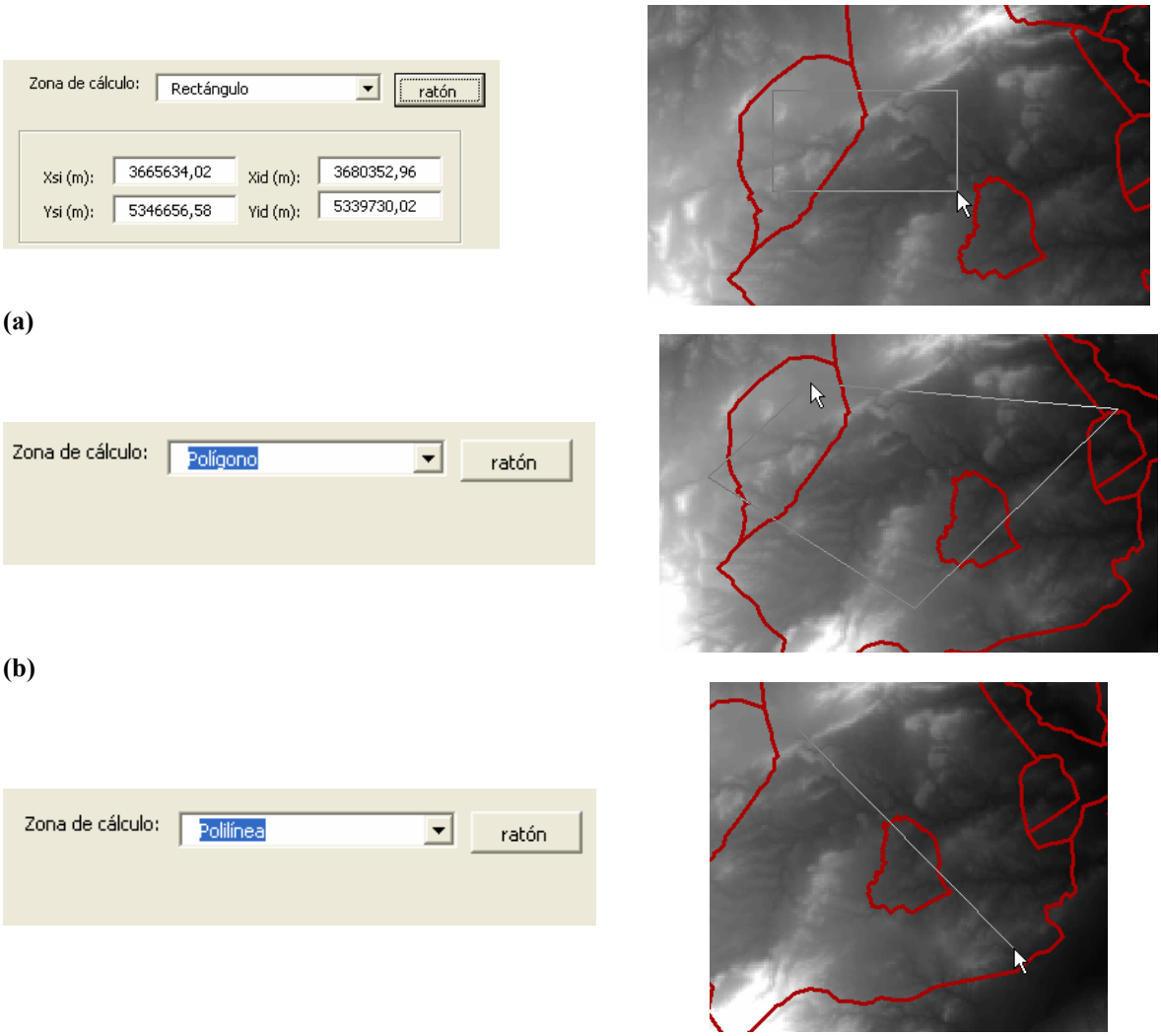


Figura 63: Zonas de cálculo. (a) Rectángulo. El rectángulo como zona de cálculo puede ser definido mediante el ratón o bien introduciendo directamente las coordenadas de sus límites. **(b) Polígono.** La

zona de cálculo puede tener una forma irregular. **(c) Polilínea.** La zona de cálculo queda definida sobre una línea.

En el caso de la elección de una Capa Vectorial como zona de cálculo, aparece en la ventana un segunda lista desplegable con las capas vectoriales cargadas en ese momento en el mapa y que pueden ser elegidas por el usuario.

Después se procede a la selección de cada una de las *features* de la capa vectorial elegida donde se van a llevar a cabo los cálculos. Para ello se pulsa el botón *Select Features* y se seleccionan con el ratón:

Figura 64: Capa Vectorial: Si la zona de calculo elegida es una capa vectorial, se puede elegir una de las que se encuentran cargadas en el mapa a través de una segunda lista desplegable.

2.3.- Si se ha seleccionado la opción de generación de informe de resultados, se activará una tercera lista desplegable en donde se seleccionará el campo de la tabla de atributos de la capa vectorial elegida que se utilizará como identificador en el informe de cada una de las *features* seleccionadas.

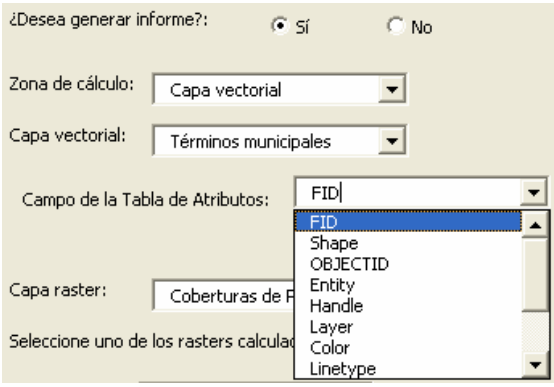


Figura 65: Campo de la Tabla de Atributos. Campos de la tabla de atributos de la capa vectorial seleccionada, se ha de seleccionar uno para identificar las *features* en el informe de resultados.

2.4.- Tipo y capa raster que se va a utilizar para el calculo de las potencias. El tipo puede ser Coberturas de Potencia, Sistemas de Potencia, Mapas individuales de DP/CE, Mapas múltiples de DP/CE u Otro tipo de capa raster guardada en disco. Después de seleccionar el tipo en la lista desplegable de ha de seleccionar el raster de la lista:

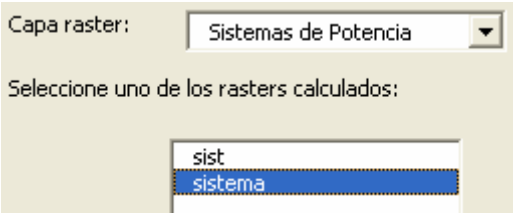


Figura 66: Capa raster. En la lista desplegable se selecciona el tipo de capa raster y en la lista de abajo la capa raster deseada de las que estan guardadas en disco.

Por último introducir el umbral de potencia deseado y pulsar *Calcular*.

2.5.- Después de esto, el programa muestra el mapa calculado y un gráfico con los porcentajes de cobertura / no cobertura sobre el territorio total seleccionado.

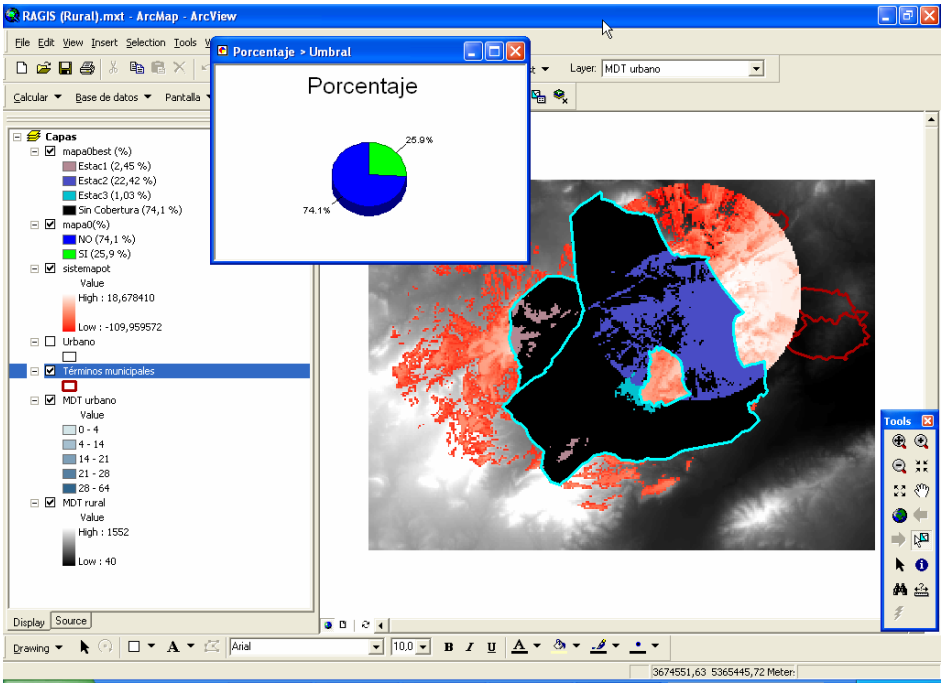


Figura 67: Mapa de % sobre Umbral.

En la carpeta llamada con el nombre del mapa que se encuentra en el directorio RAGIS\porcentajes, se crea el archivo en formato Excel con diversos resultados obtenidos tras los calculos.

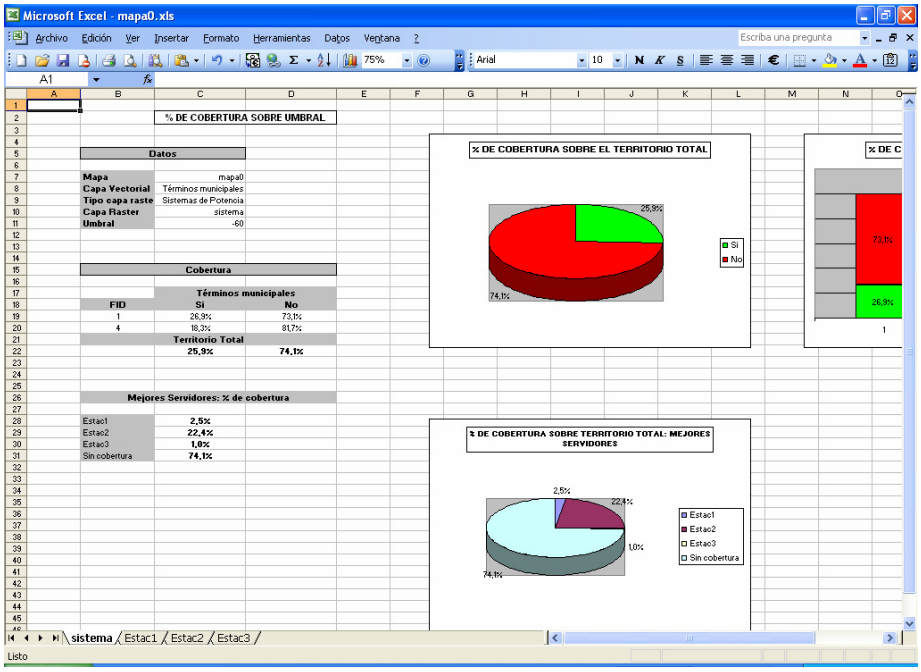


Figura 68: Informe de resultados. Archivo Excel creado en la carpeta del mapa de porcentajes con el informe de resultados obtenidos tras los calculos de los porcentajes**.**

APÉNDICE D: CÓDIGO PROGRAMADO

D.1.- Procedimientos y Funciones

- `Public Sub M_servidores()`

Procedimiento que se encarga del cálculo del raster de mejores servidores. En él se llama a las funciones `Combinaciones()` y `Ordenar()`.

- `Public Function Combinaciones(a)`

Función llamada por la anterior para la formación de la lista de combinaciones entre los emplazamientos que forman el sistema de potencia. Como parámetro de entrada recibe un array de strings con el nombre de los emplazamientos del sistema y como salida devuelve un array con las combinaciones de dichos emplazamientos.

- `Public Function Ordenar(ByVal unsortedarray)`

Función llamada por la anterior que hace la ordenación de la lista de combinaciones de tal manera que la cadena mas corta en longitud del array quede en su primera posición y la más larga en la última para una mayor claridad en la muestra de leyendas. Como parámetro de entrada recibe el array de strings desordenado y lo devuelve ordenado.

- `Public Sub MostrarMejoresServidores(ByVal filename As String, ByVal directorio As String)`

Procedimiento que muestra el raster de mejores servidores guardado en el disco duro con colores discretos aleatorios. Como parámetros de entrada recibe el nombre del raster y el directorio del disco en el que se encuentra guardado.

- `Public Sub calcular_Click()`

Código que se ejecuta al pulsar el botón Calcular de la ventana de introducción de datos de la función de % sobre Umbral. Éste procedimiento de encarga de la comprobación de todos los datos, y de las llamadas a las funciones correspondientes según sea el tipo de capa vectorial a convertir en capa raster.

- `Private Sub zona_Click()`

Parte de código del formulario de introducción de datos de la función % Sobre Umbral que se encarga de cargar los campos de la tabla de atributos de la capa vectorial seleccionada en el ComboBox `zona` en el ComboBox llamado `campos`.

- `Sub Vect_poliARaster(ByVal nombre As String, ByVal lista As String)`

Este procedimiento pasa la información vectorial de tipo Rectágulo, Línea o Polilínea a una capa de tipo raster. Para ello, los parámetros de entrada que recibe son el nombre del mapa que se va a calcular y el nombre de la capa raster seleccionada.

- `Sub VectARaster(ByVal nombre As String, ByVal lista As String, campo As String, capas() As String)`

Procedimiento análogo al anterior para el caso de información de tipo Capa Vectorial. En este caso, se unen las *features* seleccionadas por el usuario en un solo raster de territorio total. Para ello se llama a la función `FeatTotal()`. Los parámetros de entrada en este caso son: el nombre del mapa, el nombre de la capa raster seleccionada, el nombre del campo de la tabla de atributos seleccionada y un array de strings en el que se almacenaran los identificadores para cada *feature* que se usaran en el informe de resultados.

- `Function FeatTotal(pFC As IFeatureClass, pFSel As IFeatureSelection, campo As String, ByRef cadena() As String) As IfeatureClass`

Función llamada por el procedimiento anterior. Realiza la unión de todas las *features* en una sola. Además, almacena para cada *feature* su identificador, dependiendo del contenido de la variable `campo`, en la variable `capas()`. Para ello recibe como parámetro de entrada la selección de *features* y devuelve una sola *feature* que será la unión de ellas.

- `Public Sub umbral(ByVal nombre As String, ByVal c_v As String, ByVal c_r As String, ByVal lista As String, ByVal potencia As Integer, campo As String, capas() As String)`

Procedimiento que se encarga del cálculo del porcentaje de territorio con potencia recibida mayor que el umbral seleccionado. Realiza los cálculos con los rasters, y muestra el raster con los resultados por pantalla. Los parámetros de entrada que recibe son: el nombre del mapa, el nombre de la capa vectorial seleccionada, el tipo de capa raster (cobertura, sistema o mapa de DP/CE), el nombre de la capa raster, el umbral de potencia, el campo de la tabla de atributos y los identificadores correspondientes a este campo para cada *feature*.

- `Public Sub ObtenerSuperficies1(territorio As IRaster, ByRef superficies() As Variant)`

Procedimiento que obtiene las superficies de cada una de las *features* que forman el territorio total. Para ello recibe el raster de todo el territorio y un array en el que se iran almacenando dichas superficies.

- `Public Sub ObtenerSuperficies2(territorio As Variant, ByRef superficies() As Variant, ByRef suma As Variant)`

Este procedimiento obtiene la superficie del raster del territorio total a partir de la suma de las superficies de cada una de las *features*.

- `Public Sub CrearExcel(tipo As Integer, nombre As String, c_r As String, lista As String, c_v As String, umbral As String, campo As String, capas() As String, territorio As IRaster, superficies() As Variant, Optional servs As Variant)`

Procedimiento que crea el archivo en formato Excel del informe de resultados. Para ello, recibe como parámetros de entrada, un entero indicando si la capa raster es de tipo cobertura (o sistema sin el raster de mejor servidor calculado) o sistema, el nombre del mapa, el tipo de capa raster, el nombre de dicha capa raster, el nombre de la capa vectorial, el umbral de potencia, el nombre del campo de la tabla de atributos, los identificadores de cada *feature*, el raster del territorio total, un array con las superficies de cada *feature* y un array opcional, que se pasará en el caso de sistemas con mejor servidor, con los rasters de los porcentajes de territorio de cobertura calculados para cada *feature* y el territorio total por el procedimiento umbral.

D.2.- Código Fuente

```
'Este procedimiento calcula los emplazamientos pertenecientes al sistema que dan servicio en
'cada punto del mapa

Public Sub M_servidores()

        Load sis_nuevo
        Load sis_abrir

        If ((sis_nuevo.nombre = "") And (sis_abrir.nombre = "")) = False Then

             Dim file_sis As New FileSystemObject
             Dim k As Integer
             Dim j As Integer
             Dim i As Integer

             Dim l As Integer
             Dim m As Integer
             Dim n As Integer

             Dim pmxdoc As IMxDocument
             Set pmxdoc = ThisDocument
             Dim pmap As IMap
             Set pmap = pmxdoc.FocusMap

             Dim pWSFcob As IWorkspaceFactory2
             Dim pRWScob As IRasterWorkspace2
             Set pWSFcob = New RasterWorkspaceFactory
             If Not pWSFcob.IsWorkspace(rutas.coberturas) Then Exit Sub
             Set pRWScob = pWSFcob.OpenFromFile(rutas.coberturas, 0)

             Dim pWSFsis As IWorkspaceFactory2
             Dim pRWSsis As IRasterWorkspace2
             Set pWSFsis = New RasterWorkspaceFactory
             If Not pWSFsis.IsWorkspace(rutas.sistemas) Then Exit Sub
             Set pRWSsis = pWSFsis.OpenFromFile(rutas.sistemas, 0)

             Dim servidoresDS As IRasterDataset
             Dim servidores As IRaster

             'la idea es que sólo se puede calcular el best-server una vez hayamos
             'calculado el sistema de potencia

             If file_sis.FolderExists(rutas.sistemas + nombre + "\") = False Then

                  MsgBox "¡¡¡Debe calcular primero el sistema de potencia!!!"
             Else

                  'Obtenemos una lista con las posibles combinaciones de las coberturas
                  Dim lista()
                  ReDim lista(1 To (2 ^ ((UBound(cob) + 1)) - 1))
                  lista = Combinaciones(cob)

                  'Si el raster de servidores no existe se calcula. En el caso de haber 'sido
                  ya calculado anteriormente sólo se muestra por pantalla

                  If file_sis.FolderExists(rutas.sistemas + nombre + "serv") = False Then

                       Dim v_celdas As Variant
                       Dim v_indices As Variant
                       Dim v_salida
                       Dim salida_valores
                       Dim salida_indices

                       Dim palgbop As IMapAlgebraOp
                       Set palgbop = New RasterMapAlgebraOp    'Creamos el operador espacial
```

```
Dim pEnv As IRasterAnalysisEnvironment 'Definimos el espacio de
'salida
Set pEnv = palgbop
Dim pWStemp As IWorkspace
Dim pWSFtemp As IWorkspaceFactory2
Set pWSFtemp = New RasterWorkspaceFactory
Set pWStemp = pWSFtemp.OpenFromFile("c:\arcgis\RAGIS\temp\", 0)
Set pEnv.OutWorkspace = pWStemp

Dim pInputRasterDS() As IRasterDataset
Dim pInputRaster1() As IRaster
Dim pInputRaster() As IRaster
Dim pPnt() As IPnt
Dim pInputBand() As IRasterBand
Dim pInputBandCol() As IRasterBandCollection
Dim pInputRasProps() As IRasterProps
Dim pInputRawPixel() As IRawPixels
Dim pInputBlock() As IPixelBlock

ReDim pInputRasterDS(UBound(cob))
ReDim pInputRaster1(UBound(cob))
ReDim pInputRaster(UBound(cob))
ReDim pInputBand(UBound(cob))
ReDim pInputBandCol(UBound(cob))
ReDim pInputRasProps(UBound(cob))
ReDim pPnt(UBound(cob))
ReDim pInputRawPixel(UBound(cob))
ReDim pInputBlock(UBound(cob))

Dim pOrigen As IPnt
Set pOrigen = New DblPnt
pOrigen.x = 0
pOrigen.y = 0


'%%%%%%%%%%%%%%%%%%%%% DATOS DE ENTRADA %%%%%%%%%%%%%%%%%%%%%%%%

Dim RasProps As IRasterProps

Dim xmin
Dim xmax
Dim ymin
Dim ymax
Dim resolucion


For i = 0 To UBound(cob)

     Set pInputRasterDS(i) = pRWScob.OpenRasterDataset(cob(i))
     Set pInputRaster1(i) = pInputRasterDS(i).CreateDefaultRaster
     Set RasProps = pInputRaster1(i)

     If i = 0 Then
          xmin = RasProps.Extent.xmin
          xmax = RasProps.Extent.xmax
          ymin = RasProps.Extent.ymin
          ymax = RasProps.Extent.ymax
          resolucion = CDbl(RasProps.MeanCellSize.x)
     Else
          If (resolucion > RasProps.MeanCellSize.x) Then resolucion =
          RasProps.MeanCellSize.x
          If (xmin > RasProps.Extent.xmin) Then xmin =
          RasProps.Extent.xmin
          If (xmax < RasProps.Extent.xmax) Then xmax =
          RasProps.Extent.xmax
          If (ymin > RasProps.Extent.ymin) Then ymin =
          RasProps.Extent.ymin
          If (ymax < RasProps.Extent.ymax) Then ymax =
```

```
            RasProps.Extent.ymax
      End If

Next

Dim pExt As IEnvelope
Set pExt = New Envelope


For i = 0 To UBound(cob)

      Set pExt = New Envelope

      pExt.xmin = xmin
      pExt.xmax = xmax
      pExt.ymin = ymin
      pExt.ymax = ymax

      pEnv.SetExtent esriRasterEnvValue, pExt

      'Pasamos las celdas NoData a valores reales
      '"Bindamos" los rasters
      Call palgbop.BindRaster(pInputRaster1(i), "R" + Str(i + 1))
      'Realizamos la operación
      Set pInputRaster(i) = palgbop.Execute("con(isnull([R" + Str(i +
      1) + "]), -9998, [R" + Str(i + 1) + "])")
      'Obtenemos las RasterBand de cada raster y las propiedades de
      'éstas
      Set pInputBandCol(i) = pInputRaster(i)
      Set pInputBand(i) = pInputBandCol(i).item(0)
      'Obtenemos las propiedades de los Rasters de entrada
      Set pInputRasProps(i) = pInputBand(i)
      'Obtenemos los IRawPixels de los rasters de entrada
      Set pInputRawPixel(i) = pInputBand(i)
      'Obtenemos los PixelBlocks de los raster de entrada
      Set pPnt(i) = New DblPnt
      pPnt(i).SetCoords pInputRasProps(i).Width,
      pInputRasProps(i).Height
      Set pInputBlock(i) = pInputRawPixel(i).CreatePixelBlock(pPnt(i))
      pInputRawPixel(i).Read pOrigen, pInputBlock(i)
Next

Dim sistema1 As IRaster
Dim sistemaDS As IRasterDataset
Dim sistema As IRaster
Set sistemaDS = pRWSsis.OpenRasterDataset(nombre)

Set sistema1 = sistemaDS.CreateDefaultRaster
Call palgbop.BindRaster(sistema1, "S1")
Set sistema = palgbop.Execute("con(isnull([S1]), -9999, [S1])")
Dim sistemaBand As IRasterBand
Dim sistemaBandCol As IRasterBandCollection
Set sistemaBandCol = sistema
Set sistemaBand = sistemaBandCol.item(0)

Dim sistemaRasProps As IRasterProps
Set sistemaRasProps = sistemaBand

Dim sistemaRawPixel As IRawPixels
Set sistemaRawPixel = sistemaBand

Dim sistemaBlock As IPixelBlock
Set sistemaBlock = sistemaRawPixel.CreatePixelBlock(pPnt(0))

sistemaRawPixel.Read pOrigen, sistemaBlock

Dim pPropSet As IPropertySet
Set pPropSet = New PropertySet
```

```
pPropSet.SetProperty "Database", rutas.sistemas + "ficheros\"

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

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


'%%%%%%%%%%%%%%%%%%%%%%%%%%% DATOS DE SALIDA %%%%%%%%%%%%%%%%%%%%%%

'Creamos el rasterDS de salida que va a contener los servidores
Dim pOrigin As IPoint
Set pOrigin = New point
pOrigin.x = pInputRasProps(0).Extent.xmin
pOrigin.y = pInputRasProps(0).Extent.ymin

Dim outputPixelType As rstPixelType
outputPixelType = PT_LONG


Set servidoresDS = pRWSsis.CreateRasterDataset(nombre + "serv",
"GRID", _ pOrigin, pInputRasProps(0).Width, pInputRasProps(0).Height,
_ pInputRasProps(0).MeanCellSize.x, pInputRasProps(0).MeanCellSize.y,
1, _ outputPixelType, pInputRasProps(0).SpatialReference, True)


'Creamos el default raster del rasterDS de salida
Set servidores = servidoresDS.CreateDefaultRaster

'Obtenemos el array del raster de salida
Dim servidoresBandCol As IRasterBandCollection
Set servidoresBandCol = servidoresDS

Dim servidoresBand As IRasterBand
Dim servidoresRawPixel As IRawPixels
Dim servidoresBlock As IPixelBlock
Dim servidores_v

Set servidoresBand = servidoresBandCol.item(0)
Set servidoresRawPixel = servidoresBand
Set servidoresBlock = servidoresRawPixel.CreatePixelBlock(pPnt(0))

servidores.Read pOrigen, servidoresBlock

servidores_v = servidoresBlock.SafeArray(0)

Dim servidoresRasProps As IRasterProps
Set servidoresRasProps = servidoresBand

'Comparamos los Rasters de coberturas para obtener las bandas de los
'servidores
ReDim v_salida(UBound(cob))
ReDim v_celda(UBound(cob))


'Obtenemos los solapmientos de las coberturas del sistema en cada
'punto

Dim v_solap() As String
ReDim v_solap(UBound(cob))
Dim cadena As String
```

```
        Dim b As Integer
        Dim a As Integer

        For k = 0 To pInputRasProps(0).Width - 1

            For j = 0 To pInputRasProps(0).Height - 1

                If sistemaBlock.GetVal(0, k, j) <> -9999 Then

                    cadena = ""
                    For i = 0 To UBound(cob)

                        If (pInputBlock(i).GetVal(0, k, j) <> -9998) Then

                            If cadena = "" Then

                                cadena = cadena + "  " + cob(i)

                            Else

                                cadena = cadena + ", " + cob(i)

                            End If

                        End If

                    Next

                    a = 0
                    For i = 1 To UBound(lista)

                        If cadena = lista(i) Then

                            servidores_v(k, j) = i
                            a = a + 1
                            Exit For

                        End If

                    Next

                    If a = 0 Then: servidores_v(k, j) =
                    servidoresRasProps.NoDataValue

                Else

                    servidores_v(k, j) = servidoresRasProps.NoDataValue

                End If
            Next
        Next

        servidoresRawPixel.Write pOrigen, servidoresBlock

    Else
        'Si el raster de servidores ya ha sido calculado, sólo lo mostramos
        'por  pantalla
        borra.capa2 nombre + "serv"
        Set servidoresDS = pRWSsis.OpenRasterDataset(nombre + "serv")
        Set servidores = servidoresDS.CreateDefaultRaster
    End If

    Dim pRLayer As IRasterLayer
    Set pRLayer = New RasterLayer
    pRLayer.CreateFromRaster servidores

    Dim pcolor As IRgbColor
```

```
                Set pcolor = New RgbColor

                Dim pUvren As IRasterUniqueValueRenderer
                Set pUvren = New RasterUniqueValueRenderer

                Dim pren As IRasterRenderer
                Set pren = pUvren

                Set pren.raster = pRLayer.raster
                pren.Update

                Dim psym As IColorSymbol

                For j = 0 To pUvren.ClassCount(0) - 1

                     pUvren.Label(0, j) = ""
                     pUvren.Heading(0) = "Servidores"
                     pUvren.Label(0, j) = lista(pUvren.Value(0, j, 0))
                     pcolor.Red = Int(255 * Rnd)
                     pcolor.Green = Int(255 * Rnd)
                     pcolor.Blue = Int(255 * Rnd)
                     Set psym = New ColorSymbol
                     psym.Color = pcolor

                     pUvren.Symbol(0, j) = psym

                Next

                pren.Update
                Set pRLayer.Renderer = pren

                pRLayer.name = nombre

                pmap.AddLayer pRLayer

                For i = 0 To UBound(cob)

                     borra.capa2 cob(i)
                     cober.abrir_shape cob(i)

                Next

           End If

        Else

           MsgBox "Los datos del sistema no han sido cargados", vbExclamation

        End If

End Sub
```

```
'Función llamada para ordenar la lista de combinaciones de menor a mayor tamaño de las
'cadenas
        'Parámetros de entrada:
           'Unsortedarray: array de cadenas desornado
        'Parámetros de salida:
           'array de cadenas con las cadenas ordenadas de menor a mayor longitud

Public Function Ordenar(ByVal unsortedarray)

        Dim Front, Back, i, Loc
        Dim aux1, aux2, Arrsize
```

```
        Dim indexarray() As Variant

        Arrsize = UBound(unsortedarray)    'Longitud del array

        ReDim indexarray(0 To Arrsize)


        For i = 0 To Arrsize

              indexarray(i) = i + 1

        Next

        For Front = 0 To Arrsize - 1

              Loc = Front

              For Back = Front To Arrsize

                    If unsortedarray(Loc) > unsortedarray(Back) Then

                          'Obtención del elemento mayor
                          Loc = Back

                    End If
              Next

              aux1 = unsortedarray(Loc)
              aux2 = indexarray(Loc)
              unsortedarray(Loc) = unsortedarray(Front)
              indexarray(Loc) = indexarray(Front)
              unsortedarray(Front) = aux1
              indexarray(Front) = aux2

        Next

        Ordenar = indexarray

End Function
```

```
'Esta función devuelve una array de strings con todas las combinaciones posibles de
'coberturas
        'Parámetros de entrada:
              'a: array de strings del que se obtendran las combinaciones

Public Function Combinaciones(a)

        Dim listas2() As String
        ReDim listas2(1 To (2 ^ ((UBound(a) + 1)) - 1))

        For i = 1 To ((2 ^ (UBound(a) + 1)) - 1)

              For numact = 0 To UBound(a)

                    If listas2(i) = "" Then

                          If (i And (2 ^ numact)) <> 0 Then

                                listas2(i) = listas2(i) + "  " + a(numact)

                          End If
```

```
                Else

                    If (i And (2 ^ numact)) <> 0 Then

                        listas2(i) = listas2(i) + ", " + a(numact)

                    End If
                End If

            Next

        Next

        Dim u()
        ReDim u(0 To (UBound(listas2) - 1))

        Dim u2()
        ReDim u2(0 To (UBound(listas2) - 1))

        For i = 0 To UBound(listas2) - 1

            u(i) = Len(listas2(i + 1))

        Next

        u2 = Ordenar(u)

        ReDim listas(1 To UBound(listas2))

        For i = 1 To UBound(listas2)

            listas(i) = listas2(u2(i - 1))

        Next

        Combinaciones = listas

End Function
```

```
'Muestra en pantalla un rasterDataSet (en el disco duro) donde no se emplea
'una rampa de colores sino colores discretos aleatorios

Public Sub MostrarMejoresServidores(ByVal filename As String, ByVal directorio As String)

        Dim pWSF As IWorkspaceFactory
        Dim pWS As IRasterWorkspace
        Dim pRasterDS As IRasterDataset
        Dim pRL As IRasterLayer
        Dim pmxdoc As IMxDocument
        Dim pmap As IMap

        Set pmxdoc = ThisDocument
        Set pmap = pmxdoc.FocusMap

        Set pWSF = New RasterWorkspaceFactory
        Set pWS = pWSF.OpenFromFile(directorio, 0)

        Set pRasterDS = pWS.OpenRasterDataset(filename)
        Set pRL = New RasterLayer
        pRL.CreateFromDataset pRasterDS

        'Obtenemos una lista con las posibles combinaciones de las coberturas
        Dim lista()
```

```
        ReDim lista(1 To (2 ^ ((UBound(cob) + 1)) - 1))
        lista = Combinaciones(cob)

        Dim pcolor As IRgbColor
        Set pcolor = New RgbColor

        Dim pUvren As IRasterUniqueValueRenderer
        Set pUvren = New RasterUniqueValueRenderer

        Dim pren As IRasterRenderer
        Set pren = pUvren

        Set pren.raster = pRL.raster
        pren.Update

        Dim psym As IColorSymbol

             For j = 0 To pUvren.ClassCount(0) - 1

                  pUvren.Label(0, j) = ""
                  pUvren.Heading(0) = "Servidores"
                  pUvren.Label(0, j) = lista(pUvren.Value(0, j, 0))
                  pcolor.Red = Int(255 * Rnd)
                  pcolor.Green = Int(255 * Rnd)
                  pcolor.Blue = Int(255 * Rnd)
                  Set psym = New ColorSymbol
                  psym.Color = pcolor

                  pUvren.Symbol(0, j) = psym
             Next

        pren.Update
        Set pRL.Renderer = pren

        pRL.name = filename

        pmap.AddLayer pRL

End Sub
```

```
'Codigo que se ejecuta al pulsar el boton Calcular de la ventana de 'introducion de datos.
'Comprueba si faltan datos y se encarga de llamar a las demas funciones y procedimientos

Public Sub calcular_Click()

        dibuja.dibujo = 2

        If Len(municipio.nombre.text) > 8 Then

             MsgBox "El nombre del porcentaje excede de 8 caracteres" & " " & Err.Description,
             vbExclamation, "Error"
             Exit Sub

            End If

        Dim file As New FileSystemObject
        Dim path As String
        Dim path2 As String
        Dim path3 As String
        Dim path4 As String
        Dim path5 As String
```

```
Dim ok As Boolean
ok = True

path = rutas.coberturas
path2 = rutas.porcentajes
path3 = rutas.sistemas
path4 = rutas.densidad
path5 = rutas.sistemasdens

If municipio.zona2.text <> "Otro" Then
     If (municipio.lista.text = "") Then
          MsgBox "Faltan datos" & " " & Err.Description, vbExclamation, "Error"
          Exit Sub
     End If
Else
     If municipio.raster_origen.text = "" Then
          MsgBox "Faltan datos" & " " & Err.Description, vbExclamation, "Error"
          Exit Sub
     End If
End If

If municipio.nombre.text = "" Or municipio.umbral.text = "" Then

     MsgBox "Faltan datos" & " " & Err.Description, vbExclamation, "Error"
     Exit Sub

ElseIf file.FolderExists(path2 + municipio.nombre.text) Then

     adver_porc.Label1 = "El mapa '" + municipio.nombre.text + "' ya existe. ¿Desea
     reemplazarlo?"
     adver_porc.Show
     Exit Sub

ElseIf municipio.figura.text = "Polilínea" And Not
file.FileExists("c:\arcgis\RAGIS\temp\poli.shp") Then

     MsgBox "No se ha creado la polilínea" & " " & Err.Description, vbExclamation,
     "Error"
     Exit Sub

ElseIf municipio.figura.text = "Capa vectorial" And municipio.zona.text = "" Then

     MsgBox "No se ha seleccionado ninguna capa vectorial" & " " & Err.Description,
     vbExclamation, "Error"
     Exit Sub

ElseIf municipio.figura.text = "Polígono" And Not
file.FileExists("c:\arcgis\RAGIS\temp\poly.shp") Then

     MsgBox "No se ha creado el polígono" & " " & Err.Description, vbExclamation,
     "Error"
     Exit Sub

ElseIf municipio.figura.text = "Rectángulo" And Not
file.FileExists("c:\arcgis\RAGIS\temp\poly.shp") Then

     MsgBox "No se ha creado el rectángulo" & " " & Err.Description, vbExclamation,
     "Error"
     Exit Sub

Else

     Dim pmxdoc As IMxDocument
```

```
Set pmxdoc = ThisDocument
Dim pmap As IMap
Set pmap = pmxdoc.FocusMap
Dim numero As Integer
Dim layer(50) As ILayer

If municipio.figura.text = "Capa vectorial" Then

     varios.selecciona_capa municipio.zona.text

     Dim pFlayer As IFeatureLayer
     Set pFlayer = pmxdoc.SelectedLayer

     Dim pFSel As IFeatureSelection
     Set pFSel = pFlayer

     ok = True

     If pFSel.SelectionSet.count = 0 Then

          MsgBox "Seleccionar un elemento de la capa" & " " & Err.Description,
          vbExclamation, "Error"
          ok = False
          Exit Sub

     End If

End If

file.CreateFolder (path2 + municipio.nombre.text)

numero = pmap.LayerCount

pmxdoc.UpdateContents

If municipio.lista.text <> "" Then

     'Cargamos la cobertura
     If municipio.zona2.text = "Coberturas de Potencia" Then

          cober.CreateRasterDataset_muni municipio.lista.text + "pot", path

     End If

     If municipio.zona2.text = "Sistemas de Potencia" Then

          cober.CreateRasterDataset_muni municipio.lista.text + "pot", path3

     End If

     If municipio.zona2.text = "Mapas individuales de DP/CE" Then

          cober.CreateRasterDataset_muni municipio.lista.text, path4

     End If

     If municipio.zona2.text = "Mapas múltiples de DP/CE" Then

          cober.CreateRasterDataset_muni municipio.lista.text, path5
```

```
        End If

    End If

    pmxdoc.ActivatedView.Refresh


    borra.temporal

    Dim name As String
    Dim sistema As Integer

    If municipio.lista.text <> "" Then

        name = municipio.lista.text

    End If

    If municipio.zona2.text = "Coberturas de Potencia" Then

        sistema = 1

    ElseIf municipio.zona2.text = "Sistemas de Potencia" Then

        sistema = 2

    ElseIf municipio.zona2.text = "Mapas individuales de DP/CE" Then

        sistema = 3

    ElseIf municipio.zona2.text = "Mapas múltiples de DP/CE" Then

        sistema = 4

    ElseIf municipio.zona2.text = "Otro" Then

        sistema = 5
        name = municipio.nombre_raster

    End If


    If ok = True Then 'hemos seleccionado una zona

        municipio.Hide      'Se cierra la ventana y comenzamos los cálculos

        If si.Value = True Then

            Dim capas() As String

            If municipio.figura.text = "Capa vectorial" Then

                'Convertimos la feature seleccionada en raster
                porcentaje.VectARaster municipio.nombre.text, name,
                municipio.campos.text, capas
                porcentaje.umbral municipio.nombre.text, municipio.zona.text,
                municipio.zona2.text, name, municipio.umbral.text,
                municipio.campos.text, capas
```

```
                    ElseIf municipio.figura.text = "Rectángulo" Or municipio.figura.text =
                    "Polígono" Or municipio.figura.text = "Polilínea" Then

                         municipio.campos.text = ""
                         municipio.zona = ""
                         ReDim capas(0)
                         capas(0) = municipio.figura.text
                         'Convertimos la feature seleccionada en raster
                         porcentaje.Vect_poliARaster municipio.nombre.text, name
                         porcentaje.umbral municipio.nombre.text, municipio.figura.text,
                         municipio.zona2.text, name, municipio.umbral.text,
                         municipio.campos.text, capas

                    End If

               Else

                    If municipio.figura.text = "Capa vectorial" Then

                         'convertimos la feature seleccionada en raster
                         porcentaje.Feat2Raster municipio.nombre.text, name,
                         Replace(municipio.umbral.text, ".", ",")

                    ElseIf municipio.figura.text = "Rectángulo" Then

                         porcentaje.Feat2Rasterpoli municipio.nombre.text, name,
                         Replace(municipio.umbral.text, ".", ",")

                    ElseIf municipio.figura.text = "Polígono" Or municipio.figura.text =
                    "Polilínea" Then

                         'convertimos la feature seleccionada en raster
                         porcentaje.Feat2Rasterpoli municipio.nombre.text, name,
                         Replace(municipio.umbral.text, ".", ",")

                    End If

                    porcentaje.cambios municipio.nombre.text, name,
                    Replace(municipio.umbral.text, ".", ","), sistema

                    If sistema = 2 And file.FolderExists(rutas.sistemas +
                    municipio.lista.text + "best") Then

                         porcentaje.cambios2 municipio.nombre.text, name,
                         Replace(municipio.umbral.text, ".", ","), sistema

                    End If

               End If

          End If

     End If

     borra.temporal
     almacen_muni.Show

End Sub
```

```
'Carga los campos de la tabla de atributos de la capa vectorial seleccionada

Private Sub zona_Click()

        Dim pmxdoc As IMxDocument
        Set pmxdoc = ThisDocument

        Dim path As String
        Dim path2 As String
        Dim path3 As String
        Dim path4 As String
        Dim path5 As String
        Dim path6 As String
        Dim path7 As String

        path = rutas.coberturas + "ficheros\"
        path2 = rutas.sistemas + "ficheros\"
        path3 = rutas.coberturas
        path4 = rutas.sistemas
        path5 = rutas.porcentajes
        path6 = rutas.densidad
        path7 = rutas.sistemasdens

        Dim file As New FileSystemObject

        Dim pmap As IMap
        Set pmap = pmxdoc.FocusMap

        Dim numero As Integer
        Dim layer() As ILayer
        Dim capa As IFeatureLayer
        Dim capa2 As IFeatureClass

        numero = pmap.LayerCount
        ReDim layer(numero - 1)

        Dim nombre As String
        Dim posicion

        For i = 0 To numero - 1

              Set layer(i) = pmap.layer(i)

              If layer(i).name <> municipio.zona.text Then

                    nombre = layer(i).name
                    posicion = InStr(1, nombre, " ")

                    If posicion <> 0 Then
                          nombre = Left(nombre, posicion - 1)
                    End If

                    If (TypeOf layer(i) Is IFeatureLayer = False) And
                    (file.FolderExists(path7 + nombre) = False) And
                    (file.FolderExists(path6 + nombre) = False) And
                    (file.FolderExists(path5 + nombre) = False) And
                    (file.FolderExists(path3 + nombre) = False) And
                    (file.FolderExists(path4 + nombre) = False) Then

                          layer(i).Visible = True
                          pmxdoc.ActivatedView.Refresh
```

```
                Else

                     layer(i).Visible = False
                     pmxdoc.ActivatedView.Refresh

                End If

            Else

                municipio.campos.Clear

                Set capa = layer(i)
                Set capa2 = capa.FeatureClass

                For j = 0 To capa2.Fields.FieldCount - 1

                     municipio.campos.AddItem capa2.Fields.field(j).name

                Next

                municipio.campos.text = municipio.campos.List(0)

                layer(i).Visible = True
                varios.selecciona_capa layer(i).name
                pmxdoc.ActivatedView.Refresh

            End If

        Next

End Sub
```

```
'Pasa información vectorial a raster
            'Parámetros de entrada:
                     'nombre: nombre del mapa a calcular
                     'lista: nombre de la capa raster seleccionada para los calculos

Sub Vect_poliARaster(ByVal nombre As String, ByVal lista As String)

        Dim path As String

        If municipio.zona2.text = "Coberturas de Potencia" Then

            path = rutas.coberturas

        ElseIf municipio.zona2.text = "Sistemas de Potencia" Then

            path = rutas.sistemas

        ElseIf municipio.zona2.text = "Mapas individuales de DP/CE" Then

            path = rutas.densidad

        ElseIf municipio.zona2.text = "Mapas múltiples de DP/CE" Then

            path = rutas.sistemasdens
```

```
End If

Dim dCellSize As Double
dCellSize = ficheros.resolucion_raster(lista, path) 'habría que hacerlo genérico
'para cualquier mdt

Dim pmxdoc As IMxDocument
Set pmxdoc = ThisDocument

Dim pPropSet As IPropertySet
Set pPropSet = New PropertySet
pPropSet.SetProperty "Database", "c:\arcgis\RAGIS\temp\"

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

Dim pfeatcls As IFeatureClass
Dim fig As String



If municipio.figura.text = "Polilínea" Then

     fig = "poli"

Else

     fig = "poly"

End If

Set pfeatcls = pAccessFeatWS.OpenFeatureClass(fig)

Dim pScratchDS As IDataset
Set pScratchDS = pfeatcls

Dim pRWSF As IWorkspaceFactory
Set pRWSF = New RasterWorkspaceFactory

Dim pRWS As IRasterWorkspace
Set pRWS = pRWSF.OpenFromFile("c:\arcgis\RAGIS\temp\", 0)

On Error Resume Next
Dim prds As IRasterDataset
Set prds = pRWS.OpenRasterDataset(pScratchDS.name)

If Not prds Is Nothing Then

     Dim pDS As IDataset
     Set pDS = prds

     If Not pDS.CanDelete Then

          Debug.Print "unable to delete old raster dataset"
          Exit Sub
```

```
            Else

                Debug.Print "deleting old raster datset"
                pDS.Delete

            End If

        End If

        Dim pConvOp As IConversionOp
        Set pConvOp = New RasterConversionOp

        Dim pRAEnv As IRasterAnalysisEnvironment
        Set pRAEnv = pConvOp

        pRAEnv.SetCellSize esriRasterEnvValue, dCellSize

        Set prds = pConvOp.ToRasterDataset(pfeatcls, "GRID", pRWS, "1")

        Dim palgbop As IMapAlgebraOp
        Set palgbop = New RasterMapAlgebraOp

        Dim pEnv As IRasterAnalysisEnvironment
        Set pEnv = palgbop

        Set pEnv.OutWorkspace = pRWS

        Dim pRaster As IRaster
        Dim prds2 As IRasterDataset
        Set pRaster = prds.CreateDefaultRaster

        Call palgbop.BindRaster(pRaster, "raster")

        Dim out As IRaster

        Set out = palgbop.Execute("con( [raster] <> 1, 1, 1)")

        Set prds2 = pConvOp.ToRasterDataset(out, "GRID", pRWS, nombre)

        Call palgbop.UnbindRaster("raster")
        Set pRaster = Nothing
        Set out = Nothing

        Call borra.grid("c:\arcgis\RAGIS\temp\", "1")

        Dim pRLayer As IRasterLayer
        Set pRLayer = New RasterLayer
        pRLayer.CreateFromDataset prds2

        pScratchDS.Delete

End Sub
```

```
'Convierte información vectorial (capa vectorial) en raster
        'Parámetros de entrada:
             'nombre: nombre de la capa vectorial
             'lista: nombre de la capa raster
             'potencia: umbral de potencia
             'campo: nombre del campo de la tabla de atributos seleccionado
             'capas(): array que llevará el contenido correspondiente al campo
             elegido de cada feature seleccionada

Sub VectARaster(ByVal nombre As String, ByVal lista As String, campo As
String, capas() As String)

        Dim path As String

        If municipio.zona2.text = "Coberturas de Potencia" Then

             path = rutas.coberturas

        ElseIf municipio.zona2.text = "Sistemas de Potencia" Then

             path = rutas.sistemas

        ElseIf municipio.zona2.text = "Mapas individuales de DP/CE" Then

             path = rutas.densidad

        ElseIf municipio.zona2.text = "Mapas múltiples de DP/CE" Then

             path = rutas.sistemasdens

        ElseIf municipio.zona2.text = "Otro" Then

             path = municipio.directorio_raster + "\"

        End If

        Dim dCellSize As Double
        dCellSize = ficheros.resolucion_raster(lista, path)

        Dim pmxdoc As IMxDocument
        Set pmxdoc = ThisDocument

        Dim pFlayer As IFeatureLayer
        Set pFlayer = pmxdoc.SelectedLayer

        Dim pFSel As IFeatureSelection
        Set pFSel = pFlayer

        If pFSel.SelectionSet.count = 0 Then

             MsgBox "No se ha seleccionado ninguna zona" & " " &
             Err.Description, vbExclamation, "Error"
             Exit Sub

        End If

        ReDim capas(pFSel.SelectionSet.count - 1)

        Dim pScratchFC As IFeatureClass
        Set pScratchFC = FeatTotal(pFlayer.FeatureClass, pFSel, campo, capas)

        Dim pScratchDS As IDataset
```

```
        Set pScratchDS = pScratchFC

        Dim pRWSF As IWorkspaceFactory
        Set pRWSF = New RasterWorkspaceFactory

        Dim pRWS As IRasterWorkspace
        Set pRWS = pRWSF.OpenFromFile("c:\arcgis\RAGIS\temp\", 0)

        On Error Resume Next
        Dim prds As IRasterDataset
        Set prds = pRWS.OpenRasterDataset(pScratchDS.name)

        If Not prds Is Nothing Then

             Dim pDS As IDataset
             Set pDS = prds
             If Not pDS.CanDelete Then

                  Debug.Print "unable to delete old raster dataset"
                  Exit Sub

             Else

                  Debug.Print "deleting old raster datset"
                  pDS.Delete

             End If

        End If

        Dim pConvOp As IConversionOp
        Set pConvOp = New RasterConversionOp


        Dim pRAEnv As IRasterAnalysisEnvironment
        Set pRAEnv = pConvOp
        pRAEnv.SetCellSize esriRasterEnvValue, dCellSize

        Set prds = pConvOp.ToRasterDataset(pScratchFC, "GRID", pRWS, nombre)

        Dim pRLayer As IRasterLayer
        Set pRLayer = New RasterLayer
        pRLayer.CreateFromDataset prds

        pScratchDS.Delete

End Sub
```

```
'Función llamada por la anterior. Toma una selección de features y las une en una sola.
        'Parámetros de entrada:
             'pFC:
             'pFSel: Selección de features
             'campo: nombre del campo de la tabla de atributos de las features seleccionado
             'cadena: array de strings donde se almacenaran el contenido de las columnas de
             'las tablas de atributos de cada feature correspondientes al campo
             'seleccionado. Los cambios realizados en este variable se mantienen
        'Parámetros de salida:
             'Feature formada por todas las features seleccionadas por el usuario
```

```
Function FeatTotal(pFC As IFeatureClass, pFSel As IFeatureSelection, campo As String, ByRef
cadena() As String) As IFeatureClass

        Dim pSWSF As IScratchWorkspaceFactory
        Set pSWSF = New ScratchWorkspaceFactory

        Dim pFWS As IFeatureWorkspace
        Set pFWS = pSWSF.DefaultScratchWorkspace

        Dim pInDataset As IDataset
        Set pInDataset = pFC

        On Error Resume Next
        Dim pDS As IDataset
        Set pDS = pFWS.OpenFeatureClass(pInDataset.name)

        If Not pDS Is Nothing Then

             Debug.Print "deleting " & pDS.name
             pDS.Delete

        End If

        Dim pUID As New UID
        pUID.Value = "esricore.Feature"

        Dim pScratchFC As IFeatureClass
        Set pScratchFC = pFWS.CreateFeatureClass(pInDataset.name, pFC.Fields, pUID, Nothing,
        esriFTSimple, "Shape", "")

        Dim pInFCur As IFeatureCursor
        pFSel.SelectionSet.Search Nothing, False, pInFCur

        Dim pOutFCur As IFeatureCursor
        Set pOutFCur = pScratchFC.Insert(True)

        Dim pInFeat As IFeature
        Dim pOutBuff As IFeatureBuffer
        Set pOutBuff = pScratchFC.CreateFeatureBuffer
        Set pInFeat = pInFCur.NextFeature
        Dim l As Long
        Dim i As Integer

        Do While Not pInFeat Is Nothing

             For l = 0 To pOutBuff.Fields.FieldCount - 1

                  pOutBuff.Value(l) = pInFeat.Value(l)
                  If pOutBuff.Fields.field(l).name = campo Then

                       cadena(i) = pInFeat.Value(l)
                       i = i + 1

                  End If

             Next l

             pOutFCur.InsertFeature pOutBuff
             Set pInFeat = pInFCur.NextFeature

        Loop

        pOutFCur.Flush
```

```
        Set FeatTotal = pScratchFC

End Function
```

```
'Rutina que se encarga del cálculo del % sobre umbral
        'Parametros de entrada:
             'nombre: nombre del mapa de cálculo de umbral
             'c_v: nombre de la capa vectorial seleccionada (una de las capas disponibles)
             'c_r: tipo de capa raster (sistema o cobertura)
             'lista: nombre de la capa raster
             'potencia: umbral de potencia
             'campo: nombre del campo de la tabla de atributos de las features seleccionado
             'capas(): array de strings con los identificadores de cada una de las features
             'seleccionadas

Public Sub umbral(ByVal nombre As String, ByVal c_v As String, ByVal c_r As String, ByVal
lista As String, ByVal potencia As Integer, campo As String, capas() As String)

        Dim pWS As IRasterWorkspace
        Dim pWSF As IWorkspaceFactory
        Set pWSF = New RasterWorkspaceFactory
        Set pWS = pWSF.OpenFromFile(rutas.coberturas, 0)
        Dim pWS2 As IRasterWorkspace
        Set pWS2 = pWSF.OpenFromFile(rutas.sistemas, 0)
        Dim pWS3 As IRasterWorkspace
        Set pWS3 = pWSF.OpenFromFile(rutas.densidad, 0)
        Dim pWS4 As IRasterWorkspace
        Set pWS4 = pWSF.OpenFromFile(rutas.sistemasdens, 0)
        Dim pWS6 As IRasterWorkspace
        Set pWS6 = pWSF.OpenFromFile(municipio.directorio_raster, 0)

        Dim pmxdoc As IMxDocument
        Set pmxdoc = ThisDocument
        Dim pmap As IMap
        Set pmap = pmxdoc.FocusMap

        Dim rds As IRasterDataset
        Dim tipo As Integer

        If c_r = "Coberturas de Potencia" Then

             Set rds = pWS.OpenRasterDataset(lista + "pot")
             tipo = 1    'Cobertura

        ElseIf c_r = "Sistemas de Potencia" Then

             Set rds = pWS2.OpenRasterDataset(lista + "pot")
             tipo = 2    'Sistema

        ElseIf c_r = "Mapas individuales de DP/CE" Then

             Set rds = pWS3.OpenRasterDataset(lista)
             tipo = 1   'Mapa individual (Cobertura)

        ElseIf c_r = "Mapas múltiples de DP/CE" Then

             Set rds = pWS4.OpenRasterDataset(lista)
             tipo = 1   'Mapa múltiple, no se considera el caso de mejores servidores
             (Cobertura)

        ElseIf c_r = "Otro" Then

             Set rds = pWS6.OpenRasterDataset(lista)
```

```
        path = municipio.directorio_raster + "\"
        tipo = 1   'Otro tipo de capa raster, no se considera el caso de mejores
        servidores (Cobertura)

    End If

    Dim pinraster1 As IRaster
    Set pinraster1 = rds.CreateDefaultRaster

    Dim palgbop As IMapAlgebraOp
    Set palgbop = New RasterMapAlgebraOp

    Dim pExt As IEnvelope
    Set pExt = New Envelope

    Dim pInputRasProps As IRasterProps

    Dim pEnv As IRasterAnalysisEnvironment
    Set pEnv = palgbop

    Dim pWStemp As IRasterWorkspace
    Dim pWSFtemp As IWorkspaceFactory
    Set pWSFtemp = New RasterWorkspaceFactory
    Dim compru As New FileSystemObject

    If compru.FolderExists("c:\arcgis\RAGIS\temp\") = False Then

        compru.CreateFolder "c:\arcgis\RAGIS\temp\"

    End If

    Set pWStemp = pWSFtemp.OpenFromFile("c:\arcgis\RAGIS\temp\", 0)

    'Cargamos el raster de la "feature" que se encuentra en temp
    Dim rds2 As IRasterDataset
    Set rds2 = pWStemp.OpenRasterDataset(nombre)

    Dim pinraster2 As IRaster
    Set pinraster2 = rds2.CreateDefaultRaster

    Set pInputRasProps = pinraster2

    Set pExt = New Envelope
    pExt.xmin = pInputRasProps.Extent.xmin
    pExt.ymin = pInputRasProps.Extent.ymin
    pExt.xmax = pInputRasProps.Extent.xmax
    pExt.ymax = pInputRasProps.Extent.ymax
    pEnv.SetExtent esriRasterEnvValue, pExt

    Set pEnv.OutWorkspace = pWStemp

    'Antes de nada, obtenemos las superficies de cada uno de las features para después
    'calcular los porcentajes de cobertura
    Dim superficies()
    ReDim superficies(UBound(capas))

    Call ObtenerSuperficies1(pinraster2, superficies)

    'Bindamos los rasters y las variables
    Call palgbop.BindRaster(pinraster1, "cober")
    Call palgbop.BindRaster(pinraster2, "muni")

    Dim out As IRaster
    Dim out2 As IRaster
    Dim out3 As IRaster
    Dim out4 As IRaster
    Dim out5 As IRaster
```

```
'out-> ponemos a 1 los puntos donde la potencia sea mayor que la sens y a cero el
'resto
'out2-> donde out no está definida (NO DATA) ponemos un 0
'out3->ponemos a 1 todos los valores del raster del territorio total
'out4->multiplicamos out2 y out3 (cobertura->1, no cobertura->0)
'out5->multiplicamos out2 y muni (territorio con cobertura)

On Error GoTo Error

Set out = palgbop.Execute("con( [cober] >= " & potencia & ", 1, 0)")

Call palgbop.BindRaster(out, "out")
Set out2 = palgbop.Execute("con (isnull([out]), 0, [out])")

Set out3 = palgbop.Execute("con([muni] <> 1, 1, [muni])")

Call palgbop.BindRaster(out2, "out2")
Call palgbop.BindRaster(out3, "out3")

Set out4 = palgbop.Execute("[out2] * [out3]")

Set out5 = palgbop.Execute("[out2] * [muni]")

Call palgbop.BindRaster(out4, "out4")
Call palgbop.BindRaster(out5, "out5")

'raster y gráfica con aquellos lugares y porcentajes donde hay cobertura
Dim prasbc2 As IRasterBandCollection
Set prasbc2 = out4

Dim prasband2 As IRasterBand
Set prasband2 = prasbc2.item(0)

Dim layer2 As IRasterLayer
Set layer2 = New RasterLayer

layer2.CreateFromRaster out4

Dim n_pixels()
ReDim n_pixels(2)
Dim suma_total

Call ObtenerSuperficies2(out4, n_pixels, suma_total)

Dim ptable2 As ITable
Dim pdatagraph2 As IDataGraph
Dim pdatagraphprop2 As IDataGraphProperties
Dim pgraphwindow2 As IDataGraphWindow
Dim pdatagraphs2 As IDataGraphs
Dim pdatagraphpie2 As IDataGraphPieProperties

Set ptable2 = layer2

Set pdatagraph2 = New DataGraph

Set pdatagraph2.table = ptable2
pdatagraph2.FieldSet1 = "count"

pdatagraph2.name = "Porcentaje > Umbral"
Set pdatagraphprop2 = pdatagraph2
pdatagraphprop2.GraphType = esriDataGraphTypePie
pdatagraphprop2.GraphSubtype = esriDataGraphSubtypePie3DSimple
pdatagraphprop2.title = "Porcentaje"

pdatagraphprop2.ShowLegend = False

Set pdatagraphpie2 = pdatagraphprop2
pdatagraphpie2.LabelSlicesUsing = esriDataGraphPieLabelSliceUsingPercent
```

```
Set pgraphwindow2 = New DataGraphWindow
Set pgraphwindow2.DataGraph = pdatagraph2
Set pgraphwindow2.Application = Application

Set pdatagraphs2 = pmxdoc
pdatagraphs2.Add pdatagraph2

Dim prasDS2 As IRasterDataset
Set prasDS2 = prasband2.RasterDataset

Dim pTemp2 As ITemporaryDataset
Set pTemp2 = prasDS2

Dim pWS5 As IRasterWorkspace
Dim pWSF5 As IWorkspaceFactory
Set pWSF5 = New RasterWorkspaceFactory
Set pWS5 = pWSF5.OpenFromFile(rutas.porcentajes + nombre, 0)

pTemp2.MakePermanentAs nombre, pWS5, "GRID"

pdatagraph2.ExportToFile rutas.porcentajes + nombre + "\graph2.bmp"

pdatagraphs2.Remove pdatagraph2

Set ptable2 = Nothing
Set layer2 = Nothing

Dim prlayer2 As IRasterLayer
Set prlayer2 = New RasterLayer

prlayer2.CreateFromRaster prasDS2.CreateDefaultRaster

Dim pcolor As IRgbColor
Set pcolor = New RgbColor

Dim pUvren As IRasterUniqueValueRenderer
Set pUvren = New RasterUniqueValueRenderer

Dim pren As IRasterRenderer
Set pren = pUvren

Set pren.raster = prlayer2.raster
pren.Update

Dim psym As IColorSymbol

pUvren.HeadingCount = 2

Dim j As Integer

For j = 0 To 1   'sólo hay dos posibilidades: 1->cobertura, 0->no cobertura

     If j = 0 Then

          pUvren.ClassCount(j) = 1
          pUvren.AddValue j, 0, j
          pUvren.Label(j, 0) = "NO (" + CStr(Round(n_pixels(0) * 100 / suma_total,
          2)) + " %)"
          pcolor.Red = 0
          pcolor.Green = 0
          pcolor.Blue = 255
          Set psym = New ColorSymbol
          psym.Color = pcolor
          pUvren.Symbol(j, 0) = psym

     Else
```

```
            pUvren.ClassCount(j) = 1
            pUvren.AddValue j, 0, j
            pUvren.Label(j, 0) = "SI (" + CStr(Round(n_pixels(1) * 100 / suma_total,
            2)) + " %)"
            pcolor.Red = 0
            pcolor.Green = 255
            pcolor.Blue = 0
            Set psym = New ColorSymbol
            psym.Color = pcolor
            pUvren.Symbol(j, 0) = psym

      End If

Next

pren.Update
Set prlayer2.Renderer = pren

prlayer2.name = nombre + "(%)"
pmap.AddLayer prlayer2

Set praster2 = Nothing

palgbop.UnbindRaster ("out")
palgbop.UnbindRaster ("out2")
palgbop.UnbindRaster ("out3")
palgbop.UnbindRaster ("out4")

Set out = Nothing
Set out2 = Nothing
Set out3 = Nothing
Set out4 = Nothing

If tipo = 1 Then  'Si es una cobertura, simplemente creamos el excel con lo ya
'calculado

      'Creamos un archivo excel con los datos
      Call CrearExcel(tipo, nombre, c_r, lista, c_v, Format(potencia), campo, capas,
      out5, superficies)

ElseIf tipo = 2 Then  'Si es un sistema, calculamos el porcentaje de los mejores
'servidores

      Dim bestserver As New FileSystemObject

      If Not bestserver.FileExists(rutas.sistemas + lista + "bestp.aux") Then

            MsgBox "Para más información, realizar el cálculo del mejor servidor"
            'Creamos el archivo Excell
            Call CrearExcel(1, nombre, c_r, lista, c_v, Format(potencia), campo,
            capas, out5, superficies)
            Exit Sub

      End If

      Dim rdsbest As IRasterDataset
      Set rdsbest = pWS2.OpenRasterDataset(lista + "bestp")

      Dim pinraster3 As IRaster
      Set pinraster3 = rdsbest.CreateDefaultRaster

      'Bindamos los rasters y las variables
      Call palgbop.BindRaster(pinraster1, "sis")
      Call palgbop.BindRaster(pinraster3, "best")

      Dim s_out As IRaster
      Dim s_out1 As IRaster
      Dim s_out2 As IRaster
```

```
Dim s_out3 As IRaster
Dim s_out4 As IRaster

's_out -> ponemos a 1 donde se supera la sensibilidad y NO DATA donde no
's_out1 -> multiplicamos out por el mejor servidor
's_out2 -> donde s_out1 sea null lo ponemos a 9998 (valor alto aleatorio)
's_out3 -> ponemos todos los valores de muni a 1
's_out4 -> multiplicamos s_out2 por s_out3

Set s_out = palgbop.Execute("con( [sis] >= " & potencia & ", 1,
setnull([sis]))")
Call palgbop.BindRaster(s_out, "s_out")

Set s_out1 = palgbop.Execute("[s_out] * [best]")
Call palgbop.BindRaster(s_out1, "s_out1")

Set s_out2 = palgbop.Execute("con (isnull([s_out1]), 9998, [s_out1])")
Call palgbop.BindRaster(s_out2, "s_out2")

Set s_out3 = palgbop.Execute("con([muni] <> 1, 1, [muni])")
Call palgbop.BindRaster(s_out3, "s_out3")

Set s_out4 = palgbop.Execute("[s_out2] * [s_out3]")
Call palgbop.BindRaster(s_out4, "s_out4")

Call palgbop.UnbindRaster("s_out")
Call palgbop.UnbindRaster("s_out1")
Call palgbop.UnbindRaster("s_out3")

Set s_out = Nothing
Set s_out1 = Nothing
Set s_out3 = Nothing

Dim s_out5 As IRaster
Dim s_out6 As IRaster

Dim muns() As IRaster
ReDim muns(UBound(capas) + 1)

For i = 0 To UBound(capas)

     Set s_out5 = palgbop.Execute("con( [muni] <> " + Format(i + 1) + ",
     setnull([muni]), 1)")
     Call palgbop.BindRaster(s_out5, "s_out5")

     Set s_out6 = palgbop.Execute("[s_out2] * [s_out5]")
     Call palgbop.BindRaster(s_out6, "s_out6")

     Set muns(i) = s_out6

     Call palgbop.UnbindRaster("s_out5")
     Call palgbop.UnbindRaster("s_out6")

     Set s_out5 = Nothing
     Set s_out6 = Nothing

Next

Call palgbop.UnbindRaster("s_out2")
Set s_out2 = Nothing

Set muns(UBound(capas) + 1) = s_out4

Dim prasbc As IRasterBandCollection
Set prasbc = s_out4

Dim prasband As IRasterBand
Set prasband = prasbc.item(0)
```

```
Dim layer As IRasterLayer
Set layer = New RasterLayer

layer.CreateFromRaster s_out4

Dim pTable As ITable
Dim pdatagraph As IDataGraph
Dim pdatagraphprop As IDataGraphProperties
Dim pgraphwindow As IDataGraphWindow
Dim pdatagraphs As IDataGraphs
Dim pdatagraphpie As IDataGraphPieProperties

Dim pRasDS As IRasterDataset
Set pRasDS = prasband.RasterDataset

Dim pTemp As ITemporaryDataset
Set pTemp = pRasDS

pTemp.MakePermanentAs nombre + "best", pWS5, "GRID"

Set pTable = Nothing
Set layer = Nothing
Set palgbop = Nothing

Dim path2 As String
path2 = rutas.sistemas + "ficheros\"
Dim pPropSet2 As IPropertySet
Set pPropSet2 = New PropertySet
pPropSet2.SetProperty "Database", path2

Dim pAccessFact2 As IWorkspaceFactory
Set pAccessFact2 = New ShapefileWorkspaceFactory

Dim paccessws2 As IWorkspace
Set paccessws2 = pAccessFact2.Open(pPropSet2, 0)

Dim pAccessFeatWS2 As IFeatureWorkspace
Set pAccessFeatWS2 = paccessws2

Dim pfeatcls2 As IFeatureClass
Set pfeatcls2 = pAccessFeatWS2.OpenTable(lista)

Dim feat As IFeature

Dim qf As IQueryFilter
Set qf = New QueryFilter

Dim num As Long
Dim cob() As String
num = pfeatcls2.FeatureCount(qf)
ReDim cob(num)

Dim z As Integer

For z = 0 To num - 1

     Set feat = New feature
     Set feat = pfeatcls2.GetFeature(z)
     cob(z) = feat.Value(4)

Next

ReDim n_pixels(num)
suma_total = 0

Call ObtenerSuperficies2(s_out4, n_pixels, suma_total)
```

```
Dim path3 As String
path3 = rutas.porcentajes + nombre
Dim pPropSet3 As IPropertySet
Set pPropSet3 = New PropertySet
pPropSet3.SetProperty "Database", path3

Dim paccess As IWorkspace
Set paccess = pAccessFact2.Open(pPropSet3, 0)

Dim pAccessFeatW2 As IFeatureWorkspace
Set pAccessFeatW2 = paccess

Dim pFlds As IFields
Set pFlds = New esriCore.Fields

Dim pFldsEdit As IFieldsEdit
Set pFldsEdit = pFlds

Dim pFld As IField
Set pFld = New esriCore.field

Dim pfldedit As IFieldEdit
Set pfldedit = pFld

With pfldedit
     .Type = 6
     .IsNullable = False
     .name = "FID"
End With
pFldsEdit.AddField pfldedit

Set pfldedit = New esriCore.field

With pfldedit
     .length = 30
     .name = "COBERTURA"
     .Type = esriFieldTypeString
End With
pFldsEdit.AddField pfldedit

Set pfldedit = New esriCore.field

With pfldedit
     .length = 30
     .name = "VALOR"
     .Type = esriFieldTypeInteger
End With
pFldsEdit.AddField pfldedit

Dim UID As New esriCore.UID
UID.Generate

Dim EXTCLSID As New esriCore.UID
EXTCLSID.Generate

Dim ptab As ITable
Set ptab = pAccessFeatW2.CreateTable(nombre + "best", pFlds, Nothing, Nothing,
"")

Dim fila As IRow

For z = 0 To num - 1

     Set fila = New row
     Set fila = ptab.CreateRow
     fila.Value(1) = cob(z)
     fila.Value(2) = z
     fila.Store
```

```
            Next

            'Cargamos la capa en pantalla
            Dim pRLayer As IRasterLayer
            Set pRLayer = New RasterLayer
            pRLayer.CreateFromRaster pRasDS.CreateDefaultRaster

            Dim pcolor2 As IRgbColor
            Set pcolor2 = New RgbColor

            Dim pUvren2 As IRasterUniqueValueRenderer
            Set pUvren2 = New RasterUniqueValueRenderer

            Dim pren2 As IRasterRenderer
            Set pren2 = pUvren2

            Set pren2.raster = pRLayer.raster
            pren2.Update

            Dim psym2 As IColorSymbol

            pUvren2.HeadingCount = num + 1

            For j = 0 To num - 1

                 pUvren2.ClassCount(j) = 1
                 pUvren2.AddValue j, 0, j
                 pUvren2.Label(j, 0) = cob(j) + " (" + CStr(Round(n_pixels(j) * 100 /
                 suma_total, 2)) + " %)"
                 pcolor2.Red = Int(255 * Rnd)
                 pcolor2.Green = Int(255 * Rnd)
                 pcolor2.Blue = Int(255 * Rnd)
                 Set psym2 = New ColorSymbol
                 psym2.Color = pcolor2
                 pUvren2.Symbol(j, 0) = psym2

            Next

            pUvren2.ClassCount(num) = 1
            pUvren2.AddValue num, 0, 9998
            pUvren2.Label(num, 0) = "Sin Cobertura" + " (" + CStr(Round(n_pixels(j) * 100 /
            suma_total, 2)) + " %)"
            pcolor2.Red = 0
            pcolor2.Green = 0
            pcolor2.Blue = 0
            Set psym2 = New ColorSymbol
            psym2.Color = pcolor2
            pUvren2.Symbol(num, 0) = psym2

            pren2.Update
            Set pRLayer.Renderer = pren2

            pRLayer.name = nombre + "best (%)"
            pmap.AddLayer pRLayer

            'Creamos el archivo Excel
            Call CrearExcel(2, nombre, c_r, lista, c_v, Format(potencia), campo, capas,
            out5, superficies, muns)

        End If

        borra.temporal

        Exit Sub

Error:
```

```
        If Err.Number <> 0 Then

             Msg = "Error # " & Str(Err.Number) & " was generated by " _
             & Err.Source & Chr(13) & Err.Description
             MsgBox Msg, , "Error", Err.HelpFile, Err.HelpContext

        End If

End Sub
```

```
'Obtiene las superficies de cada una de las features que forman el raster del territorio
total
        'Parametros de entrada:
             'territorio: raster del territorio total seleccionado
             'superficies(): array donde se almacenaran las superficies de cada feature

Public Sub ObtenerSuperficies1(territorio As IRaster, ByRef superficies() As Variant)

        Dim territorioBC As IRasterBandCollection
        Set territorioBC = territorio

        Dim territorioB As IRasterBand
        Set territorioB = territorioBC.item(0)

        Dim tabla As ITable
        Dim ExistTable As Boolean
        territorioB.HasTable ExistTable
        If ExistTable = False Then Exit Sub
        Set tabla = territorioB.AttributeTable

        For i = 0 To UBound(superficies)

             superficies(i) = tabla.GetRow(i).Value(2)

        Next

End Sub
```

```
'Obtiene la superficie del raster del territorio total mediante la suma de las
'superficies de cada feature que lo forman
        'Parámetros de entrada:
             'territorio: raster del territorio total seleccionado
             'superficies(): array donde se almacenaran las superficies de cada feature
             'suma: superficie del territorio total (suma de todas las superficies)

Public Sub ObtenerSuperficies2(territorio As Variant, ByRef superficies() As Variant,
ByRef suma As Variant)

        Dim territorioBC As IRasterBandCollection
        Set territorioBC = territorio

        Dim territorioB As IRasterBand
        Set territorioB = territorioBC.item(0)
        suma = 0
        Dim tabla As ITable

        Set tabla = territorioB.AttributeTable
        If (UBound(superficies) = 2) And (tabla.GetRow(0).Value(1)) = 1 Then
```

```
            superficies(0) = 0
            superficies(1) = tabla.GetRow(0).Value(2)
            suma = superficies(1)

        Else

            For i = 0 To UBound(superficies)

                superficies(i) = tabla.GetRow(i).Value(2)
                suma = suma + superficies(i)

            Next

        End If

End Sub
```

```
'Crea el informe de resultados en formato Excel
        'Parametros de entrada:
            'tipo: entero que indica que tipo de umbral estamos calculando:
                '1: cobertura
                '2: sistema
            'nombre: nombre del mapa de cálculo de umbral
            'c_r: tipo de capa raster (sistema o cobertura)
            'lista: nombre de la capa raster
            'c_v: capa vectorial (una de las capas disponibles)
            'umbral: umbral de cobertura para el mapa
            'campo: nombre del campo de caracteristicas deseadas
            'capas(): Array con el nombre de las caracteristicas
            'seleccionadas
            'territorio: Raster con el porcentaje de cobertura de cada una de
            'las caracteristicas seleccionadas
            'superficies(): Array con las superficiess de cada uno de las
            'caracteristicas seleccionadas
            'servs(): (Optional) Array de Rasters con el porcentaje de
            'cobertura de los mejores servidores calculado sobre cada una de
            'las caracteristicas
            'seleccionadas (solamente necesario en el caso de sistemas) y
            'sobre territorio total

Public Sub CrearExcel(tipo As Integer, nombre As String, c_r As String, lista
As String, c_v As String, umbral As String, campo As String, capas() As
String, territorio As IRaster, superficies() As Variant, Optional servs As
Variant)

        Dim valor
        'Obtenemos la tabla del raster calculado
        Dim territorioBC As IRasterBandCollection
        Set territorioBC = territorio

        Dim territorioB As IRasterBand
        Set territorioB = territorioBC.item(0)

        Dim tabla As ITable
        Dim ExistTable As Boolean
        territorioB.HasTable ExistTable

        Dim pMathOp As IMathOp
        Set pMathOp = New RasterMathOps

        Dim pInputDataset As IGeoDataset
        Dim pOutputDataset As IGeoDataset
```

```
'Si el raster no tiene tabla de atributos (por ser FLOAT), lo pasamos
'a ENTERO y utilizamos esa tabla de atributos

If ExistTable = False Then

     Set pInputDataset = territorio
     Set pOutputDataset = pMathOp.Int(pInputDataset)
     Set territorioBC = pOutputDataset
     Set territorioB = territorioBC.item(0)

End If

Set tabla = territorioB.AttributeTable

Dim objExcel As Object
Dim objLibro As Object

On Error Resume Next

Set objExcel = GetObject(, "Excel.Application")
If Err.Number = 429 Then

     Err.Clear
     Set objExcel = CreateObject("Excel.Application")

End If

'Creamos un nuevo libro
If tipo = 2 Then    'Sistema con mejores servidores

     'Obtenemos los emplazamientos que forman el sistema
     Dim numero As Integer
     Dim coberturas() As String
     Dim path As String
     path = rutas.sistemas + "ficheros\"

     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 pAccessFeatWS As IFeatureWorkspace
     Set pAccessFeatWS = paccessws

     Dim pfeatcls As IFeatureClass
     Set pfeatcls = pAccessFeatWS.OpenFeatureClass(lista)
     numero = pfeatcls.GetFeature(0).Value(3)  'numero de
     'emplazamientos del sistema

     ReDim coberturas(numero - 1)

     For i = 0 To numero - 1

          'emplamientos que forman el sistema
          coberturas(i) = pfeatcls.GetFeature(i).Value(4)

     Next

     objExcel.SheetsInNewWorkbook = numero + 1 'Creamos el nuevo libro

Else   'Cobertura
```

```
      objExcel.SheetsInNewWorkbook = 1

End If

'Editamos la primera hoja del archivo
Set objLibro = objExcel.Workbooks.Add

objLibro.Worksheets(1).name = lista
objLibro.Worksheets(1).PageSetup.LeftFooter = "&D   RAGIS"
objLibro.Worksheets(1).PageSetup.CenterFooter = "&F"
objLibro.Worksheets(1).PageSetup.RightFooter = "Página &P de &N"

objLibro.Worksheets(1).Range("C2").Value = "% DE COBERTURA SOBRE
UMBRAL"
objLibro.Worksheets(1).Range("C2").Font.Bold = True
objLibro.Worksheets(1).Range("C2").Font.Size = 11
With objLibro.Worksheets(1).Range("C2:D2")
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .BorderAround (xlContinuous)
End With

objLibro.Worksheets(1).Range("B5").Value = "Datos"
objLibro.Worksheets(1).Range("B5").Font.Bold = True
With objLibro.Worksheets(1).Range("B5:C5")
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .Interior.Color = RGB(223, 223, 223)
      .BorderAround (xlContinuous)
End With

objLibro.Worksheets(1).Range("B7").Value = "Mapa"
objLibro.Worksheets(1).Range("B8").Value = "Capa Vectorial"
objLibro.Worksheets(1).Range("B9").Value = "Tipo capa raster"
objLibro.Worksheets(1).Range("B10").Value = "Capa Raster"
objLibro.Worksheets(1).Range("B11").Value = "Umbral"

With objLibro.Worksheets(1).Range("B7:B11")
      .Interior.Color = RGB(223, 223, 223)
      .Font.Bold = True
End With

objLibro.Worksheets(1).Range("C7").Value = nombre
objLibro.Worksheets(1).Range("C8").Value = c_v
objLibro.Worksheets(1).Range("C9").Value = c_r
objLibro.Worksheets(1).Range("C10").Value = lista
objLibro.Worksheets(1).Range("C11").Value = umbral

With objLibro.Worksheets(1).Range("C7:C11")
      .HorizontalAlignment = xlHAlignRight
      .Columns.AutoFit
End With

objLibro.Worksheets(1).Range("B15").Value = "Cobertura"
objLibro.Worksheets(1).Range("B15").Font.Bold = True
With objLibro.Worksheets(1).Range("B15:D15")
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .Interior.Color = RGB(223, 223, 223)
      .BorderAround (xlContinuous)
End With

objLibro.Worksheets(1).Range("C17").Value = c_v
objLibro.Worksheets(1).Range("C17").Font.Bold = True
With objLibro.Worksheets(1).Range("C17:D17")
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .Interior.Color = RGB(223, 223, 223)
```

```
End With

objLibro.Worksheets(1).Range("B18").Value = campo
objLibro.Worksheets(1).Range("C18").Value = "Si"
objLibro.Worksheets(1).Range("D18").Value = "No"
With objLibro.Worksheets(1).Range("B18:D18")
     .Interior.Color = RGB(223, 223, 223)
     .Font.Bold = True
     .HorizontalAlignment = xlHAlignCenter
End With

Dim cobertura     'cobertura del territorio total
cobertura = 0
Dim k As Integer
k = 1

For i = 0 To UBound(capas)

     If tabla.GetRow(k).Value(1) = i + 1 Then

          cobertura = cobertura + tabla.GetRow(k).Value(2)

          objLibro.Worksheets(1).Range("B" + Format(19 +
          i)).HorizontalAlignment = xlHAlignCenter

          'Obtenemos los valores correspondientes a las coberturas
          valor = ((tabla.GetRow(k).Value(2)) / superficies(i))
          k = k + 1

     Else

          valor = 0

     End If

     objLibro.Worksheets(1).Range("B" + Format(19 + i)).Value =
     capas(i)
     objLibro.Worksheets(1).Range("B" + Format(19 +
     i)).HorizontalAlignment = xlHAlignCenter
     objLibro.Worksheets(1).Range("C" + Format(19 + i)).Value = valor
     objLibro.Worksheets(1).Range("C" + Format(19 + i)).NumberFormat =
     "0.0%"
     objLibro.Worksheets(1).Range("D" + Format(19 + i)).Value = 1 -
     valor
     objLibro.Worksheets(1).Range("D" + Format(19 + i)).NumberFormat =
     "0.0%"
     'Centramos el texto en la celda
     objLibro.Worksheets(1).Range("C" + Format(19 +
     i)).HorizontalAlignment = xlHAlignCenter
     objLibro.Worksheets(1).Range("D" + Format(19 +
     i)).HorizontalAlignment = xlHAlignCenter

Next

objLibro.Worksheets(1).Range("B" + Format(20 + UBound(capas))).Value =
"Territorio Total"
objLibro.Worksheets(1).Range("B" + Format(20 +
UBound(capas))).Font.Bold = True
With objLibro.Worksheets(1).Range("B" + Format(20 + UBound(capas)) +
":D" + Format(20 + UBound(capas)))
     .Merge
     .HorizontalAlignment = xlHAlignCenter
     .Interior.Color = RGB(223, 223, 223)
End With

'Obtenemos la superficies del territorio total
Dim superficies_total
superficies_total = 0
```

```
    For i = 0 To UBound(capas) + 1
         superficies_total = superficies_total + tabla.GetRow(i).Value(2)
    Next

    'Escribimos los datos del territorio total
    objLibro.Worksheets(1).Range("C" + Format(20 + UBound(capas) + 1)) =
    cobertura / superficies_total
    objLibro.Worksheets(1).Range("C" + Format(20 + UBound(capas) +
    1)).NumberFormat = "0.0%"
    objLibro.Worksheets(1).Range("C" + Format(20 + UBound(capas) +
    1)).HorizontalAlignment = xlHAlignCenter
    objLibro.Worksheets(1).Range("C" + Format(20 + UBound(capas) +
    1)).Font.Bold = True

    objLibro.Worksheets(1).Range("D" + Format(20 + UBound(capas) + 1)) =
    tabla.GetRow(0).Value(2) / superficies_total
    objLibro.Worksheets(1).Range("D" + Format(20 + UBound(capas) +
    1)).NumberFormat = "0.0%"
    objLibro.Worksheets(1).Range("D" + Format(20 + UBound(capas) +
    1)).HorizontalAlignment = xlHAlignCenter
    objLibro.Worksheets(1).Range("D" + Format(20 + UBound(capas) +
    1)).Font.Bold = True

    'Centramos las columnas
    objLibro.Worksheets(1).Range("B7:" + "B" + Format(19 +
    UBound(capas))).Columns.AutoFit
    objLibro.Worksheets(1).Range("D7:" + "D" + Format(19 +
    UBound(capas))).ColumnWidth = objLibro.Worksheets(1).Range("C7:" + "C"
    + Format(19 + UBound(capas))).ColumnWidth


'************************* GRÁFICOS *************************

          '+++++++++++++++ GRÁFICO 1 +++++++++++++++++

    Dim grafico1 As ChartObject
    Dim cw As Integer
    Dim rh As Integer

    Dim izda
    Dim top
    Dim anchura
    Dim altura

    cw = objLibro.Worksheets(1).Columns(1).Width
    rh = objLibro.Worksheets(1).Rows(1).Height

    izda = cw * 7.5
    top = rh * 3
    anchura = cw * 6
    altura = rh * 18

    Set grafico1 = objLibro.Worksheets(1).ChartObjects.Add(izda, top,
    anchura, altura)

    'Tipo de gráfico
    grafico1.Chart.ChartType = xl3DPie

    'Añadimos las series al gráfico
    grafico1.Chart.SeriesCollection.Add , _
    Source:=objLibro.Worksheets(1).Range("C" + Format(20 + UBound(capas) +
    1) + ":" + "D" + Format(20 + UBound(capas) + 1)), _
    Rowcol:=xlRows, SeriesLabels:=False, _
    CategoryLabels:=False

    'Añadimos título y colores
```

```
With grafico1.Chart.SeriesCollection(1)
      .name = "Territorio total"
      .Points(1).Interior.Color = RGB(0, 255, 0)
      .Points(2).Interior.Color = RGB(255, 0, 0)
End With

With grafico1.Chart.ChartTitle
      .Caption = "% DE COBERTURA SOBRE EL TERRITORIO TOTAL"
      .Font.Size = 9
      .Font.Bold = True
      .Border.Weight = xlThin
End With

'Leyenda y mostramos etiquetas
grafico1.Chart.Axes(xlCategory).CategoryNames = Array("Si", "No")
grafico1.Chart.Elevation = 25

grafico1.Chart.SeriesCollection(1).ApplyDataLabels , _
ShowValue:=True

'+++++++++++++++++++++ GRAFICO 2 +++++++++++++++++++++

Dim grafico2 As ChartObject
Dim n_capas As Integer

n_capas = UBound(capas) + 1

Dim c As Integer

n_graf = 1

If n_capas > 10 Then n_graf = Fix(n_capas / 10)

If n_graf < n_capas / 10 Then n_graf = n_graf + 1

c = 19

For i = 0 To n_graf - 1

      If (i + 1) Mod 2 <> 0 Then 'El grafico ocupa una posición PAR
      '(arriba)

            If i = 0 Then

                  Set grafico2 =
                  objLibro.Worksheets(1).ChartObjects.Add(izda + (i + 1)
                  * (anchura) + cw, top, anchura, altura)

            Else 'El grafico ocupa una posición IMPAR (abajo)

                  Set grafico2 =
                  objLibro.Worksheets(1).ChartObjects.Add(izda + (i) *
                  (anchura) + (cw * 2), top, anchura, altura)

            End If

      Else

            If i = 1 Then

                  Set grafico2 =
                  objLibro.Worksheets(1).ChartObjects.Add(izda + (i) *
                  (anchura) + cw, top * 9, anchura, altura)

            Else

                  Set grafico2 =
                  objLibro.Worksheets(1).ChartObjects.Add(izda + (i - 1)
```

```
                    * (anchura) + (cw * 2), top * 9, anchura, altura)

                End If

            End If

            'Tipo de gráfico: Columnas
            grafico2.Chart.ChartType = xlColumnStacked

            'Añadimos los datos de las series. En este caso, creamos un
            'NUEVO gráfico por cada 10 SERIES
            If i = n_graf - 1 And (n_capas Mod 10) <> 0 Then

                grafico2.Chart.SeriesCollection.Add , _
                Source:=objLibro.Worksheets(1).Range("B" + Format(c) + ":" +
                "D" + Format(c + (n_capas Mod 10) - 1)), _
                Rowcol:=xlColumns, SeriesLabels:=False, _
                CategoryLabels:=True

            Else

                grafico2.Chart.SeriesCollection.Add , _
                Source:=objLibro.Worksheets(1).Range("B" + Format(c) + ":" +
                "D" + Format(c + 9)), _
                Rowcol:=xlColumns, SeriesLabels:=False, _
                CategoryLabels:=True

            End If

            c = c + 10

            'Establecemos los colores
            grafico2.Chart.SeriesCollection(1).Interior.Color = RGB(0, 255,
            0)

            grafico2.Chart.SeriesCollection(2).Interior.Color = RGB(255, 0,
            0)

            'Título
            grafico2.Chart.HasTitle = True

            With grafico2.Chart.ChartTitle
                .Caption = "% DE COBERTURA EN CAPA VECTORIAL"
                .Font.Size = 9
                .Font.Bold = True
                .Border.Weight = xlThin
            End With

            'Mostramos los valores de cada serie
            grafico2.Chart.SeriesCollection(1).ApplyDataLabels , _
            ShowValue:=True

            grafico2.Chart.SeriesCollection(2).ApplyDataLabels , _
            ShowValue:=True

            'Quitamos el eje y
            grafico2.Chart.HasAxis(2) = False

            'Leyenda
            grafico2.Chart.SeriesCollection(1).name = "Si"
            grafico2.Chart.SeriesCollection(2).name = "No"

            Set grafico2 = Nothing

        Next

    '*****************************************************
```

```
If tipo = 2 Then  'Sistema con mejores servidores (continuamos
'editando el excell)

     Dim valor2
     Dim municipioBC As IRasterBandCollection
     Dim municipioB As IRasterBand
     Dim tabla_mun As ITable

     For i = 0 To numero - 1 'Vemos la cobertura de cada uno de los
     'emplazamientos

          objLibro.Worksheets(i + 2).name = coberturas(i)
          objLibro.Worksheets(i + 2).PageSetup.LeftFooter = "&D
          RAGIS"
          objLibro.Worksheets(i + 2).PageSetup.CenterFooter = "&F"
          objLibro.Worksheets(i + 2).PageSetup.RightFooter = "Página
          &P de &N"

          objLibro.Worksheets(i + 2).Range("B2").Value = "% DE
          COBERTURA SOBRE UMBRAL: MEJORES SERVIDORES"
          objLibro.Worksheets(i + 2).Range("B2").Font.Bold = True
          objLibro.Worksheets(i + 2).Range("B2").Font.Size = 11
          With objLibro.Worksheets(i + 2).Range("B2:E2")
               .Merge
               .HorizontalAlignment = xlHAlignCenter
               .BorderAround (xlContinuous)
          End With

          objLibro.Worksheets(i + 2).Range("B5").Value = "Datos"
          objLibro.Worksheets(i + 2).Range("B5").Font.Bold = True
          With objLibro.Worksheets(i + 2).Range("B5:C5")
               .Merge
               .HorizontalAlignment = xlHAlignCenter
               .Interior.Color = RGB(223, 223, 223)
               .BorderAround (xlContinuous)
          End With

          objLibro.Worksheets(i + 2).Range("B7").Value = "Mapa"
          objLibro.Worksheets(i + 2).Range("B8").Value = "Capa
          Vectorial"
          objLibro.Worksheets(i + 2).Range("B9").Value = "Tipo capa
          raster"
          objLibro.Worksheets(i + 2).Range("B10").Value =
          objLibro.Worksheets(i + 2).Range("B11").Value = "Umbral"

          With objLibro.Worksheets(i + 2).Range("B7:B11")
               .Interior.Color = RGB(223, 223, 223)
               .Font.Bold = True
          End With

          objLibro.Worksheets(i + 2).Range("C7").Value = nombre
          objLibro.Worksheets(i + 2).Range("C8").Value = c_v
          objLibro.Worksheets(i + 2).Range("C9").Value = "Coberturas"
          objLibro.Worksheets(i + 2).Range("C10").Value =
          pfeatcls.GetFeature(i).Value(4)
          objLibro.Worksheets(i + 2).Range("C11").Value = umbral
          With objLibro.Worksheets(i + 2).Range("C7:C11")
               .HorizontalAlignment = xlHAlignRight
               .Columns.AutoFit
          End With

          objLibro.Worksheets(i + 2).Range("B15").Value = "Cobertura"
          objLibro.Worksheets(i + 2).Range("B15").Font.Bold = True
          With objLibro.Worksheets(i + 2).Range("B15:D15")
               .Merge
               .HorizontalAlignment = xlHAlignCenter
               .Interior.Color = RGB(223, 223, 223)
```

```
      .BorderAround (xlContinuous)
End With

objLibro.Worksheets(i + 2).Range("C17").Value = c_v
objLibro.Worksheets(i + 2).Range("C17").Font.Bold = True
With objLibro.Worksheets(i + 2).Range("C17:D17")
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .Interior.Color = RGB(223, 223, 223)
End With

objLibro.Worksheets(i + 2).Range("B18").Value = campo
objLibro.Worksheets(i + 2).Range("C18").Value = "Si"
objLibro.Worksheets(i + 2).Range("D18").Value = "No"
With objLibro.Worksheets(i + 2).Range("B18:D18")
      .Interior.Color = RGB(223, 223, 223)
      .Font.Bold = True
      .HorizontalAlignment = xlHAlignCenter
End With

objLibro.Worksheets(i + 2).Range("B" + Format(20 +
UBound(capas))).Value = "Territorio Total"
objLibro.Worksheets(i + 2).Range("B" + Format(20 +
UBound(capas))).Font.Bold = True
With objLibro.Worksheets(i + 2).Range("B" + Format(20 +
UBound(capas)) + ":D" + Format(20 + UBound(capas)))
      .Merge
      .HorizontalAlignment = xlHAlignCenter
      .Interior.Color = RGB(223, 223, 223)
End With

'Editamos cada una de las hojas que forman el archivo
For j = 0 To UBound(capas)

      Set municipioBC = servs(j)
      Set municipioB = municipioBC.item(0)

      municipioB.HasTable ExistTable

      'Si el raster no tiene tabla de atributos (por ser
      'FLOAT), lo pasamos a ENTERO y utilizamos esa tabla de
      'atributos
      If ExistTable = False Then

            Set pInputDataset = servs(j)
            Set pOutputDataset = pMathOp.Int(pInputDataset)
            Set municipioBC = pOutputDataset
            Set municipioB = municipioBC.item(0)

      End If

      Set tabla_mun = municipioB.AttributeTable

      valor2 = 0

      For k = 0 To UBound(coberturas)

            If tabla_mun.GetRow(k).Value(1) = i Then: valor2
            = ((tabla_mun.GetRow(k).Value(2)) /
            superficies(j))

      Next

      objLibro.Worksheets(i + 2).Range("B" + Format(19 +
      j)).Value = capas(j)
      objLibro.Worksheets(i + 2).Range("B" + Format(19 +
      j)).HorizontalAlignment = xlHAlignCenter
      'Obtenemos los valores correspondientes a 'las
```

```
      coberturas
      objLibro.Worksheets(i + 2).Range("C" + Format(19 +
      j)).Value = valor2
      objLibro.Worksheets(i + 2).Range("C" + Format(19 +
      j)).NumberFormat = "0.0%"
      objLibro.Worksheets(i + 2).Range("D" + Format(19 +
      j)).Value = 1 - valor2
      objLibro.Worksheets(i + 2).Range("D" + Format(19 +
      j)).NumberFormat = "0.0%"
      'Centramos el texto en la celda
      objLibro.Worksheets(i + 2).Range("C" + Format(19 +
      j)).HorizontalAlignment = xlHAlignCenter
      objLibro.Worksheets(i + 2).Range("D" + Format(19 +
      j)).HorizontalAlignment = xlHAlignCenter

Next

'Escribimos los datos del territorio total
Set municipioBC = servs(UBound(capas) + 1)
Set municipioB = municipioBC.item(0)

municipioB.HasTable ExistTable

'Si el raster no tiene tabla de atributos (por 'ser FLOAT),
lo pasamos a ENTERO y utilizamos esa 'tabla de atributos
If ExistTable = False Then

      Set pInputDataset = servs(UBound(capas) + 1)
      Set pOutputDataset = pMathOp.Int(pInputDataset)
      Set municipioBC = pOutputDataset
      Set municipioB = municipioBC.item(0)

End If

Set tabla_mun = municipioB.AttributeTable

valor2 = 0

For k = 0 To UBound(coberturas)

      If tabla_mun.GetRow(k).Value(1) = i Then: valor2 =
      ((tabla_mun.GetRow(k).Value(2)) / superficies_total)

Next

objLibro.Worksheets(i + 2).Range("C" + Format(20 +
UBound(capas) + 1)).Value = valor2
objLibro.Worksheets(i + 2).Range("C" + Format(20 +
UBound(capas) + 1)).NumberFormat = "0.0%"
objLibro.Worksheets(i + 2).Range("C" + Format(20 +
UBound(capas) + 1)).HorizontalAlignment = xlHAlignCenter
objLibro.Worksheets(i + 2).Range("C" + Format(20 +
UBound(capas) + 1)).Font.Bold = True

objLibro.Worksheets(i + 2).Range("D" + Format(20 +
UBound(capas) + 1)).Value = 1 - valor2
objLibro.Worksheets(i + 2).Range("D" + Format(20 +
UBound(capas) + 1)).NumberFormat = "0.0%"
objLibro.Worksheets(i + 2).Range("D" + Format(20 +
UBound(capas) + 1)).HorizontalAlignment = xlHAlignCenter
objLibro.Worksheets(i + 2).Range("D" + Format(20 +
UBound(capas) + 1)).Font.Bold = True

'Centramos las columnas
objLibro.Worksheets(i + 2).Range("B7:" + "B" + Format(19 +
UBound(capas))).Columns.AutoFit
objLibro.Worksheets(i + 2).Range("D7:" + "D" + Format(19 +
UBound(capas))).ColumnWidth = objLibro.Worksheets(i +
```

```
                2).Range("C7:" + "C" + Format(19 +
                UBound(capas))).ColumnWidth

            Next

'************************ GRAFICOS ************************

            For j = 2 To objLibro.Worksheets.count

                c = 19

                For i = 0 To n_graf - 1

                    If (i + 1) Mod 2 <> 0 Then  'El grafico ocupa una
                    'posición PAR (arriba)

                        If i = 0 Then

                            Set grafico2 =
                            objLibro.Worksheets(j).ChartObjects.Add(izda
                            + (i + 1 - 1) * (anchura), top, anchura,
                            altura)

                        Else  'El gráfico ocupa una posición IMPAR
                        '(abajo)

                            Set grafico2 =
                            objLibro.Worksheets(j).ChartObjects.Add(izda
                            + (i - 1) * (anchura) + cw, top, anchura,
                            altura)

                        End If

                    Else

                        If i = 1 Then

                            Set grafico2 =
                            objLibro.Worksheets(j).ChartObjects.Add(izda
                            + (i - 1) * (anchura), top * 9, anchura,
                            altura)

                        Else

                            Set grafico2 =
                            objLibro.Worksheets(j).ChartObjects.Add(izda
                            + (i - 2) * (anchura) + cw, top * 9,
                            anchura, altura)

                        End If

                    End If

                    'Tipo de gráfico: Columnas
                    grafico2.Chart.ChartType = xlColumnStacked

                    If i = n_graf - 1 And (n_capas Mod 10) <> 0 Then

                            grafico2.Chart.SeriesCollection.Add , _
                            Source:=objLibro.Worksheets(j).Range("B" +
                            Format(c) + ":" + "D" + Format(c + (n_capas
                            Mod 10) - 1)), _
                            Rowcol:=xlColumns, SeriesLabels:=False, _
                            CategoryLabels:=True

                    Else
```

```
                    grafico2.Chart.SeriesCollection.Add , _
                    Source:=objLibro.Worksheets(j).Range("B" +
                    Format(c) + ":" + "D" + Format(c + 9)), _
                     Rowcol:=xlColumns, SeriesLabels:=False, _
                    CategoryLabels:=True

              End If

              c = c + 10

              'Color, título y etiquetas de valores

              grafico2.Chart.SeriesCollection(1).Interior.Color =
              RGB(0, 255, 0)
              grafico2.Chart.SeriesCollection(2).Interior.Color =
              RGB(255, 0, 0)
              grafico2.Chart.HasTitle = True

              With grafico2.Chart.ChartTitle
                   .Caption = "MEJOR SERVIDOR: " +
                   objLibro.Worksheets(j).name
                   .Font.Size = 9
                   .Font.Bold = True
                   .Border.Weight = xlThin
              End With


              grafico2.Chart.SeriesCollection(1).ApplyDataLabels ,
              ShowValue:=True

              grafico2.Chart.SeriesCollection(2).ApplyDataLabels ,
              ShowValue:=True

              'Quitamos el eje y
              grafico2.Chart.HasAxis(2) = False

              'Leyendas
              grafico2.Chart.SeriesCollection(1).name = "Si"
              grafico2.Chart.SeriesCollection(2).name = "No"

              Set grafico2 = Nothing

         Next

    Next

'+++++++++++++++++++ GRAFICO MEJORES SERVIDORES +++++++++++++++++++

    m = 0
    Dim t As Variant
    Dim grafico3 As ChartObject

    objLibro.Worksheets(1).Range("B" + Format(25 +
    UBound(capas))).Value = "Mejores Servidores: % de cobertura"
    objLibro.Worksheets(1).Range("B" + Format(25 +
    UBound(capas))).Font.Bold = True
    With objLibro.Worksheets(1).Range("B" + Format(25 +
    UBound(capas)) + ":" + "D" + Format(25 + UBound(capas)))
         .Merge
         .HorizontalAlignment = xlHAlignCenter
         .Interior.Color = RGB(223, 223, 223)
    End With

    For i = 0 To numero - 1

         objLibro.Worksheets(1).Range("B" + Format(27 + UBound(capas)
         + m)).Value = objLibro.Worksheets(i + 2).Range("C10").Value
```

```
                objLibro.Worksheets(1).Range("B" + Format(27 + UBound(capas)
                + m)).Interior.Color = RGB(223, 223, 223)

                objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas)
                + m)).Value = objLibro.Worksheets(i + 2).Range("C" +
                Format(20 + UBound(capas) + 1)).Value
                objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas)
                + m)).NumberFormat = "0.0%"
                objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas)
                + m)).HorizontalAlignment = xlHAlignCenter
                objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas)
                + m)).Font.Bold = True

                m = m + 1
                t = t + objLibro.Worksheets(i + 2).Range("C" + Format(20 +
                UBound(capas) + 1)).Value

            Next

            objLibro.Worksheets(1).Range("B" + Format(27 + UBound(capas) +
            m)).Value = "Sin cobertura"
            objLibro.Worksheets(1).Range("B" + Format(27 + UBound(capas) +
            m)).Interior.Color = RGB(223, 223, 223)
            objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas) +
            m)).Value = 1 - t
            objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas) +
            m)).NumberFormat = "0.0%"
            objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas) +
            m)).HorizontalAlignment = xlHAlignCenter
            objLibro.Worksheets(1).Range("C" + Format(27 + UBound(capas) +
            m)).Font.Bold = True

            Set grafico3 = objLibro.Worksheets(1).ChartObjects.Add(izda, top
            * 9, anchura, altura)

            grafico3.Chart.ChartType = xl3DPie

            grafico3.Chart.SeriesCollection.Add , _
            Source:=objLibro.Worksheets(1).Range("B" + Format(27 +
            UBound(capas)) + ":" + "C" + Format(27 + UBound(capas) + m)), _
            Rowcol:=xlColumns, SeriesLabels:=False, _
            CategoryLabels:=True

            grafico3.Chart.HasTitle = True

            With grafico3.Chart.ChartTitle
                .Caption = "% DE COBERTURA SOBRE TERRITORIO TOTAL: MEJORES
                SERVIDORES"
                .Font.Size = 8
                .Font.Bold = True
                .Border.Weight = xlThin
            End With

            grafico3.Chart.Elevation = 25
            grafico3.Chart.SeriesCollection(1).ApplyDataLabels , _
            ShowValue:=True

            Set grafico3 = Nothing

'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'*********************************************************************

        End If

        'Guardamos el libro
        objLibro.SaveAs rutas.porcentajes + nombre + "\" + nombre + ".xls"
        objLibro.Close     'Cerramos el libro
```

```
        objExcel.Quit
        Set objLibro = Nothing
        Set objExcel = Nothing

End Sub
```