

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

**TRABAJO DE FIN DE MASTER**

**MODELO DE RECOGIDA Y GESTIÓN DE  
DATOS MASIVOS PROCEDENTES DE  
SENSORES EN VEHÍCULOS DE LA RED DE  
CARRETERAS VASCAS**

**Borja Alonso Urcelay**

**Grado en Ingeniería Técnica de Telecomunicaciones**

**Dirigido por Manuel Vélez 2ºAP**

**Curso 2015-2016**



**Ingeniaritza Goi Eskola Teknikoa  
Escuela Técnica Superior de Ingeniería  
Bilbao**

# Índice

## Contenido

1. Resumen .....	6
2. Lista De Ilustraciones, Tablas Y Acrónimos .....	8
2.1 Lista De Ilustraciones .....	8
2.2 Lista De Tablas .....	9
2.3 Acrónimos .....	9
3. Memoria .....	10
3.1 Objetivos Y Alcance Del Trabajo .....	10
3.2 Beneficios .....	11
3.3 Estado Del Arte .....	13
3.4 Análisis De Alternativas .....	15
Ubicación .....	15
Estaciones Base .....	15
Modelo de distribución de vehículos .....	16
Plataforma de desarrollo .....	16
3.5 Análisis De Riesgos .....	17
Riesgo Social .....	17
Limitación De Simulación .....	17
Limitación De Comparación .....	17
4. Metodología .....	18
4.1. Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas .....	18
Modelo De Cobertura De Receptores .....	19
Representación Y Sectorización De Las Carreteras .....	19
Modelo De Distribución De Vehículos .....	21
4.2. Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo .....	23
Criterios De Optimización .....	23
Algoritmo De Cobertura Y Coste .....	24

Algoritmo Metaheurístico .....	25
5. Cálculos .....	26
5.1. Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas .....	26
Modelado De Cobertura Y Carreteras .....	26
Modelado De Distribución De Vehículos .....	30
5.2. Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo .....	39
Criterios De Optimización.....	39
Algoritmo De Cobertura Y Coste .....	40
Algoritmo Metaheurístico .....	41
6. Aspectos Económicos .....	44
6.1 Presupuesto .....	44
Costes Unitarios .....	44
Mano de Obra .....	44
Recursos Materiales .....	44
Otros Gastos.....	45
Cálculo Del Presupuesto.....	45
7. Conclusiones .....	46
8. Bibliografía .....	47
Anexo I: Pliego De Condiciones. Normativa Aplicable.....	49
Condiciones Generales .....	49
Aplicación De Normas .....	49
Desarrollo Del Proyecto.....	49
Alteraciones Y Modificaciones.....	49
Materiales Auxiliares .....	49
Condiciones Técnicas .....	49
Software De Desarrollo .....	49
Condiciones Facultativas .....	50
Ejecución De La Obra.....	50
Recepción De La Obra .....	50
Plazo De Garantía .....	50

Condiciones Económicas .....	50
Liquidación En Caso De Rescisión .....	50
Fianza .....	51
Revisión De Precios .....	51
Infracciones .....	51
Condiciones Administrativas .....	51
Personal Contratado .....	51
Conservación De Las Obras .....	51
Contrato .....	51
Responsabilidades .....	52
Rescisión Del Contrato .....	52
Anexo II: Planos Y Esquemas .....	53
Código de los algoritmos desarrollados en el Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas .....	53
Sectorización .....	53
Base Datos Sectores .....	55
Subsectorización .....	57
Añadir Carreteras A Base Datos .....	59
Añadir Identificador A Carreteras .....	63
Base Datos Poblaciones .....	65
Base Datos Carreteras .....	70
Base Datos Anillos .....	72
Distribución Vehículos .....	78
Base Datos Vehículos Distribuidos .....	80
Matriz Vehículos Vs Transmisores .....	82
Matrices Alcance .....	84
Probabilidad Aceptación .....	85
Código de los algoritmos desarrollados en el Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo .....	87
Vehículos Vs Estaciones Base .....	87
Algoritmo Cálculo Coste-Cobertura .....	89

eman ta zabal zazu



Universidad Euskal Herriko  
del País Vasco Unibertsitatea

# 1. Resumen

La evolución tecnológica pone a nuestra disposición dispositivos cada vez más completos y accesibles. Estos cuentan con una gran cantidad de sensores que cuantifican y monitorizan las magnitudes físicas de su entorno. En el caso concreto de la industria automovilística, el aumento de sensores integrados es significativo. Si se consigue establecer un modelo de comunicación vehicular y desarrollar un sistema que gestione y analice la información que captan los vehículos, se abre un abanico de posibilidades: desde mejorar la seguridad vial hasta conocer con exactitud el estado del medio ambiente en los alrededores de las calzadas. Para poder diseñar estrategias óptimas es necesario el desarrollo de un *modelo de recogida y gestión de datos masivos procedentes de sensores en vehículos de la red de carreteras*, a través del cual se pueda diseñar un modelo de comunicación que posibilite la conexión de los vehículos y que muestre de forma realista la magnitud de datos con la que se va a trabajar. Este proyecto se divide en dos partes principales. En primer lugar en el desarrollo de un modelo realista de recogida de datos procedentes de los vehículos que circulan por las carreteras de la comunidad autónoma vasca (CAV). Esta primera parte fue aceptada, publicada y presentada en el *XXXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, que se celebró en Madrid en el Campud de Canotblanco de la EPS-UAM entre los días 5 y 7 de Septiembre de 2016. La segunda parte consta del estudio de optimización del modelo de red necesario para proveer de servicio a los vehículos. Esta segunda parte ha sido enviada a la tercera *International Conference on the Harmony Search Algorithm*, que tendrá lugar entre los días 22 y 24 de Febrero de 2017 en el edificio Bizkaia Aretoa de la UPV/EHU, Bilbao.

Technological evolution puts at our disposal more and more complete and accessible devices. These devices have a great amount of sensors that quantify and monitor the physical magnitudes of their environment. In the specific case of the automotive industry, there has been a significant increase of integrated sensors. If establishing a model of vehicular communication and developing a system that manages and analyzes the information that vehicles capture is reached, a wide range of possibilities is opened: from improving road safety to knowing the exact state of the environment around the roads. In order to be able to design the optimal strategies it is required the development of a *model about collection and managing of massive data coming from sensor in vehicles that travel through the highway network*, by which we will be able to design a model of communication that allows us to make the connection between vehicles and that will show on a realistic way the magnitude of the data with which the work will be done. This project is divided on two main parts. In the first place, it focuses on the development of a realistic model about collection of data coming from the vehicles that travel through the highways of the Basque

autonomous community. This first part was accepted, published and expounded on the *XXXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*. The second part consists of a study on the streamlining of the net model necessary to provide service to the vehicles. That second part has been sent to the third *International Conference on the Harmony Search Algorithm*.

Eboluzio teknologikoak geroz eta gailu eskuragarri eta osoagoak jartzen ditugure esku. Hauek, hainbat sentsore dituzte haien inguruko magnitude fisikoak kuantifikatu eta monitorizatzeko. Automobil industriaren kasu zehatzean, sentsore integratuen areagotzea adierazgarria da. Ibilgailuek biltzen duten informazioa aztertu eta kudeatzen duen sistema bat garatzea lortuz gero, hainbat aplikazioetarako aukera zabaltzen da: bideko segurtasuna hobetzetik errepide ondoko ingurugiroaren egoera zehaztasunez ezagutzera. Estrategia optimoak diseinatu ahal izateko, errepideetako ibilgailuen sentsoreetatik datozen datu kopuru handiak biltzeko eta kudeatzeko modelo bat beharrezkoa da, honen bidez komunikazio modelo bat diseinatu eta ibilgailuen arteko komunikazioa ahalbidetzeko, datuen tamaina modu erreal batean erakusten duena. Proiektu hau bi ataletan banatzen da. Lehenik eta behin, Euskal Autonomi Erkidegoko (EAE) errepideetako ibilgailuetatik datozen datuen bilketarako modelo erreal baten garapena. Lehen atal hau jadanik onetsia, argitaratua eta aurkeztua izan zen *XXXI Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*-n. Bigarren atalak ibilgailuei zerbitzua emateko beharrezkoa den sare modeloaren optimizazioaren ikerketan datza. Bigarren atal hau hirugarren *International Conference on the Harmony Search Algorithm*-era bidali da.

# 2. Lista De Ilustraciones, Tablas Y Acrónimos

## 2.1 Lista De Ilustraciones

Ilustración 1: Diagrama de flujo del modelo de recogida y análisis de datos procedentes de vehículos en una red de carreteras .....	19
Ilustración 2: Algoritmo de división del área de interés en sectores con cobertura .....	20
Ilustración 3: Algoritmo para cálculo de cobertura en función de los tipos de perfiles de las estaciones base .....	24
Ilustración 4: Funcionamiento algoritmo desarrollado en conjunto con el HSA .....	25
Ilustración 5: Mapa con estaciones base.....	27
Ilustración 6: Mapa de la CAV con la Cobertura de cada Estación Base .....	27
Ilustración 7: Sectorización Óptima.....	28
Ilustración 8: Mapa de la CAV sectorizado en zonas con al menos un transmisor (mapa de cobertura) .....	28
Ilustración 9: Subsectorización.....	29
Ilustración 10: Información almacenada en la base de datos de los subsectores .....	30
Ilustración 11: Carreteras sin filtrado .....	31
Ilustración 12: Mapa de la CAV con carreteras de interés.....	31
Ilustración 13: Mapa de la CAV con las carreteras de interés divididas en tres grupos (rojo más afluencia, azul afluencia intermedia y amarillo afluencia baja).....	32
Ilustración 14: Base de datos completada con la información de las carreteras .....	32
Ilustración 15: Mapa con poblaciones (Rojo) y el alcance de la distribución de los vehículos (Círculo).....	34
Ilustración 16: Descripción del porcentaje de vehículos en el interior de un anillo .....	34
Ilustración 17: Base de datos con la información de los vehículos en cada subsector .....	36
Ilustración 18: Curva con los resultados de las iteraciones .....	42
Ilustración 19: Resultado tras 5 iteraciones aleatorias del algoritmo de cálculo de coste y cobertura .....	43



## 2.2 Lista De Tablas

Tabla 1: División de tipos de carreteas .....	30
Tabla 2: Municipios y su respectiva población .....	33
Tabla 3: Factor corrector C .....	35
Tabla 4: Lista de municipios y la cantidad de vehículos distribuidos entre las 7:00-10:59 .....	37
Tabla 5: Lista de municipios y la cantidad de vehículos distribuidos entre las 21:00-23:59 .....	38
Tabla 6: Perfiles posibles para cada estación base .....	40
Tabla 7: Coste personal .....	44
Tabla 8: Coste material.....	45
Tabla 9: Presupuesto .....	45

## 2.3 Acrónimos

CAV: Comunidad Autónoma Vasca

Tx: Transmisor

V2I: Vehicle-to-infrastructure

V2V: Vehicle-to-vehicle

V2X: Vehicular-to-vehicular/infrastructure

M2M: Machine-to-machine

HSA: Harmony Search Algorithm

LED: Light Emission Diode

SWARS: Secure and Accurate Road Weather Services

URSI: XXXI Simposium Nacional de la Unión Científica Internacional de Radio

ICHSA: International Conference on the Harmony Search Algorithm

GPS: Global Positioning System

GLONASS: Global'naya Navigatsionnaya Sputnikovaya Sistema

## 3. Memoria

### 3.1 Objetivos Y Alcance Del Trabajo

El objetivo principal de este proyecto se puede desglosar en dos partes. La primera parte consiste en el desarrollo de un modelo de recogida de datos masivos para la monitorización de las carreteras vascas. Con este primer objetivo se pretende alcanzar una plataforma que simule de forma realista la distribución de vehículos en las carreteras de la CAV. La plataforma va a posibilitar conocer la magnitud de datos que se tratan en una red vehicular, de tal forma que permitirá el estudio de estrategias adaptadas de recogida y procesado de datos. Para la consecución de este apartado se diseña un algoritmo de distribución de vehículos a lo largo de las carreteras de la CAV que permita su monitorización y proporcionando la ubicación de cada vehículo que permita en el futuro obtener su respectiva información almacenada a través de los sensores que posee. Esta primera parte va a suponer una base inicial sobre la que poder continuar investigando en la temática del big data en el caso concreto de una red vehicular vehicle-to-infrastructure (V2I) o vehicle-to-vehicle (V2V). Pretende posibilitar el estudio, a través de simulaciones, de estrategias adaptadas que permitan optimizar el rendimiento de la red. Con el desarrollo de esta primera parte del objetivo se pretende realizar un primer paso para posteriormente optar a la convocatoria europea de proyectos Celtic-Plus [1]. En ella se va a plantear un proyecto a tres años (01/01/2017-31/12/2019) el cual contará con la participación de empresas y centros de investigación de seis países diferentes: Bélgica, Francia, España, Finlandia, Portugal y Turquía. El título del proyecto a presentar será “Secure and Accurate Road Weather Services (SWARS)”.

El segundo objetivo consiste en la búsqueda de soluciones que mejoren la eficiencia de una red inalámbrica V2I para la recogida de datos masivos con multiobjetivos metaheurísticos. En este apartado, utilizando la plataforma previamente desarrollada, se va a estudiar cómo optimizar la red. La optimización, que puede seguir numerosos criterios, se va a definir como la capacidad de ofrecer el servicio adecuado empleando para ello la menor cantidad de recursos necesarios posibles. De esta forma se tratará de no dotar de más potencia de la necesaria a las estaciones base ni de reservar un excedente del espectro radioeléctrico. Para alcanzar este objetivo se va a emplear un algoritmo del tipo *Harmony Search Algorithm (HSA)* [2] con operadores de selección por dominancia para convertirlo en multiobjetivo. En esencia será el HSA básico, pero el criterio para mantener las soluciones en la memoria de armonías se basará en rango y distancia de hacinamiento. Aunque este tipo de

algoritmos metaheurísticos no nos permitan alcanzar el resultado óptimo, sí que posibilitan acercarnos a este punto de trabajo. Para emplear ese algoritmo se va a desarrollar un código, que teniendo como entrada unos datos sobre las estaciones base, nos de como resultado el coste y la cobertura asociada a esa entrada. En colaboración con un experto en HSA podremos controlar las entradas de tal foma que el resultado final nos ofrezca soluciones de mejoras en eficiencia.

## 3.2 Beneficios

El desarrollo de este proyecto proporcionará una amplia serie de beneficios sociales, tecnológicos y económicos. La monitorización del estado de las carreteras y su entorno nos posibilitará tener a nuestro alcance una gran cantidad de métricas de múltiples orígenes, desde medioambientales hasta socioculturales.

En primer lugar se podrá conocer con precisión todo el entorno de las carreteras. Con ello será posible obtener información climatológica y ambiental. Esto facilitará labores muy diversas. Se podrá monitorizar en tiempo real una serie de parámetros que permitirán el desarrollo de estrategias personalizadas para la resolución de varios problemas. Hoy en día la preocupación por el medio ambiente no para de aumentar y se invierte cada vez más dinero en tratar de reducir la contaminación. Con los sensores de los vehículos y una red adecuada se podrá monitorizar en tiempo real el estado del aire (niveles de óxidos de nitrógeno e hidrocarburos) y aplicar medidas restrictivas en caso de que fuera necesario para poder estabilizar altos niveles. Por ejemplo, la comunidad autónoma de Madrid, en función de esos niveles, restringe la velocidad o incluso el flujo de vehículos. Pero estas medidas se toman al cabo de unos días de alcanzar los niveles peligrosos. Con este progreso se podrían acortar los tiempos de reacción notablemente. Se podrían poner en práctica estas soluciones antes de alcanzar los niveles peligrosos, solo con ver la evolución que están siguiendo.

Poder monitorizar los vehículos también simplificará la labor de conocer con precisión los flujos por las carreteras. Conociendo estos datos se podrían proponer variaciones en la organización de las carreteras, tanto estáticas (de forma permanente) como dinámicas (variaciones según el momento del día, semana, mes e incluso año). Es decir, se harán visibles algunas carencias organizativas y se podrán variar de forma permanente el sentido de algunos carriles si fuera necesario. Como modificaciones dinámicas, junto con la ayuda de semáforos estratégicamente ubicados, se podrá variar la circulación de algunos carriles en caso de que algún accidente bloqueará un sentido y el opuesto no estuviera casi ocupado, o en función de la hora del día y la demanda. Es habitual que a primera hora de la mañana se colapse un sentido mientras

que a última hora de la tarde se colapse el opuesto debido al flujo de personas que van y vienen del trabajo, respectivamente. También facilitará la obtención de datos sobre las rutas más realizadas, lo que posibilitará una mejora considerable en el diseño de las líneas de transporte público (cambios de ruta, ampliación de frecuencia en horas determinadas,...).

La mejora más importante será en el ámbito de la seguridad. Este proyecto ayudaría a crear modelos y herramientas que podrían servir para mejorar considerablemente este aspecto desde varios puntos de vista diferentes. Por un lado, nos ayudaría a conocer el estado de las carreteras en tiempo real (lluvia, viento, tráfico,...), y esto hará posible que se ajusten algunos parámetros que afecten a la circulación. Por ejemplo se propone la implantación de señales de velocidad LED, cuyos límites puedan ser modificables para aumentar la seguridad. También se podrán activar las farolas en caso de que fuese necesario y aun cuando no sea dentro del horario estipulado, mejorando la visibilidad de los conductores. Por otra parte se podrá conocer el mal estado de algunas carreteras. En muchas ocasiones es difícil para el organismo competente conocer el estado de todas las carreteras y esto podría agilizar los trámites para su reparación, convirtiendo las carreteras en un sitio un poco más seguro. Y por último, y la mejora más importante en este apartado: se podrá comunicar a los servicios de emergencia en tiempo real cuándo se ha producido un accidente, mejorando el tiempo de respuesta y aumentando las posibilidades de supervivencia de los afectados. Estas mejoras tendrían dos grandes consecuencias. La primera es la cantidad de vidas que se podrán salvar y la segunda el ahorro económico, especialmente en sanidad.

Finalmente, y aunque aún es un campo en el cual queda mucho por desarrollar, la instauración de esta red podría facilitar notablemente la implantación del vehículo autónomo. A día de hoy los vehículos autónomos toman las decisiones de circulación en función de la información captada por sus sensores. En un futuro estas decisiones podrían tomarse en colaboración con la información obtenida por otros vehículos, minimizando el error de decisión y favoreciendo un funcionamiento más seguro.

## 3.3 Estado Del Arte

Con la evolución de la tecnología, cada día tenemos a nuestra disposición más dispositivos y de menor tamaño. Estos dispositivos cuentan, en su mayoría, con una gran cantidad de sensores que cuantifican y monitorizan las magnitudes físicas de su entorno, tales como la temperatura, la presión atmosférica, la ubicación, etc. Este desarrollo provoca que los datos que se generan crezcan de manera exponencial, contribuyendo al big data [3]. En la actualidad aún no se han implementado redes ni se ha estudiado en profundidad como acceder a esta información. La gran cantidad de datos recogidos nos va a permitir, con una forma eficiente de gestionarlos y tratarlos, ser capaces de obtener datos relevantes del entorno de las carreteras.

Precisamente en la industria del automóvil es donde más ha aumentado la cantidad de sensores que se integran. Los vehículos que salen al mercado a día de hoy incorporan numerosos dispositivos capaces de monitorizar todo lo que rodea la conducción [4], desde parámetros ambientales (temperatura, contaminación [5], precipitaciones [6],...), parámetros relativos al vehículo (velocidad, revoluciones, gasolina, emisiones de gases de efecto invernadero,...), sensores de seguridad (activación de airbags, detección de volcado,...) y de ubicación (GPS, Galileo, GLONASS,...). En cambio aún no se ha analizado en detalle como implementar una red que permita tener acceso a todos los parámetros que estos sensores registran. Se podría conocer con detalle el estado de las carreteras y lo que les rodea sin necesidad de instalar y dar soporte a una serie de estaciones ubicadas a lo largo de éstas, como se hace en la actualidad.

Con la ayuda de los nuevos servicios de comunicación que se englobarán en el futuro ecosistema 5G [7] [8], que además de las comunicaciones móviles permite también la integración de IoT y de comunicaciones vehiculares V2X y comunicaciones entre máquinas machine-to-machine (M2M), los datos de los sensores podrán ser compartidos a través de internet. Estos modelos de comunicaciones móviles, que han sido ampliamente investigados y desarrollados, nos sirven de propuesta futura para la implantación de nuestra red. Dentro de las comunicaciones vehiculares V2X contamos con dos variantes. La primera consiste en la comunicación directa entre cada vehículo y la estación base, V2I. Esta es la más simple de implementar, pero a la vez resulta poco óptima en entornos con muchos vehículos, ya que ocupa demasiado ancho de banda y el receptor podría no tener capacidad para comunicarse con una cantidad demasiado grande de vehículos [9]. La segunda variante consiste en clústeres de comunicación de varios vehículos, definiendo un vehículo maestro que será el encargado de la comunicación con el receptor [10] [11]. Esta alternativa optimiza el ancho de banda y la carga que recibe la estación base, pero requiere de estrategias complejas de comunicación entre vehículos.

A la hora de plantear modelos de gestión eficientes, para la recogida de datos procedentes de un gran volumen de vehículos que se desplazan por carreteras, es de gran importancia el poder contar con herramientas capaces de estimar la distribución de vehículos a lo largo de las carreteras situadas en el área de interés. Para esta distribución existen muchos diseños y desarrollos de modelos matemáticos de muy diversa complejidad [12] [13].

Asimismo, en estos escenarios es también muy relevante el intentar minimizar los costes relacionados con la infraestructura de red que se necesita para que se establezcan las comunicaciones inalámbricas por las que transmiten la información de los sensores de los vehículos que se desplazan en una determinada área geográfica. En este sentido existen investigaciones previas sobre métodos de ahorros orientados al ahorro del coste energético [14].

Este proyecto tiene su origen en el objetivo de optar a la convocatoria europea de proyectos Celtic-Plus. En ella se va a plantear un proyecto a tres años (01/01/2017-31/12/2019) el cual contará con la participación de empresas y centros de investigación de seis países diferentes: Bélgica, Francia, España, Finlandia, Portugal y Turquía. El título del proyecto a presentar será "Secure and Accurate Road Weather Services (SWARS)". El proyecto pretende, a través de los datos meteorológicos registrados por los sensores de los vehículos, aumentar la información que las estaciones meteorológicas registran. Esta información se utilizará para gestionar los puntos de mayor riesgo en situaciones climatológicas adversas en las carreteras suburbanas. Este trabajo pretende dar respuesta a algunos de los retos planteados en el proyecto de convocatoria europea, como es la posibilidad de tener un setup de análisis realista y el de realizar estudios de configuración de red que permitan ser eficientes en la comunicación de los datos recogidos por los vehículos.

## 3.4 Análisis De Alternativas

Al realizar este proyecto han sido varias las ocasiones en las que se ha tenido que escoger una alternativa en lugar de otras. A continuación se van a describir las decisiones más importantes que se han tomado durante su ejecución.

### Ubicación

En primer lugar se debía definir la ubicación sobre la cual realizar el modelo. Aunque las alternativas eran numerosas se opta por emplear la CAV por varios motivos. Por un lado se debe a que este proyecto puede servir como preámbulo para la convocatoria europea de proyectos Celtic-Plus en el proyecto “Secure and Accurate Road Weather Services (SWARS)”. Este proyecto europeo nos acota las posibilidades a los lugares de interés de la convocatoria. El motivo que hace que se incline la balanza a favor de la CAV es que al conocer el lugar y sus instituciones sabíamos ya a dónde acudir para obtener información acerca de la ubicación de las antenas (Itelazpi, Cellnex, Ministerio del Interior,...), datos estadísticos de la población (Eustat), datos sobre el volumen de vehículos, datos sobre la geolocalización de las carreteras y los pueblos... Por tanto resulta especialmente interesante realizar el modelo sobre un lugar en el cual no se tendrá la necesidad de invertir mucho tiempo en investigar sobre sus características y además que pueda servir como base para el desarrollo del proyecto europeo. Esta ubicación también permitiría plantear las bases para que los resultados del modelo realista planteado pudieran llevar a evaluarse de forma acotada en una posible prueba piloto real que se llegara a realizar dentro del proyecto.

### Estaciones Base

En segundo lugar se plantea si emplear estaciones base ya existente de otras redes o diseñar una red desde cero, incluyendo el emplazamiento de las estaciones base. Por motivos económicos se opta por emplear una red ya existente. Con esta elección tomada queda escoger la red que mejor se nos adapte. Tenemos a nuestra disposición la red de telefonía móvil y la red de TDT. Finalmente se decide emplear las estaciones base de la red de TDT y descartar las de telefonía móvil, porque Cellnex Telecom opera una red celular de datos destinada específicamente a Internet of Things. Esta red permite la conexión de objetos a largo alcance y bajo coste. Cellnex Telecom resulta el operador ideal debido a que se va a trabajar con sensores instalados en vehículos y se pretende instaurar una red IoT con las características de tasas bajas de envío de datos y comunicación a larga distancia. Además también se emplean las estaciones base de

Itelazpi, ya que solo con las de Cellnex Telecom serían demasiado pocas como para dar cobertura a un territorio tan amplio.

## Modelo de distribución de vehículos

El modelo de distribución de vehículos también ha sido motivo de decisión. En primer lugar se contempló la posibilidad de hacerlo mediante estudios de flujos de vehículos, pero debido a la dificultad que entrañaba para modelar toda la CAV y el poco impacto positivo que tendría en los resultados finales se opta por un algoritmo que no tenga en cuenta los flujos. Es decir, el camino que vaya a trazar cada vehículo. Por el contrario se va a tener en cuenta únicamente la ubicación exacta de cada vehículo en un momento determinado, sin que exista ninguna correlación con la ubicación que tendrán los vehículos en el siguiente instante de tiempo. Para ello se decide finalmente emplear el algoritmo descrito más adelante, en el cual se redistribuyen los vehículos sin importar la ubicación previa de los mismos.

## Plataforma de desarrollo

En cuanto a la plataforma y lenguaje de desarrollo para la implantación de los algoritmos que ayuden a seleccionar una gestión eficiente de los recursos de red de conexión de datos inalámbricos se contemplaron varias alternativas (C, Java, C# y Matlab). Debido a que se planificó desde un principio la importancia que tendría el tratamiento de grandes cantidades de datos se opta por escoger el lenguaje que favoreciera su tratamiento, Matlab (Matrix Laboratory). Esta plataforma está orientada a matrices, lo cual simplifica sobremanera el diseño de un proyecto con estas características. Además, permite obtener con facilidad resultados visuales en forma de plots, lo cual resulta muy práctico al ser especialmente difícil comprobar los resultados de algunas operaciones que manipulan muchos datos.

## Algoritmo Heurístico

Para el desarrollo de la segunda parte del trabajo se contemplan varios tipos de algoritmos heurísticos que puedan servirnos en el objetivo de mejorar la eficiencia de la red inalámbrica. Entre las alternativas que se sopesaron se encuentran algoritmos genéticos, Coral Reefs Optimization, Firefly Optimization, inteligencia de enjambre... Finalmente se escoge el algoritmo *Harmony Search Algorithm* debido a que en el departamento hay un investigador especializado en él. Esto va a suponer un ahorro en tiempo, ya que con su ayuda no será necesaria la inversión temporal en recopilar información y la posterior implantación en el código desarrollado.



## 3.5 Análisis De Riesgos

La ejecución de cualquier proyecto lleva consigo una serie de riesgos que pueden afectar a su desarrollo e incluso poner fin al mismo. Identificar y evaluar los riesgos que un determinado proyecto puede suponer es una tarea que pretende reducir la posibilidad de que se produzcan dichos riesgos y, en caso de que sean inevitables, tratar de reducir su impacto estableciendo unos planes de contingencia. A continuación se detallan los riesgos del proyecto.

### Riesgo Social

Aunque las mejoras sociales que puede traer son amplísimas, no hay que olvidar que la información es poder. Si la gestión de tantísima información no fuera usada para el beneficio común, sino que fuera gestionada de forma fraudulenta, esto podría suponer un gran riesgo social para los intereses del conjunto. Es por ello que sería recomendable en un punto de maduración superior desarrollar una estrategia de control sobre la gestión de los datos capturados.

### Limitación De Simulación

Debido a la naturaleza del proyecto, una simulación, tiene inexactitudes con la realidad. Si estas desviaciones fueran excesivamente amplias, podría desvirtuar el objetivo de proporcionar un escenario realista sobre el que sustentar unos resultados que puedan ser aplicados a la realidad. Los puntos débiles de esta simulación son: la omisión de pequeñas poblaciones, la inexactitud del reparto del parque automovilístico, la limitación en el alcance de la distribución del parque automovilístico de cada población, no haber tenido en cuenta con precisión los flujos de automóviles,...

### Limitación De Comparación

Al no existir ninguna red vehicular implantada similar a la que se quiere estudiar con este proyecto no se pueden comparar los resultados modelados con datos reales obtenidos de una red equivalente. Esto puede desfavorecer aún más las limitaciones de la simulación descritas en el punto anterior, ya que no se podrá calibrar con desviaciones observadas en otros puntos geográficos.

## 4. Metodología

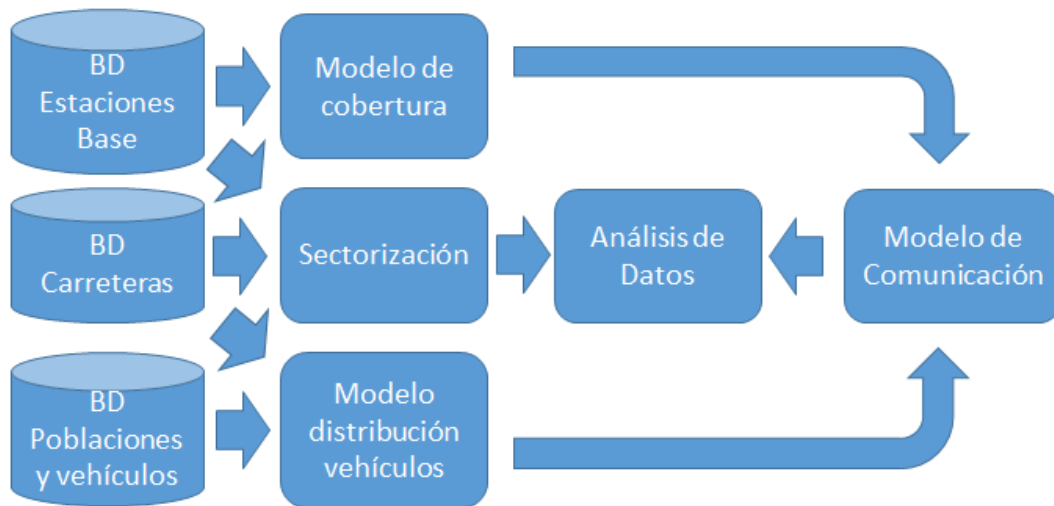
El trabajo se ha dividido en dos partes temáticas principales. La primera parte consiste en el diseño y desarrollo de un modelo de distribución de vehículos en las carreteras de la CAV. En esta parte se plantea el reto de modelar de forma realista un escenario que sea capaz de simular la obtención de datos de una red vehicular. La segunda parte, la cual se apoya en la primera, consiste en un estudio para optimizar la gestión de las estaciones base que formarán la red vehicular. Este estudio tratará de responder a la problemática de no sobredimensionar la red y ser capaz de dar cobertura a las necesidades de nuestro escenario.

### 4.1. Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas

La parte que se describe a continuación es la correspondiente a la publicada y presentada en el XXXI Simposium Nacional de la Unión Científica Internacional de Radio, URSI 2016, que se celebró en Madrid en el Campus de Cantoblanco de la EPS-UAM entre los días 5 y 7 de Septiembre de 2016 [15].

En esta sección se presenta un modelo de recogida de datos procedentes de vehículos distribuidos por la red de carreteras con el objetivo de monitorizar en tiempo real el estado de las mismas.

Este proyecto se centra principalmente en el modelado de las etapas de recogida de datos. En la Fig. 1 se presenta el diagrama de flujo del proceso global de recogida y análisis final de los datos recogidos, donde el acrónimo BD corresponde a las bases de datos, existiendo tres. Con la primera base de datos, la de las estaciones base, se modela la cobertura. Con la primera y la segunda, la de carreteras, se sectoriza el área de interés. Con la segunda y tercera, la de poblaciones y vehículos, se modela la distribución de vehículos. Con el modelo de cobertura y de distribución de vehículos, en un punto mas avanzado de maduración del proyecto, se podrá modelar la comunicación. Y por último se podrá estudiar técnicas de análisis de datos con la combinación de la sectorización y el modelo de comunicación. Esto sería para un segundo punto de maduración.



**Ilustración 1: Diagrama de flujo del modelo de recogida y análisis de datos procedentes de vehículos en una red de carreteras**

## Modelo De Cobertura De Receptores

En primer lugar es indispensable definir el método de recogida de datos. Para ello hay que definir la ubicación de las estaciones base, que actuarán como receptores de la información recopilada por los sensores. Como el objetivo es el de maximizar la utilización de los recursos que se tiene a disposición, se propone emplear la ubicación de antenas transmisoras de TDT.

Una vez obtenida la ubicación de los posibles receptores se estudia el alcance de cobertura que estos proporcionarían a las carreteras. De esta manera, se consigue alcanzar una recogida geolocalizada de datos en función de la cercanía a las diferentes estaciones base.

## Representación Y Sectorización De Las Carreteras

Conociendo la cobertura de cada receptor se propone la sectorización del territorio basada en la maximización del área de cobertura (Ilustración 2). Para ello se propone el algoritmo de sectorización de la Ilustración 2, donde Tx norte es el transmisor ubicado más al norte, Tx sur el ubicado más al sur y así sucesivamente. Los valores  $x_{min}$  y  $x_{max}$  son los valores en el eje x de los transmisores ubicados más al norte y sur, mientras que los  $y_{min}$  e  $y_{max}$  son los valores en el eje y de los transmisores ubicados más al este y oeste. El parámetro R hace referencia a la longitud de cada sector (cuadrado) de cobertura.

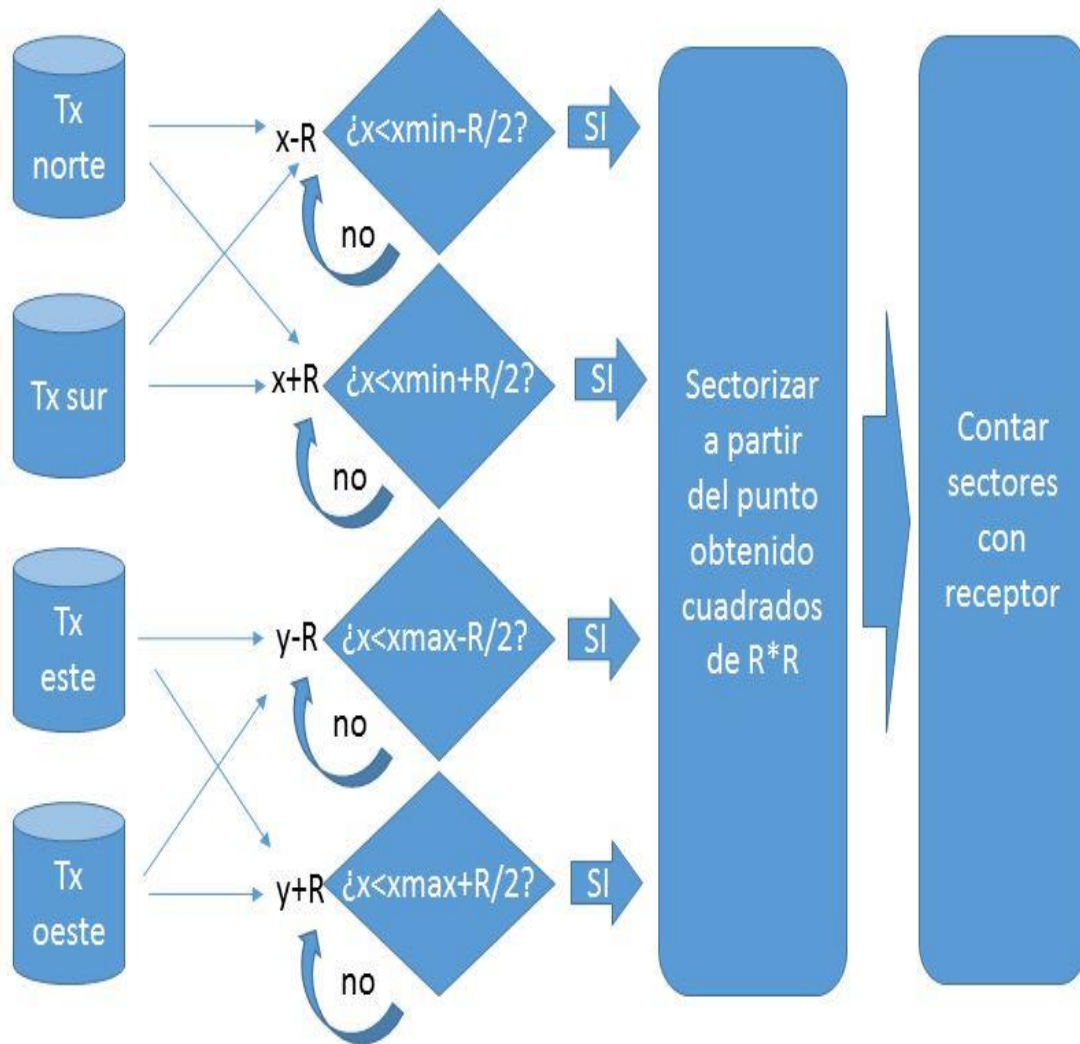


Ilustración 2: Algoritmo de división del área de interés en sectores con cobertura

Este algoritmo se ejecuta cuatro veces, con inicios de partida distintos, y después compara los resultados de los cuatro casos. En primer lugar comienza con la antena base situada más al norte (Tx norte). A partir de su latitud se va desplazando hacia el este hasta situarse a la mitad del radio de alcance de la estación base ubicada más al este. A partir de ese punto se comienza a sectorizar en cuadrados con los lados del valor del radio (R). La segunda ejecución comienza con la estación base ubicada más al sur. Repite el proceso del anterior para localizar el punto de partida, a partir del cual sectoriza el área de interés. En la tercera ejecución el proceso cambia ligeramente. Ahora se escoge la estación base ubicada más al este. En este caso, en vez de buscar la latitud más al este de una estación base, se busca superar la longitud más al norte de la estación base correspondiente. A partir de ese punto se comienza la sectorización. Por último se repite este mismo proceso desde la estación base ubicada

más al oeste. Una vez realizadas las cuatro iteraciones se comparan entre sí, escogiendo la sectorización que tenga más sectores con al menos una estación base en su interior.

Una vez sectorizado el territorio de interés es adecuado dividirlo en subsectores, con el fin de obtener mayor resolución en la información recopilada. La información de cada subsector será recogida por el receptor del sector al que pertenece.

Para conocer la ubicación de las carreteras se utilizará la base de datos desde Open Street Maps [16]. Cada polilínea del mapa viene descrita por el tipo de vía. De esta forma se pueden dividir las vías en varios grupos principales, en función de la cantidad de tráfico que se prevé que tendrán. Es importante conocer el tipo de vía que primará en cada subsector (prima el tipo de vía con mayor afluencia de tráfico). De esta forma, se conocerá aproximadamente la cantidad de información que obtendremos de cada subsector, pudiendo aplicar diferentes algoritmos de análisis y tratamiento de datos.

## Modelo De Distribución De Vehículos

El siguiente paso a dar, con la división del territorio a cubrir y el posicionamiento de los diferentes tipos de vías ya realizado, es la distribución de los vehículos. A diferencia de otros modelos matemáticos de mayor complejidad en este caso se va a plantear un modelo de simulación de tráfico simplificado. Para ello se obtiene información de los principales núcleos urbanos (cantidad de vehículos por ubicación geográfica). Utilizando las poblaciones más significativas de la región a estudiar, se crean círculos de interés con radio máximo,  $r_{max}$ , centrados en cada población. Estas áreas circulares se dividen en varios anillos concéntricos con diferentes radios interiores y exteriores. En cada anillo de radio interior,  $r_{int}$ , y radio exterior,  $r_{ext}$ , se distribuye la proporción correspondiente del total de vehículos que hay en esa población acorde a la siguiente ecuación (Ecuación 1).

$$P = \frac{\int_{r_{int}}^{r_{ext}} e^{-c*r} dr}{\int_0^{r_{max}} e^{-c*r} dr} \quad (1)$$

Donde P es la proporción de vehículos entre los radios de la integral superior, r el radio y c el coeficiente que modifica la pendiente en función del tamaño de la población. Debido a la naturaleza de las diferentes poblaciones los desplazamientos en una gran ciudad serán, en su mayoría, más cortos que en una localidad menor, ya que

la oferta de servicios es también superior. Se podrán ajustar estas diferencias con el coeficiente  $c$ , el cual suaviza o aumenta la pendiente de la función de distribución. Si  $c$  tiene un valor alto, aumenta la pendiente de la función, aumentando el porcentaje de vehículos a distribuir en los anillos más próximos al centro de la población. Si  $c$  tiene un valor bajo, la pendiente será menos pronunciada, aumentando el porcentaje de vehículos a distribuir en los anillos más alejados del centro de la población. Los vehículos se distribuyen entre las vías existentes, que se agruparán en tres grandes tipos, de forma que el tipo de vía más concurrido tendrá mayor probabilidad de que se le asignen los vehículos que el tipo de vía menos concurrido. Debido a que la cantidad de vehículos en circulación no es la misma según la hora del día se podrá aplicar un factor corrector que dependa de esta variable. El total de vehículos ( $V$ ) a distribuir de forma aleatoria en el área encerrada entre dos circunferencias será (Ecuación 2):

$$V = C * P * TV \quad (2)$$

donde  $C$  es el factor corrector, y  $TV$  el total de vehículos en cada núcleo urbano.

## 4.2. Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo

La parte que se describe a continuación es la correspondiente a la enviada al III congreso internacional del *Harmony Search Algorithm (ICHSA)* [17]. Este congreso tendrá lugar entre los días 22 y 24 de Agosto de 2017 en el edificio *Bizkaia Aretoa* de la UPV/EHU, Bilbao.

En esta segunda parte del proyecto se va a emplear el modelo previamente desarrollado para caracterizar la red vehicular. Al principio del desarrollo del modelo se consideraban las mismas características para todas las estaciones base, de tal forma que se pudiera simplificar un primer análisis de cobertura. Ahora, con la distribución de los vehículos realizada y conociendo la ubicación de las antenas base, se va a tratar de optimizar el coste variando las características de cada estación base en función de sus necesidades.

### Criterios De Optimización

Para esta optimización se van a considerar dos parámetros: potencia y capacidad. Esta elección se debe a que son los dos parámetros que con mayor facilidad pueden verse modificados. Para aumentar o disminuir la cobertura basta con hacer lo propio con la potencia, mientras que la capacidad dependerá del ancho de banda que contratemos. No se tendrá en cuenta la opción de variar la ubicación de las estaciones base, ya que supondría obra civil y, por tanto, un gran desembolso. La potencia se va a medir en el radio de cobertura que alcanza cada transmisor y la capacidad se va a medir en la cantidad de usuarios que pueden conectarse simultáneamente. El coste se calculará, por tanto, en función de ambos parámetros.

$$C = f (R, BW) \tag{3}$$

donde C es coste, R el radio de cobertura y BW el ancho de banda.

Para ello se van a definir una serie de perfiles, en el que cada cual es el resultado de una combinación de estos dos parámetros. A cada perfil se le asociará un coste,

resultado de la suma del coste de la cobertura y capacidad correspondiente que se le haya asociado.

## Algoritmo De Cobertura Y Coste

Una vez se hayan tenido en cuenta los criterios de optimización se va a desarrollar un algoritmo que calcule la cobertura en función del coste total (Ilustración 3). Para ello se introduce en el algoritmo una combinación de perfiles de las diferentes estaciones base. Con esta combinación de perfiles, que caracterizan cada estación base con un radio de cobertura y un ancho de banda, se podrá calcular el total de vehículos que tendrá cobertura. Para este cálculo se emplea el algoritmo explicado más adelante en la sección 5.2. Con él se obtiene como resultado, por cada diferente entrada, una combinación coste-cobertura, con la cual seremos capaces de discernir qué combinación de perfiles (input del algoritmo) nos conviene en función de nuestros intereses.

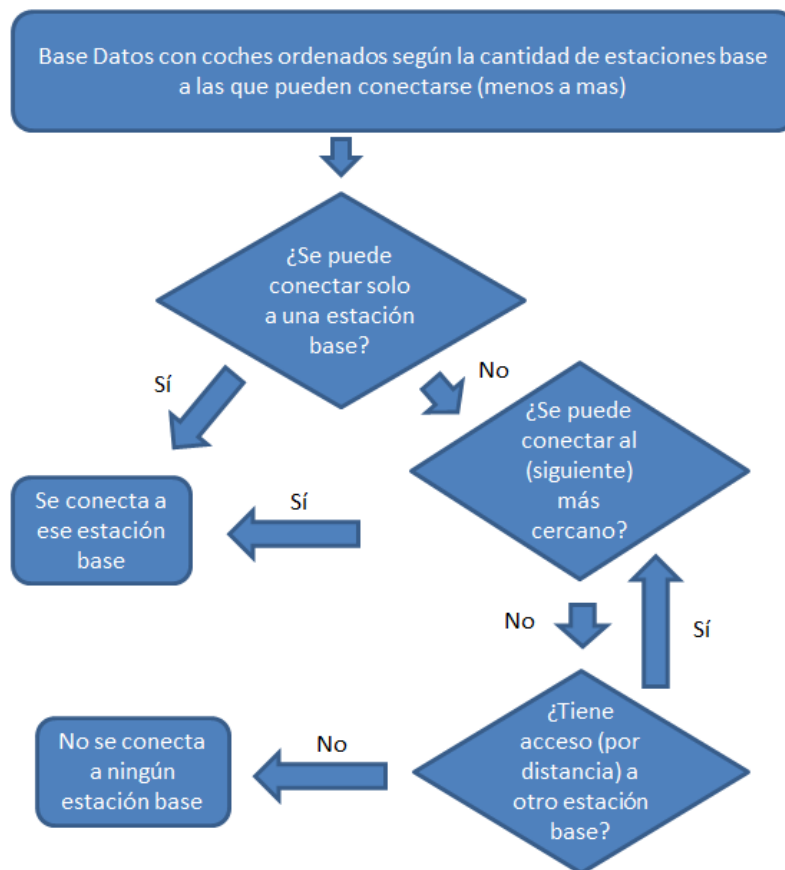


Ilustración 3: Algoritmo para cálculo de cobertura en función de los tipos de perfiles de las estaciones base



## Algoritmo Metaheurístico

La optimización se sustentará en un problema de Harmony Search Algorithm (HSA). Este algoritmo metaheurístico consiste en reiterar los cálculos hasta encontrar una solución adecuada. Para su cálculo se emplearán los perfiles previamente diseñados, de los cuales se asignará uno a cada estación base. Con cada estación base definida por un perfil concreto se calcula la cobertura que se tendrá para ese coste relativo a esos perfiles. El valor calculado se introducirá en el algoritmo HS, el cual seguirá los criterios de optimización de coste o cobertura que se le hayan programado. Si el resultado obtenido tras pasar por el algoritmo HS no fuera el buscado, volvería de nuevo a la asignación de perfiles. En caso contrario el valor de representaría en la gráfica de salida. Una vez finalizado el proceso y completada la gráfica se seleccionaría la configuración que se adaptara a nuestras exigencias de coste y/o cobertura.

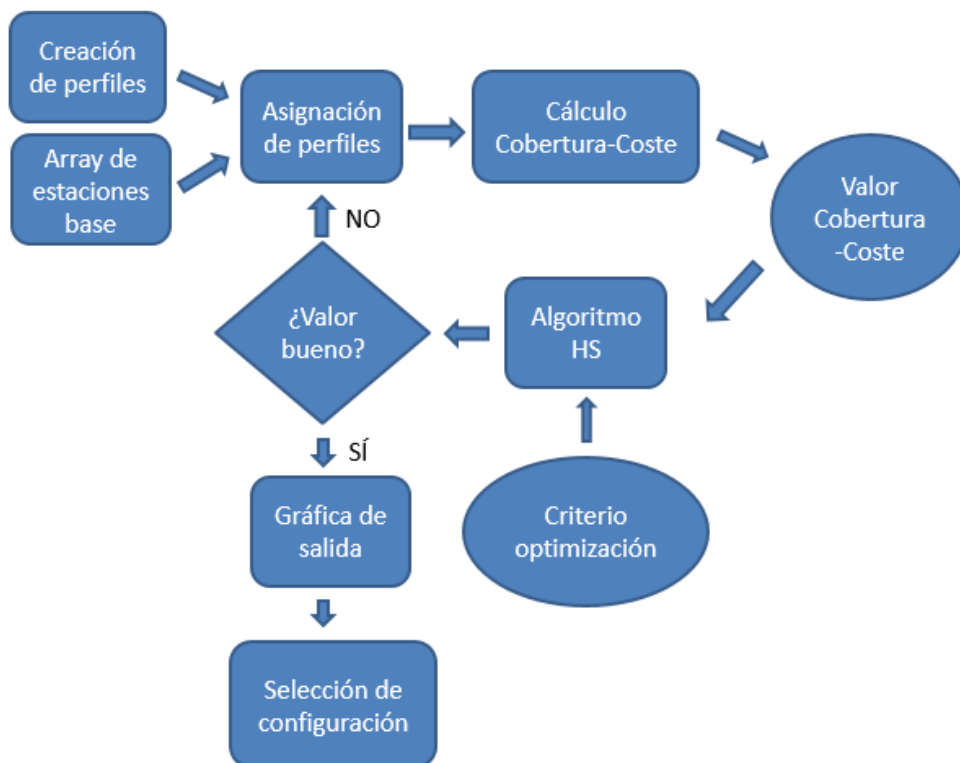


Ilustración 4: Funcionamiento algoritmo desarrollado en conjunto con el HSA

## 5. Cálculos

Este apartado, al igual que el anterior, se divide en las dos partes temáticas que abarca el proyecto. En primer lugar se describe el proceso llevado a cabo para diseñar y desarrollar el modelo de recogida de datos masivos procedentes de la red de carreteras. En este bloque se definen las estaciones base que se emplearán para la red y los núcleos urbanos que tendrán peso en la distribución de los vehículos a lo largo de las carreteras. En segundo lugar se describe el estudio llevado a cabo para optimizar la infraestructura de red empleando métodos metaheurísticos multiobjetivo. En este segundo bloque se analiza cómo maximizar la cobertura de nuestro escenario reduciendo los costes variables de la red.

### 5.1. Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas

La parte que se describe a continuación es la correspondiente a la publicada y presentada en el XXXI Simposium Nacional de la Unión Científica Internacional de Radio, URSI 2016, que se celebró en Madrid en el Campus de Cantoblanco de la EPS-UAM entre los días 5 y 7 de Septiembre de 2016.

Todos los códigos de los algoritmos descritos en este apartado (5.1) están en el Anexo II.

#### Modelado De Cobertura Y Carreteras

Siguiendo el método expuesto en el apartado anterior se comienza definiendo el mapa de estaciones base. Para ello se decide obtener la ubicación de los diferentes transmisores de TDT de la base de datos de Cellnex, antigua Abertis Telecom [18]. Una vez se ubican en el mapa estos 228 transmisores de TDT (Ilustración 3) se define un radio de cobertura de 10km para cada transmisor. Se elige este radio, porque se plantea un escenario realista para la futura conectividad V2X. Esta tecnología plantea conectividades de largo alcance. Se opta por definir un mismo rango de cobertura para todos los transmisores, simplificando así el modelo inicial. Mostrando en el mapa de la CAV los transmisores con su respectiva área de cobertura se aprecia que la cobertura será escasa en algunos puntos de Álava (Ilustración 5). Se obtiene la ubicación de las

antenas de TDT que pertenecen a Itelazpi [19]. Con otros casi 200 transmisores, 413 en total, se consigue ampliar la cobertura. Se alcanza a dar cobertura a la CAV casi en su totalidad, limitando las zonas que quedaban descubiertas (Ilustración 6).



Ilustración 5: Mapa con estaciones base

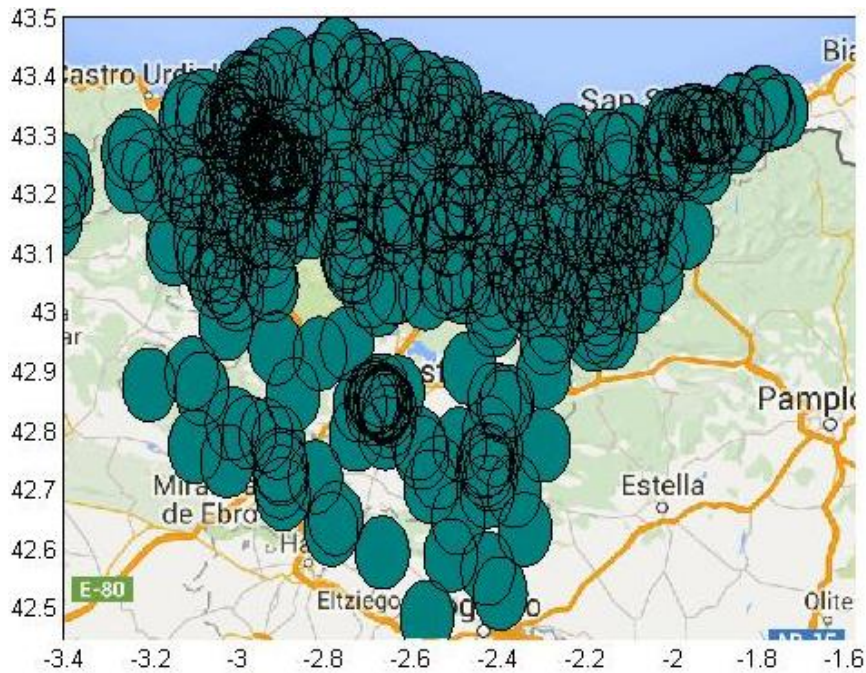


Ilustración 6: Mapa de la CAV con la Cobertura de cada Estación Base

Con las estaciones base ya definidas se procede a sectorizar el mapa en celdas. En primer lugar se decide sectorizar en celdas cuadradas, cada una con unas dimensiones de 20kmx20km (Ilustración 7). Esto se debe a que el alcance de las anteriores es de 10km, por tanto la celda será de las dimensiones del diámetro. Para optimizar la sectorización se emplea el algoritmo maximización de área explicado en el apartado 4.1. Con él se llega al resultado de que se maximiza el área de cobertura al emplear el transmisor situado más al norte, dando como resultado el siguiente mapa (Ilustración 8).



Ilustración 7: Sectorización Óptima

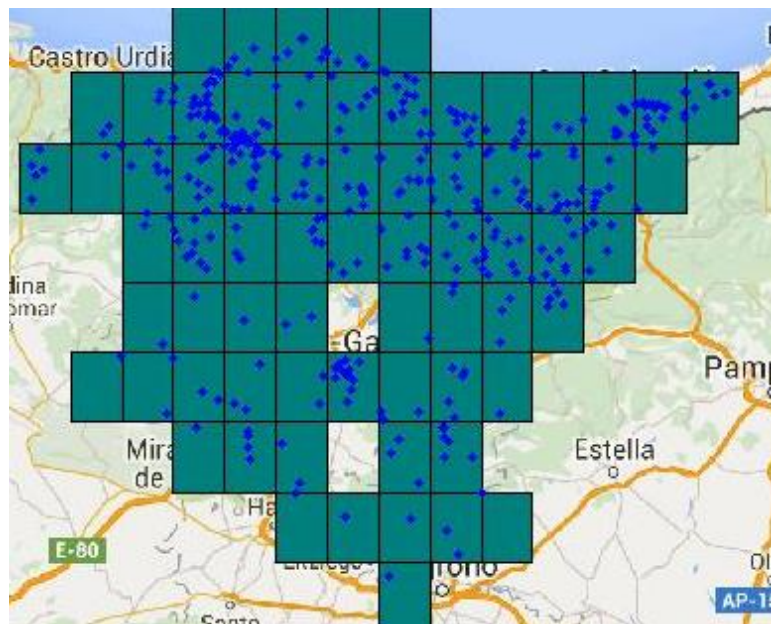


Ilustración 8: Mapa de la CAV sectorizado en zonas con al menos un transmisor (mapa de cobertura)



Con este resultado se genera una base de datos con una entrada para sector, en la cual se almacena, además de su identificador, su ubicación y si en ese sector hay cobertura o no. Para poder aumentar la resolución de la información que se va a recibir de los vehículos, se subsectoriza cada sector en 100 subsectores. Cada subsector tendrá una forma cuadrada de dimensiones 2kmx2km (Ilustración 9). Con estos nuevos datos se amplía la base de datos, ahora con una entrada por cada subsector (Ilustración 10). En ella se almacena la misma información que en la anterior, solo que ahora relativa a cada subsector. Como se puede apreciar en la ilustración 8 cada entrada de la base de datos tiene un id, el cual identifica a qué sector corresponde, un id\_sub, el cual identifica a qué subsector corresponde dentro del sector que identifica el id. Los parámetros  $x_{ini}$ ,  $x_{fin}$ ,  $y_{ini}$  e  $y_{fin}$  marcan los límites del subsector, siendo el punto  $(x_{ini}, y_{ini})$  el extremo superior izquierda y el punto  $(x_{fin}, y_{fin})$  el extremo inferior derecha. El último parámetro, cobertura, indica con un '1' si dentro de ese sector hay al menos una estación base, es decir, si en ese sector hay cobertura. En caso contrario el valor será '0'.

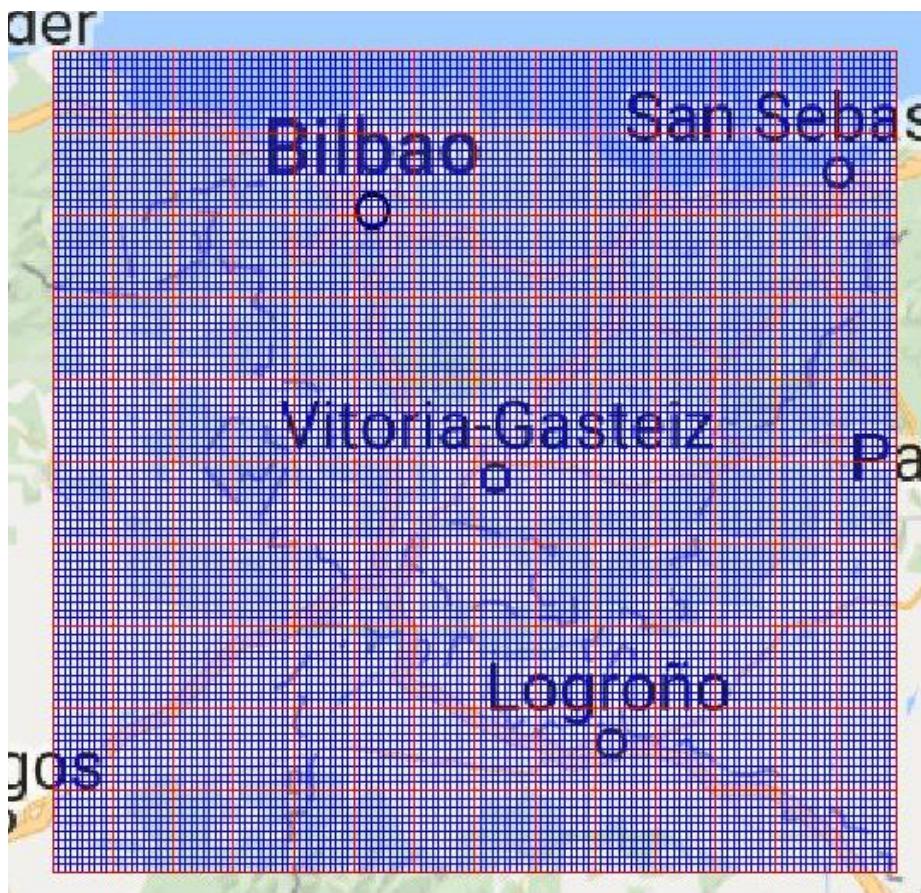


Ilustración 9: Subsectorización

Field ▲	Value	Min	Max
id	27	27	27
id_sub	80	80	80
x_ini	-1.9911	-1.9911	-1.9911
x_fin	-1.9787	-1.9787	-1.9787
y_ini	43.3034	43.3034	43.3034
y_fin	43.2911	43.2911	43.2911
cobertura	1	1	1

Ilustración 10: Información almacenada en la base de datos de los subsectores

## Modelado De Distribución De Vehículos

El siguiente paso a dar es situar las carreteras sobre nuestro escenario. Para ello se opta por descargar la información de la base de datos Open Street Maps. A través de ella se obtienen las diferentes polilíneas, las cuales incluyen la información de la ubicación de las carreteras y el tipo que son. En total se dispone de 34 tipos de vías y de más de 540.000 puntos en total (Ilustración 11). De entre todos los tipos de carreteras que hay se escogen únicamente las que son de nuestro interés (Ilustración 12). El resto, como por ejemplo bus-stop, pedestrian, etc., se descartan. Las escogidas se dividen en tres grupos, tal y como se indica en la tabla 2, en función de la afluencia de tráfico que tendrá cada una (Ilustración 13).

Tabla 1: División de tipos de carretera

Afluencia	Baja	Media	Alta
Vía	Tertiary Tertiary_link	Residential Road Secondary Secondary_link	Motorway Motorway_link Primary Primary_link



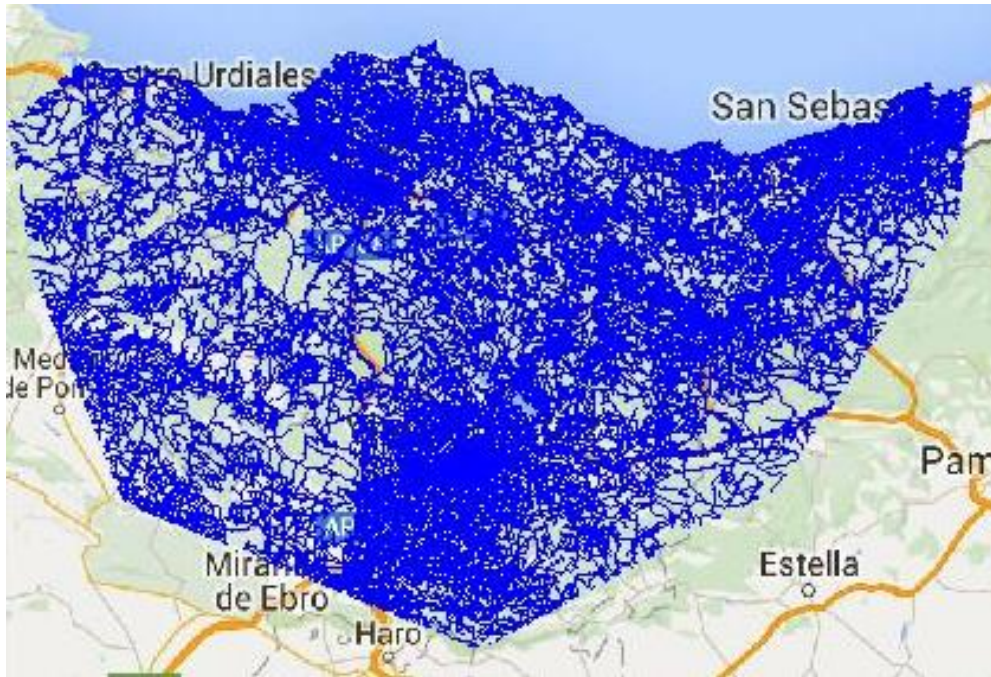


Ilustración 11: Carreteras sin filtrado



Ilustración 12: Mapa de la CAV con carreteras de interés

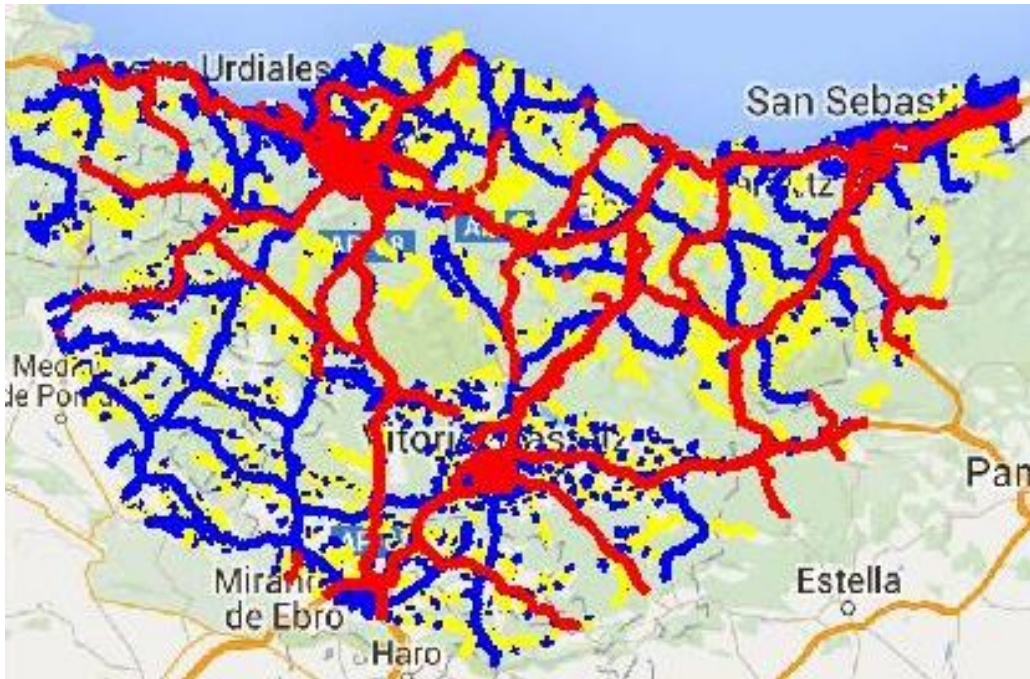


Ilustración 13: Mapa de la CAV con las carreteras de interés divididas en tres grupos (rojo más afluencia, azul afluencia intermedia y amarillo afluencia baja)

Una vez hecha la división se completa la base de datos de los subsectores. En esta base de datos se le añade a cada subsector los datos de las carreteras que están en su interior: identificador, ubicación y tipo de tráfico (Ilustración 14). La forma de añadirlos es mediante 4 arrays. El primer array, `datos_carretera_id`, es el identificador de cada punto de carretera. Los dos siguientes arrays, `datos_carretera_X` y `datos_carretera_Y`, son las coordenadas de cada punto de carretera. Y por último, el array `datos_carretera_tipo`, es el tipo (1, 2 o 3) de carretera según afluencia.

Field ▲	Value	Min	Max
id	28	28	28
id_sub	71	71	71
x_ini	-1.9787	-1.9787	-1.9787
x_fin	-1.9664	-1.9664	-1.9664
y_ini	43.3034	43.3034	43.3034
y_fin	43.2911	43.2911	43.2911
cobertura	1	1	1
datos_carretera_id	<1x1405 double>	1060	345130
datos_carretera_X	<1x1405 double>	-1.9787	-1.9664
datos_carretera_Y	<1x1405 double>	43.2911	43.3034
datos_carretera_tipo	<1x1405 double>	2	3

Ilustración 14: Base de datos completada con la información de las carreteras



Con las carreteras ya definidas y situadas en nuestro modelo, el siguiente paso es distribuir de forma realista los vehículos. Para ello se obtiene una base de datos del organismo Eustat con las poblaciones más habitadas (>15.000 personas) y su parque automovilístico correspondiente. En total se trata de veintinueve poblaciones (Tabla 2) repartidas entre las tres provincias, entre las cuales poseen casi un millón de vehículos (946.694). Aunque el dato concreto de los vehículos en cada población no se encontró, sí se encontró la cantidad de vehículos existentes en cada una de las tres provincias (Bizkaia: 493.456, Gipuzkoa: 304.802 y Araba: 148.436). Se decidió dividir de forma proporcional la cantidad de coches de cada provincia entre las diferentes poblaciones de cada una en función de su población. Es decir, si por ejemplo el total de la población de Bizkaia fuera de 100 personas y la población de un núcleo urbano fuera 20, a esa población se le asignará el 20% de los vehículos totales existentes en Bizkaia.

**Tabla 2: Municipios y su respectiva población**

Ciudad	Población	Ciudad	Población	Ciudad	Población
Bilbao	343.234	Leioa	30.642	Lasarte-Oria	18.444
Vitoria-Gasteiz	240.699	Galdakao	29.284	Laudio /Llodio	18.404
Donostia/San Sebastián	180.291	Durango	29.109	Amorebieta-Etxano	18.355
Barakaldo	98.964	Sestao	27.578	Hondarribia	16.999
Getxo	78.124	Eibar	27.204	Mungia	16.965
Irun	59.624	Erandio	23.950	Bermeo	16.752
Portugalete	46.700	Zarautz	22.915	Gernika-Lumo	16.493
Santurtzi	45.962	Arrasate/Mondragón	21.933	Pasaia	16.418
Basauri	41.174	Hernani	19.923	Ermua	16.036
Errenteria	39.813	Tolosa	18.993	-	-

Para distribuir los vehículos se opta por dividir el área de influencia de cada una de las 29 poblaciones en 25 anillos concéntricos, cada uno con 1km de diferencia entre rint y rext (Ilustración 15). En cada anillo se distribuirá la proporción de área bajo la curva entre los dos radios que forman el anillo respecto al total del área bajo la

curva (Ilustración 16). En el área encerrada en cada anillo se distribuirán los vehículos de la siguiente forma. La cantidad de vehículos asignada a cada área entre anillos se distribuyen de forma aleatoria, teniendo la vía de afluencia alta y media tres y dos veces, respectivamente, más probabilidades de que se les asigne un vehículo que la de afluencia baja.

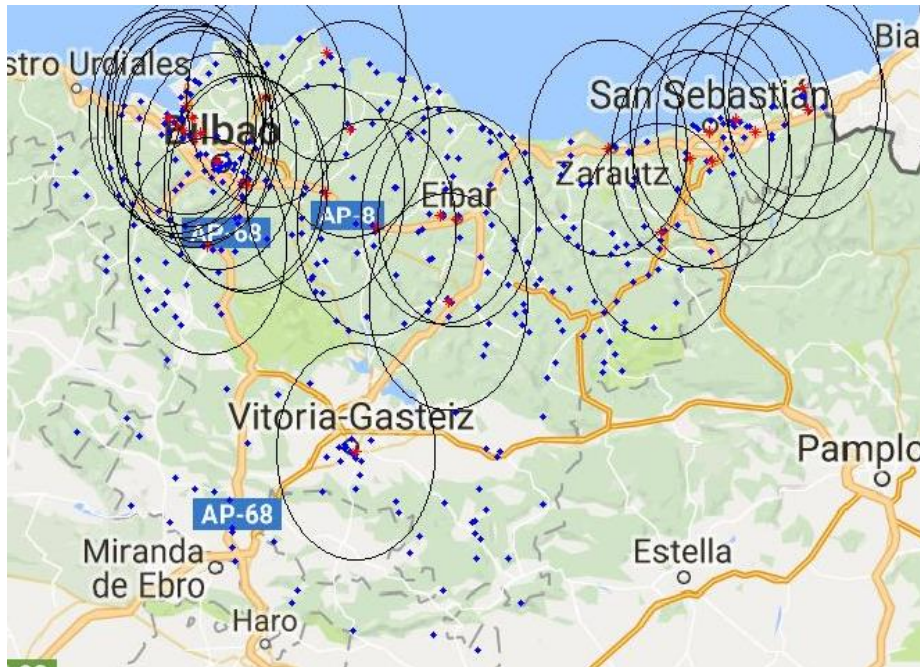


Ilustración 15: Mapa con poblaciones (Rojo) y el alcance de la distribución de los vehículos (Circulo)

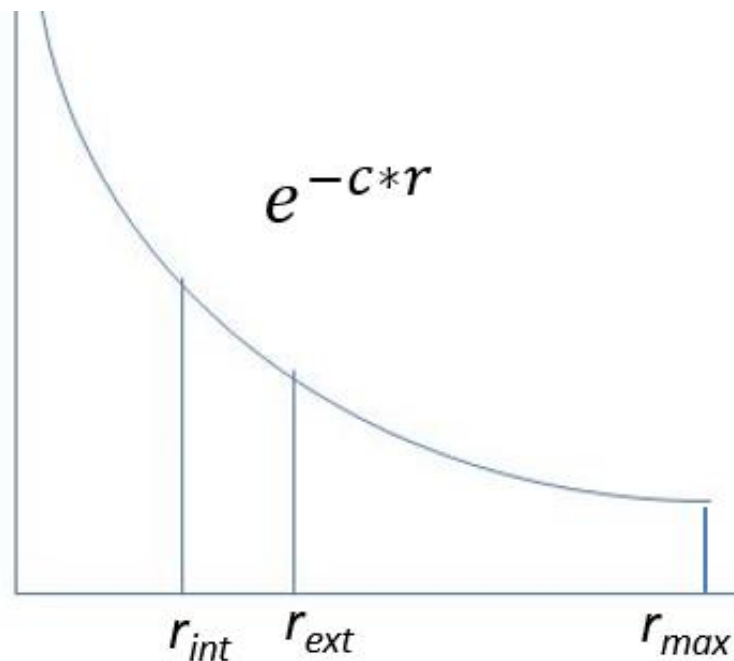


Ilustración 16: Descripción del porcentaje de vehículos en el interior de un anillo

Debido a que no se emplean en todo momento todos los vehículos del parque automovilístico, se propone emplear el factor corrector de la ecuación 2. Este factor estima qué porcentaje del total de vehículos estará en circulación en cada periodo horario. Para este caso concreto se emplearán los valores de la tabla 3. Estos datos se han estimado conociendo las horas de mayor afluencia de vehículos y calculando un porcentaje aproximado del parque automovilístico movilizado para los desplazamientos necesarios.

**Tabla 3: Factor corrector C**

Hora	Factor corrector
00:00-6:59	0.3
7:00-10:59	0.7
11:00-12:59	0.5
13:00-15:59	0.6
16:00-17:59	0.5
18:00-20:59	0.6
21:00-23:59	0.4

Se multiplica el factor corrector por el porcentaje correspondiente a cada área y éste por la cantidad total de vehículos de la población. Así obtenemos el número total de vehículos a distribuir de forma aleatoria en el área en un momento determinado. Además, para corregir posibles errores al no tener en cuenta las poblaciones más pequeñas, se distribuirán de forma aleatoria en todo el mapa un 10% del total de vehículos distribuidos con el algoritmo. Es decir, si suponemos que la hora simulada es entre las 7 y las 10, se añadirán 66.269 ( $0.1 \times 0.7 \times 946694$ ) vehículos extra.

Finalmente se creará una base de datos que almacene por cada subsector la información correspondiente a todos los vehículos que se encuentren en él, de tal forma que simplifique notablemente la manipulación de datos. Esta base de datos

permitirá saber cuántos vehículos hay en cada punto de carretera (puntos pertenecientes a las polilíneas), puntos ubicados en el subsector almacenado. En la ilustración 17 se puede ver los datos del subsector 41 perteneciente al sector 31. Se puede observar cómo se ha añadido un array (`datos_carretera_vehículos`) respecto a la ilustración 12, la cual almacena la información sobre la cantidad, organizado por cada punto de carretera, de los vehículos dentro de ese subsector. En este caso concreto se observa como el punto de carretera con más vehículos es de 8 y el de menos de 0. El objetivo sería comenzar el estudio de estrategias de recolección y tratamiento de datos masivos a partir de ella.

Field ▲	Value	Min	Max
id	31	31	31
id_sub	41	41	41
x_ini	-3.3367	-3.3367	-3.3367
x_fin	-3.3244	-3.3244	-3.3244
y_ini	43.2170	43.2170	43.2170
y_fin	43.2047	43.2047	43.2047
cobertura	1	1	1
datos_carretera_id	<1x13 double>	411891	411903
datos_carretera_X	<1x13 double>	-3.3367	-3.3336
datos_carretera_Y	<1x13 double>	43.2048	43.2078
datos_carretera_tipo	<1x13 double>	1	1
datos_carretera_vehiculos	<1x13 double>	0	8

Ilustración 17: Base de datos con la información de los vehículos en cada subsector

Se puede comprobar cómo el número de vehículos distribuidos coincide con los vehículos que deberían distribuirse. En las siguientes dos tablas disponemos de dos ejemplos en dos diferentes periodos de tiempo. En la tabla 3 los datos corresponden a la hora punta, es decir, entre las 7:00 y las 10:59. Se comprueba cómo efectivamente se distribuye en total un 70 % del millón de vehículos que existe en la CAV. En la tabla 4 los datos se corresponden al periodo horario entre las 21:00 y las 23:59. Se comprueba de nuevo como efectivamente en este caso los vehículos distribuidos son el 40% del total. Tanto el 70% como el 40% son los datos obtenidos de la tabla 2, la tabla que define el factor corrector C.

Tabla 4: Lista de municipios y la cantidad de vehículos distribuidos entre las 7:00-10:59

Ciudad	Vehículos
Bilbao	144.462
Vitoria-Gasteiz	103.417
Donostia / San Sebastián	93.125
Barakaldo	41.654
Getxo	32.881
Irun	30.797
Portugalete	19.654
Santurtzi	19.345
Basauri	17.330
Errenteria	20.566
Leioa	12.897
Galdakao	12.325
Durango	12.251
Sestao	11.608
Eibar	14.050
Erandio	10.079
Zarautz	11.836
Arrasate/Mondragón	11.328
Hernani	10.290
Tolosa	9.810
Lasarte-Oria	9.528
Laudio/Llodio	7.908
Amorebieta-Etxano	7.726
Hondarribia	8.780
Mungia	7.140
Bermeo	7.051
Gernika-Lumo	6.942
Pasaia	8.482
Ermua	6.750
<b>Total</b>	<b>710.012</b>

Tabla 5: Lista de municipios y la cantidad de vehículos distribuidos entre las 21:00-23:59

<b>Ciudad</b>	<b>Vehículos</b>
Bilbao	82.550
Vitoria-Gasteiz	59.095
Donostia / San Sebastián	53.214
Barakaldo	23.802
Getxo	18.789
Irun	17.598
Portugalete	11.231
Santurtzi	11.054
Basauri	9.903
Errenteria	11.752
Leioa	7.370
Galdakao	7.043
Durango	7.001
Sestao	6.633
Eibar	8.029
Erandio	5.759
Zarautz	6.763
Arrasate/Mondragón	6.473
Hernani	5.880
Tolosa	5.606
Lasarte-Oria	5.445
Laudio/Llodio	4.519
Amorebieta-Etxano	4.415
Hondarribia	5.017
Mungia	4.080
Bermeo	4.029
Gernika-Lumo	3.967
Pasaia	4.847
Ermua	3.857
<b>Total</b>	<b>405.721</b>

## 5.2. Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo

La parte que se describe a continuación es la correspondiente a la enviada al III congreso internacional del *Harmony Search Algorithm (ICHSA)*. Este congreso tendrá lugar entre los días 22 y 24 de Agosto de 2017 en el edificio *Bizkaia Aretoa* de la UPV/EHU, Bilbao.

Todos los códigos de los algoritmos descritos en este apartado (5.2) están en el Anexo II.

### Criterios De Optimización

En primer lugar se van a definir las características de los diferentes perfiles posibles que pueda tomar cada transmisor. Estos perfiles, como se ha comentado en el apartado anterior, serán una combinación de dos datos: potencia y capacidad. Esta elección se debe a que son los dos parámetros que podemos controlar con mayor facilidad. Variar el radio de cobertura es tan simple como modificar la potencia que se le inyecta a la antena, mientras que variar el ancho de banda disponible dependerá del espectro que se contrate. Por motivos económicos se descarta contemplar la opción de variar la ubicación de las estaciones base, ya que supondría una costosa obra civil. De esta forma estamos buscando minimizar el coste variable, es decir, el coste que deriva del uso de las estaciones base. La elección del radio de cobertura y del ancho de banda de cada estación base será determinante a la hora de calcular el coste, objeto de optimización en este caso. Para poder tener un criterio de optimización económico lo primero que es necesario definir es la fórmula que calcule el coste en función de ambos parámetros (Fórmula 4).

$$C = 0,5 * BW + 10 * \pi * R^2 \quad (4)$$

donde C es el coste, BW la capacidad medida en coches y R el radio de cobertura en km.

El siguiente paso consiste en la definición de los perfiles (Tabla 5). Para ello se van a combinar tres posibles valores de cada parámetro. Para la capacidad los posibles valores serán 1000, 2000 y 5000 vehículos y para la cobertura serán 5, 10 y 30 km de radio. Además se considerará un perfil en el cual ambos valores sean 0, y por tanto el coste será 0. Este perfil representa el caso posible de que una estación base se encuentre apagada. Se muestran los perfiles en la tabla 4.

**Tabla 6: Perfiles posibles para cada estación base**

Perfil	Radio	Coches	Coste (€)
1	5	1000	1285,40
2	5	2000	1785,40
3	5	5000	3285,40
4	10	1000	3641,59
5	10	2000	4141,59
6	10	5000	5641,59
7	30	1000	28774,33
8	30	2000	29274,33
9	30	5000	30774,33
10	0	0	0,00

## Algoritmo De Cobertura Y Coste

Una vez definidos los perfiles y conociendo la cantidad exacta de estaciones base, se desarrolla un algoritmo con las siguientes características (Ilustración 3). El algoritmo, mediante la entrada de una combinación de perfiles para cada estación base, debe ser capaz de calcular la cobertura que ese caso concreto dará a un coste asignado. Es decir, si introducimos en el algoritmo un array de 413 posiciones (la cantidad de estaciones base) y en cada posición hay un número del 1 al 10 (los posibles perfiles), el algoritmo debe calcular la cobertura que se tendrá para ese caso concreto. El cálculo del coste total de ese caso es tan simple como acceder a la tabla con los perfiles del array y sumar su coste. Repitiendo este cálculo con varias entradas podremos acercarnos al resultado óptimo. El algoritmo que se emplea para conocer la cobertura es el siguiente.

En primer lugar se ordenan los vehículos en la base de datos. El criterio para ordenarlos, de menor a mayor, es la cantidad de estaciones base a las cuales pueda conectarse. Es decir, el vehículo que se encuentre situado dentro del área de cobertura de las menos estaciones bases será el primero. Esto se debe a que un vehículo que se



encuentre en el área de cobertura de una única estación base solo podrá conectarse a esa estación. De tal forma que los siguientes vehículos, que podían conectarse a más de una estación base, no limiten el acceso a los que únicamente puedan conectarse a una sola estación base. Se comienza, coche a coche, con el criterio de decisión. Los vehículos que solo están en el área de cobertura de una estación se conectan a esa en caso de que sea posible, ya que las estaciones base tendrán un límite de vehículos que se puedan conectar. Después se aplicará el algoritmo para los vehículos que estén situados en el área de cobertura de dos estaciones base. En este caso tratarán de conectarse a la estación base más cercana. En caso de no ser posible intentará conectarse a la otra estación base. Si no pudiera, porque ambas estaciones base han alcanzado su límite de capacidad, este vehículo computará como un vehículo sin cobertura. El proceso se repite con los vehículos que están en el área de cobertura de tres estaciones base y así sucesivamente. Cuando el algoritmo finaliza podremos saber cuántos vehículos se han podido conectar y cuántos no. De esta forma podremos asociar un valor de cobertura al valor del coste definido por el array de perfiles.

## Algoritmo Metaheurístico

Por tanto, introduciendo un array con 413 posiciones en el algoritmo, en las cuales habrá un perfil para cada estación base, se obtendrá como resultado una cobertura y un coste, asociado a ese array. Esto da lugar a una combinatoria muy elevada,  $10^{413}$ , de los diferentes perfiles que se pueden asignar a cada uno de las estaciones base. A cada combinación de perfiles le corresponde un único valor coste-cobertura. Tras repetir el proceso numerosas veces se obtendrá como resultado una gran cantidad de combinaciones de 413 perfiles, de los cuales conoceremos coste y cobertura. Con estos datos podremos escoger la combinación de 413 perfiles que más se ajuste a nuestros intereses. Por ejemplo podremos optimizar la red con un criterio de minimizar el coste, siempre y cuando se supere el 90% de cobertura. O se puede optimizar con un criterio de máximo presupuesto, en el cual únicamente podamos gastar un dinero limitado y queramos maximizar la cobertura sin superar ese coste. Si se representaran los resultados de cada iteración en una gráfica se obtendría un resultado como el de la ilustración 18. Donde el eje y es cobertura, el eje x coste, los puntos A y B son dos combinaciones que forman parte del Pareto y el punto C es un punto del interior.

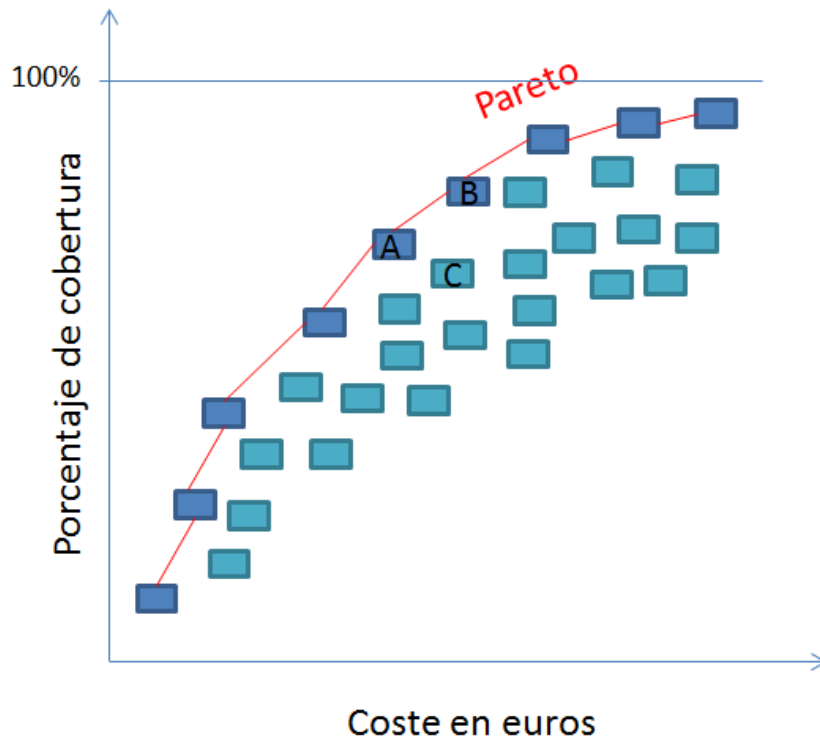
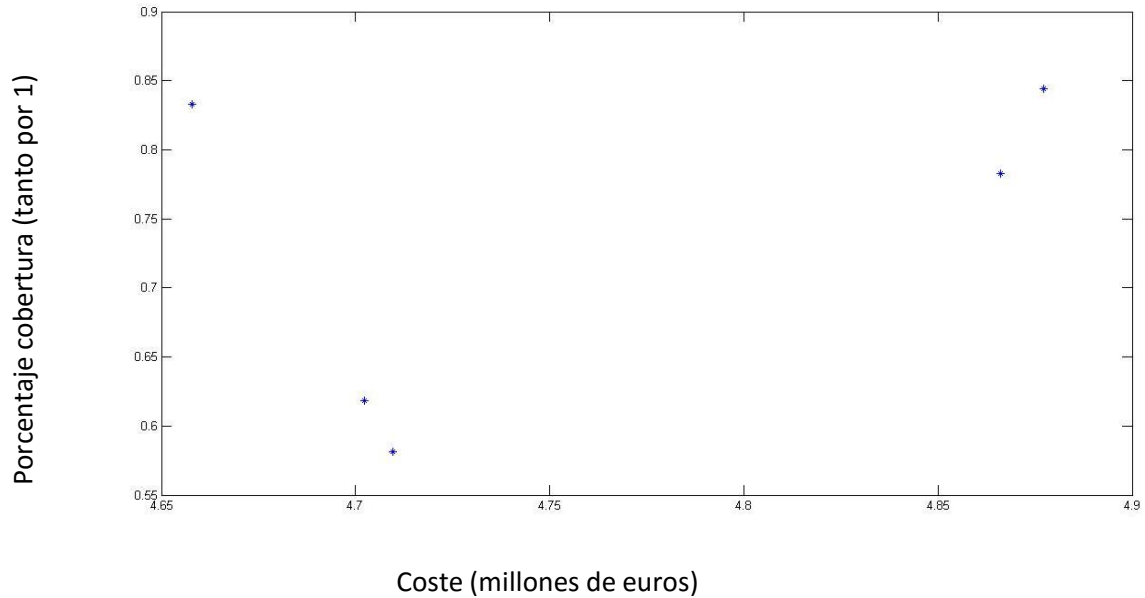


Ilustración 18: Curva con los resultados de las iteraciones

Cada punto de la gráfica representa una iteración del algoritmo. En él se obtiene como resultado una cobertura para un coste dado. Evidentemente, según cómo se diseñen los diferentes perfiles de las estaciones base, podría darse el caso que para un coste superior se tuviera una cobertura inferior. Esos casos son los que están por debajo del Pareto, como es el caso del punto C. En cambio los puntos del Pareto son los que optimizan ambos parámetros, es decir, para un coste dado maximizan la cobertura que se pueda obtener con ese límite. Por tanto, si quisiéramos la cobertura que ofrece el punto B, tendríamos que invertir su coste. En cambio, si ese coste fuera superior al que se puede invertir, habría que escoger el punto A, el cual tiene menor cobertura, pero menor coste también.

La implementación iterativa del algoritmo HS, en base a los datos de cobertura y de coste proporcionados por el algoritmo desarrollado en este trabajo, es algo que queda fuera del alcance de este proyecto, debido principalmente a la elevada complejidad computacional que implica el elevado número de iteraciones que realiza para buscar soluciones adecuadas. A modo de ejemplo, en la ilustración 19 se presenta un ejemplo de 5 iteraciones aleatorias iniciales en las que se han elegido unos perfiles iniciales para la red de 413 estaciones base y se han ido obteniendo diferentes valores

para la relación cobertura-coste.



**Ilustración 19: Resultado tras 5 iteraciones aleatorias del algoritmo de cálculo de coste y cobertura**

En esta gráfica podemos apreciar 5 puntos, los cuales unen un porcentaje de cobertura (eje y) y un coste (eje x). En este caso el punto que más se aproxima a una posible eficiencia del Pareto será el punto situado más a la izquierda. Este punto presenta una cobertura del casi el 85% para un coste algo superior a los 4,65 millones de euros. El resto de puntos ofrece una cobertura inferior para un coste superior, siendo por tanto puntos de trabajo de una eficiencia inferior.

## 6. Aspectos Económicos

### 6.1 Presupuesto

#### Costes Unitarios

En este apartado se valorarán los costes unitarios de cada uno de los elementos que toman parte en este proyecto.

#### Mano de Obra

En este apartado se mostrará una tabla en la que aparece la relación integrante del proyecto-coste unitario de mano de obra por hora. El coste de los salarios está extraído de las Disposiciones Generales del Departamento de Industria, Innovación, Comercio y Turismo.

Tabla 7: Coste personal

Nombre	Cargo	Horas	Coste (€/hora)	Coste (€)
Alonso, Borja	Desarrollador del proyecto	600	25	15.000
Vélez, Manolo	Jefe de proyecto	20	50	1.000
Sobrón, Iker	Diseñador de proyecto	50	35	1.750

#### Recursos Materiales

A continuación se muestra una lista de los recursos materiales empleados en la elaboración del proyecto, indicando el coste de cada uno de ellos.

**Amortizables:** Incluiremos en esta partida todo el material existente en el grupo de investigación y que por tanto, no tenemos que adquirir. Los materiales están repetidos porque cada uno de los programadores hará uso de un ordenador que tiene sus propias licencias. Son los siguientes:

**Tabla 8: Coste material**

Objeto	Precio (€)	Amortización (meses)	Coste (€)
Ordenador Acer	720	36	120
Impresora	300	36	50

**Fungibles:** En esta categoría incluimos todos aquellos bienes y recursos que no van a poder ser reutilizados tras realizar el proyecto. Por ejemplo, en este grupo podemos encontrar material de oficina, luz, gasolina... El coste de este tipo de bienes se ha estimado en 500 euros, cantidad que se tendrá en cuenta para calcular el presupuesto del proyecto.

## Otros Gastos

Aquí incluiremos otros costes derivados del proyecto que a priori no podemos detallar. En este apartado de gastos estarían los Viajes de representación, comidas por trabajo... En principio vamos a asignar a este concepto unos gastos de 1500 euros que habrá que contabilizar en el presupuesto final.

## Cálculo Del Presupuesto

**Tabla 9: Presupuesto**

Concepto	Coste (€)
RR.HH.	17.750,00
Recursos Materiales	170,00
Otros Gastos	1500,00
Fondos de Contingencia (5%)	1.000,00
<b>Presupuesto Total</b>	<b>20.420,00</b>

## 7. Conclusiones

En el presente documento se expone un proyecto consistente en un modelo de recogida y gestión de datos masivos procedentes de sensores en vehículos de la red de carreteras vascas. Tras la elaboración del mismo se llega a las siguientes conclusiones.

En primer lugar el desarrollo de este proyecto surge de la necesidad de modelar de forma realista un escenario que permita simular la distribución de los vehículos en la red de carreteras de la CAV de tal forma que sea posible conocer la magnitud de datos con la cual se va a trabajar. Debido a que no existe un escenario real con una implantación similar, es necesario contar con esta plataforma que sirva como base para un posterior desarrollo de estrategias adaptadas, que permita optimizar la red vehicular. Además este proyecto aborda la tarea de optimizar el modelo de red necesario para proveer de servicio a los vehículos, consiguiendo un rendimiento alto teniendo en cuenta las características de la red.

Por otra parte gracias al trabajo de este proyecto se puede conseguir mejorar significativamente aspectos sociales muy importantes. La seguridad vial, que a día de hoy supone uno de los índices más altos de siniestralidad de la población, puede verse mejorada de forma notable aplicando diferentes estrategias que se sustenten en este proyecto. También puede suponer una revolución en el cuidado del medioambiente, ya que nos va a proporcionar una serie de datos de gran significado para poder abordar la contaminación.

Este proyecto es además fácilmente escalable a otros escenarios tanto a nivel local (ciudades) o regional (otras provincias o comunidades autónomas). Esto permite tener una penetración muy alta en el mercado sin casi aumentar los costes en desarrollo, posibilitando una implementación masiva de redes vehiculares optimizadas. También es escalable a otros tipos de redes, con lo cual la optimización podría realizarse también en redes de telefonía.

## 8. Bibliografía

- [1] <https://www.celticplus.eu/>, 29/09/2016
- [2] Zong Woo Geem, Joong Hoon Kim and G. V. Loganathan, "TECHNICAL ARTICLE A New Heuristic Optimization Algorithm: Harmony Search"
- [3] C. K. S. Leung, R. K. MacKinnon and F. Jiang, "Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data," *2014 IEEE International Congress on Big Data*, Anchorage, AK, 2014, pp. 315-322.
- [4] M. Kulkarni, S. M. Sundaram and V. Diwakar, "Development of sensor and optimal placement for smoke detection in an electric vehicle battery pack," *Transportation Electrification Conference (ITEC), 2015 IEEE International*, Chennai, 2015, pp. 1-3.
- [5] R. S. Mohan, R. Sachin and U. Sakthivel, "Vehicular Ad Hoc Network Based Pollution Monitoring in Urban Areas," *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*, Mathura, 2012, pp. 214-217.
- [6] Y. Higuchi and H. Hontani, "A System of Networked Car-Mounted Sensors for Measuring Road Surfaces," 2006 SICE-ICASE International Joint Conference, Busan, 2006, pp. 1310-1315.
- [7] M. R. Palattella et al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510-527, March 2016. doi: 10.1109/JSAC.2016.2525418
- [8] A. Gohil, H. Modi and S. K. Patel, "5G technology of mobile communication: A survey," *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on*, Gujarat, 2013
- [9] Kyoungsoo Bok, Seungwan Hong, Jongtae Lim and Jaesoo Yoo, "A multiple RSU scheduling for V2I-based data services," *2016 International Conference on Big Data and Smart Computing (BigComp)*, Hong Kong, 2016, pp. 163-168.
- [10] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro and E. Cerqueira, "Towards software-defined VANET: Architecture and services," *Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean*, Piran, 2014, pp. 103-110.
- [11] J. J. Kponyo, Y. Kuang, E. Zhang and K. Domenic, "VANET Cluster-on-Demand Minimum Spanning Tree (MST) Prim clustering algorithm," *Computational Problem-solving (ICCP), 2013 International Conference on*, Jiuzhai, 2013, pp. 101-104.
- [12] R. Li, W. Feng, Z. Xu and S. Pan, "A mathematical simulation model for access traffic of geospatial data," *Computer Science & Education (ICCSE), 2012 7th International Conference on*, Melbourne, VIC, 2012, pp. 1127-1129.



- [13] S. Busanelli, G. Ferrari and R. Gruppini, "Performance analysis of broadcast protocols in VANETs with Poisson vehicle distribution," *ITS Telecommunications (ITST), 2011 11th International Conference on*, St. Petersburg, 2011, pp. 133-138.
- [14] S. Willeke, G. Ullmann and P. Nyhuis, "Method for an Energy-Cost-Oriented Manufacturing Control to Reduce Energy Costs: Energy Cost Reduction by Using a New Sequencing Method," *2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*, Jeju, 2016, pp. 1-5.
- [15] <http://rfcas.eps.uam.es/ursi2016/>, 29/09/2016
- [16] Open Streets Maps, <https://www.openstreetmap.org>, 29/09/2016
- [17] <http://www.ichsa2017.com/>, 29/09/2016
- [18] Itelazpi, <http://www.itelazpi.eus/>, 29/09/2016
- [19] Cellnex, <https://www.cellnextelecom.com/>, 29/09/2016

# Anexo I: Pliego De Condiciones. Normativa Aplicable

## Condiciones Generales

### Aplicación De Normas

El presente pliego de condiciones pretende establecer las condiciones de instalación y uso del producto, con el fin de obtener el mayor rendimiento haciendo un buen uso del mismo.

### Desarrollo Del Proyecto

El desarrollo del proyecto depende entre otras consideraciones, de la interpretación técnica de los documentos que lo comprenden. Y estará al cargo del mismo el director del proyecto. El contratista estará obligado a recurrir al director del proyecto con cualquier duda o aclaración que surgiera durante su ejecución. El contratista se hará responsable de cualquier fallo o mala ejecución del proyecto motivado por la omisión de la obligación y por lo tanto las modificaciones pertinentes correrán a cargo del contratista.

### Alteraciones Y Modificaciones

El director técnico tiene la facultad de introducir modificaciones de acuerdo con su propio criterio, siempre que se cumplan las condiciones técnicas referidas en el presente proyecto y de forma que no varíe el importe del mismo.

### Materiales Auxiliares

Los materiales auxiliares correrán a cargo del contratista, quien proveerá de todos los que hicieran falta para la realización del proyecto.

## Condiciones Técnicas

### Software De Desarrollo

Para la realización del proyecto se establece como plataforma de desarrollo el software MatLab. Se ha de emplear la versión MATLAB 7.13 (R2011b) o superior.

## Condiciones Facultativas

### Ejecución De La Obra

La realización del proyecto se llevará a cabo entre los meses de marzo a septiembre, del año 2016, ambos inclusive; como se establece en el contrato firmado por ambas partes.

Cuando el ritmo normal de trabajo establecido por el contratista no sea normal o bien por petición de alguna de las partes se podrá preparar una programación de inspecciones obligatorias de acuerdo con el ritmo de la obra.

### Recepción De La Obra

Una vez finalizado el desarrollo del proyecto se hace entrega al director técnico que lo revisará y lo entregará a la empresa; a partir de este día comienza el plazo de garantía

### Plazo De Garantía

El plazo de garantía será de 1 mes, octubre de 2016.

## Condiciones Económicas

### Liquidación En Caso De Rescisión

Siempre que se requiera rescindir el contrato por las causas mencionadas en las condiciones administrativas o bien por acuerdo entre ambas partes se abandonará a la empresa el proyecto y el material adquirido para su desarrollo.

Cuando se rescinda el contrato llevará implícito la retención de la fianza para los posibles gastos de conservación en el plazo de garantía y los derivados del mantenimiento.

## Fianza

En el contrato se deberá fijar la fianza que el contratista deberá disponer en garantía del cumplimiento o se convendrá una retención sobre los pagos realizados a cargo de obra ejecutada. De no estipularse la fianza en el contrato, se entiende que se adoptará como garantía una retención del 5% sobre el pago a cargo del contratista.

## Revisión De Precios

En el contrato formalizado se establecerá si el contratista tiene derecho a la revisión de precios y la manera de aplicarlos. En defecto de esta última, se aplicará a juicio del director técnico algunos de los criterios oficiales aceptados.

## Infracciones

Por motivos de retardo en los plazos de entrega se podrán establecer penalizaciones que tendrán un coste y un retraso fijado en el contrato.

## Condiciones Administrativas

### Personal Contratado

La persona que realice el desarrollo del proyecto deberá disponer de todos los permisos necesarios para la ejecución.

### Conservación De Las Obras

La conservación del proyecto será la obligación del contratista, realizando las inspecciones necesarias para un buen seguimiento de la obra y su conservación, hasta la fecha de la recepción definitiva por el contratista. Irán a su cargo los gastos propios del mantenimiento.

### Contrato

El contrato se realizará mediante un documento privado, que podrá elevarse a escritura pública por petición de una de las dos partes.  
La totalidad de los documentos que componen el proyecto técnico incorporados al contrato y tanto el contratista como el director deberán firmar en testimonio de que los conocen y los aceptan.

## Responsabilidades

El contratista es el responsable de la ejecución del proyecto en las condiciones establecidas.

## Rescisión Del Contrato

Causas de rescisión; se consideran causas suficientes para la rescisión del contrato las siguientes:

- Primera: muerte o incapacitación del contratista
- Segunda: la quiebra del contratista.
- Tercera: modificaciones del proyecto cuando se produzcan alteraciones superiores al 20% del valor contratado.
- Cuarta: el no comienzo del proyecto en el plazo estipulado cuando sea por causas ajenas a la propiedad.
- Quinta: incumplimiento de las condiciones del contrato cuando ello implique mala fe.
- Sexta: finalización del plazo de ejecución sin que el proyecto responda a las exigencias de producción.
- Séptima: subcontratar la totalidad o parte del desarrollo, sin la autorización del director técnico del proyecto y la propiedad.

# Anexo II: Planos Y Esquemas

## Código de los algoritmos desarrollados en el Modelo De Recogida De Datos Masivos Para La Monitorización Ambiental De La Red De Carreteras Vascas

### Sectorización

```
%%Se abre xlsx con ubicaciones de los transmisores
file='REPETIDORES_TDT.xlsx';
transmisores= xlsread(file);
transmisores=transmisores';

%%Cargamos ubicacion trasmisores
latrepy=transmisores(4,:);
lonrepx=transmisores(8,:);

%%Sectorizar MAPA
%Ubicación más alta
[valalt,posalt]=max(latrepy);
wa=lonrepx(posalt);
r=10/81;
%%limite por la derecha dentro de nuestro rango
while wa>-3.5
    wa=wa-r;
end

%%Ubicamos los transmisores en una matriz
matriz=ubi_trans(wa,latrepy(posalt),r,lonrepx,latrepy);

figure(2)
%plot(lonrepx,latrepy, '.');hold on;
ylim([42.3 43.5]);
xlim([-3.3 -1.7]);
conty=0;contx=0;

load('db_distribucion_sectores');%&

aaa=zeros(1,140);
total_coches=zeros(1,140);
for ii=1:14000
    suma=sum(db_distribucion_sectores(ii).distCoches(3,:));
```

```

total_coches(1,db_distribucion_sectores(ii).id)=total_coches(1,db_distribucion_sectores(ii).id)+suma;
    if suma >0;
        aaa(1,db_distribucion_sectores(ii).id)=1;
    end
end
cont=0; cont2=0;
for i=latrepy(posalt):-r:42.3
    conty=conty+1;
    for j=wa:+r:-1.8+r
        contx=contx+1;
        conty
        contx
        if matriz(conty,contx)~=0
            cont2=cont2+1;
            rectangle('Position',[j-r/2 i-r/2 r r],'FaceColor',[0
.5 .5]); hold on;
            if aaa(1,(conty-1)*14+contx)==1
                rectangle('Position',[j-r/2 i-r/2 r
r],'FaceColor',[1 0 1]); hold on;
                cont=cont+1;
            end
        end
    end
    contx=0;
end
cont
cont2
plot(lonrepx,latrepy,'.');hold on;
plot(db_carreteras(2,202838),db_carreteras(3,202838),'*r');hold on;
plot_google_map
%%%primera columna matriz todo 0s
matriz=matriz(1:end,2:end);
save('matriz','matriz');
  
```



## Base Datos Sectores

```

clear
load('matriz');
%%%Generar Base Datos
%%%Se abre xlsx con ubicaciones de los transmisores
file='REPETIDORES_TDT.xlsx';
transmisores= xlsread(file);
transmisores=transmisores';

%%%Cargamos ubicacion trasmisores
latrepy=transmisores(4,:);
lonrepx=transmisores(8,:);

%%%Sectorizar MAPA
%Ubicación más alta
[valalt,posalt]=max(latrepy);
wa=lonrepx(posalt);
r=10/81;
%%%limite por la derecha dentro de nuestro rango
while wa>-3.5
    wa=wa-r;
end
n=1;
for i=wa-r/2:+r:-1.8+r
    columnas(n)=i;
    n=n+1;
end
%%%Crea valores frontera fila
n=1;
for i=latrepy(posalt)+r/2:-r:42.3-r/2
    filas(n)=i;
    n=n+1;
end
ny=0;n=0;
for i=latrepy(posalt):-r:42.3
    ny=ny+1;nx=0;
    for j=wa+r:+r:-1.8+r
        nx=nx+1;n=n+1;
        sectores(n).id=n;
        sectores(n).x_ini=columnas(nx);
        sectores(n).x_fin=columnas(nx+1);
        sectores(n).y_ini=filas(ny);
        sectores(n).y_fin=filas(ny+1);
        if matriz(ny,nx)~=0
            sectores(n).cobertura=1;
        end
        if matriz(ny,nx)==0
            sectores(n).cobertura=0;
        end
    end
end
end
save('DBsectores', 'sectores');
  
```

```
% figure(1)
% ylim([42.45 43.5]);
% xlim([-3.3 -1.7]);
% for i=latrepy(posalt):-r:42.3
%     for j=wa:+r:-1.8
%         rectangle('Position',[j-r/2 i-r/2 r r]);
%     end
% end
% plot_google_map
%
% col=columnas(2:end);fil=filas(2:end);
% for i=1:length(lonrepx)
%     mcol=find(col>lonrepx(i));
%     nfil=find(fil<latrepy(i));
%     nfil=nfil(1);
%     mcol=mcol(1);
%     tx(i).id=i;
%     tx(i).sector=mcol+(nfil-1)*(length(columnas)-2);
%     tx(i).x_lon=lonrepx(i);
%     tx(i).y_lat=latrepy(i);
%     plot(lonrepx(i),latrepy(i),'.');hold on;
% end
% save('DBtx', 'tx');
```

## Subsectorización

```

load('DBsectores')
% for i=1:length(sectoresocupados)
%     ocupacion(i)=sectoresocupados(i).ocupado;
% end
%
% sum(ocupacion)

cont=0;
for i=1:length(sectores)

    id(i)=sectores(i).id;
    x_inicio(i)=sectores(i).x_ini;
    x_finfin(i)=sectores(i).x_fin;
    y_inicio(i)=sectores(i).y_ini;
    y_finfin(i)=sectores(i).y_fin;
    cobertura(i)=sectores(i).cobertura;
    x_sep=abs(x_inicio-x_finfin)/10;
    y_sep=abs(y_inicio-y_finfin)/10;
    n=0;
    y_ini=y_inicio+y_sep;
    x_ini=x_inicio-x_sep;
    for j=1:10
        y_ini(i)=y_ini(i)-y_sep(i);
        for k=1:10
            cont=cont+1;
            n=n+1;
            id_sub(cont)=n;
            x_ini(i)=x_ini(i)+x_sep(i);
            subsector(cont).id=i;
            subsector(cont).id_sub=n;
            subsector(cont).x_ini=x_ini(i);
            subsector(cont).x_fin=x_ini(i)+x_sep(i);
            subsector(cont).y_ini=y_ini(i)+y_sep(i);
            subsector(cont).y_fin=y_ini(i);
            subsector(cont).cobertura=cobertura(i);
            pos=[x_ini(i) y_ini(i)-y_sep(i) x_sep(i) y_sep(i)];
            rectangle('Position',pos,'EdgeColor','b');hold on
        end
        x_ini=x_inicio(i)-x_sep;
    end
end

save('DBsubsectores_PorSectores', 'subsector');

cont=0;
for m=1:10
    for k=1:10
        for i=1:14
            for j=1:10
                cont=cont+1;
                n(cont)=j+100*(i-1)+10*(k-1)+1400*(m-1);
            end
        end
    end
end
  
```

```
                end
            end
        end
    end
end
for i=1:length(subsector)
    subsector_xy(i)=subsector(n(i));
end

save('DBsubsectores_PorOrdenXY', 'subsector_xy');

%%%Visualizacion, comprobación visual%%%
for i=1:length(sectores)
    id(i)=sectores(i).id;
    x_origen(i)=sectores(i).x_ini;
    y_origen(i)=sectores(i).y_fin;
end
r=10/81;
for i=1:length(x_origen)
    pos=[x_origen(i) y_origen(i) r r];
    rectangle('Position',pos,'EdgeColor','r');hold
on;% 'FaceColor',[0 .5 .5],
end
plot_google_map
```

## Añadir Carreteras A Base Datos

```

car = shaperead('roads');
type={car.type};
%%tipo 1
t1_1 = strmatch('motorway', type)';
t1_2 = strmatch('motorway_link', type)';
t1_3 = strmatch('primary', type)';
t1_4 = strmatch('primary_link', type)';
t1=[t1_1 t1_2 t1_3 t1_4];
clear t1_1;clear t1_2;clear t1_3;clear t1_4;
%%tipo 2
t2_1 = strmatch('residential', type)';
t2_2 = strmatch('road', type)';
t2_3 = strmatch('secondary', type)';
t2_4 = strmatch('secondary_link', type)';
t2=[t2_1 t2_2 t2_3 t2_4];
clear t2_1;clear t2_2;clear t2_3;clear t2_4;
%%tipo 3
t3_1 = strmatch('tertiary', type)';
t3_2 = strmatch('tertiary_link', type)';
t3=[t3_1 t3_2];
clear t3_1;clear t3_2;
clear type;

figure(1)
for i=1:length(t3)
    plot( car(t3(i)).X, car(t3(i)).Y);hold on;
end
plot_google_map

load('DBsubsectores_PorOrdenXY')

x_max=subsector_xy(1,140).x_fin;
x_ini=subsector_xy(1,1).x_ini;
x_fin=subsector_xy(1,1).x_fin;
x_sep=abs(x_ini-x_fin);

y_max=subsector_xy(1,14000).y_fin;
y_ini=subsector_xy(1,1).y_ini;
y_fin=subsector_xy(1,1).y_fin;
y_sep=abs(y_ini-y_fin);

%%valores x e y
conty=0;contx=0;
for j=y_ini-y_sep:-y_sep:y_max-y_sep
    conty=conty+1;
    lonrepy(conty)=j;
end
for i=x_ini+x_sep:x_sep:x_max+x_sep
    contx=contx+1;
    lonrepx(contx)=i;
  
```

```

end
clear conty;clear contx;
clear y_max; clear y_ini;clear y_fin;clear y_sep;
clear x_max; clear x_ini;clear x_fin;clear x_sep;
clear j; clear i;

cont_err=0;
cont_3=0;
for ii=1:length(t3)
  for jj=1:length(car(t3(ii)).X)
    try
      cont_3=cont_3+1;
      mcol=find(lonrepx>car(t3(ii)).X(jj));
      nfil=find(latrepy<car(t3(ii)).Y(jj));
      nfil=nfil(1);
      mcol=mcol(1);
      n_subsector=(nfil-1)*140+mcol;
      datos_carreteras_3_X(cont_3)=car(t3(ii)).X(jj);
      datos_carreteras_3_Y(cont_3)=car(t3(ii)).Y(jj);
      datos_carreteras_3_tipo(cont_3)=3;
      datos_carreteras_3_subsector(cont_3)=n_subsector;

      catch me
        cont_err=cont_err+1;
      end
    end
  end
end
cont_2=0;
for ii=1:length(t2)
  for jj=1:length(car(t2(ii)).X)
    try
      cont_2=cont_2+1;
      mcol=find(lonrepx>car(t2(ii)).X(jj));
      nfil=find(latrepy<car(t2(ii)).Y(jj));
      nfil=nfil(1);
      mcol=mcol(1);
      n_subsector=(nfil-1)*140+mcol;
      datos_carreteras_2_X(cont_2)=car(t2(ii)).X(jj);
      datos_carreteras_2_Y(cont_2)=car(t2(ii)).Y(jj);
      datos_carreteras_2_tipo(cont_2)=2;
      datos_carreteras_2_subsector(cont_2)=n_subsector;

      catch me
        cont_err=cont_err+1;
      end
    end
  end
end
cont_1=0;
for ii=1:length(t1)
  for jj=1:length(car(t1(ii)).X)
    try
      cont_1=cont_1+1;
      mcol=find(lonrepx>car(t1(ii)).X(jj));
      nfil=find(latrepy<car(t1(ii)).Y(jj));
      nfil=nfil(1);
      mcol=mcol(1);
      n_subsector=(nfil-1)*140+mcol;

```

```

    datos_carreteras_1_X(cont_1)=car(t1(ii)).X(jj);
    datos_carreteras_1_Y(cont_1)=car(t1(ii)).Y(jj);
    datos_carreteras_1_tipo(cont_1)=1;
    datos_carreteras_1_subsector(cont_1)=n_subsector;

    catch me
      cont_err=cont_err+1;
    end
  end
end

clear cont_1;clear cont_2;clear cont_3;
clear mcol;clear nfil;
clear lonrepx;clear latrepy;

datos_carreteras_X=[datos_carreteras_1_X datos_carreteras_2_X
datos_carreteras_3_X];
datos_carreteras_Y=[datos_carreteras_1_Y datos_carreteras_2_Y
datos_carreteras_3_Y];
datos_carreteras_tipo=[datos_carreteras_1_tipo
datos_carreteras_2_tipo datos_carreteras_3_tipo];
datos_carreteras_subsector=[datos_carreteras_1_subsector
datos_carreteras_2_subsector datos_carreteras_3_subsector];

clear datos_carreteras_1_X; clear datos_carreteras_2_X; clear
datos_carreteras_3_X;
clear datos_carreteras_1_tipo; clear datos_carreteras_2_tipo; clear
datos_carreteras_3_tipo;
clear datos_carreteras_1_Y; clear datos_carreteras_2_Y; clear
datos_carreteras_3_Y;
clear datos_carreteras_1_subsector; clear
datos_carreteras_2_subsector; clear datos_carreteras_3_subsector;

% figure(2)
% for i=1:length(datos_carreteras_3_X)
%   i
%   plot(datos_carreteras_3_X(i), datos_carreteras_3_Y(i));hold
on;
% end
% plot_google_map

for i=1:length(subsector_xy)
  index=find(datos_carreteras_subsector==i);
  if length(index)~=0
    for j=1:length(index)
      datos_carretera_X(j)=datos_carreteras_X(index(j));
      datos_carretera_Y(j)=datos_carreteras_Y(index(j));

datos_carretera_tipo(j)=datos_carreteras_tipo(index(j));
    end
    subsector_xy(i).datos_carretera_X=datos_carretera_X;
    subsector_xy(i).datos_carretera_Y=datos_carretera_Y;
    subsector_xy(i).datos_carretera_tipo=datos_carretera_tipo;
    clear datos_carretera_X;
    clear datos_carretera_Y;
  }
end

```



```
clear datos_carretera_tipo;  
end  
  
end  
  
% save('DBsubsectores_PorSectoros_Prueba', 'subsector_xy');
```

## Añadir Identificador A Carreteras

```
load('DBsubsectores_PorOrdenXY')
load('DB_carreteras')

x_max=subsector_xy(1,140).x_fin;
x_ini=subsector_xy(1,1).x_ini;
x_fin=subsector_xy(1,1).x_fin;
x_sep=abs(x_ini-x_fin);

y_max=subsector_xy(1,14000).y_fin;
y_ini=subsector_xy(1,1).y_ini;
y_fin=subsector_xy(1,1).y_fin;
y_sep=abs(y_ini-y_fin);

%%valores x e y
conty=0;contx=0;
for j=y_ini-y_sep:-y_sep:y_max-y_sep
    conty=conty+1;
    latrepy(conty)=j;
end
for i=x_ini+x_sep:x_sep:x_max+x_sep
    contx=contx+1;
    lonrepx(contx)=i;
end
clear conty;clear contx;
clear y_max; clear y_ini;clear y_fin;clear y_sep;
clear x_max; clear x_ini;clear x_fin;clear x_sep;
clear j; clear i;

datos_carreteras_id=db_carreteras(1,:);
datos_carreteras_X=db_carreteras(2,:);
datos_carreteras_Y=db_carreteras(3,:);
datos_carreteras_tipo=db_carreteras(4,:);
cont_err=0;
for i=1:length(db_carreteras)
    i/length(db_carreteras)
    try
        mcol=find(lonrepx>db_carreteras(2,i));
        nfil=find(latrepy<db_carreteras(3,i));
        nfil=nfil(1);
        mcol=mcol(1);
        n_subsector=(nfil-1)*140+mcol;
        datos_carreteras_subsector(i)=n_subsector;

    catch me
        cont_err=cont_err+1;
    end
end

clear mcol;clear nfil;
clear lonrepx;clear latrepy;
```

```
% figure(2)
% for i=1:length(datos_carreteras_3_X)
%     i
%     plot(datos_carreteras_3_X(i), datos_carreteras_3_Y(i));hold
on;
% end
% plot_google_map

for i=1:length(subsector_xy)
    i/length(subsector_xy)
    index=find(datos_carreteras_subsector==i);
    if length(index)~=0
        for j=1:length(index)
            datos_carretera_id(j)=datos_carreteras_id(index(j));
            datos_carretera_X(j)=datos_carreteras_X(index(j));
            datos_carretera_Y(j)=datos_carreteras_Y(index(j));

datos_carretera_tipo(j)=datos_carreteras_tipo(index(j));

            end
            subsector_xy(i).datos_carretera_id=datos_carretera_id;
            subsector_xy(i).datos_carretera_X=datos_carretera_X;
            subsector_xy(i).datos_carretera_Y=datos_carretera_Y;
            subsector_xy(i).datos_carretera_tipo=datos_carretera_tipo;
            clear datos_carretera_id;
            clear datos_carretera_X;
            clear datos_carretera_Y;
            clear datos_carretera_tipo;
        end
    end

end

save('DBsubsectores_Completo', 'subsector_xy');
```

## Base Datos Poblaciones

```

car = shaperead('roads');
type={car.type};
%%tipo 1
t1_1 = strmatch('motorway', type)';
t1_2 = strmatch('motorway_link', type)';
t1_3 = strmatch('primary', type)';
t1_4 = strmatch('primary_link', type)';
t1=[t1_1 t1_2 t1_3 t1_4];
clear t1_1;clear t1_2;clear t1_3;clear t1_4;
tipo_1=3*ones(1,length(t1));
%%tipo 2
t2_1 = strmatch('residential', type)';
t2_2 = strmatch('road', type)';
t2_3 = strmatch('secondary', type)';
t2_4 = strmatch('secondary_link', type)';
t2=[t2_1 t2_2 t2_3 t2_4];
clear t2_1;clear t2_2;clear t2_3;clear t2_4;
tipo_2=2*ones(1,length(t2));
%%tipo 3
t3_1 = strmatch('tertiary', type)';
t3_2 = strmatch('tertiary_link', type)';
t3=[t3_1 t3_2];
clear t3_1;clear t3_2;
tipo_3=ones(1,length(t3));
clear type;
t_tot=[t1 t2 t3];
tipos=[tipo_1 tipo_2 tipo_3];
clear t1;clear t2;clear t3;
clear tipo_1;clear tipo_2;clear tipo_3;
clear tipo_1;clear tipo_2;clear tipo_3;

load('db_carreteras')
vehiculos = xlsread('Datos_Vehiculos')';

for j=1:length(vehiculos)
    cont=0;
    for i=1:length(db_carreteras)
        try
            r=((db_carreteras(2,i)-
vehiculos(2,j))^2+(db_carreteras(3,i)-vehiculos(3,j))^2)^0.5;
            r_km=r*81;

            if r_km>0 && r_km<25
                switch j%numero de ciudad
                    case 1
                        cont=cont+1;
                        ciudadl.id(cont)=db_carreteras(1,i);
                        ciudadl.X(cont)=db_carreteras(2,i);
                        ciudadl.Y(cont)=db_carreteras(3,i);
                        ciudadl.distancia(cont)=r_km;
                        ciudadl.tipo(cont)=db_carreteras(4,i);

```

```
case 2
    cont=cont+1;
    ciudad2.id(cont)=db_carreteras(1,i);
    ciudad2.X(cont)=db_carreteras(2,i);
    ciudad2.Y(cont)=db_carreteras(3,i);
    ciudad2.distancia(cont)=r_km;
    ciudad2.tipo(cont)=db_carreteras(4,i);
case 3
    cont=cont+1;
    ciudad3.id(cont)=db_carreteras(1,i);
    ciudad3.X(cont)=db_carreteras(2,i);
    ciudad3.Y(cont)=db_carreteras(3,i);
    ciudad3.distancia(cont)=r_km;
    ciudad3.tipo(cont)=db_carreteras(4,i);
case 4
    cont=cont+1;
    ciudad4.id(cont)=db_carreteras(1,i);
    ciudad4.X(cont)=db_carreteras(2,i);
    ciudad4.Y(cont)=db_carreteras(3,i);
    ciudad4.distancia(cont)=r_km;
    ciudad4.tipo(cont)=db_carreteras(4,i);
case 5
    cont=cont+1;
    ciudad5.id(cont)=db_carreteras(1,i);
    ciudad5.X(cont)=db_carreteras(2,i);
    ciudad5.Y(cont)=db_carreteras(3,i);
    ciudad5.distancia(cont)=r_km;
    ciudad5.tipo(cont)=db_carreteras(4,i);
case 6
    cont=cont+1;
    ciudad6.id(cont)=db_carreteras(1,i);
    ciudad6.X(cont)=db_carreteras(2,i);
    ciudad6.Y(cont)=db_carreteras(3,i);
    ciudad6.distancia(cont)=r_km;
    ciudad6.tipo(cont)=db_carreteras(4,i);
case 7
    cont=cont+1;
    ciudad7.id(cont)=db_carreteras(1,i);
    ciudad7.X(cont)=db_carreteras(2,i);
    ciudad7.Y(cont)=db_carreteras(3,i);
    ciudad7.distancia(cont)=r_km;
    ciudad7.tipo(cont)=db_carreteras(4,i);
case 8
    cont=cont+1;
    ciudad8.id(cont)=db_carreteras(1,i);
    ciudad8.X(cont)=db_carreteras(2,i);
    ciudad8.Y(cont)=db_carreteras(3,i);
    ciudad8.distancia(cont)=r_km;
    ciudad8.tipo(cont)=db_carreteras(4,i);
case 9
    cont=cont+1;
    ciudad9.id(cont)=db_carreteras(1,i);
    ciudad9.X(cont)=db_carreteras(2,i);
    ciudad9.Y(cont)=db_carreteras(3,i);
    ciudad9.distancia(cont)=r_km;
    ciudad9.tipo(cont)=db_carreteras(4,i);
case 10
```

```
cont=cont+1;
ciudad10.id(cont)=db_carreteras(1,i);
ciudad10.X(cont)=db_carreteras(2,i);
ciudad10.Y(cont)=db_carreteras(3,i);
ciudad10.distancia(cont)=r_km;
ciudad10.tipo(cont)=db_carreteras(4,i);
case 11
cont=cont+1;
ciudad11.id(cont)=db_carreteras(1,i);
ciudad11.X(cont)=db_carreteras(2,i);
ciudad11.Y(cont)=db_carreteras(3,i);
ciudad11.distancia(cont)=r_km;
ciudad11.tipo(cont)=db_carreteras(4,i);
case 12
cont=cont+1;
ciudad12.id(cont)=db_carreteras(1,i);
ciudad12.X(cont)=db_carreteras(2,i);
ciudad12.Y(cont)=db_carreteras(3,i);
ciudad12.distancia(cont)=r_km;
ciudad12.tipo(cont)=db_carreteras(4,i);
case 13
cont=cont+1;
ciudad13.id(cont)=db_carreteras(1,i);
ciudad13.X(cont)=db_carreteras(2,i);
ciudad13.Y(cont)=db_carreteras(3,i);
ciudad13.distancia(cont)=r_km;
ciudad13.tipo(cont)=db_carreteras(4,i);
case 14
cont=cont+1;
ciudad14.id(cont)=db_carreteras(1,i);
ciudad14.X(cont)=db_carreteras(2,i);
ciudad14.Y(cont)=db_carreteras(3,i);
ciudad14.distancia(cont)=r_km;
ciudad14.tipo(cont)=db_carreteras(4,i);
case 15
cont=cont+1;
ciudad15.id(cont)=db_carreteras(1,i);
ciudad15.X(cont)=db_carreteras(2,i);
ciudad15.Y(cont)=db_carreteras(3,i);
ciudad15.distancia(cont)=r_km;
ciudad15.tipo(cont)=db_carreteras(4,i);
case 16
cont=cont+1;
ciudad16.id(cont)=db_carreteras(1,i);
ciudad16.X(cont)=db_carreteras(2,i);
ciudad16.Y(cont)=db_carreteras(3,i);
ciudad16.distancia(cont)=r_km;
ciudad16.tipo(cont)=db_carreteras(4,i);
case 17
cont=cont+1;
ciudad17.id(cont)=db_carreteras(1,i);
ciudad17.X(cont)=db_carreteras(2,i);
ciudad17.Y(cont)=db_carreteras(3,i);
ciudad17.distancia(cont)=r_km;
ciudad17.tipo(cont)=db_carreteras(4,i);
case 18
cont=cont+1;
```

```
ciudad18.id(cont)=db_carreteras(1,i);
ciudad18.X(cont)=db_carreteras(2,i);
ciudad18.Y(cont)=db_carreteras(3,i);
ciudad18.distancia(cont)=r_km;
ciudad18.tipo(cont)=db_carreteras(4,i);
case 19
cont=cont+1;
ciudad19.id(cont)=db_carreteras(1,i);
ciudad19.X(cont)=db_carreteras(2,i);
ciudad19.Y(cont)=db_carreteras(3,i);
ciudad19.distancia(cont)=r_km;
ciudad19.tipo(cont)=db_carreteras(4,i);
case 20
cont=cont+1;
ciudad20.id(cont)=db_carreteras(1,i);
ciudad20.X(cont)=db_carreteras(2,i);
ciudad20.Y(cont)=db_carreteras(3,i);
ciudad20.distancia(cont)=r_km;
ciudad20.tipo(cont)=db_carreteras(4,i);
case 21
cont=cont+1;
ciudad21.id(cont)=db_carreteras(1,i);
ciudad21.X(cont)=db_carreteras(2,i);
ciudad21.Y(cont)=db_carreteras(3,i);
ciudad21.distancia(cont)=r_km;
ciudad21.tipo(cont)=db_carreteras(4,i);
case 22
cont=cont+1;
ciudad22.id(cont)=db_carreteras(1,i);
ciudad22.X(cont)=db_carreteras(2,i);
ciudad22.Y(cont)=db_carreteras(3,i);
ciudad22.distancia(cont)=r_km;
ciudad22.tipo(cont)=db_carreteras(4,i);
case 23
cont=cont+1;
ciudad23.id(cont)=db_carreteras(1,i);
ciudad23.X(cont)=db_carreteras(2,i);
ciudad23.Y(cont)=db_carreteras(3,i);
ciudad23.distancia(cont)=r_km;
ciudad23.tipo(cont)=db_carreteras(4,i);
case 24
cont=cont+1;
ciudad24.id(cont)=db_carreteras(1,i);
ciudad24.X(cont)=db_carreteras(2,i);
ciudad24.Y(cont)=db_carreteras(3,i);
ciudad24.distancia(cont)=r_km;
ciudad24.tipo(cont)=db_carreteras(4,i);
case 25
cont=cont+1;
ciudad25.id(cont)=db_carreteras(1,i);
ciudad25.X(cont)=db_carreteras(2,i);
ciudad25.Y(cont)=db_carreteras(3,i);
ciudad25.distancia(cont)=r_km;
ciudad25.tipo(cont)=db_carreteras(4,i);
case 26
cont=cont+1;
ciudad26.id(cont)=db_carreteras(1,i);
```



```
ciudad26.X(cont)=db_carreteras(2,i);
ciudad26.Y(cont)=db_carreteras(3,i);
ciudad26.distancia(cont)=r_km;
ciudad26.tipo(cont)=db_carreteras(4,i);
case 27
    cont=cont+1;
    ciudad27.id(cont)=db_carreteras(1,i);
    ciudad27.X(cont)=db_carreteras(2,i);
    ciudad27.Y(cont)=db_carreteras(3,i);
    ciudad27.distancia(cont)=r_km;
    ciudad27.tipo(cont)=db_carreteras(4,i);
case 28
    cont=cont+1;
    ciudad28.id(cont)=db_carreteras(1,i);
    ciudad28.X(cont)=db_carreteras(2,i);
    ciudad28.Y(cont)=db_carreteras(3,i);
    ciudad28.distancia(cont)=r_km;
    ciudad28.tipo(cont)=db_carreteras(4,i);
case 29
    cont=cont+1;
    ciudad29.id(cont)=db_carreteras(1,i);
    ciudad29.X(cont)=db_carreteras(2,i);
    ciudad29.Y(cont)=db_carreteras(3,i);
    ciudad29.distancia(cont)=r_km;
    ciudad29.tipo(cont)=db_carreteras(4,i);
end
end
catch me
    disp('falla')
end
end
end
clear j; clear ii; clear jj;

db_urbe=[ciudad1 ciudad2 ciudad3 ciudad4 ciudad5 ciudad6 ciudad7
ciudad8 ciudad9 ciudad10...
    ciudad11 ciudad12 ciudad13 ciudad14 ciudad15 ciudad16 ciudad17
ciudad18 ciudad19 ciudad20...
    ciudad21 ciudad22 ciudad23 ciudad24 ciudad25 ciudad26 ciudad27
ciudad28 ciudad29];

save('DB_urbe', 'db_urbe');
```

## Base Datos Carreteras

```

car = shaperead('roads');
type={car.type};
%%%tipo 1
t1_1 = strmatch('motorway', type)';
t1_2 = strmatch('motorway_link', type)';
t1_3 = strmatch('primary', type)';
t1_4 = strmatch('primary_link', type)';
t1=[t1_1 t1_2 t1_3 t1_4];
clear t1_1;clear t1_2;clear t1_3;clear t1_4;
tipo_1=3*ones(1,length(t1));
%%%tipo 2
t2_1 = strmatch('residential', type)';
t2_2 = strmatch('road', type)';
t2_3 = strmatch('secondary', type)';
t2_4 = strmatch('secondary_link', type)';
t2=[t2_1 t2_2 t2_3 t2_4];
clear t2_1;clear t2_2;clear t2_3;clear t2_4;
tipo_2=2*ones(1,length(t2));
%%%tipo 3
t3_1 = strmatch('tertiary', type)';
t3_2 = strmatch('tertiary_link', type)';
t3=[t3_1 t3_2];
clear t3_1;clear t3_2;
tipo_3=ones(1,length(t3));
clear type;
t_tot=[t1 t2 t3];
tipos=[tipo_1 tipo_2 tipo_3];
clear t1;clear t2;clear t3;
clear tipo_1;clear tipo_2;clear tipo_3;

%%%Base de datos de carreteras
db_carreteras=0;
cont=0;
for i=1:length(t_tot)
    for j=1:length(car(t_tot(i)).X)-1
        try
            cont=cont+1;
            db_carreteras(1,cont)=cont;
            db_carreteras(2,cont)=car(t_tot(i)).X(j);
            db_carreteras(3,cont)=car(t_tot(i)).Y(j);
            db_carreteras(4,cont)=tipos(i);
        catch me
        end
    end
end

%%%Base de datos de vehiculos y distancias
repetidores = xlsread('REPETIDORES_TDT.xlsx')';

ejey_rep=repetidores(4,:);
ejex_rep=repetidores(8,:);
db_carreterasVsTx=zeros(length(db_carreteras),length(ejey_rep));
  
```

```
for j=1:length(db_carreteras)
    r_km=zeros(1,length(ejey_rep));
    for i=1:length(ejey_rep)
        r=((db_carreteras(2,j)-
ejex_rep(i))^2+(db_carreteras(3,j)-ejey_rep(i))^2)^0.5;
        r_km(i)=r*81;
    end
    db_carreterasVsTx(j,:)=r_km;
end

db_carreterasVsTx=cast(db_carreterasVsTx,'single');
save('DB_carreteras','db_carreteras')
save('DB_carreterasVsTx','db_carreterasVsTx')
```

## Base Datos Anillos

```

clear all;

load('DB_urbe');
poblaciones = xlsread('Poblaciones');
vehiculos = xlsread('Datos_Vehiculos');

for i=1:29;
% i=1;
i
    cont=zeros(25,1);
    for jj=1:length(db_urbe(i).distancia)
        if db_urbe(i).distancia(jj) < 1
            cont(1,1)=cont(1,1)+1;
            anillos_dist(1,cont(1,1))=db_urbe(i).distancia(jj);
            anillos_X(1,cont(1,1))=db_urbe(i).X(jj);
            anillos_Y(1,cont(1,1))=db_urbe(i).Y(jj);
            anillos_tipo(1,cont(1,1))=db_urbe(i).tipo(jj);
            anillos_id(1,cont(1,1))=db_urbe(i).id(jj);
        end
        if db_urbe(i).distancia(jj) > 1 & db_urbe(i).distancia(jj)
< 2
            cont(2,1)=cont(2,1)+1;
            anillos_dist(2,cont(2,1))=db_urbe(i).distancia(jj);
            anillos_X(2,cont(2,1))=db_urbe(i).X(jj);
            anillos_Y(2,cont(2,1))=db_urbe(i).Y(jj);
            anillos_tipo(2,cont(2,1))=db_urbe(i).tipo(jj);
            anillos_id(2,cont(2,1))=db_urbe(i).id(jj);
        end
        if db_urbe(i).distancia(jj) > 2 & db_urbe(i).distancia(jj)
< 3
            cont(3,1)=cont(3,1)+1;
            anillos_dist(3,cont(3,1))=db_urbe(i).distancia(jj);
            anillos_X(3,cont(3,1))=db_urbe(i).X(jj);
            anillos_Y(3,cont(3,1))=db_urbe(i).Y(jj);
            anillos_tipo(3,cont(3,1))=db_urbe(i).tipo(jj);
            anillos_id(3,cont(3,1))=db_urbe(i).id(jj);
        end
        if db_urbe(i).distancia(jj) > 3 & db_urbe(i).distancia(jj)
< 4
            cont(4,1)=cont(4,1)+1;
            anillos_dist(4,cont(4,1))=db_urbe(i).distancia(jj);
            anillos_X(4,cont(4,1))=db_urbe(i).X(jj);
            anillos_Y(4,cont(4,1))=db_urbe(i).Y(jj);
            anillos_tipo(4,cont(4,1))=db_urbe(i).tipo(jj);
            anillos_id(4,cont(4,1))=db_urbe(i).id(jj);
        end
        if db_urbe(i).distancia(jj) > 4 & db_urbe(i).distancia(jj)
< 5
            cont(5,1)=cont(5,1)+1;
            anillos_dist(5,cont(5,1))=db_urbe(i).distancia(jj);
            anillos_X(5,cont(5,1))=db_urbe(i).X(jj);

```

```

    anillos_Y(5,cont(5,1))=db_urbe(i).Y(jj);
    anillos_tipo(5,cont(5,1))=db_urbe(i).tipo(jj);
    anillos_id(5,cont(5,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 5 & db_urbe(i).distancia(jj)
< 6
    cont(6,1)=cont(6,1)+1;
    anillos_dist(6,cont(6,1))=db_urbe(i).distancia(jj);
    anillos_X(6,cont(6,1))=db_urbe(i).X(jj);
    anillos_Y(6,cont(6,1))=db_urbe(i).Y(jj);
    anillos_tipo(6,cont(6,1))=db_urbe(i).tipo(jj);
    anillos_id(6,cont(6,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 6 & db_urbe(i).distancia(jj)
< 7
    cont(7,1)=cont(7,1)+1;
    anillos_dist(7,cont(7,1))=db_urbe(i).distancia(jj);
    anillos_X(7,cont(7,1))=db_urbe(i).X(jj);
    anillos_Y(7,cont(7,1))=db_urbe(i).Y(jj);
    anillos_tipo(7,cont(7,1))=db_urbe(i).tipo(jj);
    anillos_id(7,cont(7,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 7 & db_urbe(i).distancia(jj)
< 8
    cont(8,1)=cont(8,1)+1;
    anillos_dist(8,cont(8,1))=db_urbe(i).distancia(jj);
    anillos_X(8,cont(8,1))=db_urbe(i).X(jj);
    anillos_Y(8,cont(8,1))=db_urbe(i).Y(jj);
    anillos_tipo(8,cont(8,1))=db_urbe(i).tipo(jj);
    anillos_id(8,cont(8,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 8 & db_urbe(i).distancia(jj)
< 9
    cont(9,1)=cont(9,1)+1;
    anillos_dist(9,cont(9,1))=db_urbe(i).distancia(jj);
    anillos_X(9,cont(9,1))=db_urbe(i).X(jj);
    anillos_Y(9,cont(9,1))=db_urbe(i).Y(jj);
    anillos_tipo(9,cont(9,1))=db_urbe(i).tipo(jj);
    anillos_id(9,cont(9,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 9 & db_urbe(i).distancia(jj)
< 10
    cont(10,1)=cont(10,1)+1;
    anillos_dist(10,cont(10,1))=db_urbe(i).distancia(jj);
    anillos_X(10,cont(10,1))=db_urbe(i).X(jj);
    anillos_Y(10,cont(10,1))=db_urbe(i).Y(jj);
    anillos_tipo(10,cont(10,1))=db_urbe(i).tipo(jj);
    anillos_id(10,cont(10,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 10 & db_urbe(i).distancia(jj)
< 11
    cont(11,1)=cont(11,1)+1;
    anillos_dist(11,cont(11,1))=db_urbe(i).distancia(jj);
    anillos_X(11,cont(11,1))=db_urbe(i).X(jj);
    anillos_Y(11,cont(11,1))=db_urbe(i).Y(jj);
    anillos_tipo(11,cont(11,1))=db_urbe(i).tipo(jj);
    anillos_id(11,cont(11,1))=db_urbe(i).id(jj);
  end

```

```
end
if db_urbe(i).distancia(jj) > 11 & db_urbe(i).distancia(jj)
< 12
    cont(12,1)=cont(12,1)+1;
    anillos_dist(12,cont(12,1))=db_urbe(i).distancia(jj);
    anillos_X(12,cont(12,1))=db_urbe(i).X(jj);
    anillos_Y(12,cont(12,1))=db_urbe(i).Y(jj);
    anillos_tipo(12,cont(12,1))=db_urbe(i).tipo(jj);
    anillos_id(12,cont(12,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 12 & db_urbe(i).distancia(jj)
< 13
    cont(13,1)=cont(13,1)+1;
    anillos_dist(13,cont(13,1))=db_urbe(i).distancia(jj);
    anillos_X(13,cont(13,1))=db_urbe(i).X(jj);
    anillos_Y(13,cont(13,1))=db_urbe(i).Y(jj);
    anillos_tipo(13,cont(13,1))=db_urbe(i).tipo(jj);
    anillos_id(13,cont(13,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 13 & db_urbe(i).distancia(jj)
< 14
    cont(14,1)=cont(14,1)+1;
    anillos_dist(14,cont(14,1))=db_urbe(i).distancia(jj);
    anillos_X(14,cont(14,1))=db_urbe(i).X(jj);
    anillos_Y(14,cont(14,1))=db_urbe(i).Y(jj);
    anillos_tipo(14,cont(14,1))=db_urbe(i).tipo(jj);
    anillos_id(14,cont(14,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 14 & db_urbe(i).distancia(jj)
< 15
    cont(15,1)=cont(15,1)+1;
    anillos_dist(15,cont(15,1))=db_urbe(i).distancia(jj);
    anillos_X(15,cont(15,1))=db_urbe(i).X(jj);
    anillos_Y(15,cont(15,1))=db_urbe(i).Y(jj);
    anillos_tipo(15,cont(15,1))=db_urbe(i).tipo(jj);
    anillos_id(15,cont(15,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 15 & db_urbe(i).distancia(jj)
< 16
    cont(16,1)=cont(16,1)+1;
    anillos_dist(16,cont(16,1))=db_urbe(i).distancia(jj);
    anillos_X(16,cont(16,1))=db_urbe(i).X(jj);
    anillos_Y(16,cont(16,1))=db_urbe(i).Y(jj);
    anillos_tipo(16,cont(16,1))=db_urbe(i).tipo(jj);
    anillos_id(16,cont(16,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 16 & db_urbe(i).distancia(jj)
< 17
    cont(17,1)=cont(17,1)+1;
    anillos_dist(17,cont(17,1))=db_urbe(i).distancia(jj);
    anillos_X(17,cont(17,1))=db_urbe(i).X(jj);
    anillos_Y(17,cont(17,1))=db_urbe(i).Y(jj);
    anillos_tipo(17,cont(17,1))=db_urbe(i).tipo(jj);
    anillos_id(17,cont(17,1))=db_urbe(i).id(jj);
end
if db_urbe(i).distancia(jj) > 17 & db_urbe(i).distancia(jj)
< 18
```

```

    cont(18,1)=cont(18,1)+1;
    anillos_dist(18,cont(18,1))=db_urbe(i).distancia(jj);
    anillos_X(18,cont(18,1))=db_urbe(i).X(jj);
    anillos_Y(18,cont(18,1))=db_urbe(i).Y(jj);
    anillos_tipo(18,cont(18,1))=db_urbe(i).tipo(jj);
    anillos_id(18,cont(18,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 18 & db_urbe(i).distancia(jj)
< 19
    cont(19,1)=cont(19,1)+1;
    anillos_dist(19,cont(19,1))=db_urbe(i).distancia(jj);
    anillos_X(19,cont(19,1))=db_urbe(i).X(jj);
    anillos_Y(19,cont(19,1))=db_urbe(i).Y(jj);
    anillos_tipo(19,cont(19,1))=db_urbe(i).tipo(jj);
    anillos_id(19,cont(19,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 19 & db_urbe(i).distancia(jj)
< 20
    cont(20,1)=cont(20,1)+1;
    anillos_dist(20,cont(20,1))=db_urbe(i).distancia(jj);
    anillos_X(20,cont(20,1))=db_urbe(i).X(jj);
    anillos_Y(20,cont(20,1))=db_urbe(i).Y(jj);
    anillos_tipo(20,cont(20,1))=db_urbe(i).tipo(jj);
    anillos_id(20,cont(20,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 20 & db_urbe(i).distancia(jj)
< 21
    cont(21,1)=cont(21,1)+1;
    anillos_dist(21,cont(21,1))=db_urbe(i).distancia(jj);
    anillos_X(21,cont(21,1))=db_urbe(i).X(jj);
    anillos_Y(21,cont(21,1))=db_urbe(i).Y(jj);
    anillos_tipo(21,cont(21,1))=db_urbe(i).tipo(jj);
    anillos_id(21,cont(21,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 21 & db_urbe(i).distancia(jj)
< 22
    cont(22,1)=cont(22,1)+1;
    anillos_dist(22,cont(22,1))=db_urbe(i).distancia(jj);
    anillos_X(22,cont(22,1))=db_urbe(i).X(jj);
    anillos_Y(22,cont(22,1))=db_urbe(i).Y(jj);
    anillos_tipo(22,cont(22,1))=db_urbe(i).tipo(jj);
    anillos_id(22,cont(22,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 22 & db_urbe(i).distancia(jj)
< 23
    cont(23,1)=cont(23,1)+1;
    anillos_dist(23,cont(23,1))=db_urbe(i).distancia(jj);
    anillos_X(23,cont(23,1))=db_urbe(i).X(jj);
    anillos_Y(23,cont(23,1))=db_urbe(i).Y(jj);
    anillos_tipo(23,cont(23,1))=db_urbe(i).tipo(jj);
    anillos_id(23,cont(23,1))=db_urbe(i).id(jj);
  end
  if db_urbe(i).distancia(jj) > 23 & db_urbe(i).distancia(jj)
< 24
    cont(24,1)=cont(24,1)+1;
    anillos_dist(24,cont(24,1))=db_urbe(i).distancia(jj);
    anillos_X(24,cont(24,1))=db_urbe(i).X(jj);

```

```

        anillos_Y(24,cont(24,1))=db_urbe(i).Y(jj);
        anillos_tipo(24,cont(24,1))=db_urbe(i).tipo(jj);
        anillos_id(24,cont(24,1))=db_urbe(i).id(jj);
    end
    if db_urbe(i).distancia(jj) > 24 & db_urbe(i).distancia(jj)
< 25
        cont(25,1)=cont(25,1)+1;
        anillos_dist(25,cont(25,1))=db_urbe(i).distancia(jj);
        anillos_X(25,cont(25,1))=db_urbe(i).X(jj);
        anillos_Y(25,cont(25,1))=db_urbe(i).Y(jj);
        anillos_tipo(25,cont(25,1))=db_urbe(i).tipo(jj);
        anillos_id(25,cont(25,1))=db_urbe(i).id(jj);
    end
end

    end

    Urbes_anillos_db(i).distancia=db_urbe(i).distancia;
    Urbes_anillos_db(i).X=db_urbe(i).X;
    Urbes_anillos_db(i).Y=db_urbe(i).Y;
    Urbes_anillos_db(i).tipo=db_urbe(i).tipo;
    Urbes_anillos_db(i).distanciaanillos=anillos_dist;
    Urbes_anillos_db(i).Xanillos=anillos_X;
    Urbes_anillos_db(i).Yanillos=anillos_Y;
    Urbes_anillos_db(i).tipoanillos=anillos_tipo;
    Urbes_anillos_db(i).idanillos=anillos_id;

    clear anillos_dist; clear anillos_tipo; clear anillos_X; clear
anillos_Y; clear anillos_id;
end

save('DB_Urbes_anillos', 'Urbes_anillos_db');

% array_distribucion=zeros(1,sum(db_urbe(i).tipo));
% pos=0;
% for ii=1:length(db_urbe(i).tipo)
%     cont=db_urbe(i).tipo(ii);
%     while cont > 0
%         pos=pos+1;
%         cont=cont-1;
%         array_distribucion(pos)=ii;
%     end
% end

% % %
% % %
% % %
% % % for i=1:25
% % %     coef(i)=((-i^2/50+i)-(-(i-1)^2/50+i-1))/12.5;
% % % end;%Porcentaje de vehiculos del total en cada radio
% % %
% % % % cont=0;
% % % % for i=i:length(db_urbe)
% % % %     for j=1:25

```



```
% % % i=1;clear puntos; clear ubicacion_vehiculos;cont=0;clear
resultado;%j=1;
% % %     puntos=find(db_urbe(i).distancia>j-1 &
db_urbe(i).distancia<j);
% % %     ubicacion_vehiculos=1+round((length(puntos)-
1)*rand(1,round(coef(j)*vehiculos(1,i))));
% % %     for ii=1:length(puntos)
% % %         cont=cont+1;
% % %         a=find(ubicacion_vehiculos==ii);
% % %         resultado(1,cont)=length(a);%cantidad vehiculos
en ese punto
% % %
resultado(2,cont)=db_urbe(i).X(puntos(ii));%ubicacion x
% % %
resultado(3,cont)=db_urbe(i).Y(puntos(ii));%ubicacion y
% % %
resultado(4,cont)=db_urbe(i).distancia(puntos(ii));%distancia
% % %
resultado(5,cont)=db_urbe(i).tipo(puntos(ii));%tipo carretera
% % %     resultado(6,cont)=i;%ciudad
% % %     end
% % %     end
% % % % end
% % % % length(resultado(1,:))/vehiculos(1,1)
% % % % max(resultado(1,:))
```

## Distribución Vehículos

```

load('DB_Urbes_anillos');
poblaciones = xlsread('Poblaciones');
vehiculos = xlsread('Datos_Vehiculos');
Db_dist_coches=Urbes_anillos_db;

%%%parametro corrector en funcion de la hora
c=1;

for i=1:25
    %Porcentaje de vehiculos del total en cada radio
    coef(i)=((-i^2/50+i)-(-(i-1)^2/50+i-1))/12.5;
end

%max(Urbes_anillos_db(1,1).distanciaanillos(3,:))
% i=1;j=1;
for i=1:29
    i
    vehi=zeros(25,length(Urbes_anillos_db(1,i).tipoanillos(1,:)));
    for j=1:25
        clear a; clear array_distribucion; clear pos; clear cont;
        a=round(sum(Urbes_anillos_db(1,i).tipoanillos(j,:)));
        array_distribucion=zeros(2,a);
        pos=0;
        %aumentar probabilidades según tipo carretera
        for ii=1:length(Urbes_anillos_db(1,i).tipoanillos(j,:))
            cont=Urbes_anillos_db(1,i).tipoanillos(j,ii);
            while cont > 0
                pos=pos+1;
                cont=cont-1;
                %array con la posicion de cada
                array_distribucion(1,pos)=ii;
            end
        end
        %reparto aleatorio de vehiculos
        for jj=1:round(vehiculos(7,i)*coef(j)*c)
            b=length(array_distribucion(1,:));
            aa=1;
            pos2=round((b-aa)*rand(1,1) + aa);
            while pos2==0
                pos2=round((b-aa)*rand(1,1) + aa);
            end
            array_distribucion(2,pos2)=array_distribucion(2,pos2)+1;
        end
        %ubicacion de los vehiculos
        for ii=1:length(array_distribucion(2,:))
            vehi(j,array_distribucion(1,ii))=vehi(j,array_distribucion(1,ii))+a
            rray_distribucion(2,ii);
        end
    end
end
  
```

```
Db_dist_coches(1,i).coches=vehi;  
clear vehi;  
end
```

```
save('Db_dist_coches', 'Db_dist_coches');  
clear i; clear j; clear jj; clear ii; clear cont; clear coef; clear  
pos; clear pos_2;
```

## Base Datos Vehículos Distribuidos

```

load('DB_dist_coches');
load('DBsubsectores_Completo');
vehiculos = xlsread('Datos_Vehiculos');

db_distribucion_sectores=subsector_xy;

%%%total coches, para reparto 10% aleatorio
tot_coches=sum(vehiculos(7,1:29));

for i=1:14000

db_distribucion_sectores(1,i).datos_carretera_vehiculos=zeros(1,length(db_distribucion_sectores(1,i).datos_carretera_id));
end

%%%arrays con limites x e y entre subsectores
for i=1:141
    array_xini(1,i)=subsector_xy(1,i).x_ini;
    if i==141
        array_xini(1,i)=subsector_xy(1,i-1).x_fin;
    end
    if i<101
        array_yini(1,i)=subsector_xy(1,(i-1)*140+1).y_ini;
    end
    if i==101
        array_yini(1,i)=subsector_xy(1,i-1).y_fin;
    end
end
end

contaerrores=0;
for i=1:29
    i
    for ii=1:25
        fin=find(Db_dist_coches(1,i).Xanillos(ii,:)==0,1)-1;
        for iii=1:fin
            pos=0;
            try

jx=find(array_xini>Db_dist_coches(1,i).Xanillos(ii,iii),1);

jy=find(array_yini<Db_dist_coches(1,i).Yanillos(ii,iii),1);
            n=jx-1+(jy-2)*140;

            pos=find(subsector_xy(1,n).datos_carretera_id(1,:)==Db_dist_coches(1,i).idanillos(ii,iii),1);

            db_distribucion_sectores(1,n).datos_carretera_vehiculos(1,pos)=...

            db_distribucion_sectores(1,n).datos_carretera_vehiculos(1,pos)+Db_dist_coches(1,i).coches(ii,iii);
            catch me
                contaerrores=contaerrores+1;
            end
        end
    end
end

```

```
end

end
end
end

for i=1:round(0.1*tot_coches)
    b=14000;
    aa=1;
    pos=round((b-aa)*rand(1,1) + aa);

    while
        isempty(db_distribucion_sectores(1,pos).datos_carretera_vehiculos)
            b=14000;
            aa=1;
            pos=round((b-aa)*rand(1,1) + aa);
        end

    b=length(db_distribucion_sectores(1,pos).datos_carretera_vehiculos)
    ;
    aa=1;
    pos2=round((b-aa)*rand(1,1) + aa);

    db_distribucion_sectores(1,pos).datos_carretera_vehiculos(pos2)=db_
    distribucion_sectores(1,pos).datos_carretera_vehiculos(pos2)+1;
end
save('db_distribucion_sectores', 'db_distribucion_sectores');
```

## Matriz Vehículos Vs Transmisores

```

load('db_distribucion_sectores');
load('DB_carreterasVsTx');

lim_km=20;

db_carreterasVsTx_binario=zeros(length(db_carreterasVsTx(:,1)),length(db_carreterasVsTx(1,:)));
for i=1:length(db_carreterasVsTx(1,:))
    for j=1:length(db_carreterasVsTx(:,1))
        if db_carreterasVsTx(j,i) < lim_km
            db_carreterasVsTx_binario(j,i)=1;
        end
    end
end

cont=0;
for i=1:14000
    if
~isempty(db_distribucion_sectores(1,i).datos_carretera_vehiculos)
        for
ii=1:length(db_distribucion_sectores(1,i).datos_carretera_vehiculos)
            cont=cont+1;

array_vehiculos(1,cont)=db_distribucion_sectores(1,i).datos_carretera_id(1,ii);

array_vehiculos(2,cont)=db_distribucion_sectores(1,i).datos_carretera_vehiculos(1,ii);
        end
    end
end

matriz_coches_tx=zeros(sum(array_vehiculos(2,:)),1+length(db_carreterasVsTx(1,:)));
con=0;
n=length(array_vehiculos(1,:));
for i=1:length(array_vehiculos(1,:))
    i/n
    for j=1:array_vehiculos(2,i)
        con=con+1;
        %%id de la carretera
        matriz_coches_tx(con,end)=array_vehiculos(1,i);
        %%distancias frente a los transmisores
        matriz_coches_tx(con,1:end-1)=db_carreterasVsTx_binario(array_vehiculos(1,i),:);
    end
end
m=round(length(matriz_coches_tx(:,1))/2);
matriz_coches_tx_1=matriz_coches_tx(1:m,:);
matriz_coches_tx_2=matriz_coches_tx(m+1:end,:);
  
```

```
save('matriz_coches_tx_1_20km','matriz_coches_tx_1');
save('matriz_coches_tx_2_20km','matriz_coches_tx_2');

%%%DEMASIADO LENTO%%%
% cont=0;
% for i=1:14000
%     if
~isempty(db_distribucion_sectores(1,i).datos_carretera_vehiculos)
%         for
ii=1:length(db_distribucion_sectores(1,i).datos_carretera_vehiculos
)
%             pos=0;
%
pos=find(db_carreterasVsTx(db_distribucion_sectores(1,i).datos_carretera_id(1,ii),:)<10);
%             pos(1,end+1)=414;
%             for
j=1:db_distribucion_sectores(1,i).datos_carretera_vehiculos(1,ii)
%                 cont=cont+1;
%                 for jj=1:length(pos)
%                     matriz_coches_tx(cont,pos(jj))=1;
%                 end
%             end
%         end
%     end
% end
% end
```

## Matrices Alcance

```
%%%Crear Matriz de alcance para diferentes distancias%%  
  
% load('DB_carreterasVsTx');  
  
% distancias=[5 10 20 40];  
  
%%%Se reordenan las distancias de menor a mayor  
%%%Se guarda la posición de DB_carreterasVsTx, que coincide con el  
id  
val_orden=zeros(length(db_carreterasVsTx(:,1)),length(db_carreteras  
VsTx(1,:)));  
pos_orden=zeros(length(db_carreterasVsTx(:,1)),length(db_carreteras  
VsTx(1,:)));  
for i=1:length(db_carreterasVsTx(:,1))  
    i=length(db_carreterasVsTx(:,1))  
    [val_orden(i,:),pos_orden(i,:)]=sort(db_carreterasVsTx(i,:));  
    pos_orden=cast(pos_orden,'single');  
end  
save('val_orden','val_orden');  
save('pos_orden','pos_orden');  
  
% pos_orden=cast(pos_orden,'single');
```



## Probabilidad Aceptación

```

%%Parametros
Coches_max=164000;%max coches por Tx
prob_med=1;
cont=0;

load('matriz_coches_tx_1_20km');
load('matriz_coches_tx_2_20km');
matriz_coches_tx=[matriz_coches_tx_1(:,1:end-1);
matriz_coches_tx_2(:,1:end-1)];
id_coches_tx=[matriz_coches_tx_1(:,end);
matriz_coches_tx_2(:,end)];

%%%Cuantos coches tiene cada transmisor
cantidad_coches_tx=zeros(1,length(matriz_coches_tx(1,:)));
for i=1:length(matriz_coches_tx(1,:))
    cantidad_coches_tx(1,i)=sum(matriz_coches_tx(:,i));
end
%%%A cuantos transmisores tiene acceso cada coche
cantidad_txs_coche=zeros(1,length(matriz_coches_tx(:,1)));
for i=1:length(matriz_coches_tx(:,1))
    cantidad_txs_coche(1,i)=sum(matriz_coches_tx(i,:));
end

matriz_probabilidades=2*matriz_coches_tx-1;

while prob_med >0.3
    cont=cont+1
    Coches_max=Coches_max+1000;
    for i=1:length(matriz_coches_tx(:,1))
        %
            i
            for j=1:length(matriz_coches_tx(1,:))
                if matriz_probabilidades(i,j)~-1
                    matriz_probabilidades(i,j)=max(1-
(Coches_max/cantidad_coches_tx(j)),0);
                end
            end
        end
    end

    probabilidad_aceptacion=zeros(1,length(matriz_coches_tx(:,1)));
    for i=1:length(matriz_coches_tx(:,1))
        %
            i
            for j=1:length(matriz_coches_tx(1,:))
                if matriz_probabilidades(i,j)~-1

                    probabilidad_aceptacion(1,i)=probabilidad_aceptacion(1,i)+matriz_
                    probabilidades(i,j);
                end
            end
        end
    end
end
  
```

```
probabilidad_aceptacion(1,i)=probabilidad_aceptacion(1,i)/cantida  
d_txs_coche(1,i);  
    end  
  
    sum=0;  
    cont_cobertura=0;  
    cont_no_cobertura=0;  
    for i=1:length(probabilidad_aceptacion)  
        if probabilidad_aceptacion(i)<1  
            sum=sum+probabilidad_aceptacion(i);  
            cont_cobertura=cont_cobertura+1;  
        else  
            cont_no_cobertura=cont_no_cobertura+1;  
        end  
    end  
  
    end  
  
    prob_med=(sum+cont_no_cobertura)/(cont_cobertura+cont_no_cobertur  
a);  
  
end
```

# Código de los algoritmos desarrollados en el Estudio Para Optimizar La Infraestructura De Red Para La Recogida De Datos Masivos Procedentes De Vehículos Empleando Métodos Metaheurísticos Multiobjetivo

## Vehículos Vs Estaciones Base

```
%%CARGAR DATOS NECESARIOS%%
load('db_distribucion_sectores');
load('db_carreterasVsTx');

%%DEFINICION DE PERFILES%%
dis1=5;
dis2=10;
dis3=30;
cap1=1000;
cap2=2000;
cap3=5000;
perfiles=[dis1 dis1 dis1 dis2 dis2 dis2 dis3 dis3 dis3 0; cap1 cap2
cap3 cap1 cap2 cap3 cap1 cap2 cap3 0; 0 0 0 0 0 0 0 0 0 0 ];
dis=[5 10 30];
cap=[1000 2000 5000];
%Se añade el coste a los perfiles%
cont=1;
for i=1:3
    for j=1:3
        perfiles(3,cont)=0.5*cap(j)+10*pi*dis(i)^2;
        cont=cont+1;
    end
end
clear dis; clear dis1; clear dis2; clear dis3; clear cap; clear cap1;
clear cap2; clear cap3;
clear cont; clear i; clear j;

%%ARRAY COCHES VS TX%%
cont=1;
for i=1:14000
    if
~isempty(db_distribucion_sectores(1,i).datos_carretera_vehiculos)
        for
ii=1:length(db_distribucion_sectores(1,i).datos_carretera_vehiculos)
            cont=cont+1;
            array_vehiculos(1,cont)=db_distribucion_sectores(1,i).dato
s_carretera_id(1,ii);
            array_vehiculos(2,cont)=db_distribucion_sectores(1,i).dato
s_carretera_vehiculos(1,ii);
        end
    end
end
```

```
end
end
clear i; clear ii; clear cont;

matriz_coches_tx=zeros(sum(array_vehiculos(2,:)),1+length(db_carreterasVsTx(1,:)));
con=0;
for i=1:length(array_vehiculos(1,:))
    for j=1:array_vehiculos(2,i)
        con=con+1;
        %%id de la carretera
        matriz_coches_tx(con,end)=array_vehiculos(1,i);
        %%distancias frente a los transmisores
        matriz_coches_tx(con,1:end-
1)=db_carreterasVsTx(array_vehiculos(1,i),:);
    end
end
clear con; clear i; clear j; clear ans; clear n;
matriz_coches_tx=single(matriz_coches_tx);

%%%ARRAY ENTRADA%%%
array_entrada=round(9*rand(1,413))+1;
```

## Algoritmo Cálculo Coste-Cobertura

```

%%%MATRIZ BINARIA SEGUN ARRAY ENTRADA%%%
m=length(matriz_coches_tx(:,1));
matriz_temporal=zeros(m,1);
matriz_coches_tx_final=matriz_coches_tx;

limites_tx=zeros(1,413);

for i=1:length(matriz_coches_tx(1,:))-1
    limites_tx(i)=perfiles(2,array_entrada(i));
    matriz_temporal=matriz_coches_tx_final(:,i);
    matriz_temporal(matriz_temporal>perfiles(1,array_entrada(i)))=0;
    matriz_coches_tx_final(:,i)=matriz_temporal;
    clear matriz_temporal;
end
matriz_coches_tx_final(:,end)=matriz_coches_tx(:,end);
clear cont; clear i; clear j; clear ans; clear m;%clear
matriz_coches_tx, clear ans;

%%%ARRAY DE TOTAL DE TX A LOS QUE SE PUEDE CONECTAR%%%
n_tx=zeros(length(matriz_coches_tx_final(:,end)),1);
for i=1:length(matriz_coches_tx_final(:,1))
    %encontrar la cantidad de valores diferentes a cero en el array
    %ultima columna es la cantidad de Tx a los que se puede conectar
    n_tx(i,1)=length(find(matriz_coches_tx_final(i,1:end-1)));
end
matriz_coches_tx_final(:,end+1)=n_tx;
clear i; clear n_tx;

%%%ORDENAR%%%
%Con el array n_tx se ordena ¿funcion?
%array con las posiciones segun orden de Tx
[~,pos]=sort(matriz_coches_tx_final(:,end));

%%%SE ASIGNAN COCHES A TX%%%
%Se crea un array con el contador
%una nueva columna en _final_v2-->  0!= cobertura(valor numero de Tx),
0=no
%cobertura
matriz_coches_tx_final(:,end+1)=zeros(length(matriz_coches_tx_final(:,
end)),1);%un nuevo valor, cobertura o no cobertura
for i=1:length(matriz_coches_tx_final(:,1))
    i
    pos_tx=find(matriz_coches_tx_final(pos(i),1:end-3)>0);
    for j=1:length(pos_tx)
        if limites_tx(pos_tx(j))>1
            limites_tx(pos_tx(j))=limites_tx(pos_tx(j))-1;
            matriz_coches_tx_final(pos(i),end)=1;
            break
        end
    end
end
end
end

```

```
clear i; clear j; clear n; clear pos_tx; clear pos;

%%%SE CALCULA LA COBERTURA Y COSTE%%%
%se emplea el mismo metodo que antes para saber si es o no un valor 0
%se obtiene la relacion cobertura-coste
coches_cobertura=sum(matriz_coches_tx_final(:,end));
coches_totales=length(matriz_coches_tx_final(:,end));
porcentaje_cobertura=coches_cobertura/coches_totales;
%Coste
coste_total=0;
for i=1:413
    coste_total=coste_total+perfiles(3,array_entrada(i));
end
clear i;
```