



Trabajo Fin de Máster

**Estudio de coberturas radioeléctricas proporcionadas
por el sistema inalámbrico LoRa en Cartagena e
implementación de una red LoRa propia.**



AUTOR: María García
DIRECTORES: Leandro Juan Llácer
Jose María Molina García Pardo
Junio / 2020

Autora:	María García
E-mail del Autor:	maria101010@hotmail.es
Directores:	Leandro Juan Llácer Jose María Molina García Pardo
E-mails directores:	leandro.juan@upct.es josemaria.molina@upct.es
Título del TFG:	Estudio de coberturas radioeléctricas proporcionadas por el sistema inalámbrico LoRa en Cartagena e implementación de una red LoRa propia.
Resumen:	<p>En el municipio de Cartagena se están llevando a cabo por parte del ayuntamiento acciones encaminadas a convertir la ciudad en una Smart City. Una de ellas consiste en el despliegue de una red inalámbrica para recolectar la información recogida por unos sensores (temperatura, humedad, etc.) y enviarla a un centro de control para su gestión.</p> <p>Uno de estos sistemas es LoRa que consiste en estaciones base que cubren un área de la ciudad, una serie de sensores, los cuales se comunican con las estaciones base vía radio (banda de frecuencias de 868MHz). Seguidamente, estos datos son recogidos, para su posterior análisis, en una plataforma llamada LoRaWan que actúa como servidor.</p> <p>La planificación de estos sistemas de radiocomunicaciones se realiza mediante herramientas informáticas. Una de estas herramientas es RADIOGIS, desarrollada por el grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena (UPCT). RADIOGIS está basada en el Sistema de Información Geográfica (SIG) ArcGIS de ESRI e incorpora funcionalidades y modelos de propagación para el cálculo de cobertura radioeléctrica.</p> <p>En este trabajo se ha implementado un prototipo LoRa que permite realizar campañas de medida de este sistema. Se han llevado a cabo medidas de potencia recibida desde estaciones base reales desplegadas en la ciudad de Cartagena por el ayuntamiento. Estas medidas se han comparado con modelos de propagación implementados en la herramienta de cálculo de coberturas RADIOGIS.</p> <p>El trabajo se ha realizado en la Cátedra de Infraestructuras Municipales Ayuntamiento de Cartagena – UPCT en la línea L3. Implantación de sistemas de control de servicios públicos y valorización de herramientas de gestión municipales.</p>
Titulación:	Máster en Ingeniería en Sistemas de Telecomunicación.
Departamento:	Tecnologías de la Información y las Comunicaciones.
Fecha de Presentación:	Junio 2020.

Me gustaría dedicar este TFM a mi familia por brindarme la oportunidad de estudiar esta magnífica carrera.

Agradecimientos

En primer lugar, me gustaría agradecer a Leandro Juan Llácer y a Jose María Molina García Pardo por la confianza depositada en mí durante este proyecto. Gracias por vuestra valiosa ayuda. Definitivamente no sólo me habéis brindado las herramientas necesarias para la realización de este Trabajo Final de Máster, sino que también habéis hecho que me enamore aún más de la carrera que elegí estudiar.

En segundo lugar, agradecer a mis padres y hermano, por apoyarme en cada momento de mi vida y por creer incondicionalmente en mí. Por inculcarme los valores que me definen y hacerme consciente de que con esfuerzo todas las metas son alcanzables. ¡Sin vosotros nada de esto hubiera sido posible!

Agradecer a mis abuelos, por ser un ejemplo constante de lucha y superación. A mi abuelo Enrique, que, aunque no estés, siempre te llevo presente.

Gracias Juan Antonio, por llevar 7 años apoyándome y no haber dejado de hacerlo nunca. Por demostrar que la lejanía no es un obstáculo y hacerme incondicionalmente feliz a pesar de ello. Por enseñarme valores de los que hoy en día estoy orgullosa y por demostrarme que el amor verdadero existe.

A Águeda y Juan, por ese cariño que me proporcionáis. Por mirarme y hacerme sentir que puedo conseguir todo lo que me proponga. Por cuidarme y quererme como vuestra propia hija. ¡Gracias!

Por último, a vosotros, Marta y Pedro, por ser unos amigos maravillosos, por esa alegría y bondad de la que me habéis hecho partícipe durante un año entero ¡y lo que nos queda juntos! Paula, Mery, Jaime, Javi por ser lo más bonito que me llevo de Madrid. Gracias por hacer esta experiencia única. Y a mis amigos de toda la vida, porque parte de lo que soy es gracias a vosotros.

¡Gracias a todos!

Índice

Capítulo 1.	10
Introducción y Objetivos	10
1. Introducción y Objetivos	11
1.1. Fases	11
1.2. Contenido	14
1.3. Herramientas utilizadas	14
Capítulo 2.	17
Estado del arte	17
2.1. Introducción	18
2.2. LoRa	18
2.3. Entorno de trabajo LoRa	20
2.3.1. MTMDK-ST-MDOT	20
2.3.2. MTDOT-BOX	21
2.3.3. The Things Network	21
Capítulo 3.	23
Implementación de una red LoRa	23
3.1. Introducción	24
3.2. Configuración de MTDOT-BOX	24
3.3. Configuración del Gateway	25
3.4. Preparación de los nodos	29
3.4.1. Creación de una aplicación en TTN	29
3.4.2. Registro de los nodos sensores en TTN	30
3.4.3. Configuración de los nodos sensores	31
Capítulo 4.	34
Cálculo de cobertura mediante RADIOGIS	34
4.1. ¿Qué es RADIOGIS?	35
4.2. Funcionalidades de RADIOGIS	35
4.3. Gestión de base de datos	36
4.4. Modelos de propagación	38
4.4.1. COST-231	39
4.4.2. Xia-Bertoni	40
4.5. Modelo Digital del Terreno de edificios de Cartagena	42
4.5.1. Obtención del Mapa de Cartagena	42
4.5.2. Cálculo de la altura de los edificios	44

4.6. Cálculo de Coberturas red LoRa en Cartagena.....	47
Capítulo 5.	51
Campaña de medidas y comparación de modelos	51
5.1. Recogida de datos	52
5.2. Interfaz gráfica y resultados.....	52
Capítulo 6.	57
Conclusiones y líneas futuras	57
6.1. Conclusiones.....	58
6.2. Líneas futuras	58
Capítulo 7.	59
Bibliografía	59
Anexo A	61
A.1. ¿Qué es ArcGIS?	62
A.2. ¿Qué es un Sistema de Información Geográfica?	63
A.2.1. Funciones de un SIG.....	64
A.2.2. Cuestiones a las que puede responder un SIG	64
A.3. DESKTOP GIS	65
A.3.1. ArcMap.....	65
A.3.2. ArcToolbox	66
A.4. Tipos de datos de un SIG.....	67
Anexo B. Programas desarrollados	69
B.1. Recogida de datos en tiempo real en Java.....	70
B.2. Decodificación datos JSON.....	71
B.3. Análisis de las medidas. Interfaz gráfica.	72
B.4. Conversión de datos UTM a DEC exportando fichero excel.	81

Índice de tablas

Tabla 1: Gateways LoRa.	19
Tabla 2: Características físicas del MTMDK-ST-MDOT [8].....	20
Tabla 3: Comandos AT.....	32
Tabla 4: Estudio de pérdidas.	47

Índice de figuras

Figura 1.1: Diagrama de Gantt.	13
Figura 1.2: Entorno de ArcMap.	15
Figura 1.3: Putty.	15
Figura 1.4: Interfaz gráfica.	16
Figura 2.1: Red LoRa [3].	19
Figura 2.3: Kerlink IBTS Compact [3].	19
Figura 2.2: Kerlink Station [3].	19
Figura 2.4: Router Multitech Multiconnect Conduit (MTCDTIP-LEU1-270L-868 MULTITECH) [6].	20
Figura 2.5: MTMDK-ST-MDOT [8].	20
Figura 2.7: MTDOT-BOX [7].	21
Figura 2.6: Batería MTDOT-BOX [7].	21
Figura 3.1: Conexión de cables.	24
Figura 3.2: Configuración de MTDOT-BOX.	25
Figura 3.3: Tarjeta SIM del Gateway	26
Figura 3.4: Antena LoRa del Gateway	26
Figura 3.5: Gateway conectado al Router.	26
Figura 3.6: Ventana Putty para conexión SSH.	27
Figura 3.7: Manera en Linux de entrar a un archivo.	27
Figura 3.8: Descarga del programa 'packet-forwarder'.	28
Figura 3.9: Configuración Packet-Forwarder a través de Putty.	28
Figura 3.10: Consola TTN de la Pasarela.	29
Figura 3.11: Creación de una aplicación en TTN.	30
Figura 3.12: Ventanas de interés de la aplicación en TTN.	30
Figura 3.13: Registro de nodo TTN.	31
Figura 3.14: Envío de datos.	32
Figura 3.15: Activación de nodo en consola TTN.	33
Figura 3.16: Estructura del mensaje.	33
Figura 4.1. Aspecto general de RADIOGIS.	35
Figura 4.2. Cobertura consultada en base de datos.	37
Figura 4.3. Tabla de atributos de la cobertura radioeléctrica.	37
Figura 4.4: Base de datos de emplazamientos.	38
Figura 4.5: Modelo de propagación COST-231 [13].	39
Figura 4.6: Modelo de propagación Xia-Bertoni [14].	41
Figura 4.7: Entorno de la Sede electrónica del Catastro.	43

Figura 4.8: Difusión de datos Catastrales.....	43
Figura 4.9: Elección de capas vectoriales.....	44
Figura 4.10: Municipio de Cartagena cargado en el entorno de ArcMap.....	44
Figura 4.11: Ventana Add Field.....	45
Figura 4.12: Ventana Field Calculator para el cálculo de pisos.....	45
Figura 4.13: Cálculo de alturas de los edificios.....	46
Figura 4.14: Municipio de Cartagena en 3D.....	46
Figura 4.15: Cálculo de cobertura.....	47
Figura 4.16: Cobertura generada en el Barrio Peral.....	48
Figura 4.17: Cobertura generada en la Alameda.....	49
Figura 4.18: Mapa base con el nombre de las calles del municipio.....	49
Figura 5.1: Medidas en formato JSON.....	52
Figura 5.2: Ventana de inicio de la Interfaz gráfica.....	53
Figura 5.3: Interfaz.....	53
Figura 5.4: Medidas tomadas a partir del Gateway de San Miguel.....	54
Figura 5.5: Resultados modelo COST-231.....	55
Figura 5.6: Resultados modelo Xia -Bertoni.....	56
Figura 5.7: Resultados COST-231 y Xia- Bertoni.....	56
Figura A.1: ArcObjects es una librería común para los cuatro componentes ArcGIS [5].....	62
Figura A.2: ArcObjects es empleado por desarrolladores GIS en conjunto con los otros cuatro productos [17].....	63
Figura A.3: Ciclo de vida de un SIG [17].....	63
Figura A.4: Interfaz gráfica de ArcMap.....	66
Figura A.5: Interfaz Data View [9].....	66
Figura A.7: Interfaz gráfica de ArcToolbox.....	67
Figura A.9: Modelo raster de Cartagena.....	68
Figura A.8: Modelo vectorial de una parte del municipio de Cartagena.....	68
Figura A.10: geografía real a modelar; información geográfica en formato vectorial; e información geográfica en formato raster [21].....	68

Capítulo 1.

Introducción y Objetivos

1. Introducción y Objetivos

Para generar un desarrollo urbano sostenible es necesario introducir conceptos y herramientas que nos ayuden a mejorar la vida de las personas, este es el principal desafío y la razón por la cual, se están introduciendo las denominadas Smart Cities [1] en la sociedad.

Las TIC, conocidas como tecnologías de la información y de la comunicación, son usadas en diversas ciudades con el objetivo de proveer a estas de infraestructuras que garanticen un desarrollo sostenible, un incremento de la calidad de vida de los ciudadanos y una mayor eficacia de los recursos disponibles, dando lugar a lo conocido como Smart City [1].

La automatización de procesos, el control total de la ciudad, la detección automática de averías de los servicios públicos, la prevención de desastres naturales o la optimización del tráfico urbano son también posibles gracias al “Big Data” [2], siendo este un análisis masivo de datos.

Por todo ello, en el municipio de Cartagena se están llevando a cabo por parte del ayuntamiento acciones encaminadas a convertir la ciudad en una Smart City [1]. Una de ellas consiste en el despliegue de una red inalámbrica para recolectar la información recogida por unos sensores (temperatura, humedad, etc.) y enviarla a un centro de control para su gestión. El sistema que está desplegando el ayuntamiento es LoRa [3], que consiste en estaciones base que cubren un área de la ciudad, una serie de sensores, los cuales se comunican con las estaciones base vía radio (banda de frecuencias de 868MHz). Seguidamente, estos datos son recogidos, para su posterior análisis, en una plataforma llamada LoRaWan que actúa como servidor.

La planificación de estos sistemas de radiocomunicaciones se realiza mediante herramientas informáticas. Una de estas herramientas es RADIOGIS [4], desarrollada por el grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena (UPCT). RADIOGIS está basada en el Sistema de Información Geográfica (SIG) ArcGIS de ESRI [5] e incorpora funcionalidades y modelos de propagación para el cálculo de cobertura radioeléctrica.

El objetivo general del proyecto es un análisis teórico y experimental de una red LoRa en la ciudad de Cartagena. Los objetivos específicos del proyecto son:

- Conocimiento de los recursos de los que dispone Cartagena.
- Conocer el funcionamiento del sistema LoRa.
- Configurar un prototipo de sistema LoRa para llevar a cabo campañas de medidas.
- Disponer de un modelo digital del terreno de los edificios.
- Estimar la cobertura radioeléctrica.
- Campaña de medidas a partir del sistema LoRa.
- Comparar medidas con los modelos de propagación existentes.

El trabajo se ha realizado en la Cátedra de Infraestructuras Municipales Ayuntamiento de Cartagena – UPCT en la línea L3. Implantación de sistemas de control de servicios públicos y valorización de herramientas de gestión municipales.

1.1. Fases

Las fases que se siguieron en la realización de este proyecto fueron:

- Estado del arte de los sistemas LoRa y su funcionamiento.
- Estudio de RADIOGIS.
- Investigación de posibles soluciones para la obtención de la capa vectorial del municipio de Cartagena. Descarga de este a través de la sede electrónica del Catastro y su integración en ArcGIS.
- Configuración del Gateway LoRa y conexión con la plataforma LoRaWAN.
- Configuración de nodos y comunicación de estos con la red LoRa. Activación del sensor en la plataforma LoRaWAN.
- Creación de una clase en Java con la finalidad de recoger información que proviene de la comunicación entre el sensor LoRa (UPCT) y los Gateways del ayuntamiento a partir de Apache HTTP Client.
- Creación de una clase en Java para la decodificación de la información.
- Procesamiento de datos enviados por el sensor al Gateway (pasarela).
- Creación de la base de datos de estaciones base LoRa como capa vectorial en ArcGIS.
- Cálculo de cobertura con RADIOGIS.
- Campaña de medidas en Cartagena.
- Creación de una interfaz gráfica mediante Python.

Planificador de proyectos

Seleccione un periodo para resaltarlo a la derecha. A continuación hay una leyenda que describe el gráfico.

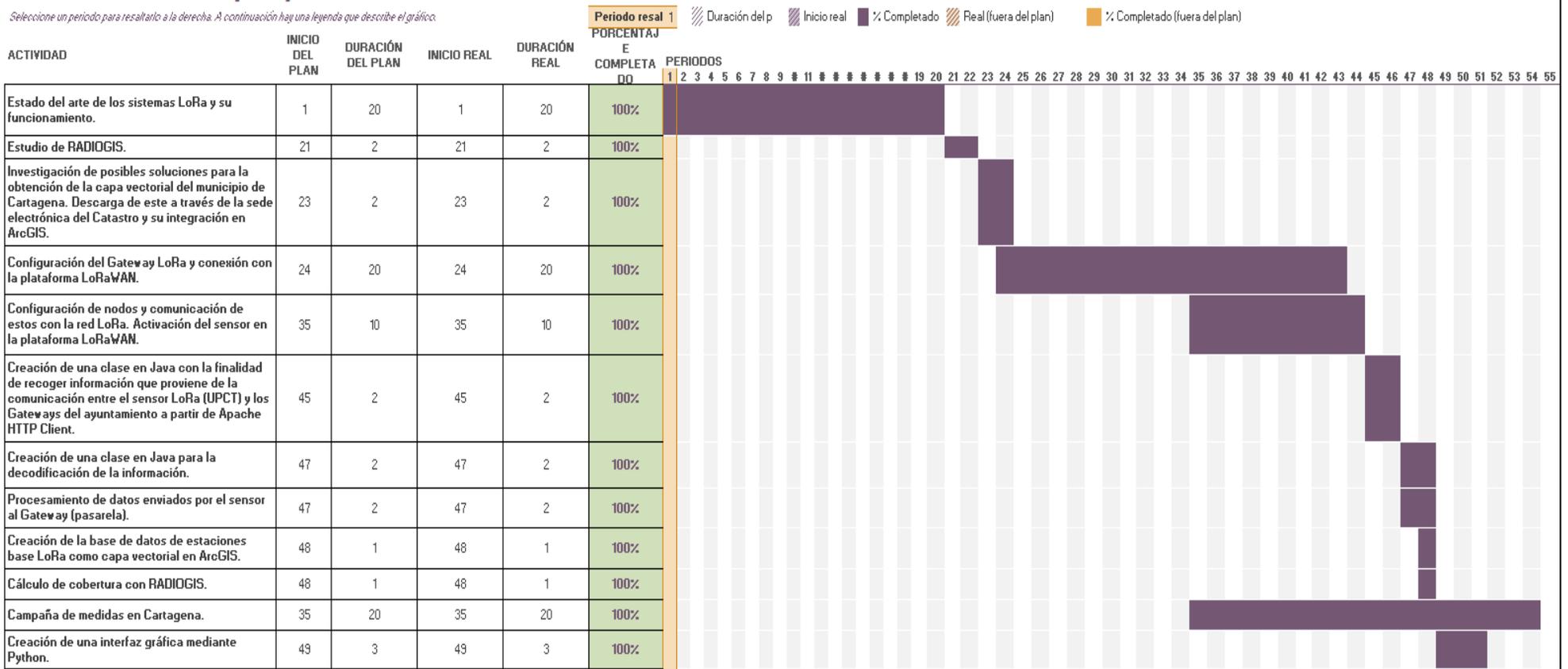


Figura 1.1: Diagrama de Gantt.

1.2. Contenido

Además de este capítulo de introducción y objetivos, este trabajo fin de máster está compuesto por 6 capítulos.

El **segundo capítulo**, estudia la tecnología LoRa utilizada en la realización de este proyecto. De esta manera, se presentarán algunos conocimientos con la finalidad de una buena comprensión de los capítulos posteriores.

El **tercer capítulo**, describe el proceso de implantación del sistema LoRa *end to end*.

El **cuarto capítulo**, describe en qué consiste la herramienta RADIOGIS desarrollada por el grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la UPCT, con sus principales componentes y funcionalidades. Además, expone los diferentes modelos de propagación que existen, focalizando la atención, tanto en el modelo COST-231 como en el de Xia-Bertoni. Finalmente describe paso por paso la metodología seguida dentro del entorno de RADIOGIS con la finalidad de calcular coberturas en todos los puntos del municipio donde se encuentra dispuesta una estación base.

El **quinto capítulo**, explica cómo se han recogido los datos a ser procesados con la finalidad de realizar un análisis de resultados.

En el **sexto capítulo**, se exponen las distintas conclusiones recogidas durante la ejecución de este proyecto, además de cuáles podrían ser las líneas futuras en una posterior continuación de este.

El **séptimo capítulo**, recoge todas las referencias bibliográficas empleadas en el desarrollo de este proyecto.

Por último, se incluyen 2 anexos. El **anexo A**, versa sobre los Sistemas de Información Geográfica (SIG o GIS, de Geographic Information Systems). Primeramente, se describe qué son los SIG y para qué se emplean. Seguidamente se nombran las diferentes aplicaciones de escritorio que presenta GIS. Para terminar, se comenta los distintos tipos de datos en un GIS. El **anexo B**, muestra los diferentes programas desarrollados durante este proyecto.

1.3. Herramientas utilizadas

En este apartado se van a exponer las distintas herramientas utilizadas durante la realización de este proyecto.

- Sistema de Información Geográfica, **ArcGis 9.1** [5] del cual se ha empleado la aplicación ArcMap (ver Figura 1.2).

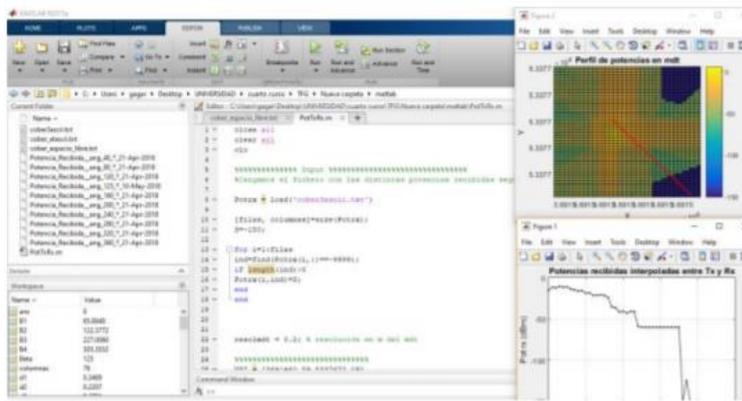


Figura 1.2: Entorno de ArcMap.

- Herramienta **RADIOGIS**, desarrollada por el grupo de Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena (UPCT).
- **MTCDD-LEU1-247A-EU-GB**. Gateway utilizado con la finalidad de desplegar una red LoRaWAN [6].
- **MTDOT-BOX-G-868**. Dispositivo de largo alcance programable desarrollado para funcionar con cualquier gateway compatible con LoRaWAN, incluyendo MultiConnect Conduit de MultiTech [7].
- **MTMDK-XDOT-EU1-A00**. Transceptor LoRa empleado como medio para la configuración del sensor [8].
- Software **Putty** o **Tera Term**. Software que permitirá al usuario establecer una conexión SSH. (Fig. 1.3)

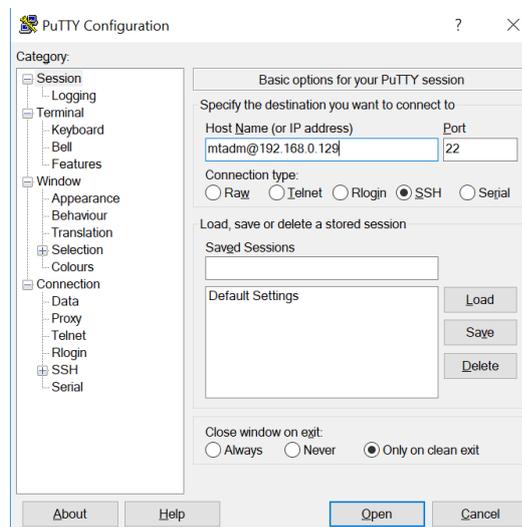


Figura 1.3: Putty.

- **FortiClient VPN**. Activa la VPN cedida por el ayuntamiento, la cual permitirá visualizar la información que los gateways del ayuntamiento están recibiendo de los sensores.
- **IntelliJ IDEA Community Edition 2019.1.3**. Software utilizado para poder obtener en tiempo real la información de potencia y geolocalización Sensor-Gateway.
- **Matlab 2015** y **Python**. Lenguajes utilizados para conseguir generar una interfaz gráfica. Esta será capaz de cotejar la información de dos ficheros con la finalidad de realizar un análisis de resultados. (Fig. 1.4)

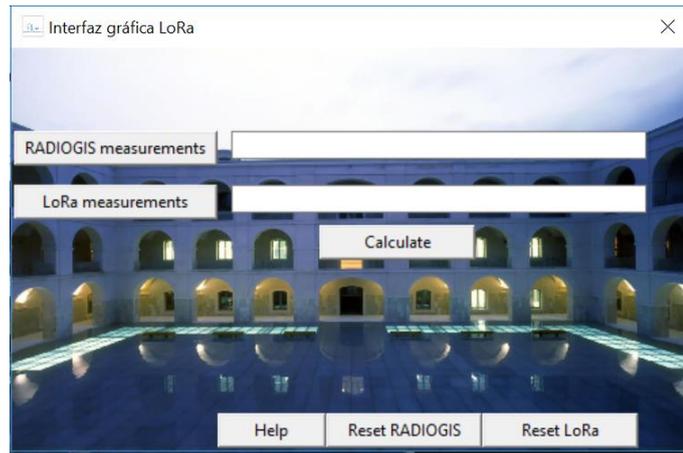


Figura 1.4: Interfaz gráfica.

Capítulo 2.

Estado del arte

2.1. Introducción

El término *Internet of Things* (IoT) [9], es utilizado para definir la conexión de objetos a Internet, permitiendo compartir diversa información mediante la interoperación de varios sistemas de comunicación, plataformas y protocolos. Esta conexión se puede realizar de varias maneras. De aquí que el IoT sea una buena idea de implementar en nuestro entorno, dando lugar a un aumento en el grado de 'Smart City', también denominada ciudad inteligente [1].

Asimismo, el municipio de Cartagena se va a abastecer de tecnología LoRa para proveer de un análisis masivo de datos (Big Data) a partir de diversos sensores (humedad, temperatura, proximidad etc).

En este capítulo se va a estudiar la tecnología utilizada en la realización de este proyecto. Esta es denominada LoRa [3]. De esta manera, se presentarán algunos conocimientos con la finalidad de una buena comprensión de los capítulos posteriores.

Actualmente, LoRa es conocida como una de las principales tecnologías dentro de las LPWAN (*Low-Power Wide Area Network*), gracias a su diversidad de ventajas, tales como prestar una comunicación bidireccional, la facilidad de interoperabilidad entre sensores sin necesidad de instalaciones complejas y la libertad que ofrece a usuarios, desarrolladores y empresas para el despliegue de IoT.

2.2. LoRa

Una red LoRa está formada por tres elementos esenciales (Fig 2.1).

- **Dispositivos finales.** Son dispositivos utilizados con el objetivo de conectar objetos a la red LoRa. Recogen la información específica del objeto y la transmiten a la estación base (Gateway).
- **Pasarelas.** Estaciones base LoRa. Éstas reciben información de distintos dispositivos finales (sensores de humedad, de proximidad, etc.). Tras obtener esta información, es enviada a servidores de red.
- **Servidores de red.** Estos equipos reciben la información proveniente de los dispositivos finales, tras esto, se encargan del procesamiento de estos datos, así como de la gestión y configuración de la red.

Entre las principales ventajas del sistema LoRa se encuentran [3]:

- Alta tolerancia a las interferencias.
- Alta sensibilidad para recibir datos (-168 dBm).
- Bajo consumo.
- Largo alcance (10-20 km).
- Baja transferencia de datos (hasta 255 bytes).
- Frecuencia de trabajo: 868 MHz en Europa.
- Bidireccionalidad.

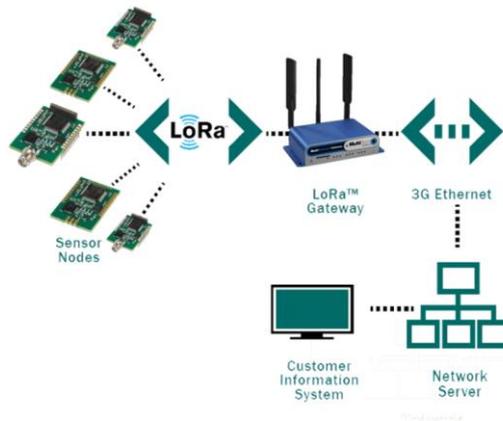


Figura 2.1: Red LoRa [3].

Todas estas ventajas hacen que ésta sea una tecnología ideal para aplicaciones en Smart Cities. Por ello, actualmente, el Ayuntamiento de Cartagena ha decidido implantar Gateways propios abarcando gran parte del municipio. En la Tabla 1 se pueden observar tres modelos diferentes de estaciones base [3].

	Potencia máxima de Tx (PIRE)	Sensibilidad (dBm)	Bandas LORA (MHz)	Ganancia (dBi)	Antena
Kerlink Station (Fig. 1.9)	27	-141	863-870	3	Omnidireccional
Kerlink IBTS Compact (Fig. 1.10)	30	-141	863-870	3	Omnidireccional
Router Multitech Multiconnect Conduit (MTCDTIP-LEU1-270L-868 MULTITECH) (Fig. 1.11)	27	-141	863-870	3	Omnidireccional

Tabla 1: Gateways LoRa.

A continuación, se presenta la imagen de cada LoRa mencionado anteriormente,



Figura 2.2: Kerlink Station [3].



Figura 2.3: Kerlink IBTS Compact [3].



Figura 2.4: Router Multitech Multiconnect Conduit (MTCDTIP-LEU1-270L-868 MULTITECH) [6].

2.3. Entorno de trabajo LoRa

2.3.1. MTMDK-ST-MDOT

MultiConnect mDot Micro Developer Kit (Fig. 2.5) es una tarjeta de desarrollo y programación de microprocesadores. Este dispositivo presenta un formato USB, permitiendo al usuario conectar de una manera rápida y sencilla un MultiConnect mDot Box y comenzar a desarrollar su propia aplicación. El diseño es ideal para conectarse a un ordenador portátil y realizar pruebas de alcance de la red LoRa [3].

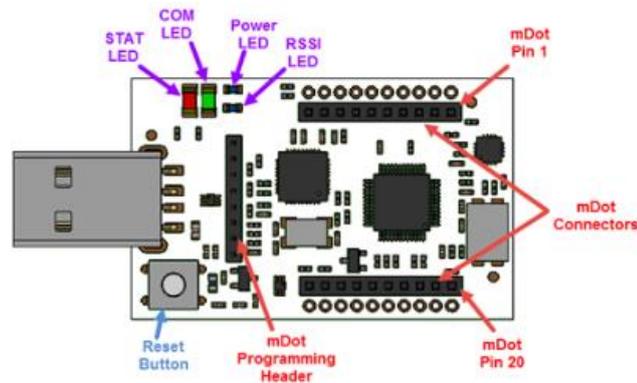


Figura 2.5: MTMDK-ST-MDOT [8].

A continuación, en la tabla 3 se muestra una breve descripción de las características físicas del MTMDK-ST-MDOT.

LED	Descripción
STAT	LED 1 Estado, luz roja parpadea cuando se enciende el dispositivo
COM	LED 2 Comunicación, luz verde que muestra cuando el dispositivo se está comunicando
PWR	LED 3, luz azul que indica cuando la tarjeta está conectada a la corriente
RSSI	LED 4, indicador de intensidad de señal recibida

Tabla 2: Características físicas del MTMDK-ST-MDOT [8].

2.3.2. MTDOT-BOX

MultiConnect mDot Box [7] es un dispositivo de largo alcance programable, el cual, como veremos en capítulos posteriores, ha sido usado con la finalidad de validar campañas de medidas de potencia con distintos modelos de propagación. Este dispositivo cuenta con sensores incorporados que muestran varios parámetros, entre ellos se encuentran: temperatura, luz ambiental, presión, elevación, RSSI, ruido, además de la información del acelerómetro de tres ejes. MDot Box fue desarrollado para funcionar con cualquier gateway compatible con LoRaWAN, incluyendo MultiConnect Conduit de MultiTech.

MDot Box está equipado con el MultiConnect mDot, un módulo de bajo coste programable LoRaWAN, incorporado dentro de un chasis de plástico. Este funciona mediante una batería (Fig. 2.6) y una pantalla LCD para mostrar la información del menú y de la navegación del dispositivo, así como datos de red, modo de prueba, GPS y estado de los sensores. También se incluyen dos botones para controlar las funciones de navegación en el menú y dos LEDs para mostrar el estado de la red y del GPS (Fig. 2.7).

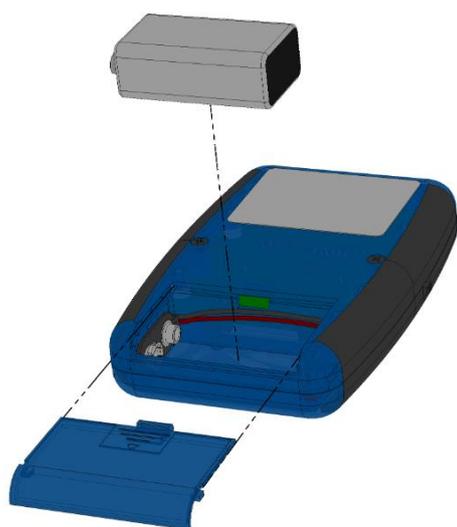


Figura 2.6: Batería MTDOT-BOX [7].



Figura 2.7: MTDOT-BOX [7].

MDot Box proporciona una gran cantidad de información, como la ubicación, factor de dispersión, rendimiento, indicador de intensidad de señal recibida (RSSI), hora, fecha, actividad y nivel de potencia en distintas ubicaciones, permitiendo a las empresas valorar una buena localización para las estaciones base. Los datos recogidos por este dispositivo llegarán en tiempo real a una plataforma elegida por el cliente.

2.3.3. The Things Network

TTN (The Things Network) [10] (Fig. 2.8) es una red IoT que permite que dispositivos de baja potencia utilicen pasarelas de largo alcance para conectarse a una red descentralizada de código

abierto con la finalidad de intercambiar datos con las aplicaciones. Esta red permite al usuario conectarse a la red y registrar diversos tipos de Gateway compatibles con LoRaWAN.

TTN proporciona un back-end de servidores de aplicación que gestionan los gateways, además de una plataforma que va a permitir al usuario crear aplicaciones, registrar dispositivos, y recibir datos en tiempo real.

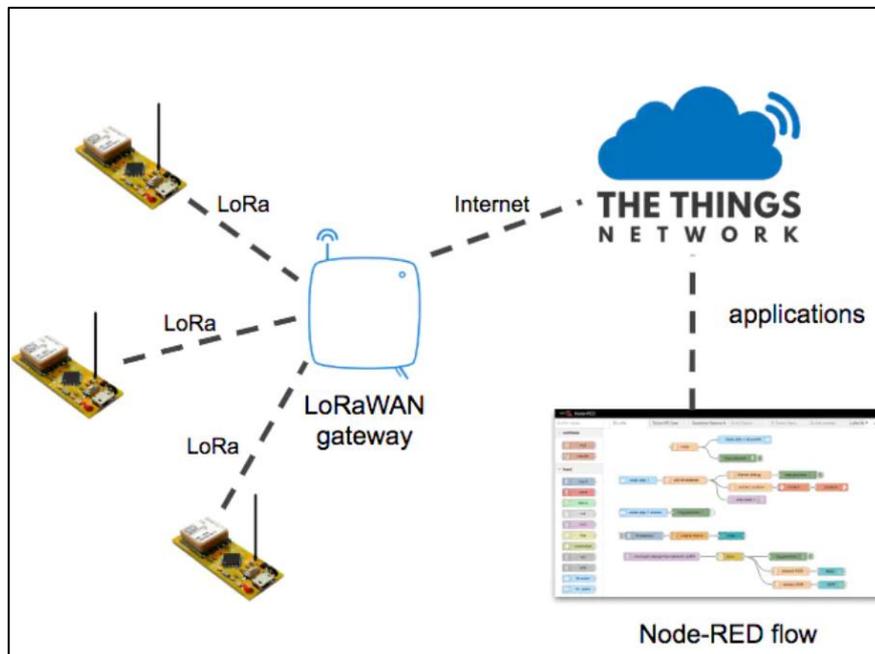


Figura 2.8: The Things Network [10].

Capítulo 3.

Implementación de una red LoRa

3.1. Introducción

Durante este apartado se va a describir el proceso de implantación del sistema LoRa *end to end*. Empezando por la configuración de los distintos sensores y gateway y terminando por la comunicación entre ellos.

3.2. Configuración de MTDOT-BOX

Tras registrar el dispositivo en la plataforma LoRaWAN, mDot Box necesita ser configurado con algunos de los parámetros generados en la aplicación *'The Things Network'* que debemos de crear. Esta configuración será realizada mediante dos dispositivos indistintamente, UDK2 o MDK. Nota: muy importante conectar los 8 cables como se muestran en la Fig. 3.1.

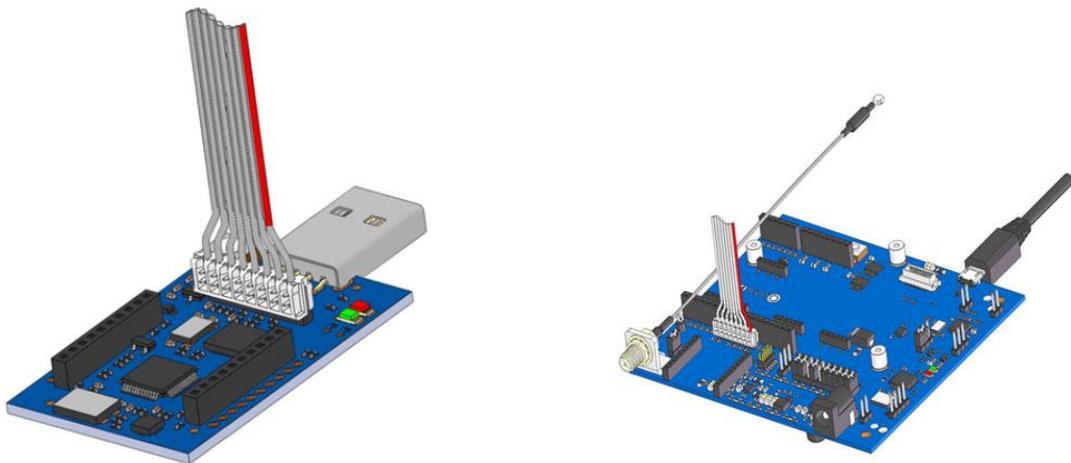


Figura 3.1: Conexión de cables.

En este proyecto se ha utilizado MDK como puente para poder configurar mDot Box, conectándose de la manera que se visualiza en la Fig. 3.2. Tras establecer la comunicación serie con el MTMDK-ST-MDOT a partir de la herramienta *'Putty'*.



Figura 3.2: Configuración de MTDOT-BOX.

Una vez se pone en marcha el equipo, aparece en la pantalla varias opciones de menú [2]:

- **View Data:** muestra los datos enviados previamente.
- **Select Region:** permite seleccionar la región adecuada en función de su frecuencia. La región es seleccionada al inicio, pero es necesario cambiar ésta, si se desea utilizar el dispositivo fuera del área originalmente seleccionada.
- **Survey Single:** el dispositivo envía un mensaje de solicitud al gateway y éste devuelve una respuesta. El usuario puede probar la conectividad con la pasarela a diferentes velocidades de datos y combinaciones de potencia.
- **Survey Sweep:** la conectividad con el gateway se prueba en una gama de niveles de potencia y velocidades de datos. El usuario puede configurar las velocidades de datos mínima y máxima, además de los niveles de potencia que desea utilizar para los barridos utilizando el Modo configuración.
- **LoRa demo:** esta modalidad permite visualizar y recopilar información recogida por los sensores en tiempo real. Los datos del sensor también se envían al gateway periódicamente a un intervalo definido por el usuario.
- **Configuration:** modificar los ajustes del dispositivo mediante comandos AT.

3.3. Configuración del Gateway

MultiTech MultiConnect Conduit (Gateway): utilizado con el fin de desplegar una red LoRaWAN. Es la pasarela industrial más configurable y escalable para aplicaciones IoT industriales. Permite conectarse a Internet haciendo uso de tecnología inalámbrica celular como 2G, 3G o 4G-LTE, WiFi o Ethernet por cable. Además, hay disponibles una gran cantidad de accesorios de expansión denominados mCard™. Se trata de pequeñas tarjetas electrónicas que proveen de conexión LoRa, Ethernet, GPIO o Serie, lo cual permite una gran flexibilidad de utilización [6].

En este caso, tan solo ha hecho falta insertar una tarjeta SIM (Fig. 3.3) suministrada por el proveedor (Movistar), además de anclar una antena LoRa al conector de la esquina superior del Gateway (Fig. 3.4).



Figura 3.3: Tarjeta SIM del Gateway

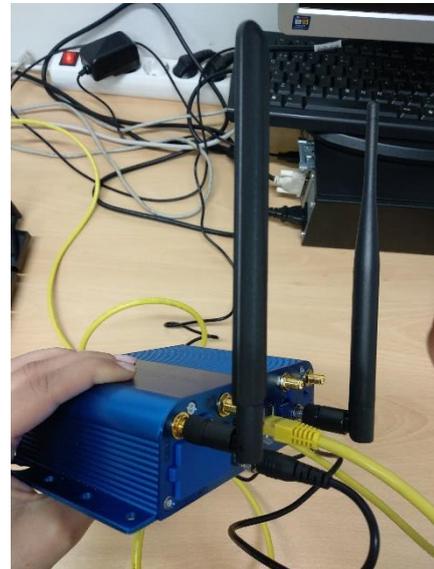


Figura 3.4: Antena LoRa del Gateway

Adicionalmente, se provee de una conexión a Internet mediante un cable Ethernet LAN convencional conectado a una de las bocas del router, proveedor de acceso a Internet. Este router le asignará una IP dinámica de red (Fig. 3.5).



Figura 3.5: Gateway conectado al Router.

El MultiConnect Conduit es el modelo Linux. Este proyecto se ha realizado con su versión 3.2.0. Por consiguiente, para ser capaz de acceder a él, se usa el protocolo SSH a través del software Tera Term o PuTTY, los cuales permiten al usuario conectarse a una consola de comandos, con la cual, podrá interactuar con el sistema operativo y poder instalar los archivos y carpetas necesarias para conectar éste con la plataforma de red en la nube LoRaWAN.

A continuación, para acceder al Conduit es necesario conocer la IP que le ha asignado el router a partir de Ethernet. Una IP es una dirección de red única para identificar a cada uno de los dispositivos conectados. Consta de 4 números separados por puntos simples. En este caso, el router ha asignado al Gateway la IP 192.168.0.129.

Tras asignar la IP del Conduit comentada anteriormente, se utilizará el software PuTTY sobre Windows para establecer una conexión SSH con él (Fig. 3.6).

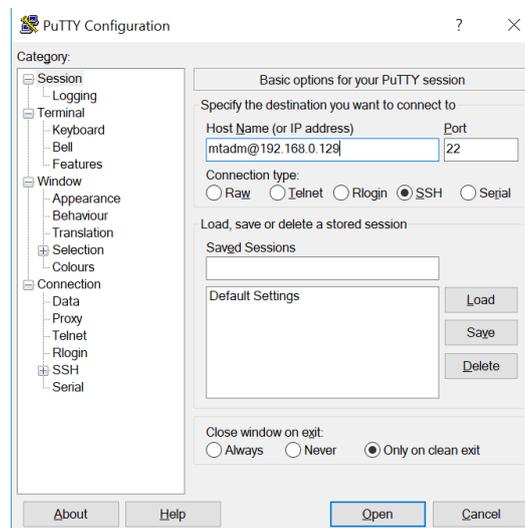


Figura 3.6: Ventana Putty para conexión SSH.

Por defecto, las credenciales del modelo mLinux son: 'mtadm@IP' y contraseña: 'root', donde una vez iniciada sesión, se puede cambiar esta contraseña. Seguidamente a partir del comando de Linux: 'date' se comprueba que el Gateway tiene establecida la fecha y hora europea. En este caso, no hubo que cambiarla puesto que era correcta.

Seguidamente, se configura la red. A partir del comando 'vi' o 'nano' se va a poder modificar un archivo dentro del sistema operativo de Linux. De esta manera, modificamos el documento 'interfaces', el cual se encuentra en la siguiente ruta: /etc/network/interfaces (Figura 3.7).

```
mtcdt:~$ cd /etc/network
mtcdt:/etc/network$ sudo nano interfaces
```

Figura 3.7: Manera en Linux de entrar a un archivo.

Tras esto, es importante saber que es necesario conectar este dispositivo con Internet. De este modo, aunque el Gateway vaya conectado por Ethernet al router, si realizamos un ping a 8.8.8.8 no va a funcionar. Por tanto, se establece como puerta de enlace predeterminada de nuestro

Gateway la dirección IP del router que presenta salida hacia Internet. Además, se configura el servidor DNS del Gateway como el de Google, cuya IP es: 8.8.8.8 o 8.8.4.4. Para ello, se debe establecer la siguiente línea:

```
post-up echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

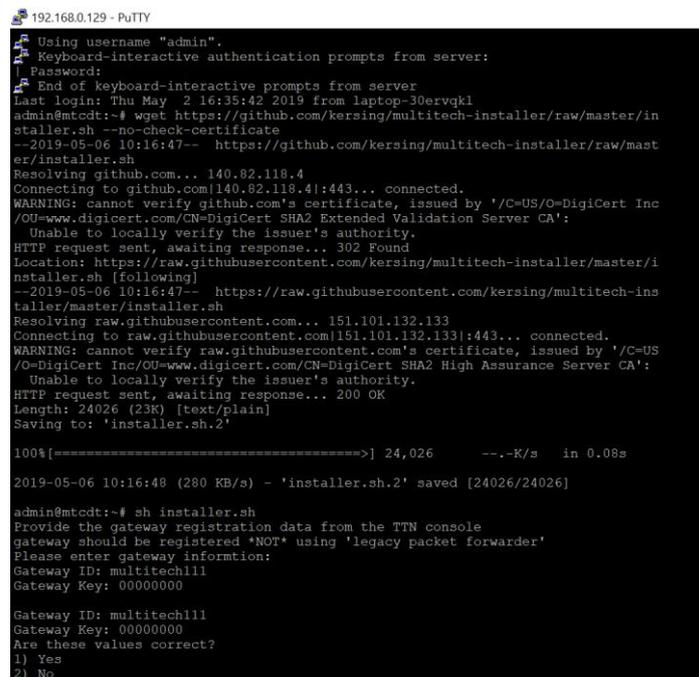
Finalmente, una vez comprobado que el ping a 'google.com' funciona, se procede a la descarga e instalación de un programa llamado 'packet-forwarder', el cual se encargará de recoger los paquetes LoRa que lleguen por cualquiera de los 7 canales de escucha a la plataforma de red LoRaWAN. Este programa se encuentra en el repositorio de Github. Por lo que, gracias a la conexión con Internet se puede proceder a la descarga del programa (Fig. 3.8).



```
Using username "mtadm".
mtadm@192.168.0.129's password:
Last login: Wed Jun 19 11:04:55 2019 from 192.168.0.106
mtodt:~$ sudo wget https://github.com/kersing/multitech-installer/raw/master/installer.sh --no-check-certificate
```

Figura 3.8: Descarga del programa 'packet-forwarder'.

Una vez descargado 'packet-forwarder', se procede a ejecutar el instalador. Este pedirá que se introduzcan una serie de parámetros de configuración del Gateway en el servidor (Fig. 3.9). Estos parámetros son proporcionados por la plataforma LoRaWAN proveedora donde previamente el usuario habrá creado una cuenta y registrado el Gateway. Si en algún momento de la instalación, se produjese algún error tipo, no poder ejecutar una acción por no tener permisos, no olvidar utilizar el comando 'sudo'. Este es una utilidad del sistema operativo de Linux, que permite ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura, convirtiéndose así temporalmente en superusuario.



```
192.168.0.129 - PuTTY
Using username "admin".
Keyboard-interactive authentication prompts from server:
Password:
End of keyboard-interactive prompts from server
Last login: Thu May 2 16:35:42 2019 from laptop-30ervqkl
admin@mtodt:~$ wget https://github.com/kersing/multitech-installer/raw/master/installer.sh --no-check-certificate
--2019-05-06 10:16:47-- https://github.com/kersing/multitech-installer/raw/master/installer.sh
Resolving github.com... 140.82.118.4
Connecting to github.com[140.82.118.4]:443... connected.
WARNING: cannot verify github.com's certificate, issued by '/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 Extended Validation Server CA':
Unable to locally verify the issuer's authority.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/kersing/multitech-installer/master/installer.sh [following]
--2019-05-06 10:16:47-- https://raw.githubusercontent.com/kersing/multitech-installer/master/installer.sh
Resolving raw.githubusercontent.com... 151.101.132.133
Connecting to raw.githubusercontent.com[151.101.132.133]:443... connected.
WARNING: cannot verify raw.githubusercontent.com's certificate, issued by '/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA':
Unable to locally verify the issuer's authority.
HTTP request sent, awaiting response... 200 OK
Length: 24026 (23K) [text/plain]
Saving to: 'installer.sh.2'

100%[=====>] 24,026 --.-K/s in 0.08s

2019-05-06 10:16:48 (280 KB/s) - 'installer.sh.2' saved [24026/24026]

admin@mtodt:~$ sh installer.sh
Provide the gateway registration data from the TTN console
gateway should be registered 'NOT' using 'legacy packet forwarder'
Please enter gateway information:
Gateway ID: multitech111
Gateway Key: 00000000

Gateway ID: multitech111
Gateway Key: 00000000
Are these values correct?
1) Yes
2) No
```

Figura 3.9: Configuración Packet-Forwarder a través de Putty.

Para verificar el comportamiento, el estado y tráfico de la pasarela, se observa la consola de TTN, donde el estado del dispositivo debe estar como 'connected' (Fig. 3.10).

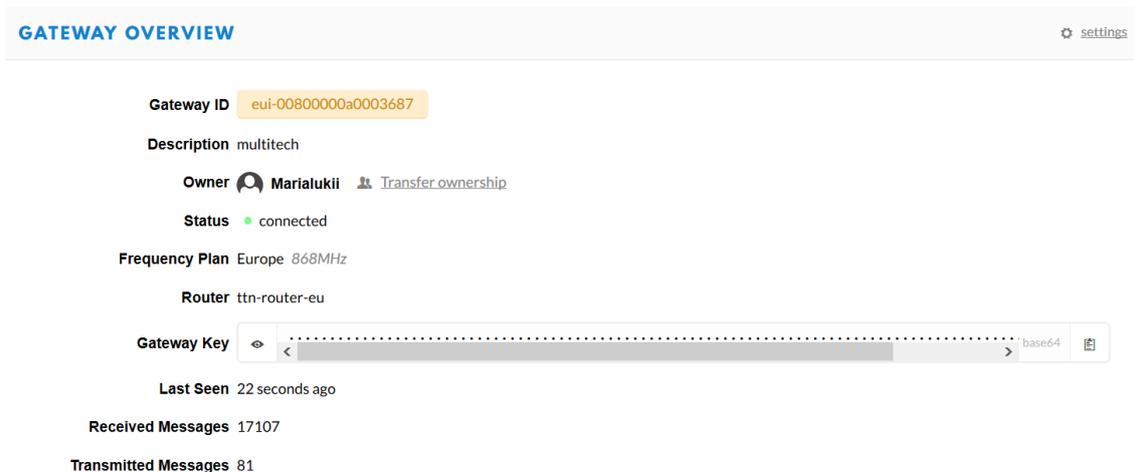


Figura 3.10: Consola TTN de la Pasarela.

3.4. Preparación de los nodos

Los nodos deben de ser preparados físicamente para poder cargarles el software que van a llevar instalado y que les permitirá conectarse y comunicarse eficazmente en la red. Para ello es preciso crear previamente una aplicación en el servidor donde se podrán registrar los nodos en ella, además de visualizar en tiempo real el tráfico ascendente y descendente de los datos.

3.4.1. Creación de una aplicación en TTN

En este apartado, se va a crear una aplicación paso por paso en la página web '**The Things Networks (TTN)**', la cual proporcionará el servidor. Al acceder con el usuario anteriormente creado al registrar el Gateway, en la pestaña de aplicaciones aparece la opción de añadir una aplicación (Fig. 3.11). Haciendo click en ella, se muestra una pantalla donde se piden como datos: un identificador, una descripción y el controlador desde el que se desea registrar la aplicación (en este caso "ttn-handler-eu" ya que la gateway está en Europa). El campo de "Application EUI" se deja vacío, puesto que, lo genera directamente el servidor.

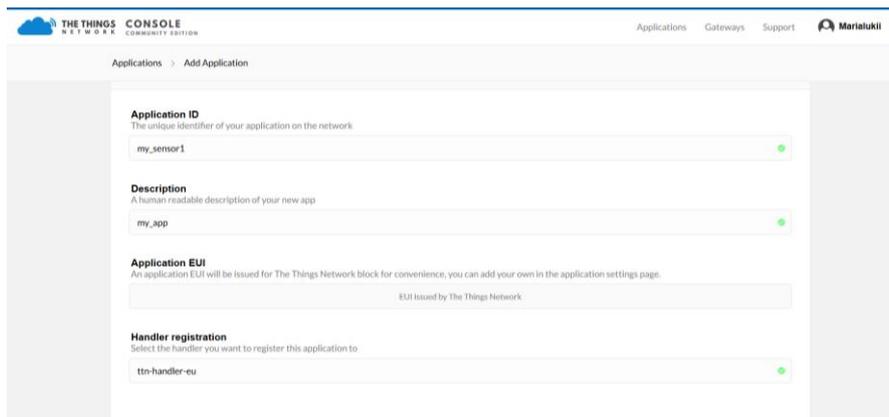


Figura 3.11: Creación de una aplicación en TTN.

Una vez creada la aplicación, se mostrará información de interés sobre esta (Fig. 3.12)

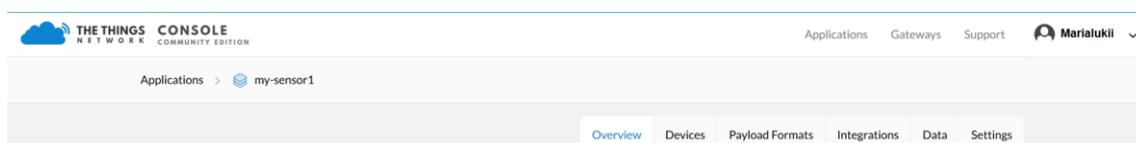


Figura 3.12: Ventanas de interés de la aplicación en TTN.

- **Overview:** se puede consultar el valor de App EUI y Access Keys necesarios para la posterior conexión de los nodos en la red.
- **Devices:** integración de los diferentes nodos que se quieren conectar a la red LoRa.
- **Payload Formats:** permite personalizar la función de decodificación para mostrar el resultado en formato carácter y de esta manera, entender el mensaje. Esta función se programa utilizando el lenguaje JavaScript.
- **Integrations:** es la forma más sencilla de conectar los dispositivos a una aplicación. Data Storage, TTN Mapper, etc.
- **Data:** permite observar en tiempo real los datos enviados y recibidos tanto en la estación base como en el sensor.

3.4.2. Registro de los nodos sensores en TTN

Una vez generada la aplicación en TTN se deben registrar los nodos que van a ser utilizados en la red LoRa. Para ello, se pincha en la pestaña “Devices”, la cual permite tanto observar los dispositivos registrados en esta aplicación como registrar uno nuevo. Simplemente basta con incluir como campos de información un identificador y el EUI del dispositivo, ya que el EUI de la aplicación lo completa TTN automáticamente y la App Key la generará al finalizar el registro del dispositivo (Fig. 3.13). Para conocer la EUI del dispositivo, se requiere utilizar el comando ‘AT+DI?’ de la Tabla 3 a partir del software Putty.

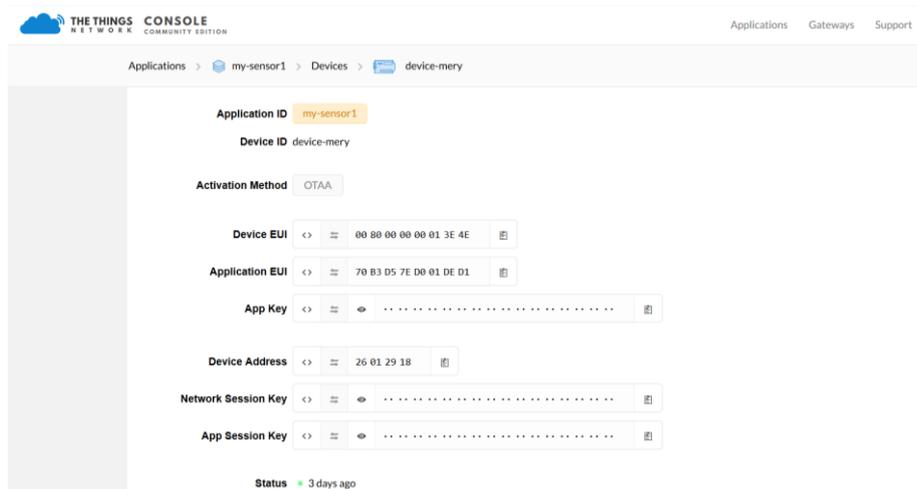


Figura 3.13: Registro de nodo TTN.

Otro campo a tener en cuenta es el de método de activación. Son procedimientos utilizados para la configuración y registro de los dispositivos finales: Over The Air Activation (OTAA) y Activation By Personalization (ABP) [10].

- Over The Air Activation (OTAA). Utilizado para este proyecto. Este método es mucho más fiable y seguro que ABP, puesto que, las claves de sesión se negocian con cada activación. Para que el dispositivo final se conecte a la red LoRa se deben integrar en el nodo una serie de parámetros necesarios para el procedimiento de conexión como son: un identificador global único (DevEUI) que identifica al nodo, el identificador de aplicación (AppEUI) y una llave criptográfica (AppKey). El procedimiento de activación comienza con el envío por parte del dispositivo final de una mensaje MAC de petición de unión, denominado join-request, entonces el servidor de red responde con un mensaje MAC de aceptación de unión join.accept.
- Activation By Personalization (ABP). Este método es mucho menos fiable y seguro que OTAA. En el dispositivo final se establece la dirección del dispositivo y las dos llaves de sesión. Todo el tráfico entre un dispositivo final y un servidor de red se cifra y se firma mediante la utilización combinada de dos llaves: la llave de sesión de aplicación (AppSkey) y la llave de sesión de red (NwkSkey).

3.4.3. Configuración de los nodos sensores

Para configurar los nodos sensores, es necesario realizar una comunicación serie con el nodo.

En primer lugar, se descargará un driver de la página oficial de Multitech. De esta manera, se detectará el puerto serie. Una vez detectado, se comprobará cual es el puerto exacto donde se encuentra conectado el dispositivo (Panel de control → Administrador de dispositivos).

Seguidamente, se ha decidido conectar el nodo a la red a través de comandos AT, donde podemos adquirir el manual de estos comandos de una manera sencilla en la guía de referencia proporcionada por esta empresa. Por consiguiente, importamos el firmware que nos va a permitir ejecutar dichos comandos, el cual se compilará dentro de la plataforma de 'arm mbed'.

Una vez compilado, se integra el programa a la placa MTDOT permitiéndonos conectar el dispositivo final a la red LoRa a partir de los comandos presentados en la Tabla 3.

Comandos AT	Función
AT+DI?	Conocer Identificador sensor
AT+NI=0, APP EUI (16-bits)	APP EUI generada
AT+NK=0, APP Key (16-bits)	APP KEY generada
AT&W	Save settings
ATZ	restart
AT+JOIN	Conectarse a la red

Tabla 3: Comandos AT.

Tras conectar este dispositivo a la red, se observa en la consola TTN un símbolo en forma de rayo, con el que nos indica que la conexión ha sido establecida con éxito (Fig. 3.15). Tras verificar la conectividad se enviarán una serie de datos (Fig. 3.14), los cuales serán recibidos por la pasarela y almacenados en los servidores de TTN.



Figura 3.14: Envío de datos.

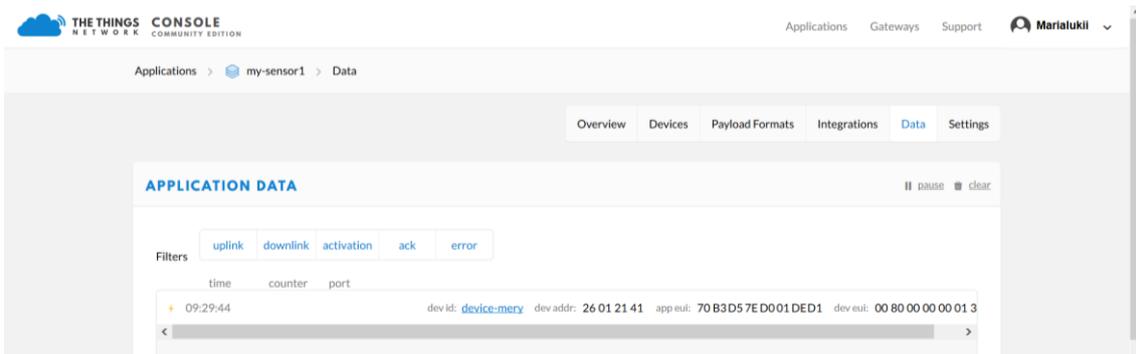


Figura 3.15: Activación de nodo en consola TTN.

En la Fig. 3.15 se observa la estructura del mensaje que el nodo ha enviado al Gateway. Este mensaje se encuentra en formato JSON. Se reciben los niveles de RSSI (Received Signal Strength Indicator), geolocalización y señal a ruido (SNR), el cual indica la calidad relativa de una señal recibida.

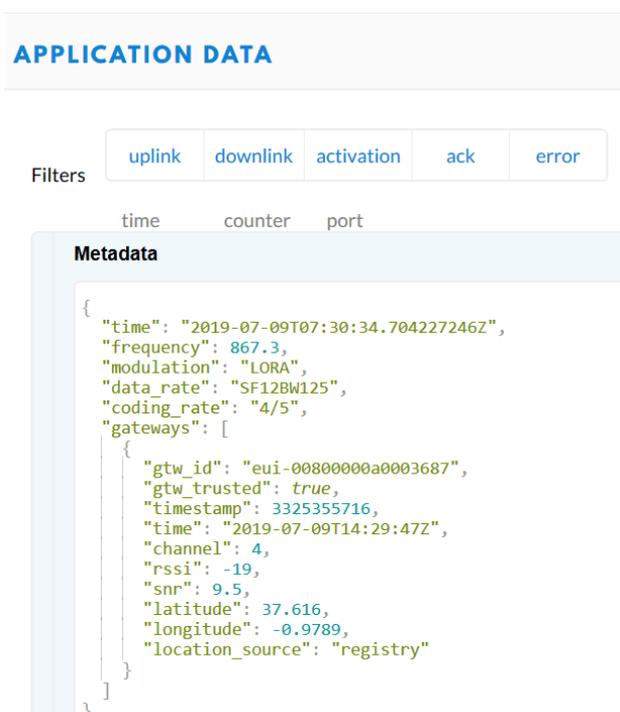


Figura 3.16: Estructura del mensaje.

Capítulo 4.
Cálculo de cobertura mediante
RADIOGIS

4.1. ¿Qué es RADIOGIS?

RADIOGIS [4] es una herramienta desarrollada por el grupo de investigación Sistemas de Comunicaciones Móviles (SiCoMo) de la Universidad Politécnica de Cartagena. Esta herramienta está basada en el sistema de información geográfica (SIG) ArcGIS de ESRI [5]. En el Anexo A se hace una introducción a los SIGs y se describen las herramientas de ArcGIS que se han utilizado en este trabajo.

RADIOGIS es incorporada de una manera rápida y sencilla en el entorno de **ArcMap** [5]. Presenta diversas funcionalidades, tales como, generar coberturas radioeléctricas a partir de distintos modelos de propagación, almacenar estas coberturas generadas 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.

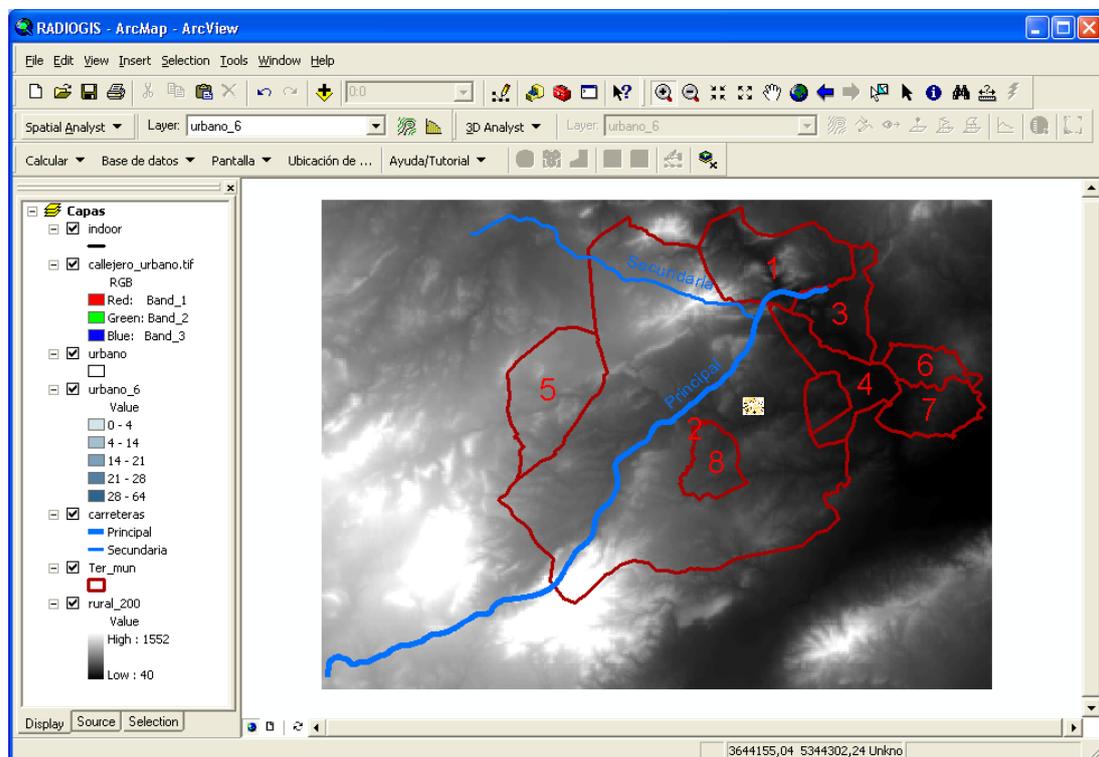


Figura 4.1. Aspecto general de RADIOGIS.

A continuación, se describen las distintas funcionalidades de RADIOGIS.

4.2. Funcionalidades de RADIOGIS

La extensión RADIOGIS incluye las siguientes funcionalidades [4]:

- Cálculo de coberturas radioeléctricas a partir de diversos modelos de propagación. Se crean dos capas, una ráster que contiene el valor de cobertura de potencia en cada punto de la

zona, y otra vectorial que contiene la tabla de atributos de la cobertura. Tras el cálculo, estas dos capas se añadirán automáticamente al mapa.

- Añadir al mapa coberturas radioeléctricas previamente calculadas que se encuentren en la base de datos de coberturas.
- Eliminar coberturas radioeléctricas tanto del mapa como de la base de datos de coberturas.
- Cálculo del porcentaje de potencia que supera un umbral sobre una zona del mapa aportado por una estación base, así como el cálculo del mejor servidor de un sistema formado por varias coberturas radioeléctricas desde diversos emplazamientos.
- Opción de “zoom” dentro del mapa.
- Identificar cualquier punto del mapa obteniendo información de los elementos de las distintas capas vectoriales del mapa que se encuentran en ese punto.
- Medir distancias entre distintos puntos del mapa y el área encerrada en los límites de un polígono.
- Crear nuevos emplazamientos.
- Añadir emplazamientos al mapa que se encuentren guardados en la base de datos.
- Eliminar emplazamientos tanto del mapa como de la base de datos de emplazamientos.
- Subir o descargar archivos del servidor (coberturas, emplazamientos, diagramas de antenas, etc.).

4.3. Gestión de base de datos

A continuación, en este apartado, se especificarán las características más importantes de RADIOGIS de una manera más detallada, los cuales se han empleado en el desarrollo de este proyecto.

RADIOGIS [4] incluye diversas funcionalidades tal y como se ha comentado en el apartado anterior, encontrándose entre ellas la gestión de base de datos. En todas las bases de datos la extensión de RADIOGIS introduce las opciones de ‘consultar’, ‘copiar’, ‘eliminar’ y algunas particularidades para cada una de las bases de datos.

Este apartado se va a centrar en ‘base de datos de coberturas radioeléctricas’ y ‘base de datos de emplazamientos’.

La **base de datos de coberturas radioeléctricas**, que se muestra en la figura 4.2, contiene registros con las distintas coberturas radioeléctricas calculadas y almacenadas. Cada cobertura radioeléctrica tiene asociada una estructura de tipo raster donde se almacenan los valores de potencia para cada una de las celdas de la zona donde se ha calculado la cobertura y una estructura de tipo vectorial (figura 4.3) cuya tabla de atributos contiene información utilizada durante la simulación de la cobertura. Para ver estos parámetros ver la Tabla 4.

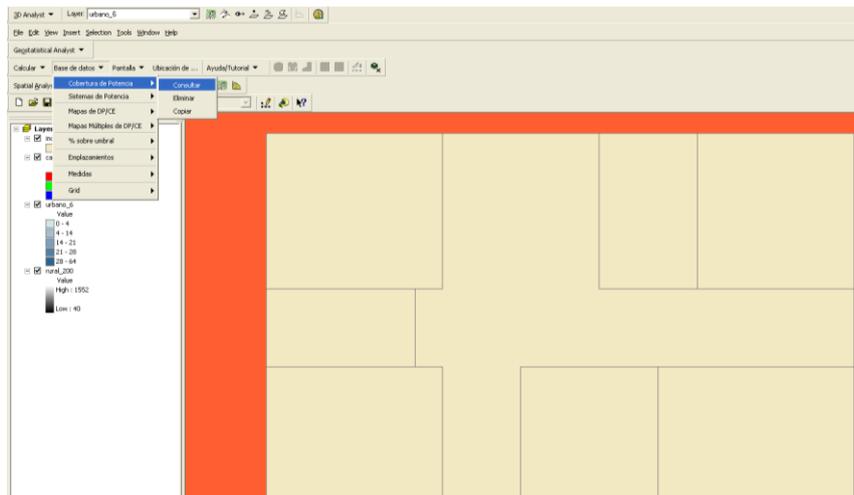


Figura 4.2. Cobertura consultada en base de datos.

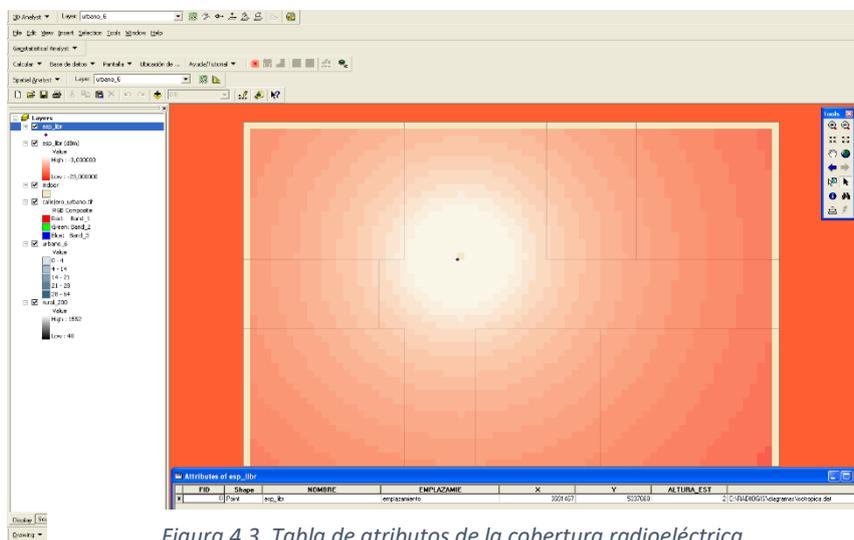


Figura 4.3. Tabla de atributos de la cobertura radioeléctrica.

En el caso de la **base de datos de emplazamientos**, el usuario, puede definir tanto emplazamientos individuales como grupos de emplazamientos aprovechando las estructuras de datos que ofrece Arcview, tal y como se observa en la Fig. 4.4:

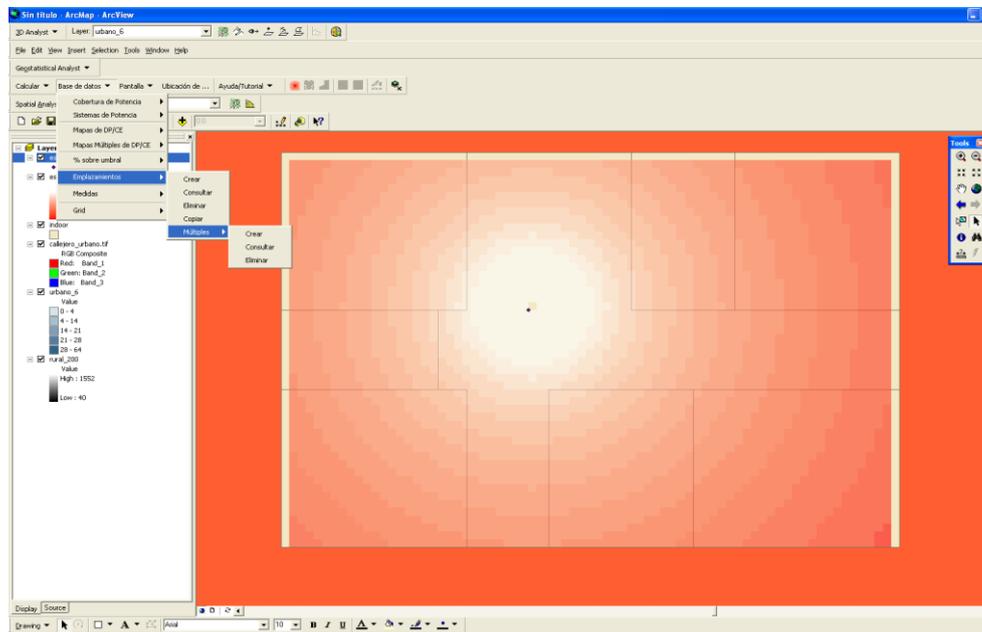


Figura 4.4: Base de datos de emplazamientos.

4.4. Modelos de propagación

Los modelos de propagación tienen como objetivo el cálculo de las pérdidas de propagación. Estas pérdidas son una primera estimación del valor de la señal en el receptor y son pérdidas en gran escala. De este modo más específico podemos considerarlas como el valor medio de pérdidas que se emplearía en un modelo estadístico. Sobre el valor estimado por el modelo existirán variaciones aleatorias provocadas por el carácter aleatorio de los elementos del canal. Por todo ello, a la pérdida estimada en el canal se le denomina pérdida básica de propagación [11], [12].

Entre el transmisor y el receptor existen numerosos elementos tales como edificios, árboles, señales de tráfico y un sinfín de obstáculos que influyen en las pérdidas.

Atendiendo al entorno de propagación, los modelos se clasifican en [12]:

- **Modelos de entornos rurales.** Estos modelos se utilizan en espacios abiertos en los que el grado de urbanización es reducido e inexistente. Estos modelos se suelen utilizar en zonas montañosas en el diseño de radioenlaces entre antenas. Modelos de este tipo son: el Bullington, el EMP, el Wilkerson, el Epstein-Peterson, el Deygout, y el modelo generalizado UITR-526.
- **Modelos de entornos urbanos macrocelulares.** Este tipo de modelos sirven para predecir las pérdidas de propagación en células grandes de los sistemas de comunicaciones móviles. Algunos de ellos sirven también para entornos rurales, en

particular para pueblos y pequeñas urbanizaciones. Estos modelos se conocen como: Okumura-Hata, Walfisch-Bertoni, Walfisch-Ikegami o COST-231.

- **Modelos de entornos urbanos microcelulares.** Al igual que los anteriores son empleados en sistemas de comunicaciones móviles, pero cuando las células son más pequeñas. El método empleado para evaluar las pérdidas suele ser diferente de los métodos de las macrocélulas. Se conocen algunos modelos de este tipo: modelo de Kaji-Akeyama, Har-Xia-Bertoni, Lanzado de rayos.
- **Modelos en entorno de interiores.** Son modelos adecuados para las células de tamaño más reducido, es decir, para aquellas situadas dentro de los edificios. Algunos son similares a los modelos de microcélulas. Los modelos en este tipo de entornos se clasifican en dos grupos: modelos empíricos y semi-deterministas.

En los siguientes apartados describimos con más detalles los modelos que se han utilizado para estimar la cobertura de los emplazamientos LoRa.

4.4.1. COST-231

Este modelo semi-empírico va a ser utilizado para calcular las pérdidas de propagación en un entorno urbano, teniendo en cuenta la disposición de edificaciones. Debido a diversas correcciones empíricas sobre modelos anteriores, se ha llegado a la conclusión de que el modelo COST-231 es más preciso [12], siendo este el modelo más utilizado para el cálculo de pérdidas de propagación en entornos urbanos.

A medida que el transmisor está más cerca a la altura del tejado de los edificios, la predicción de pérdidas será peor, llegando a tener un rendimiento muy pobre para transmisores situados por debajo de ese nivel.

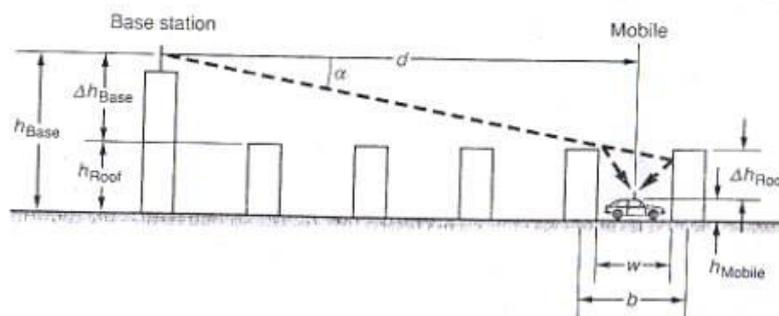


Figura 4.5: Modelo de propagación COST-231 [13].

Tal y como se observa en la Fig 4.5, este modelo supone que existe un rayo principal con ángulo de incidencia α . Este rayo se propaga desde el primer edificio donde está situada la antena transmisora, hasta el edificio más cercano al receptor. Cuando el rayo incidente se difracta en el último edificio, origina dos componentes. La primera llega directamente al receptor y la segunda lo hace tras producirse una reflexión en el edificio más cercano a él.

Para caracterizar el entorno urbano, el método utiliza los siguientes parámetros:

- h_r , expresa en metros la altura media de los edificios.
- Δh_b , es la altura de la estación base sobre los edificios, en metros.
- Δh_r , es la altura media de los edificios sobre la del móvil, en metros.
- w , es la anchura de la calle dónde está situado el móvil.
- b , es la anchura entre los centros de los edificios.
- h_m , es la altura sobre el suelo de la antena móvil, en metros.
- d , distancia entre transmisor y receptor.
- α , ángulo de incidencia del rayo principal, en grados.

Si existe visión directa entre la antena transmisora y el receptor. Las pérdidas de propagación en espacio libre se calcularán de la siguiente manera [12].

$$L_o = 32,4 + 20 \cdot \log_{10}(f(\text{MHz})) + 20 \cdot \log_{10}(R(\text{km})), \text{ si } R < 20 \text{ m} \quad (1)$$

$$L_o = 42,6 + 20 \cdot \log_{10}(f(\text{MHz})) + 26 \cdot \log_{10}(R(\text{km})), \text{ si } R \geq 20 \text{ m} \quad (2)$$

En cambio, si no existe visión directa, habrá que sumar a las pérdidas de espacio libre dos componentes más.

$$L(\text{dB}) = L_o + L_{msd} + L_{rts} \quad (3)$$

siendo:

- L_o , las pérdidas en espacio libre, en dB.
- L_{rts} , las pérdidas por difracción y dispersión del tejado a la calle, en dB.
- L_{md} , las pérdidas por difracción múltiple, en dB.

4.4.2. Xia-Bertoni

El modelo Xia-Bertoni (Fig. 4.6) es un modelo analítico que permite estimar la pérdida básica de propagación en entornos urbanos y suburbanos. El modelo Xia-Bertoni puede ser empleado teniendo en cuenta tres casuísticas en base a la disposición de la antena transmisora en el entorno [12].

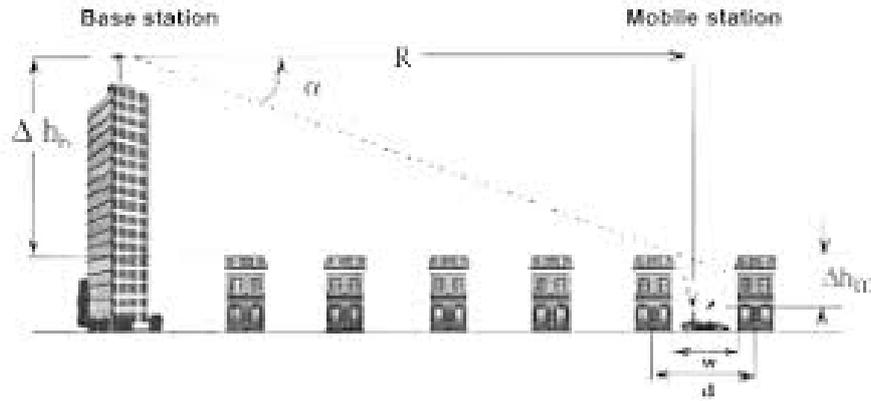


Figura 4.6: Modelo de propagación Xia-Bertoni [14].

Este modelo explica las pérdidas de propagación a partir de la suma de las pérdidas debidas al espacio libre, por la difracción múltiple debida a los edificios entre el transmisor y el receptor, y por la última difracción hasta el receptor.

$$L \text{ (dB)} = L_0 + L_{rts} + L_{md} \quad (4)$$

donde:

- L_0 , las pérdidas en espacio libre, en dB.
- L_{rts} , las pérdidas por difracción y dispersión del tejado a la calle, en dB.
- L_{md} , las pérdidas por difracción múltiple, en dB.

Se pueden distinguir **tres casuísticas** dependiendo de la disposición de la antena transmisora [19]:

- I. Si la estación base se encuentra cercana a la altura media de los edificios, las pérdidas de propagación se calcularán de la siguiente manera:

$$L = -10 \log_{10} \left(\frac{\lambda}{2\pi R \sqrt{2}} \right)^2 - 10 \log_{10} \left(\frac{\lambda}{2\pi^2 r} \left(\frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right) - 10 \log_{10} \left(\frac{d}{R} \right)^2 \quad (5)$$

siendo:

- R , la distancia entre el transmisor y el receptor, en km.
- $\theta = \tan^{-1} \left(\frac{\Delta h_m}{x} \right)$
- $r = \sqrt{(\Delta h_m)^2 + (x)^2}$
- Δh_m la diferencia entre la altura media de los edificios y la altura de la antena transmisora, en metros.
- x , la distancia entre la antena transmisora y el edificio que produce la última difracción, en metros.

- d , la distancia media de separación entre edificios.
- II. Si la estación base se encuentra por encima de la altura media de los edificios:

$$L = -10 \log_{10} \left(\frac{\lambda}{4\pi R} \right)^2 - 10 \log_{10} \left(\frac{\lambda}{2\pi^2 r} \left(\frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right) \quad (6)$$

$$- 10 \log_{10} \left[(2.35)^2 \left(\frac{\Delta h_B}{R} \sqrt{\frac{d}{\lambda}} \right)^{1.8} \right]$$

donde:

- Δh_B es la altura de la antena de la estación base con respecto a la altura media de los edificios.
 - $Q^2 = \left[(2.35)^2 \left(\frac{\Delta h_B}{R} \sqrt{\frac{d}{\lambda}} \right)^{1.8} \right]$, siendo Q el valor al cual convergen las pérdidas por difracción múltiple (L_{rts}) cuando se considera incidencia de onda plana sobre el primer edificio y un número grande de éstos últimos, tal y como enuncian Walfisch-Bertoni en [Wal88].
- III. Si la estación base se encuentra por debajo de la altura media de los edificios:

$$L = -10 \log_{10} \left(\frac{\lambda}{2\pi R \sqrt{2}} \right)^2 - 10 \log_{10} \left(\frac{\lambda}{2\pi^2 r} \left(\frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right) \quad (7)$$

$$- 10 \log_{10} \left\{ \left[\frac{d}{2\pi(R-d)} \right]^2 \frac{\lambda}{\sqrt{(\Delta h_B)^2 + d^2}} \left(\frac{1}{\phi} - \frac{1}{2\pi + \phi} \right)^2 \right\}$$

siendo:

- $\phi = \tan^{-1} \left(\frac{\Delta h_B}{d} \right)$

4.5. Modelo Digital del Terreno de edificios de Cartagena

4.5.1. Obtención del Mapa de Cartagena

Se requiere obtener el mapa del Municipio de Cartagena para el cálculo de cobertura radioeléctrica que va a proporcionar cada estación base LoRa, las cuales están distribuidos de tal manera que abarcan gran parte del municipio. Para ello, se accede a la sede electrónica del Catastro, donde a partir de una firma electrónica propia del usuario, se procederá a la descarga de la capa vectorial de Cartagena (Shapefile).

En primer lugar, tras acceder al portal de la sede electrónica del Catastro aparecerá el siguiente entorno (Fig. 4.7):



Figura 4.7: Entorno de la Sede electrónica del Catastro.

Seguidamente, se hace click en la opción “*Difusión de datos Catastrales*”, la cual nos reedirá a una ventana (Fig. 4.8) con varias posibilidades a descargar, entre ellas “*Descarga de cartografía vectorial*”

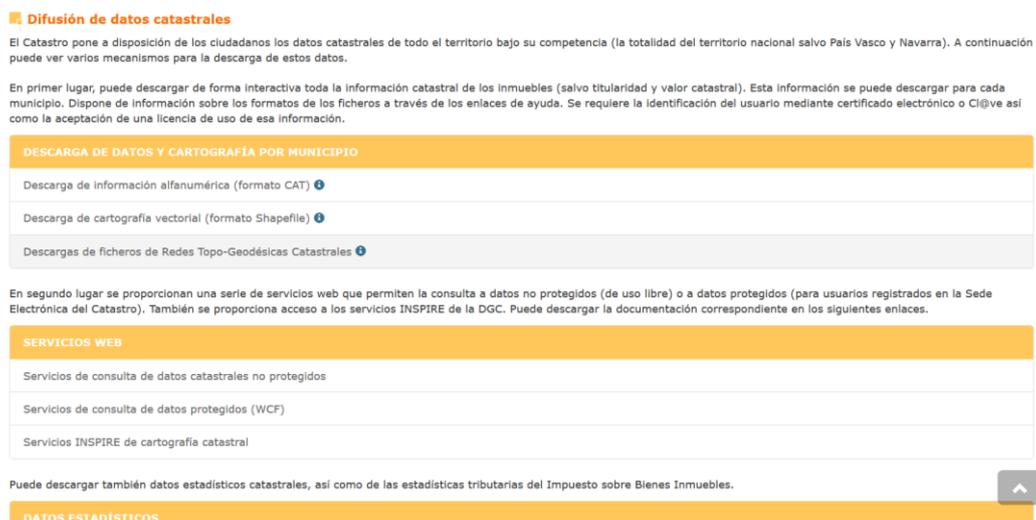


Figura 4.8: Difusión de datos Catastrales.

Tras acceder a dicho enlace, se debe verificar la firma electrónica del usuario para proceder a la descarga de la capa vectorial deseada.

De este modo, se podrá elegir la provincia y el municipio (Fig. 4.9), en nuestro caso, se escogerá Municipio ‘*Cartagena*’ y Tipología ‘*Urbana con Historia*’ para poder recurrir a una mejor visualización de la distribución geográfica de Cartagena.

Provincia:

Municipio:

Tipología: Urbana sin Historia Rústica sin Historia
 Urbana con Historia Rústica con Historia

Figura 4.9: Elección de capas vectoriales.

Finalmente, se procederá a la descarga de todas las capas disponibles, con la finalidad de integrar la capa vectorial 'CONSTRU' en el entorno de ArcMap.

4.5.2. Cálculo de la altura de los edificios

En el shapefile mencionado anteriormente, existe una columna denominada 'CONSTRU', la cual contiene los distintos volúmenes construidos en Cartagena, recopilados en números romanos y en combinación con varios códigos y signos. Se trata de una gran cantidad de información dentro de cada celda. Por ejemplo, -I+II+TZA describe una parte de un edificio que tiene una planta subterránea, dos plantas sobre rasante y, en lo más alto, una terraza (TZA).

De este modo, se deberán interpretar correctamente los datos para el posterior cálculo de altura de los edificios de una manera efectiva.

Una vez expuesto en que consiste la capa 'CONSTRU', se procede a calcular el número de pisos que presenta cada edificación. Para ello, se comentan una serie de pasos a seguir:

1. Abrir entorno de ArcMap y cargar la capa shapefile 'CONSTRU' como se observa en la Figura 4.10.

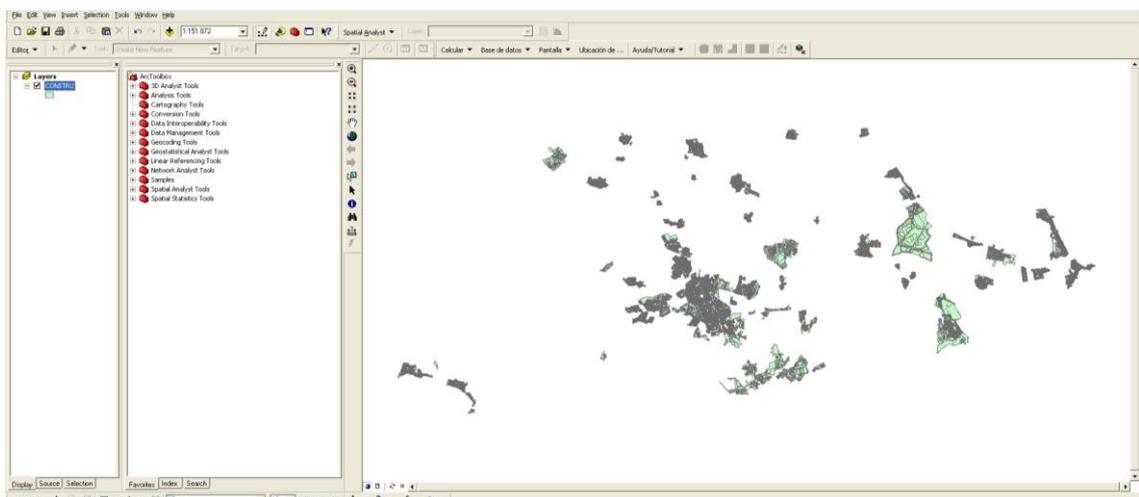


Figura 4.10: Municipio de Cartagena cargado en el entorno de ArcMap.

2. Dentro de la tabla de atributos de dicha capa, se crea un nuevo campo (denominado pisos), para ello, **Options** → **Add Field**. A continuación, emergerá la ventana que aparece en la figura 4.11.

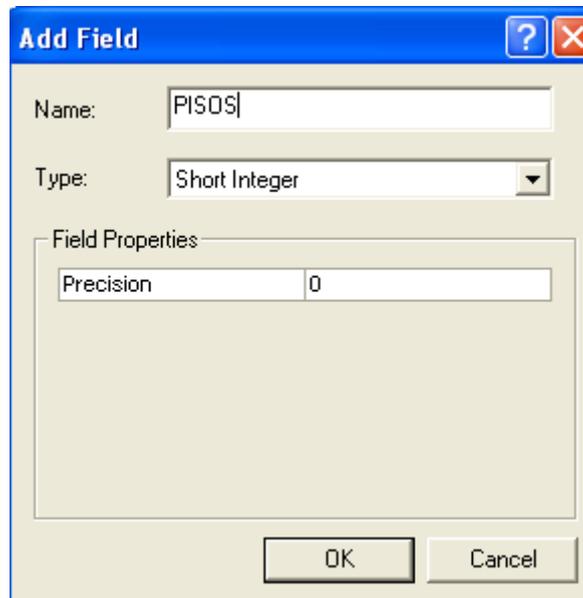


Figura 4.11: Ventana Add Field.

3. A continuación, se cargará un código programado mediante el lenguaje técnico de Visual Basic, a partir del cual, se generará la columna “PISOS” que dará como resultado el número de plantas correspondientes a cada construcción. **Calculate Values** → **Field Calculator** (Fig. 4.12).

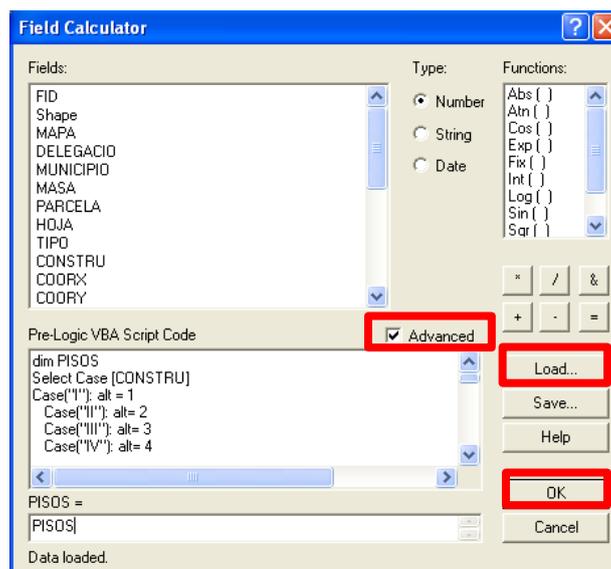


Figura 4.12: Ventana Field Calculator para el cálculo de pisos.

4. Para formalizar los cambios que se han realizado no olvidar darle click a 'Stop Editing'.
5. Finalmente, una vez generada la columna 'PISOS' se va a calcular las alturas correspondientes a cada edificio. Para ello, se tiene en cuenta que la altura media de cada planta es equivalente a un valor medio de 3 metros. De este modo, se multiplicará dicho valor por la columna 'PISOS', como se observa en la Figura 4.13.

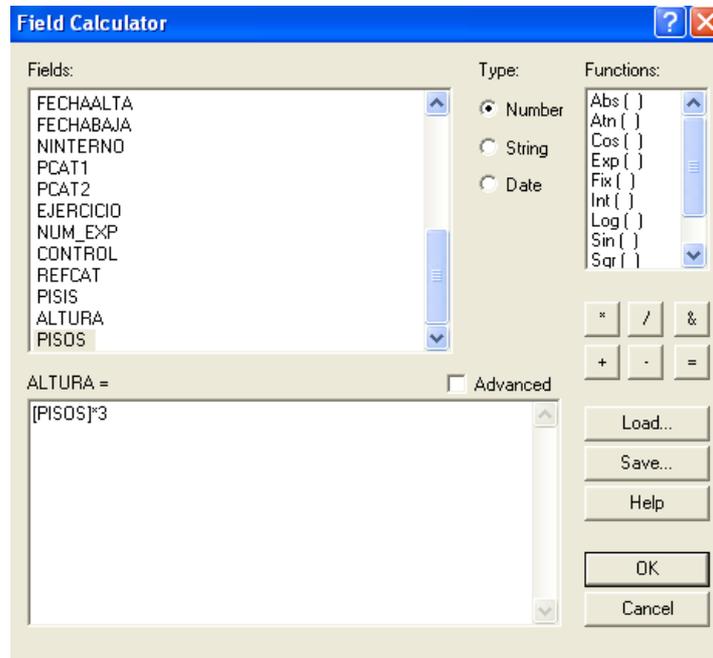


Figura 4.13: Cálculo de alturas de los edificios.

Una vez calculadas las alturas, se puede observar en la figura 4.14 una vista 3D de Cartagena mediante la extensión 'ArcScene'.



Figura 4.14: Municipio de Cartagena en 3D.

4.6. Cálculo de Coberturas red LoRa en Cartagena

En este apartado, se procede al cálculo de la cobertura (Figura 4.15) desde los emplazamientos LoRa. Para ello, seleccionamos **Calcular -> Cobertura de Potencia -> Datos Cobertura -> Nueva:**

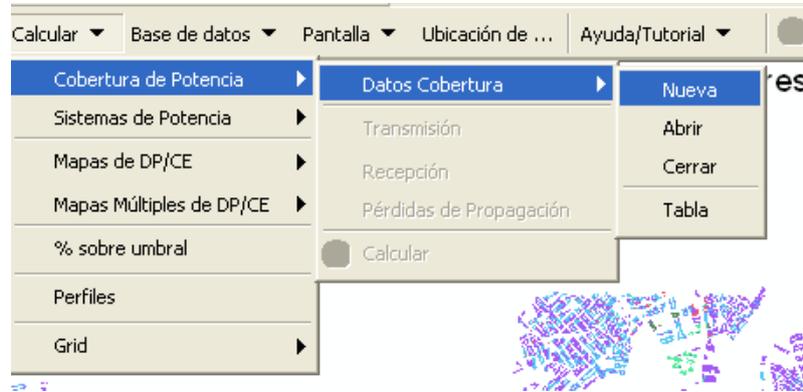


Figura 4.15: Cálculo de cobertura.

Una vez puesto nombre a la cobertura que se quiere generar se procede a la selección de parámetros para la simulación. Estos parámetros van a depender del tipo de Gateway, además es necesario realizar un Balance de enlaces ascendente y descendente para saber cuál es el más desfavorable (Tabla 4). En función de los resultados obtenidos, se escogió aquel enlace cuyas pérdidas máximas compensables son menores, el ascendente en este caso. De esta forma, a la hora de realizar los cálculos de cobertura deberemos utilizar como parámetros de transmisión (PIRE, Ganancia de antena, ...) los de la estación móvil y como parámetros de recepción (ganancia de antena, pérdidas en cables, sensibilidad del receptor) los de la estación base

Transmitter	Uds	Relation	DL (BS->Sensor)	UL (Sensor->BS)
Power Tx	dBm	Ptx	30,0	14,0
Gain Tx	dBi	Gtx	3,0	3,0
Losses Tx	dB	Ltx	2,0	0,0
PIRE	dBm	Ptx + Gtx - Ltx	31,0	17,0
Channel				
Fade Margin	dB	FM	7,7	7,7
Interference Margin	dB	IM	3,0	3,0
Receiver				
Gain Rx	dBi	Grx	3,0	3,0
Losses Rx		Lrx	0,0	2,0
Sensitivity	dBm	S	-141,0	-141,0
Lmax	dB	PIRE-FM-IM+Gr-Lr-S	164,3	148,3

Tabla 4: Estudio de pérdidas.

A continuación, se van a mostrar dos coberturas generadas a partir de los gateways instalados tanto en el Barrio Peral (Fig. 4.16) como en la Alameda (Fig. 4.17).

Datos a utilizar en RADIOGIS:

Transmisión:

- Potencia isotrópica radiada equivalente PIRE: 17 dBm

Recepción:

- Ganancia Gr=3 dB
- Pérdidas Lr=2 dB
- Sensibilidad S. Estimar la cobertura para el peor caso. S = -141 dBm

Pérdidas de propagación.

Estación Base:

- Emplazamiento:

La estación base está ubicada en las siguientes coordenadas:

Nombre	Coordenada (X)	Coordenada (Y)
pru_BP	677203,88	4166044,07

- Altura: 8 m
- Diagrama de radiación: isotrópico

Estación móvil:

- Altura: 1 m
- Diagrama de radiación: isotrópico

Parámetros comunes:

- Frecuencia: 868 MHz
- Método: Cost-231
- Resolución: 2 metros

Zona de cálculo

- Radio: 1000 m

Seguidamente, se muestra la cobertura generada (Fig. 4.16).

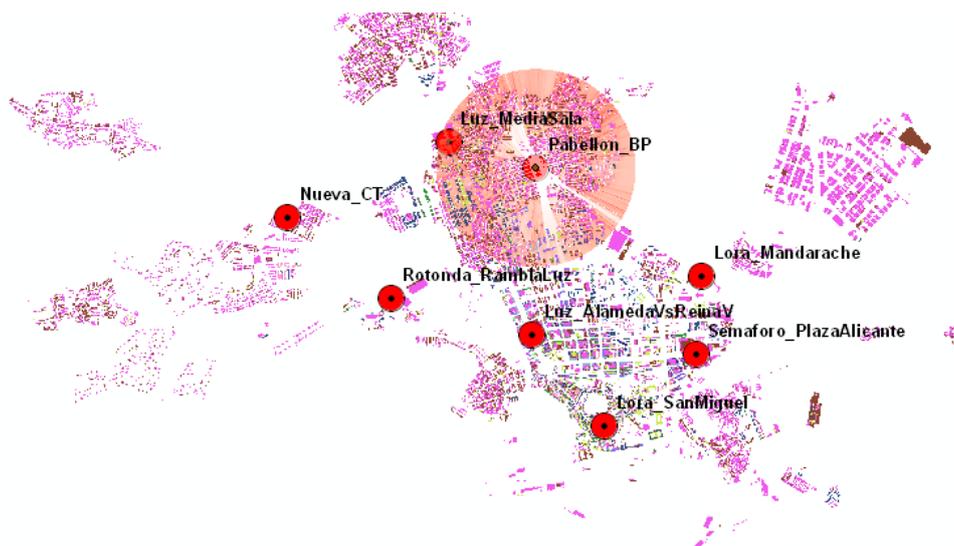


Figura 4.16: Cobertura generada en el Barrio Peral.

Siguiendo la misma metodología expuesta anteriormente, se calcula la cobertura del sistema LoRa situado en la Alameda (Fig. 4.17).

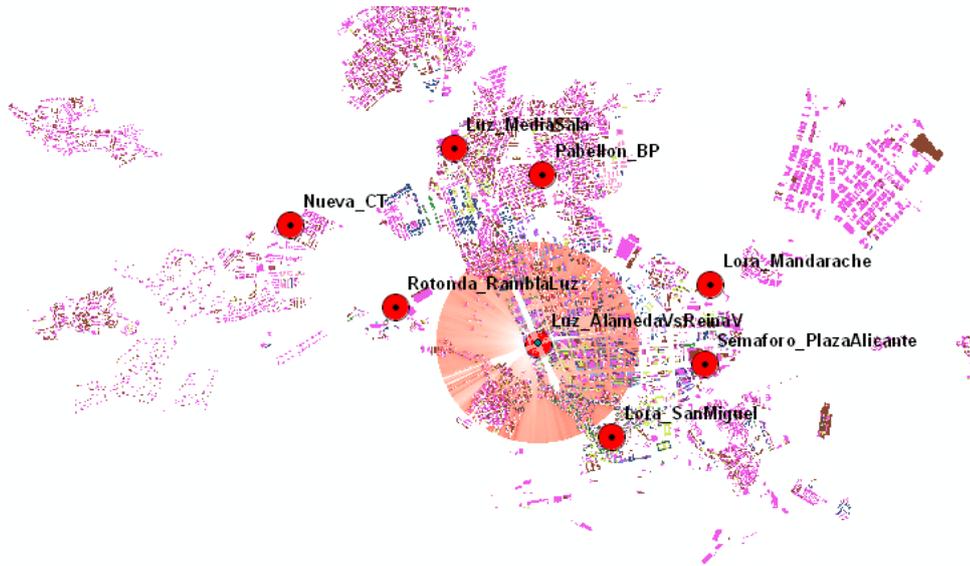


Figura 4.17: Cobertura generada en la Alameda.

Tras el cálculo de coberturas, se requiere añadir las calles pertenecientes a este municipio, para que el usuario sea capaz de reconocer a simple vista donde se está obteniendo cobertura.

Para ello, se establece un mapa mundial como base (Fig. 4.18). Los mapas base sirven como mapa de referencia en el que se va a superponer datos de capas operativas, realizar análisis y visualizar información geográfica.

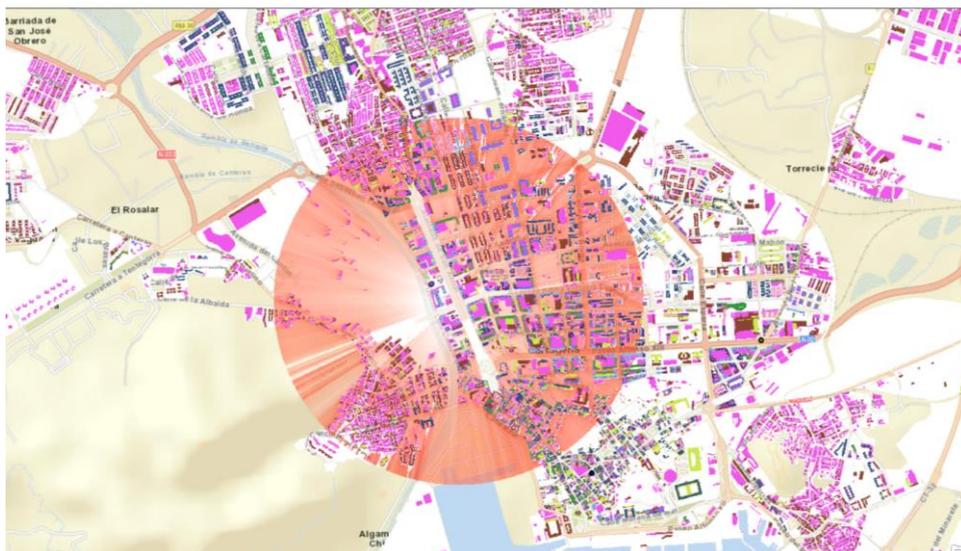


Figura 4.18: Mapa base con el nombre de las calles del municipio.

Capítulo 5.

Campaña de medidas y comparación de modelos

5.1. Recogida de datos

Tras visualizar la información recibida en el servidor de TTN (ver capítulo 2), se recogen estos datos en tiempo real para un posterior procesamiento de ellos. Asimismo, se ha realizado una clase en Java a partir de HttpClient, cliente con los métodos REST habituales. Esta clase puede ser encontrada en el anexo B de este proyecto fin de máster.

La URI nos permite modificar o manipular los objetos en REST. Esta es el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado.

Para la transferencia de datos en este sistema, se aplican acciones concretas (POST, GET, PUT y DELETE) sobre los recursos.

El servicio HttpClient devuelve la respuesta en formato JSON, tal y como se observa en la Fig. 5.1. Es por ello, que se ha elaborado una segunda clase (punto 2 del Anexo B) que permitiese decodificar este fichero, además de recoger únicamente aquellos campos que nos eran de utilidad, como, por ejemplo: RSSI, ruido y geolocalización.



```
medidas - Notepad
File Edit Format View Help
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
8:42Z"}},{"txInfo":{"dr":0,"frequency":86850000}}}}
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
03:07Z"}},{"txInfo":{"dr":0,"frequency":86850000}}}}
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
12"},{"gatewayID":"000024b0b0311d2","lorasnr":-8.8,"location":{"altitude":8,"latitude":0,"longitude\
{"result":{"type":"join","payloadJSON":{"applicationID":"1","applicationName":"app-test","devAddr":"e7c
{"result":{"type":"error","payloadJSON":{"applicationID":"1","applicationName":"app-test","devEUI":"008
{"result":{"type":"uplink","payloadJSON":{"adr":false,"applicationID":"1","applicationName":"app-test",\
```

Figura 5.1: Medidas en formato JSON.

5.2. Interfaz gráfica y resultados

A continuación, se detalla la interfaz gráfica (punto 3 del Anexo B) desarrollada con la finalidad de realizar un análisis de resultados cotejando los datos recogidos del propio entorno de RADIOGIS y los medidos propiamente con el sistema LoRa.

En primer lugar, para iniciar la aplicación de forma segura, es necesario introducir una contraseña previamente establecida (Fig. 5.2). En caso de integrar una contraseña incorrecta, emergerá una ventana indicándolo.

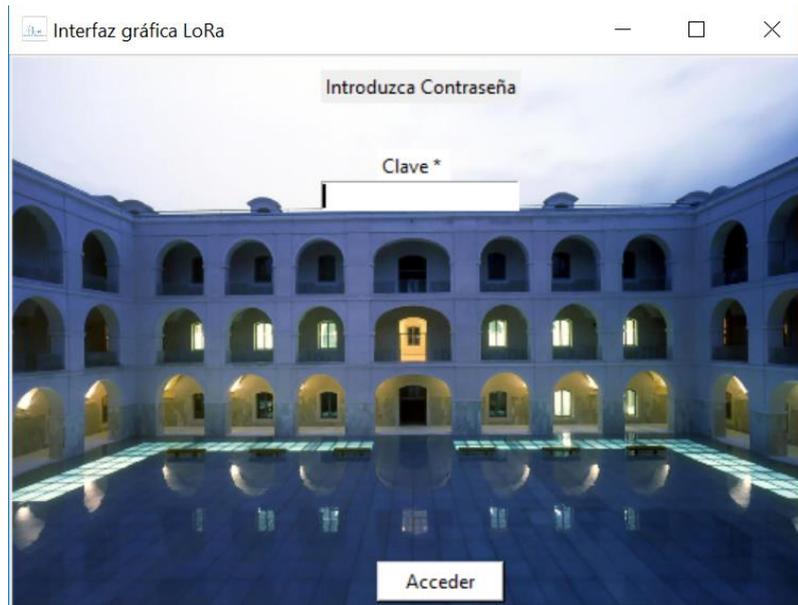


Figura 5.2: Ventana de inicio de la Interfaz gráfica.

Una vez introducida la clave correcta, la interfaz redirige al usuario a otra ventana, la cual le permitirá integrar tanto las medidas obtenidas en RADIOGIS como por el sistema LoRa (Fig. 5.3).



Figura 5.3: Interfaz.

En la esquina inferior derecha, se pueden observar 3 botones:

- Reset RADIOGIS: Borra el archivo que se haya añadido en RADIOGIS measurements.
- Reset LoRa: Borra el documento añadido en LoRa measurements.
- Help: Redirige al usuario a la página web de SICOMO.

Una vez termine la interfaz de realizar los cálculos solicitados, aparecerá una ventana donde el usuario puede elegir el lugar de exportación del fichero Excel con los resultados de la ejecución. El formato será previamente establecido.

En la Fig. 5.4 pueden verse las medidas tomadas a partir del Gateway de la calle San Miguel en Cartagena (Fig. 5.4).

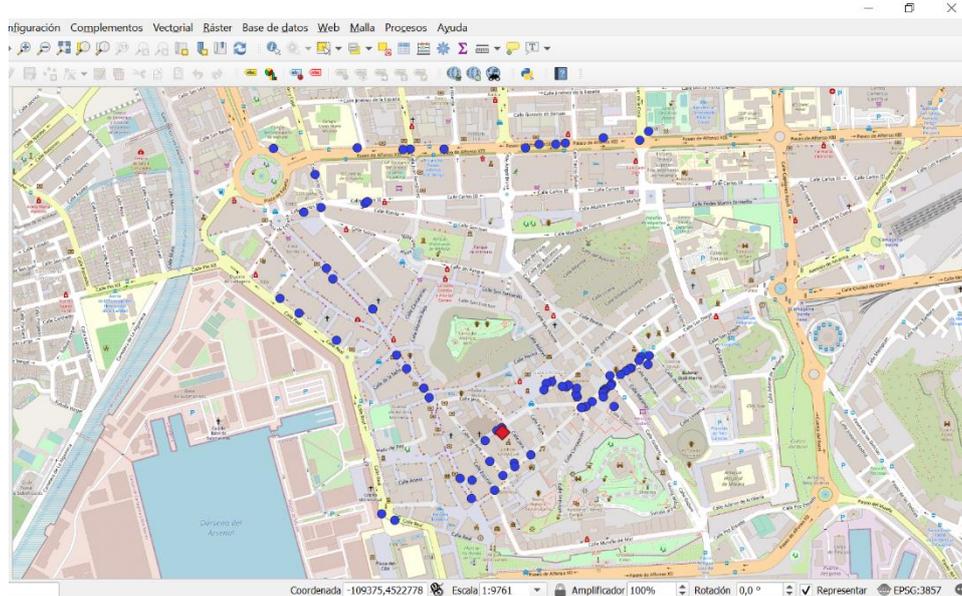


Figura: 5.4: Medidas tomadas a partir del Gateway de San Miguel.

RADIOGIS genera la cobertura de estos sistemas mediante dos modelos de propagación (COST-231 y Xia-Bertoni). En la Fig. 5.5 se observan los resultados obtenidos con el modelo de propagación COST-231. En cambio, en la Fig. 5.6 con Xia-Bertoni. Se han representado las rectas de regresión de las medidas y de los modelos de propagación utilizando el método de ajuste FI (Floating Intercept) [12].

$$RSSI(dBm) = \alpha + n10 \log_{10} d + X\sigma \quad (8)$$

donde:

α (en dBm) y n (pendiente) son parámetros de ajuste.

d es la distancia entre transmisor y receptor en metros.

$X\sigma$ una variable aleatoria gaussiana de media cero y desviación estándar σ en dBm.

Se ha procedido a estudiar el error medio (ME), error cuadrático medio (RMSE) y la desviación estándar (Std) de cada uno de ellos.

- El error cuadrático medio compara un valor predicho con un valor real [15].
- La desviación estándar (Std) es la medida de dispersión más común, que indica qué tan dispersos están los datos con respecto a la media. Mientras mayor sea la desviación estándar, mayor será la dispersión de los datos [16].

$$ME = \frac{1}{m} \sum_{i=1}^m (RSSI_{Pi} - RSSI_{mi}) \quad (9)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (RSSI_{Pi} - RSSI_{mi})^2} \quad (10)$$

$$Std = \sqrt{RMSE - ME^2} \quad (11)$$

Siendo:

- $RSSI_{Pi}$ = medida estimada en dBm de la potencia con RADIOGIS.
- $RSSI_{mi}$ = medida real en dBm de la potencia con el sistema LoRa.
- m = número total de muestras medidas.

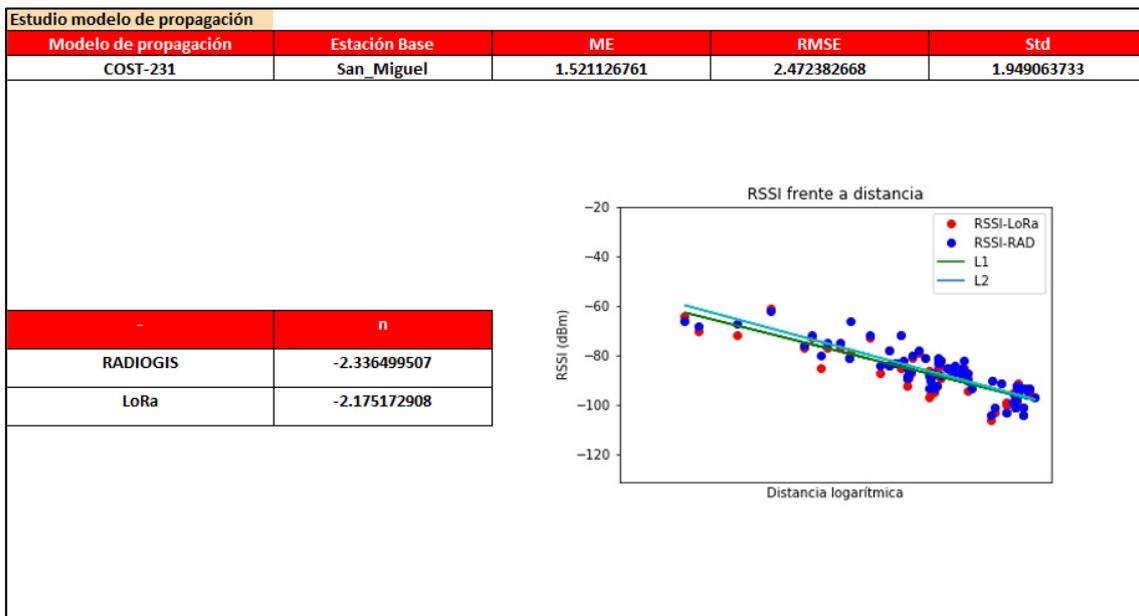


Figura 5.5: Resultados modelo COST-231.

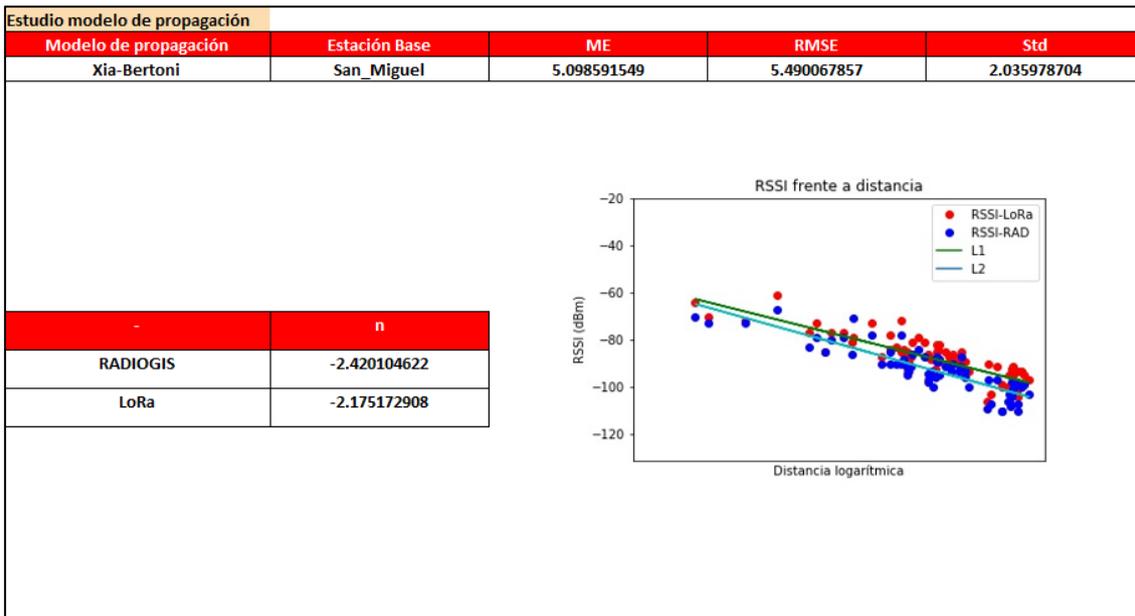


Figura 5.6: Resultados modelo Xia -Bertoni.

De los resultados, se desprende que el modelo COST-231 se ajusta mejor que el modelo de Xia-Bertoni. La pendiente de la recta de regresión para el COST-231 es de -2.34 y la de Xia-Bertoni es de -2.42, siendo la de las medidas de -2.17. Además, tanto la media como la desviación típica del error de las medidas respecto al modelo COST-231 se ajustan mejor que para el modelo de Xia-Bertoni. Estos resultados constatan que para una estación base situada por encima de la altura media de los edificios es más adecuado aplicar el modelo COST-231 para estimar la cobertura radioeléctrica. La figura 5.7 muestra un resumen realizado en Power BI de lo comentado anteriormente.

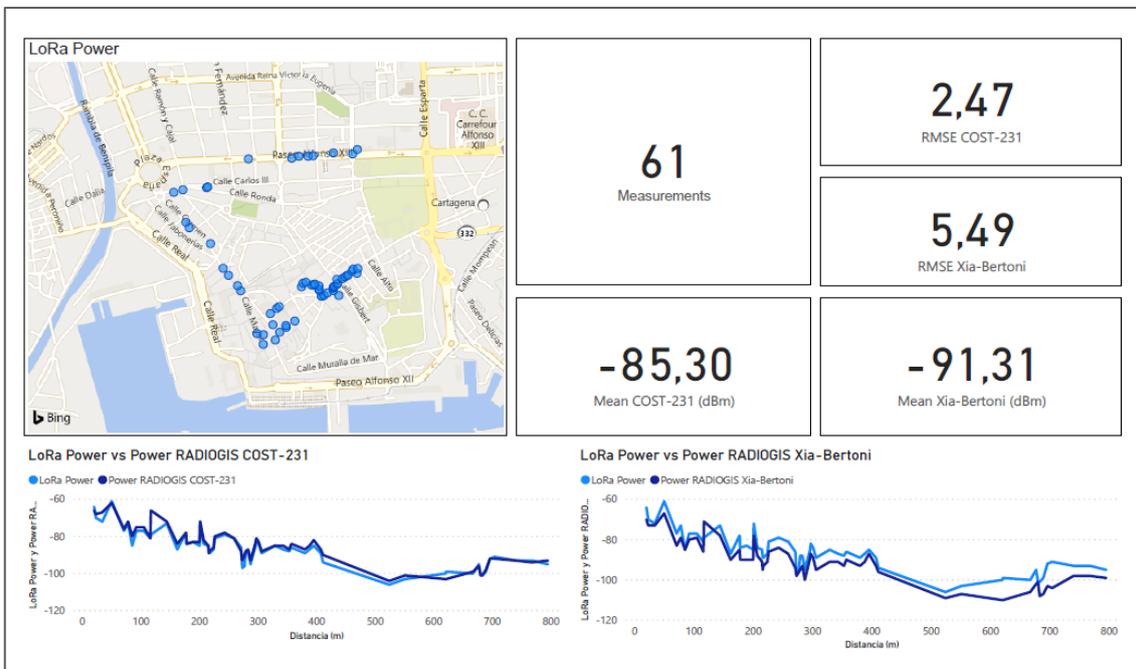


Figura 5.7: Resultados COST-231 y Xia- Bertoni.

Capítulo 6.
Conclusiones y líneas futuras

6.1. Conclusiones

Tras finalizar este proyecto podemos dar por cumplido el objetivo de éste, el cual era un análisis teórico y experimental de una red LoRa en la ciudad de Cartagena. A lo largo de este documento se ha detallado cómo se debe configurar el equipamiento LoRa y se ha presentado una campaña de medidas realizada en la ciudad de Cartagena.

Con la herramienta RADIOGIS, se han realizado diversas simulaciones de coberturas con los modelos COST-231 y Xia-Bertoni para cotejar los resultados obtenidos con las medidas reales llevadas a cabo. Las comparaciones se hicieron a partir de la interfaz gráfica desarrollada en Python, que nos permitió obtener diversas gráficas relacionando los diversos modelos entre sí.

Los resultados obtenidos muestran que, si la estación base se encuentra por encima de la altura media de los edificios, el modelo de propagación COST231 es más adecuado que el de Xia-Bertoni para estimar la cobertura radioeléctrica. En concreto, para la estación de San Miguel de nuestro sistema LoRa, las medias del error son 1.52 y 5.1, respectivamente, y las desviaciones típicas del error 1.95 y 2.04, respectivamente.

6.2. Líneas futuras

A partir de este proyecto se pueden hablar de varias líneas de trabajo derivadas de este.

La primera de ellas podría ser realizar una campaña de medidas sistemática a partir de las estaciones base situadas en los distintos emplazamientos dispuestos por el ayuntamiento de Cartagena, con la finalidad de ajustar los distintos modelos.

Otro proyecto derivado podría ser una recogida masiva de datos a partir de la configuración de distintos sensores (proximidad, temperatura, humedad, etc.) y reportando estos a una página web creada.

Finalmente, se podría realizar una colaboración con el ayuntamiento para optimizar el tráfico a partir de semáforos inteligentes. Con la ayuda del sistema LoRaWAN se monitorizarían los datos generados por cada uno de los elementos de la red, en este caso semáforos. Esta red estaría compuesta por una serie de controladores que se instalarán en cada uno de los nodos finales y por otro lado uno o varios concentradores (Gateway) que se encargarán de la recopilación de datos.

Capítulo 7.

Bibliografía

BIBLIOGRAFÍA

- [1] <https://www.fundacionendesa.org/es/recursos/a201908-smart-city>
- [2] <https://www.powerdata.es/big-data>
- [3] «¿Qué es LoRa?», Automatización del Internet de las Cosas, S.L. [En línea]. Disponible en: <http://gestion.alfaiot.com:8069/blog/our-blog-1/post/que-es-lora-2>. [Accedido: 03-mar-2019].
- [4] Fco. Javier Ponce Juan, José María Molina García-Pardo, José Víctor Rodríguez, Leandro Juan Llácer, "RAGIS: herramienta de gestión y de cálculo de cobertura radioeléctrica basada en sistemas de información geográfica", Asociación Hispanoamérica de Centros de Investigación y Empresas de Telecomunicaciones, vol. 101, pp. 59-63, Enero 2005.
- [5] http://downloads.esri.com/support/documentation/ao_/What_is_ArcGIS_Sep2005.pdf
- [6] <https://www.multitech.com/brands/multiconnect-conduit>
- [7] <https://www.multitech.com/models/99999210LF>
- [8] <https://www.multitech.com/documents/publications/data-sheets/86002182.pdf>
- [9] <https://www.sap.com/spain/trends/internet-of-things.html>
- [10] <https://www.thethingsnetwork.org/>
- [11] José María Hernando Rábanos, Comunicaciones Móviles, Ed. Centro de estudios Ramón Areces, 1997.
- [12] José María Hernando Rábanos, Transmisión por Radio, Ed. Centro de estudios Ramón Areces, 1997.
- [13] <http://kilyos.ee.bilkent.edu.tr/~microwave/programs/wireless/prop/costWI.htm>
- [14] Sergio Gallego Martínez, " Comparación de modelos teóricos para el análisis de las pérdidas por difracción múltiple de la señal radioeléctrica con medidas experimentales en entornos urbanos ", p. 18.
- [15] <https://acolita.com/que-es-el-error-cuadratico-medio-rmse/>
- [16] <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/data-concepts/what-is-the-standard-deviation/>
- [17] https://www.icog.es/TyT/files, Libro_SIG.pdf, p. 60.
- [18] «Funcionamiento | Sistemas de Información Geográfica». [En línea]. Disponible en: http://sig.cea.es/como_funcionan. [Accedido: 18-abr-2019].
- [19] «¿Qué es un SIG?», SISTEMAS DE INFORMACIÓN GEOGRÁFICA, 07-jun-2010. .
- [20] «Qué es ArcMap—Ayuda | ArcGIS for Desktop». [En línea]. Disponible en: <http://desktop.arcgis.com/es/arcmap/10.3/main/map/what-is-arcmap-.htm>. [Accedido: 18-abr-2019].
- [21] http://concurso.cnice.mec.es/cnice2006/material121/unidad2/td_sig.htm

Anexo A

A.1. ¿Qué es ArcGIS?

ArcGIS es un “software” de Sistema de Información Geográfica diseñado por la empresa californiana *Environmental Systems Research Institute* (ESRI) para trabajar a nivel multiusuario. Permite recopilar, organizar, administrar, analizar, compartir y distribuir información geográfica.

Este sistema está disponible en cualquier lugar a través de navegadores Web, dispositivos móviles como smartphones y equipos de escritorio. Se compone de cuatro partes [5]:

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

ArcGIS está basado en ArcObjects™, una librería modular común que comparten todos los componentes software del GIS. Incluye una amplia variedad de componentes programables, que van desde objetos como geometrías individuales, hasta mapas enteros que interactúan con documentos ArcMap. Los desarrolladores de GIS tienen a su disposición la librería ArcObjects para crear plataformas alternativas de software GIS, basándose en los cuatro productos anteriores. En las figuras A.1 y A.2 se pueden ver más detalladamente todos estos productos.



Figura A.1: ArcObjects es una librería común para los cuatro componentes ArcGIS [5].

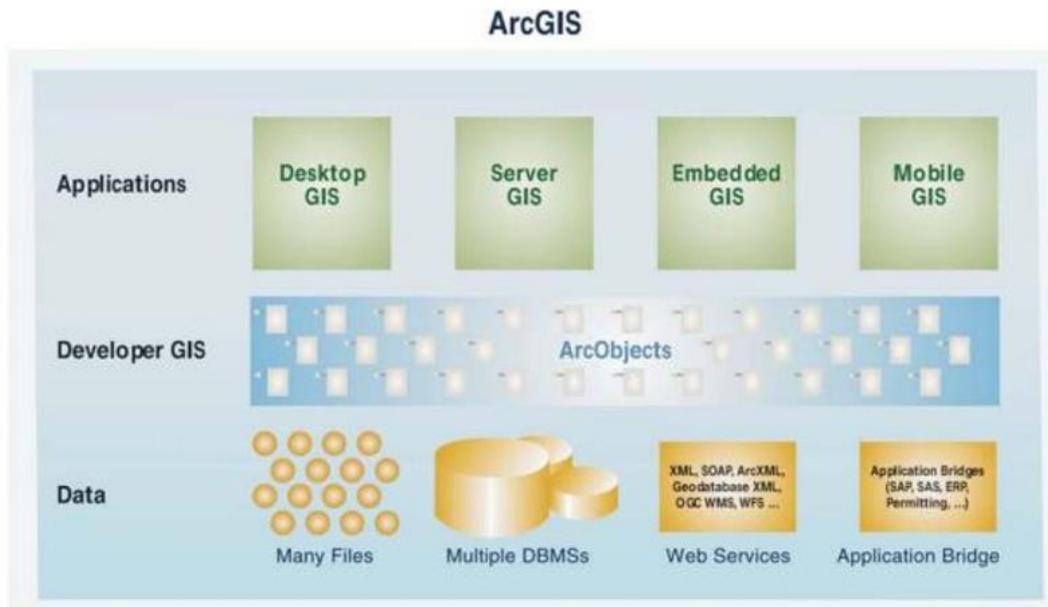


Figura A.2: ArcObjects es empleado por desarrolladores GIS en conjunto con los otros cuatro productos [17].

A.2. ¿Qué es un Sistema de Información Geográfica?

Los Sistemas de Información Geográfica (SIG) (Fig. A.3) son una tecnología que forma parte del ámbito más extenso de los Sistemas de Información. El contexto general en el que surgen es el de la “sociedad de la información”, en la que resulta esencial la disponibilidad rápida de información, para resolver problemas y contestar a las preguntas de modo inmediato [17] .

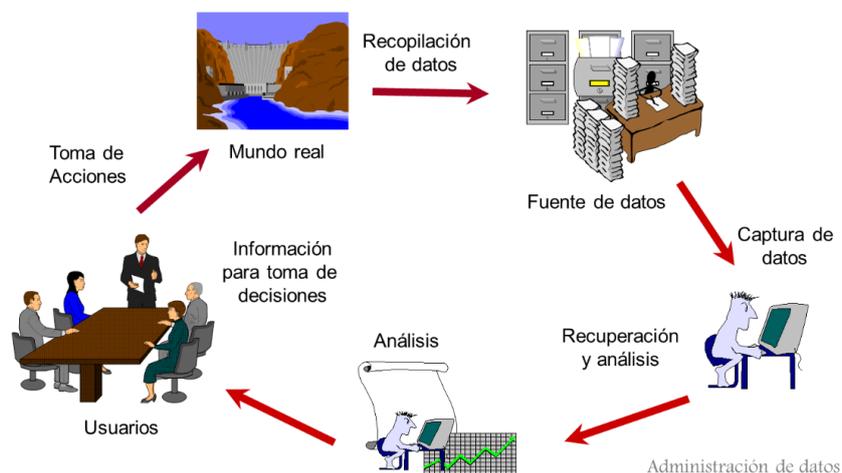


Figura A.3: Ciclo de vida de un SIG [17].

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

Miles de organizaciones de muchos sectores diferentes emplean ArcGIS, que es el sistema de información geográfica desarrollado por la empresa ESRI.

En general se usa ArcGIS como SIG porque permite:

- Resolver gran cantidad de problemas.
- Tomar mejores decisiones.
- Planificar adecuadamente.
- Utilizar los recursos de una manera más eficiente.
- Administrar y ejecutar las operaciones de una forma más eficaz.
- Comunicar de forma más efectiva.
- Educar y motivar a otros.

A.2.1. Funciones de un SIG

Los SIG operan como una base de datos geográfica asociada a los objetos existentes en un mapa digital, dando respuesta a las consultas interactivas de los usuarios, analizando y relacionando diferentes tipos de información con una sola localización geográfica.

Las principales funciones de un Sistema de Información Geográfica son [18] :

- **Entrada de datos.** Debe tener capacidad de recibir datos de distintas maneras (digitalización directa, entrada de coordenadas en archivos digitales y teledetección).
- **Output.** Es la representación gráfica y cartográfica en papel y digital (no es el eje central de un SIG).
- **Manipulación de los datos.** Debe proveer herramientas para el manejo de información; por ejemplo, funciones para la transformación matemática de coordenadas, reformato e importación y exportación de datos.
- **Análisis.** Es la función más importante. Se trata de usar la información disponible para producir nueva información.

A.2.2. Cuestiones a las que puede responder un SIG

Algunas de las preguntas que pueden responder un SIG son, ordenadas de mayor a menor complejidad [19]:

1. Localización. Apuntando con el cursor sobre la pantalla se puede obtener información sobre lo que hay en un lugar determinado (por ejemplo, cuánta población escolar habita en una sección censal). Se trata simplemente realizar una consulta en la que es necesario relacionar la información cartográfica con la base de datos de atributos.

2. Condición. A partir de unas condiciones previamente especificadas, el sistema debe indicar dónde se cumplen o no esas condiciones (por ejemplo, indicar dónde encontramos un lago para pescar situado a menos de 50 kilómetros de nuestra casa y rodeado de bosques).

3. Tendencias. En esta pregunta lo fundamental es la comparación entre situaciones temporales distintas, si bien para ello se pueden incluir condiciones (por ejemplo, cuántas hectáreas de

naranjos se encuentran a menos de 200 metros de una carretera, en una fecha dada y veinte años después). Ello supone trabajar con varios mapas de la misma zona referidos a fechas distintas.

4. Rutas. El sistema puede calcular el camino óptimo (el más corto, más barato o más rápido) entre dos puntos a través de una red (por ejemplo, entre el lugar donde se ha producido una catástrofe y el hospital más próximo).

5. Pautas. Ciertas regularidades espaciales pueden ser detectadas con la ayuda de un SIG (por ejemplo, qué patrones de distribución espacial presentan los casos de cáncer en torno a una central nuclear en la que se ha producido un accidente).

6. Modelos. Se pueden generar modelos para simular el efecto que producirían posibles fenómenos o actuaciones en el mundo real (por ejemplo, que sucedería si se construyera un nuevo tramo de autopista o si se produjera un aumento de dos metros en el nivel de las aguas del mar).

A.3. DESKTOP GIS

El Desktop GIS (-escritorio SIG-) es el nivel primario desde el que los profesionales compilan, crean y utilizan la información geográfica. Si usamos GIS en un proyecto o en un entorno multiusuario, se pueden usar estas tres aplicaciones: ArcCatalog, ArcMap y ArcToolbox.

- La aplicación de **ArcCatalog** sirve para establecer diseños base, recordar y ver metadatos.
- **ArcMap** se utiliza para todas las tareas de mapeo y edición, así como el análisis basado en mapas.
- **ArcToolbox** se usa para la conversión de datos y geoprocesamiento.

A.3.1. ArcMap

ArcMap es la herramienta principal de ArcGIS para escritorio, ésta cuenta con un entorno visual el cual permite realizar una representación cartográfica, edición, análisis y administración de datos [20].

Algunas de las funcionalidades de ArcMap son:

- Trabajar con mapas.
- Imprimir mapas.
- Compilar y editar Datasets.
- Utilizar el geoprocesamiento para automatizar y realizar análisis.
- Organizar y administrar geodatabases y documentos de ArcGIS.
- Publicar documentos de mapa como servicios de mapas mediante ArcGIS for Server.
- Permite administrar usuarios.

El entorno de ArcMap está distribuido de la siguiente manera (Fig. A.4):

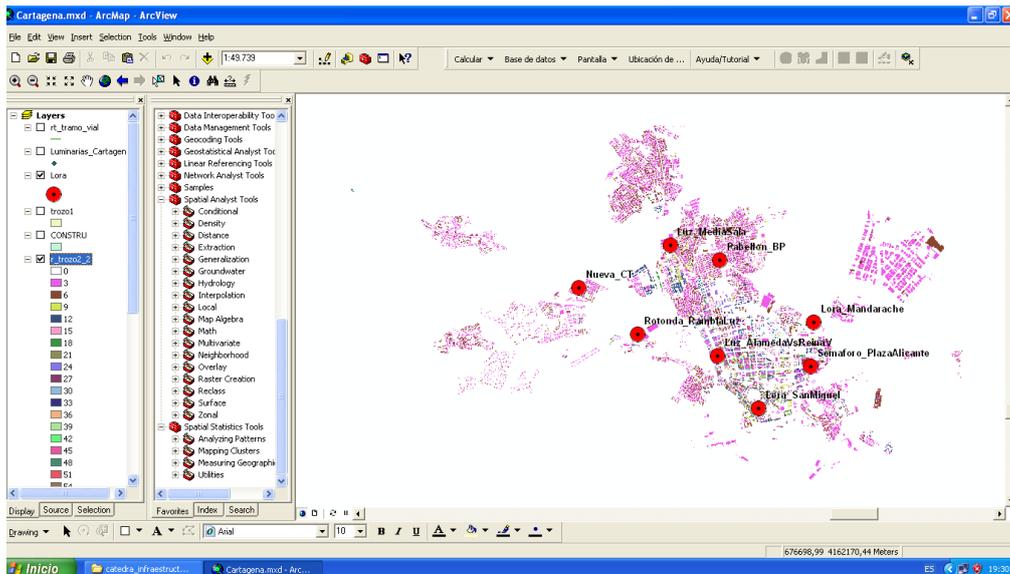


Figura A.4: Interfaz gráfica de ArcMap.

Dicha aplicación de escritorio ofrece dos tipos de vistas de mapa: Data View y Layout View.

En primer lugar, Data View sirve para mostrar, hacer cambios, introducir datos, hacer búsquedas geográficas o en las tablas de atributos, mientras que, por otro lado, la vista Layout View es la interfaz gráfica que sirve para producir mapas y gráficas. En las figuras A.5 y A.6 se muestran respectivamente estas dos interfaces.

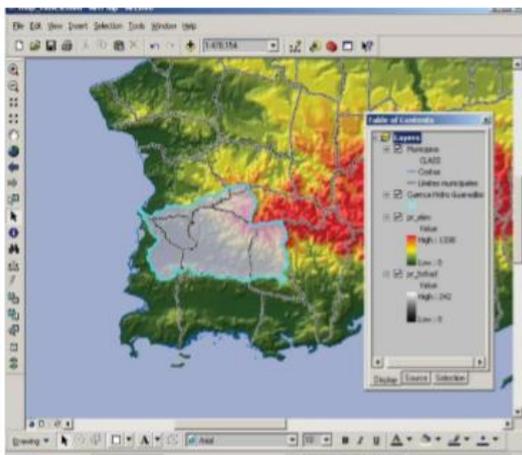


Figura A.5: Interfaz Data View [9].

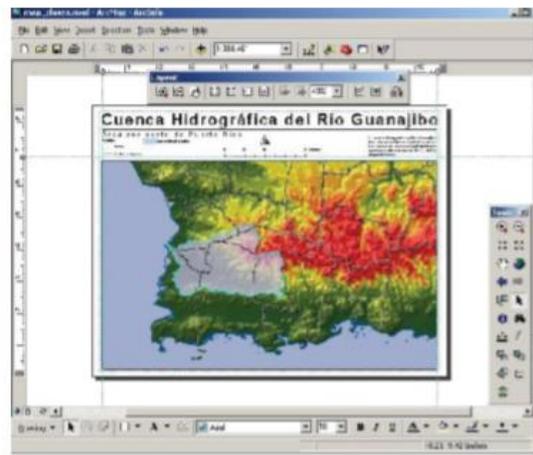


Figura A.6: Interfaz Layout View [9].

A.3.2. ArcToolbox

ArcToolbox provee un ambiente de trabajo para desarrollar operaciones de geoprocésamiento (Figura A.7). Estas operaciones se encuentran disponibles en librerías ordenadas por temas y en general cuentan con menús de ayuda (Wizards) y herramientas paso a paso para que las operaciones tanto de geoprocésamiento como de conversión de datos sean simples de utilizar. Dichas herramientas son usadas para [5]:

- Gestión de datos.
- Conversión de datos.
- Procesamiento de coberturas.
- Análisis vectorial.
- Geocodificación.
- Referenciación lineal.
- Cartografía.
- Análisis estadístico.

ArcToolbox está integrado en ArcCatalog y ArcMap y está disponible en ArcView, ArcEditor y ArcInfo.

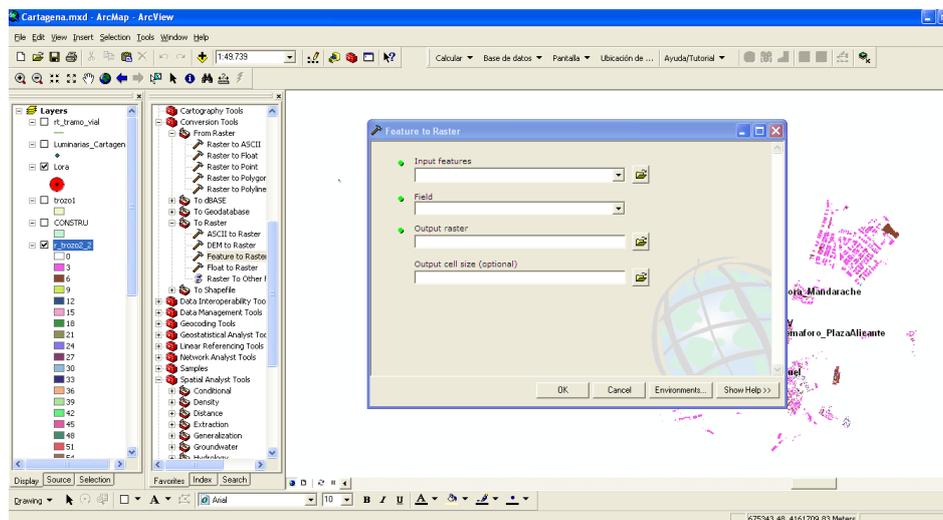


Figura A.7: Interfaz gráfica de ArcToolbox.

Las herramientas de Arctoolbox pueden ser utilizadas a través de los distintos asistentes o mediante una línea de comandos donde se especifica el nombre del comando y sus correspondientes parámetros. Arctoolbox además permite crear nuevos modelos automatizados a partir de las herramientas existentes.

A.4. Tipos de datos de un SIG

Los Sistemas de Información Geográfica manejan dos tipos de información principalmente: la información espacial o geográfica y la información descriptiva. A su vez, la información geográfica se puede almacenar de dos formas: empleando formato raster o formato vectorial [17].

La información espacial o información geográfica, es la que describe la localización y la forma de diversas características geográficas (orografía, vías de comunicación, ríos, etc.). Esta información se puede almacenar en dos formatos: tipo raster o tipo vectorial.

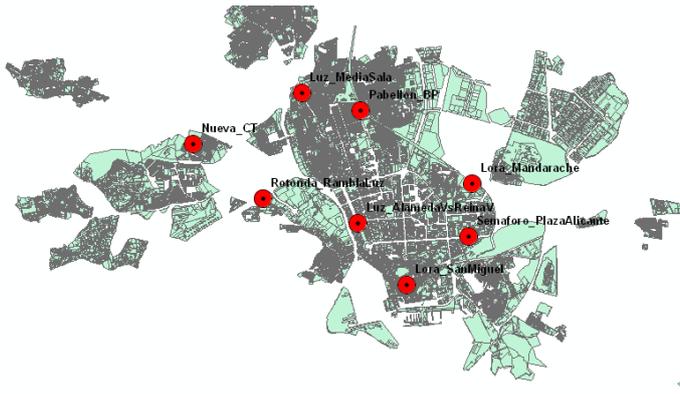


Figura A.8: Modelo vectorial de una parte del municipio de Cartagena

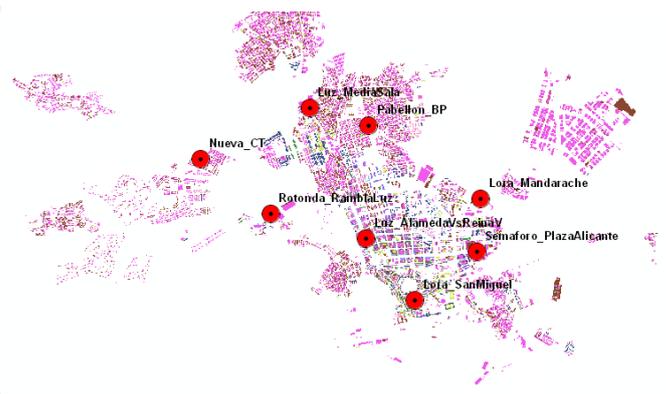


Figura A.9: Modelo raster de Cartagena.

El modelo raster, mostrado en la figura de la derecha (Fig. A.9), presupone el dividir el espacio geográfico en elementos discretos, de forma regular, contigua y mutuamente exclusiva e indivisible.

Este modelo es conocido por la existencia de una red formada por celdas o cuadrículas, conocidas como píxel, los cuales presentan una información concreta (color, altitud, etc), siendo la altura de los edificios la información perteneciente en cada píxel, en este caso.

A continuación, como se muestra en la figura A.10, en el modelo vectorial las líneas y puntos son los elementos principales del sistema. La principal diferencia con respecto a un archivo vectorial es que el archivo ráster almacena píxel mientras en el vectorial almacena coordenadas de los vértices de cada elemento geométrico.

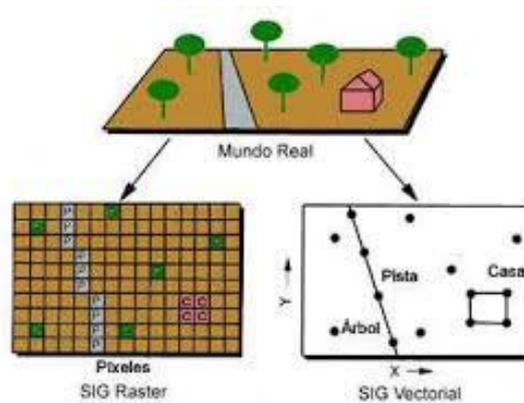


Figura A.10: geografía real a modelar; información geográfica en formato vectorial; e información geográfica en formato raster [21].

Anexo B. Programas desarrollados

B.2. Decodificación datos JSON.

```
package com.maria.tfm;

import com.google.gson.*;
import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.LinkedHashMap;
import java.util.Map;

public class ReadJson_v2 {
    public static void main(String [] arg) throws IOException {
        JsonParser parser = new JsonParser();
        Map<String, BufferedWriter> map2bwFile = new LinkedHashMap<>();
        File ruta = new File("C:\\Users\\MarÃ-a GarcÃ-a\\Desktop\\ayunta\\MarÃ-a
CÃ¡tedra\\medidas");
        File archivo = new File(ruta.getAbsolutePath() + File.separator + "medidas.txt");
        FileReader fr = new FileReader(archivo);
        BufferedReader br = new BufferedReader(fr);
        String linea;
        while ((linea = br.readLine()) != null) {
            JsonObject gsonObj = parser.parse(linea).getAsJsonObject();
            JsonObject resultObj = gsonObj.get("result").getAsJsonObject();
            String type = resultObj.get("type").getAsString();
            if (type.equalsIgnoreCase("error"))
                continue;
            String payload = resultObj.get("payloadJSON").getAsString();
            JsonObject payloadJSON = parser.parse(payload).getAsJsonObject();
            JsonElement rxJSON = payloadJSON.get("rxInfo");
            if(rxJSON == null)
                continue;
            JSONArray rxArray = rxJSON.getAsJsonArray();
            for(JsonElement element : rxArray)
            {
                JsonObject elementJSON = element.getAsJsonObject();
                String gatewayId = elementJSON.get("gatewayID").getAsString();
                File gatewayFile = new File(ruta.getAbsolutePath() + File.separator + "medidas_" +
gatewayId + ".txt");
                if(!gatewayFile.exists())
                {
                    Files.createFile(Paths.get(gatewayFile.toURI()));
                    BufferedWriter bw = new BufferedWriter(new
FileWriter(gatewayFile.getAbsolutePath()));
                    bw.write("RSSI SNR LATITUD LONGITUD");
                    bw.newLine();
                    bw.flush();
                    map2bwFile.put(gatewayFile.getName(),bw);
                }
                BufferedWriter writer = map2bwFile.get(gatewayFile.getName());
```

B.3. Análisis de las medidas. Interfaz gráfica.

```
from tkinter import *
from tkinter import ttk
from PIL import Image, ImageTk
from contra_incorrecta import error_key
from programa import programa

## Función Clave
def onEnter(event):
    Comprobar()

def Comprobar():
    Clave1=verifica_clave.get()
    Key=[]
    Key="SICOMO"
    if Key!=Clave1:
        error_key(verifica_clave)
    else:
        programa(ventana_login)

ventana_login = Tk()
ventana_login.title("Interfaz gráfica LoRa")
ventana_login.geometry("500x350")
ventana_login.tk.call('wm', 'iconphoto', ventana_login._w,
ImageTk.PhotoImage(file='images/gif.GIF'))

##Carga la imagen del Banco Santander
def resize_image(event):
    new_width = event.width
    new_height = event.height
    image = copy_of_image.resize((new_width, new_height))
    photo = ImageTk.PhotoImage(image)
    ventana_login.config(image = photo)
    ventana_login.image = photo

image = Image.open('images/upct.jpg')
copy_of_image = image.copy()
photo = ImageTk.PhotoImage(image)
ventana_login = ttk.Label(ventana_login, image = photo)
ventana_login.bind('<Configure>', resize_image)
ventana_login.pack(fill=BOTH, expand = YES)

verifica_clave=StringVar()
global entrada_login_clave
```

```

i=Label(ventana_login, text="Introduzca Contraseña")
i.place(x=195,y=10)
c=Label(ventana_login, text="Clave * ",bg="white")
c.place(x=230,y=60)

entrada_login_clave=Entry(ventana_login,textvariable= verifica_clave, show='*')
entrada_login_clave.place(x=195,y=80)

button=Button(ventana_login, text="Acceder", width=10,height=1,bg="white",
command=Comprobar)
button.place(x=230,y=320)
entrada_login_clave.bind("<Return>", onEnter)

ventana_login.mainloop()
from tkinter import *
from PIL import Image, ImageTk
import webbrowser
from tkinter import filedialog
from tkinter import ttk
import tkinter as tk
import pandas as pd
import xlrd
import threading
import numpy as np
import math
import matplotlib.pyplot as plt
from tkinter import messagebox as MessageBox

def programa(ventana_login):

    global Inputs
    global Inputs2
    global ventana

    ventana=Toplevel(ventana_login)
    ventana.transient(ventana_login)
    ventana.title("Interfaz gráfica LoRa")
    ventana.geometry("500x300")
    ventana.tk.call('wm', 'iconphoto', ventana._w,
ImageTk.PhotoImage(file='images/gif.GIF'))

    def resize_image(event):
        new_width = event.width
        new_height = event.height
        image = copy_of_image.resize((new_width, new_height))
        photo = ImageTk.PhotoImage(image)

```

```

ventana.config(image = photo)
ventana.image = photo

image = Image.open('images/upct.jpg')
copy_of_image = image.copy()
photo = ImageTk.PhotoImage(image)
ventana = Label(ventana, image = photo)
ventana.bind('<Configure>', resize_image)
ventana.pack(fill=BOTH, expand = YES)

Inputs=StringVar()
Inputs2=StringVar()

button_help=Button(ventana, text="Help",
width=10,height=1,command=Help_dir,cursor="hand2").place(x=150,y=270)
button_mesa=Button(ventana, text="Calculate",
width=15,height=1,command=Calcular,cursor="hand2").place(x=225,y=130)
button_resetPar=Button(ventana, text="Reset RADIOGIS",
width=15,height=1,command=Reset_Inputs1,cursor="hand2").place(x=230,y=270)
button_resetPar=Button(ventana, text="Reset LoRa",
width=15,height=1,command=Reset_Inputs2,cursor="hand2").place(x=345,y=270)
button6=Button(ventana, text="RADIOGIS measurements",
width=20,height=1,command=File1,cursor="hand2").place(x=0,y=60)

Parametros_entrada=ttk.Entry(ventana,textvariable=Inputs,width=50).place(x=160,y=60)
button6=Button(ventana, text="LoRa measurements",
width=20,height=1,command=File2,cursor="hand2").place(x=0,y=100)

Parametros_entrada=ttk.Entry(ventana,textvariable=Inputs2,width=50).place(x=160,y=100)

def Help_dir():
    webbrowser.open_new_tab('http://sicomo.upct.es/index.php/es/lineas-
investigacion/20-modelos-de-propagacion')

def Reset_Inputs1():
    Inputs.set("")
def Reset_Inputs2():
    Inputs2.set("")

def File1():
    global FicheroRADIOGIS
    global filas,columnas

```

```

filename= filedialog.askopenfilename(initialdir = "/",title = "Select
file",filetypes = [("All files", "*. *")])
File=filename
File=File.replace('/', '\\')
Inputs.set(File)
FicheroRADIOGIS=np.loadtxt(File)
[filas, columnas]=FicheroRADIOGIS.shape

def File2():

    global FicheroLora
    global filas2,columnas2

    filename= filedialog.askopenfilename(initialdir = "/",title = "Select
file",filetypes = [("All files", "*. *")])
    File2=filename
    File2=File2.replace('/', '\\')
    Inputs2.set(File2)
    FicheroLora=np.loadtxt(File2,skiprows=0)
    [filas2, columnas2]=FicheroLora.shape
    FicheroLora=FicheroLora[:,[0,2,3]]

def Calcular():
    def Calculo1():
        S=0

        for i in range(len(FicheroRADIOGIS)):
            for j in range(len(FicheroRADIOGIS[i])):
                if FicheroRADIOGIS[i][j]==-9999:
                    FicheroRADIOGIS[i][j]=S

    resolmdt = 2

    VSI = np.array([676849.59, 4162426.67])

    #Posición del transmisor
    Tx= np.array([677849.59, 4163426.67])

    #Calculamos la posición de los vértices de la matriz

    VSD = (VSI[0]+(columnas-1)*resolmdt, VSI[1])
    VII = (VSI[0], VSI[1]-(filas-1)*resolmdt)
    VID = (VSI[0]+(columnas-1)*resolmdt ,VSI[1]-(filas-1)*resolmdt)

    VSI2 = (676359.79+resolmdt, 4163056.76-resolmdt)
    VSD2 = (VSI[0]+(columnas-2)*resolmdt, VSI[1]-resolmdt)
    VII2 = (VSI[0]+resolmdt ,VSI[1]-(filas-2)*resolmdt)
    VID2 = (VSI[0]+(columnas-2)*resolmdt, VSI[1]-(filas-2)*resolmdt)

    coor_x_rad=[]

```

```

    coor_y_rad=[]
    coor_x1=[]
    coor_y1=[]
    Pot_Lora=[]
    Pot_RAD=[]
    d_TxRx=[]
    ind1=[]
    ind2=[]

    ##Se calcula todas las coordenadas geográficas perteneciente a cada punto
de
    ##la matriz RADIOGIS
    for i in range(len(FicheroRADIOGIS)):
        for j in range(len(FicheroRADIOGIS[i])):
            coor_x_rad.append(round(VSI[0]+(j-1)*resolmdt))
            coor_y_rad.append(round(VSI[1]+(j-1)*resolmdt))

    ##Saber si el número es Par o impar en las coordenadas del Lora

    coor_x=FicheroLora[:,1]
    coor_y=FicheroLora[:,2]

    for n in coor_x[0:len(coor_x)]:
        if n%2 == 0:
            coor_x1.append(n)
        else:
            coor_x1.append(n+1)

    for m in coor_y[0:len(coor_y)]:
        if m%2 == 0:
            coor_y1.append(m+1)
        else:
            coor_y1.append(m)

    ##Veo que coordenadas coinciden tanto en la matriz(LoRa) como en la matriz
    ##(RADIOGIS)y posteriormente guardo el punto de potencia correspondiente
a
    ##esa localización

    for t in range(len(coor_y1)):
        if ((coor_x1[t] in coor_x_rad) and (coor_y1[t] in coor_y_rad)):
            ind1.append(coor_x_rad.index(coor_x1[t]))
            ind2.append(coor_y_rad.index(coor_y1[t]))
            Pot_Lora.append(FicheroLora[t,0])
            Pot_RAD.append(FicheroRADIOGIS[ind1[t],ind2[t]])
            d_TxRx.append(round(math.sqrt((coor_x1[t]-Tx[0])**2+(coor_y[t]-
Tx[1])**2)))

    #Calculamos ME(dB),RMSE(dB)y Std
    resta=[]

```

```

ME=[]
resta_cuad=[]
RMSE=[]
Std=[]
dTxRx_log=[]
d3_log=[]
L1=[]
L2=[]

n=filas2
for p in range(len(coor_y1)):
    resta.append(abs(Pot_RAD[p]-Pot_Lora[p]))
    ME=(1/n)*sum(resta)
    resta_cuad.append(abs(Pot_RAD[p]-Pot_Lora[p])**2)
    RMSE=math.sqrt(1/n*(sum(resta_cuad)))
    Std=math.sqrt(RMSE**2-ME**2)
Modelo_Propagacion=['COST-231']
Estacion_Base=['San_Miguel']
ME_df= [ME]
RMSE_df=[RMSE]
Std_df=[Std]

df1=pd.DataFrame(list(zip(Modelo_Propagacion,Estacion_Base,ME_df,RMSE_df,
Std_df)), columns = ['Modelo de propagación','Estación Base','ME','RMSE','Std'])

file=filedialog.asksaveasfilename(filetypes=(("Excel files", "*.xlsx"),
("All files", "*.*")))

writer = pd.ExcelWriter(file, engine='xlsxwriter')

df1.to_excel(writer, sheet_name='LoRa vs RADIOGIS',startrow=1,startcol=0)

workbook = writer.book
worksheet = writer.sheets['LoRa vs RADIOGIS']

##Creo formato de celdas
header_format = workbook.add_format({'font_name': 'Calibri',
'bold': True,
'text_wrap': True,
'valign': 'top',
'align': 'center',
'fg_color': '#FF0000',
'border': 1,
'font_color': 'white'})

cell_format = workbook.add_format({'font_name': 'Calibri',

```

```

        'bold': True,
        'text_wrap': True,
        'valign': 'top',
        'align': 'center',
        'border': 1,
        'font_color': 'black'})

cell_format2 = workbook.add_format({'font_name': 'Calibri',
        'bold': True,
        'text_wrap': True,
        'valign': 'top',
        'align': 'center',
        'border': 0,
        'font_color': 'black',
        'bg_color': '#FFFFFF'})

cell_format_titulo = workbook.add_format({'font_name': 'Calibri', 'bold':
True, 'font_color': 'black', 'bg_color': 'fbddb1'})

worksheet.write_string(0,1,'Estudio          modelo          de
propagación',cell_format_titulo)

worksheet.set_column('G:XFD', None, None, {'hidden': True})

worksheet.set_default_row(hide_unused_rows=True)

for row in range(3, 19):
    worksheet.set_row(row, 22)

worksheet.set_column('C:F',22, cell_format)
worksheet.set_column('C1:F1',22, cell_format2)
worksheet.set_column('D10:F12',22, cell_format2)

worksheet.set_column('B:B',27, cell_format)
for i in range(3,9):
    worksheet.set_row(i, 22, cell_format2)
for i in range(12,19):
    worksheet.set_row(i, 22, cell_format2)

for col_num, value in enumerate(df1.columns.values):
    worksheet.write(1, col_num + 1, value, header_format)

```

```

worksheet.set_tab_color('blue')

##Calculo regresión lineal y pinto la gráfica
for p in range(len(d_TxRx)):
    dTxRx_log.append(math.log10(d_TxRx[p]))
    if (dTxRx_log[p]>2 and dTxRx_log[p]<3):
        d3_log.append(dTxRx_log[p])

param= np.polyfit(dTxRx_log,Pot_Lora,1)
param_2=np.polyfit(dTxRx_log,Pot_RAD,1)
LO=param[1]
n=param[0]
LO_2=param_2[1]
n_2=param_2[0]
Pendiente=[]
Pendientes=[]
Pendientes.append(n_2)
Pendientes.append(n)

Pendiente=['RADIOGIS','LoRa']

df=pd.DataFrame(list(zip(Pendiente,Pendientes)), columns = ['-','n'])

df.to_excel(writer, sheet_name='LoRa vs RADIOGIS',startrow=9,startcol=0)

for col_num, value in enumerate(df.columns.values):
    worksheet.write(9, col_num + 1, value, header_format)

worksheet.set_column('A:A', None, None, {'hidden': True})

for p in range(len(dTxRx_log)):
    L1.append(LO+n*dTxRx_log[p])
    L2.append(LO_2+n_2*dTxRx_log[p])

fig = plt.figure()
plt.plot (dTxRx_log,Pot_Lora,'ro')
plt.xlim(1 ,3)
plt.ylim(-131 ,-20)
plt.xlabel ('Distancia logarítmica')
plt.ylabel ('RSSI (dBm)')
plt.title ('RSSI frente a distancia')

```

```

plt.plot(dTxRx_log,Pot_RAD,"bo",)
plt.plot(dTxRx_log,L1,'g','linewidth',2)
plt.plot(dTxRx_log,L2,'c','linewidth',2)
plt.legend(('RSSI-LoRa', 'RSSI-RAD', 'L1','L2'),
           prop = {'size': 10}, loc='upper right')

plt.savefig('figura.jpg')
worksheet.insert_image('E7', 'images/figura.jpg', {'x_scale': 0.55, 'y_scale':
0.55})

writer.save()

ventana_pb.destroy()
messagebox.showinfo(message=" Output generado.", title="Output")

ventana_pb=Toplevel(ventana)
ventana_pb.title("Procesando...")
ventana_pb.geometry("400x200")
ventana_pb.tk.call('wm', 'iconphoto', ventana_pb._w,
PhotoImage(file='images/gif.gif'))
progressbar = ttk.Progressbar(ventana_pb, mode="indeterminate")
progressbar.place(x=120, y=100, width=200)
progressbar.start()
Label(ventana_pb, text=" Generando resultados." +'\n'+ " Por favor, espere
unos instantes.",font=('Helvetica', 10, 'bold')).place(x=90,y=30)
imagenmundo = tk.PhotoImage(file ="images/mundo.PNG")
imagenmundo = imagenmundo.subsample(4,4)
Label(ventana_pb, image=imagenmundo).place(x=200,y=140)

hilo2 = threading.Thread(name='Calcular', target=Calculo1)

hilo2.start()
ventana_pb.mainloop()

```

B.4. Conversión de datos UTM a DEC exportando fichero excel.

```
import utm
import pandas as pd
import numpy as np
import math

## Integramos excel en Python
df = pd.read_excel('PBI_tfm.xlsx')
## Situación del transmisor
Tx= np.array([677849.59, 4163426.67])
## Calculamos las distancias
d_TxRx=[]
for i in range(len(df['X'])):
    d_TxRx.append(round(math.sqrt((df['X'].iloc[i]-Tx[0])**2+(df['Y'].iloc[i]-Tx[1])**2)))

##Convertimos la localización de UTM a dec
lista=[]
lat=[]
long=[]

x=[678203,678200,678183,678179,678162,678158,678150,678138,678114,678118,
678115,678124,678104,678101,678100,678098,678100,678074,678060,678049,
678050,678040,678036,678037,678014,678019,678005,677980,677968,677962,67
7867,677857,677830,677842,677936,677898,677900,677872,677854,677802,6778
02,677774,677700,677686,677646,677622,677565,677472,677457,677402,677441,
677543,677548,678190,678170,678087,678000,677978,677939,677908,677721]

y=[4163629,4163608,4163626,4163617,4163599,4163598,4163592,4163584,41635
80,4163562,4163558,4163510,4163550,4163547,4163545,4163541,4163531,41635
20,4163509,4163505,4163505,4163530,4163536,4163550,4163553,4163557,41635
53,4163564,4163560,4163544,4163456,4163448,4163425,4163377,4163395,41633
75,4163367,4163345,4163311,4163291,4163333,4163337,4163521,4163543,41635
87,4163617,4163723,4163790,4163814,4163942,4163954,4163965,4163969,41641
45,4164125,4164128,4164114,4164111,4164110,4164101,4164094]

for i in range(0,len(x)):
    lista.append(utm.to_latlon(x[i], y[i], 30, 'S'))

for n in range(0,len(lista)):
    lat.append(lista[n][0])
    long.append(lista[n][1])

df_tablas=pd.DataFrame(list(zip(lat,long,d_TxRx)),columns=['Latitud','Longitud','dist
ancia_TxRx'])
df_final = pd.concat([df, df_tablas],axis=1)

df_final.to_excel("Tabla_PowerBI.xlsx")
```

