

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



"SISTEMA EXPERTO PARA LA
RECOGIDA INTELIGENTE DE
BASURAS"

TRABAJO FIN DE GRADO

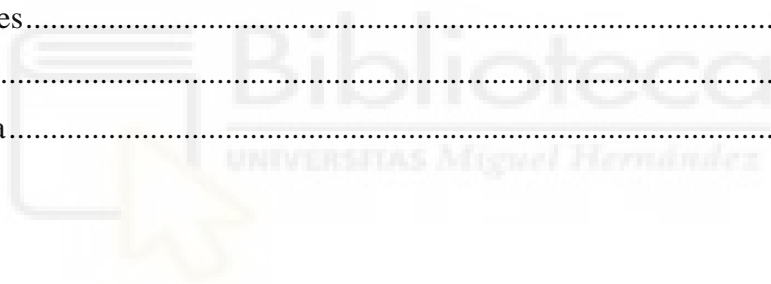
Septiembre - 2023

AUTOR: Alberto Botella Puig

DIRECTOR: José Luis Sainz-Pardo Auñón

ÍNDICE

Resumen	7
Abstract	9
1. Introducción.....	11 - 39
1.1. Introducción al proyecto.....	11 - 14
1.2. Smart Cities	15 - 22
1.3. Gestión inteligente de residuos	23 - 39
2. Material y métodos.....	41 - 50
2.1. Introducción.....	41
2.2. Travelling Salesman Problem.....	42
2.3. Formulación del TSP.....	43 - 47
2.4. Librería Lpsolve.....	48
2.5. Algoritmo.....	49 - 50
3. Resultados y discusión	52 - 53
4. Conclusiones.....	55
5. Anexos.....	57 - 65
6. Bibliografía.....	67 - 70



ÍNDICE DE FIGURAS

Figura 1: Smart City.....	15
Figura 2: Características de una Smart City	17
Figura 3: Contenedores elevados con mecanismos hidráulicos	24
Figura 4: Contenedores elevados con grúas	25
Figura 5: Diferencia entre contenedores soterrados con y sin plataforma	25
Figura 6: Recogida neumática	27
Figura 7: Funcionamiento sensor láser	29
Figura 8: Modelo VL53L1X	30
Figura 9: Funcionamiento sensor ultrasónico.....	30
Figura 10: Modelo HC-SR04	31
Figura 11: Funcionamiento sensor inductivo	31
Figura 12: Funcionamiento sensor de microondas	32
Figura 13: Sigfox	34
Figura 14: LoRa y LoRaWAN	35
Figura 15: Estructura del LoRa	36
Figura 16: NB-IoT	37
Figura 17: TSP.....	42
Figura 18: Ejemplo TSP en España	43
Figura 19: Matriz de incidencia del grafo	44
Figura 20: Función objetivo	45
Figura 21: Restricciones.....	45
Figura 22: Bucles / Subtours	46
Figura 23: Primera manera de romper un bucle	47
Figura 24: Segunda manera de romper un bucle	47
Figura 25: Algoritmo del programa	49
Figura 26: Croquis de los contenedores de Santa Pola	52
Figura 27: Solución del ejemplo.....	53

ÍNDICE DE TABLAS

Tabla 1: Ventajas y desventajas de los diversos sensores	33
Tabla 2: Características de las diferentes comunicaciones inalámbricas	38
Tabla 3: Características de las diferentes bases de datos	39
Tabla 4: Matriz no simétrica.....	57





RESUMEN

En la actualidad, existen muchos problemas medioambientales a los que se les debe buscar soluciones cuando antes. Es por ello por lo que, proyectos con la finalidad de, por ejemplo, reducir la contaminación acústica o atmosférica, entre otros, están a la orden del día.

En el presente proyecto, se pretende introducir una idea para lograr tal reducción en la contaminación que se sufre en las ciudades, proponiendo una monitorización de los contenedores de reciclaje. El objetivo tras esta monitorización consistiría en aumentar la ineficiencia que existe en la recogida de dichos contenedores debido al seguimiento lineal que tienen. Ya que siguen una hoja de rutas por la que pasan por todos los contenedores designados, independientemente de si están llenos, o no.

Dicha monitorización permitiría saber el estado en el que se encontrarían los contenedores, permitiendo observar y decidir cuáles deben ser, o no, recogidos cuando se realice la ruta con el camión de reciclaje. Esto permitiría una optimización de la ruta que siguen estos, consiguiendo un ahorro en gasolina y en ruido acústico provocado por el camión, en grandes ciudades, sobre todo.

La idea consistiría en implantar sensores a los contenedores de reciclaje, de tal manera que recojan datos sobre el volumen de los mismos. Dichos datos serían enviados vía Internet hacia una base de datos que estaría a disposición o de los servicios municipales o de la empresa encargada de dicho trabajo. Con esos datos, se decidirían qué contenedores no se deben recoger, imponiendo un umbral que será decidido por la entidad encargada de la recogida. Por último, un programa recogerá la valoración final, y buscará la ruta óptima que debe seguir el camión para recoger los contenedores de interés.



ABSTRACT

Currently, there are many environmental problems to which solutions must be sought as soon as possible. That is why, projects with the purpose of, for example, reducing acoustic or atmospheric pollution, among others, are the order of the day.

In the present project, it is intended to introduce an idea to achieve such a reduction in the pollution suffered in cities, proposing a monitoring of recycling containers. The objective after this monitoring would be to increase the inefficiency that exists in the collection of said containers due to the linear monitoring that they have. Since they follow a route sheet through which they go through all the designated containers, regardless of whether they are full or not.

Said monitoring would allow knowing the state in which the containers would be, allowing observing and deciding which ones should be, or not, collected when the route is made with the recycling truck. This would allow optimization of the route followed by these, achieving savings in gasoline and acoustic noise caused by the truck, especially in large cities.

The idea would consist of implanting sensors in recycling containers, in such a way that they collect data on their volume. Said data would be sent via the Internet to a database that would be available to either the municipal services or the company in charge of said work. With these data, they would decide which containers should not be collected, imposing a threshold that will be decided by the entity in charge of the collection. Lastly, a program will collect the final evaluation, and will search for the most optimal route that the truck must follow to collect the containers of interest.



1. INTRODUCCIÓN

1.1. INTRODUCCIÓN AL PROYECTO

La preocupación medioambiental ha pasado a tener un papel muy importante en todos los gobiernos del mundo, llegando al punto en el que se considera un problema de seguridad para los países. La contaminación es algo que está muy presente en nuestro día a día: contaminación lumínica, acústica, atmosférica, etc. Esto ha provocado que los gobiernos mundiales deban buscar e implantar nuevas soluciones para intentar que este problema no vaya a peor.

En esta insaciable búsqueda por intentar buscar mejoras a este problema, surge un concepto nuevo, el cual ya se ha empezado a aplicar en numerosas ciudades de todo el mundo: la Smart City. La idea que existe tras la Smart City es la de hacer que las ciudades se vuelvan tecnológicas, es decir, hacer que las ciudades hagan uso de tecnologías como ICT, o *big data*, las cuales son capaces de gestionar absolutamente todo de una manera sostenible y eficaz. Con ello, el objetivo final de una Smart City sería la reducción en el consumo de energía, reducción de las emisiones de CO₂ a la atmósfera y el aumento del bienestar de sus habitantes, entre otros.

La cantidad de propuestas que se pueden llegar a ofrecer en una Smart City es gigante. Es por ello por lo que, en este proyecto, vamos a proponer una propuesta, centrándonos en un aspecto muy concreto dentro de la Smart City: la recogida de residuos reciclables. Es bastante visible que existe una ineficiencia en la recogida de residuos, ya sea porque siguen rutas muy específicas, sin tener en cuenta la capacidad, en el momento, de los contenedores; o por las numerosas rutas que, por ello, se deben hacer para poder abarcar todos los contenedores de la ciudad.

Con el objetivo de lograr solucionar tales problemas, en este proyecto se proponen diversas soluciones. Una de ellas, tiene relación con la recogida neumática, la cual consiste en crear una red de tuberías, bajo el suelo, las cuales se encargan de recoger los desechos tirados a dichos contenedores y los transportar a la instalación encargada de reciclarlos. También, se proponen diversos tipos de contenedores que están enterrados bajo tierra los cuales ofrecen una serie de ventajas muy a tener en cuenta, como, por ejemplo, el impacto visual que tienen en las calles.

Por último, otra de las soluciones que se propone tiene relación con la optimización de la recogida, la cual será en la que nos centremos el resto del proyecto. Esta solución consiste en hacer uso de un sistema de monitorización, el cual permita llevar un seguimiento de la capacidad de cada contenedor por separado; de tal manera que seamos capaces de valorar y decidir qué contenedores son los que van a recogerse y cuáles no. Esto ya no solo permitirá una mayor optimización a la hora de recoger, sino que, también, obtendremos un ahorro en el uso de la gasolina de los camiones, evitando así perder tiempo innecesario. Además, también contribuirá a la mejora tanto en la contaminación atmosférica como en la acústica, al hacer un uso menor de los mismos. Dicho sistema se dividirá en dos partes fundamentales.

La primera parte estará ligada al propio contenedor. En este, se instalarán módulos con sensores volumétricos encargados de detectar cuál es la cantidad de residuos que se han acumulado. Los cuáles, posteriormente, harán uso de un método de transmisión para enviar la información que se han recogido a una base de datos.

La segunda parte está dedicada a la base de datos encargada de recoger toda la información. La información recogida ayudará a decidir un “umbral” el cuál se usará para poder elegir qué contenedores son los que se deben recoger y, además, se hará uso del *big data* para poder tener registros históricos de cada zona, lo cual nos ayudará a poder adelantarnos a situaciones en las que exista mucha mayor probabilidad de que los contenedores se llenen. Por ejemplo, en casos como ciudades turísticas en verano. Una vez elegidos los contenedores de interés, se meterán todos los datos en un programa que, con la localización de cada contenedor, creará una ruta óptima para ir a cada contenedor de interés y lo enseñará en una aplicación para los trabajadores.

A continuación, se hará un breve repaso sobre los temas que se van a desarrollar en el documento, para dar una idea general de lo que se va a exponer:

Introducción

Se empezará el proyecto realizando una introducción de todo el tema del que abarcará el mismo, haciendo hincapié en el nuevo concepto emergente: la Smart City; explicando beneficios y características, junto a algunos ejemplos de ciudades reales consideradas como *Smart Cities*.

Posteriormente, proseguiremos con la introducción a las soluciones de las que hemos hablado con anterioridad, entrando más a fondo sobre el funcionamiento, además de las ventajas y desventajas que se deben tener en cuenta a la hora de decidir su uso. De una misma manera, se hará, exactamente lo mismo, con la recogida neumática.

Por último, entraremos en el principal punto del proyecto, el sistema de monitorización. En éste, explicaremos con detalle todos los materiales que hemos escogido para su estudio, dando detalles de estos, y explicando cuáles serían los factibles para el sistema y cuáles no lo serían. Seguiríamos con la conexión inalámbrica hacia la base de datos, en la cual se propone diversos transportes de información de manera inalámbrica que serían factibles de implantar junto a los sensores. Para finalizar, se hablará de la base de datos, la cual se ha elegido una ya que es una base de datos gratuita y que actualmente es usado en muchas partes del mundo.

Material y métodos

En esta parte del proyecto, nos adentraremos en la parte final del sistema, centrándonos en el funcionamiento del programa encargado de encontrarnos la ruta óptima de los contenedores de interés. Introduciremos los grafos y el problema del TSP (*Travelling Salesman Problem*), en el cual nos apoyamos para hacer el programa.

Se explicarán la formulación que se ha seguido y pequeños detalles, como la eliminación de bucles, que nos pueden molestar en la búsqueda de las rutas. Y, por último, terminaremos explicando en profundidad el algoritmo que se ha desarrollado para hacer este programa.

Resultados y discusión

Para finalizar este proyecto, haremos uso de un ejemplo real de una ruta que siguen aquellos encargados de recoger los contenedores de reciclaje en Santa Pola. De tal manera que se propondrá un caso en el cual hemos implantado una serie de sensores en los contenedores y nos han dicho cuáles son los contenedores de interés que se deben recoger en ese momento; teniendo que obtener la ruta óptima para pasar por esos contenedores haciendo uso del programa que se ha desarrollado.



1.2. SMART CITIES

En los últimos años, ha estado habiendo, y sigue a día de hoy, un gran crecimiento en la población mundial. Esto ha provocado que se haya llegado a superar la población urbana respecto a la rural, haciendo que las ciudades tengan un mayor impacto a nivel social y económico en los países. Esta evolución que se está viviendo, plantea un cambio muy importante respecto a las gestiones que se deben hacer de las infraestructuras dentro de las ciudades.

Es aquí donde entra el concepto de *Smart City*. Como tal, este concepto no tiene aún, hoy en día, una definición concreta. Sin embargo, aunque existen muchas explicaciones sobre este concepto, podemos hacer una simple explicación de lo que trata el concepto. Una *Smart City* es un lugar en el que se han hecho más flexibles, eficientes y sostenibles las redes y servicios tradicionales, haciendo uso de tecnologías digitales y de telecomunicaciones; mejorando las operaciones con el objetivo de beneficiar a sus habitantes.



Figura 1: Smart City

Fuente: <https://canteradeempresas.com/que-son-las-smart-cities-y-que-importancia-tienen-en-nuestro-proyecto/>

Las *Smart Cities* hacen gran uso de diversas tecnologías como, por ejemplo, sensores, redes de comunicación y análisis de datos; con el objetivo de poder recopilar información sobre lo que ocurre en la ciudad. Cosas como el tráfico, la calidad del aire que se respira, el uso de la energía, la gestión de los residuos, etc. Toda esta información es recopilada, y se usa para mejorar la seguridad, movilidad, eficiencia energética y otros aspectos de la vida urbana. Además, también se intenta buscar la involucración de los ciudadanos en la toma de decisiones y en la gestión de la ciudad, fomentando, de esta manera, su participación y la transparencia a la hora de tomar decisiones.

Beneficios de una *Smart City*

Mientras la definición del concepto de *Smart City* sigue decidiéndose, hay una cosa que ha quedado clara. La tecnología de una *Smart City* hace que la ciudad sea más efectiva y eficiente, lo cual es necesario debido al rápido crecimiento de la población. Es por ello por lo que vamos a detallar, de manera breve, algunas de las ventajas que nos podrían ofrecer:

i. Tomas de decisiones basadas en datos

El avance en *big data* y en IoT, ha permitido a las ciudades acceso a información que nunca habían tenido disponible, permitiendo crear numerosas estrategias en base a la masiva información que pueden recibir. Esto permite casos como, por ejemplo, de realizar estrategias policiales en zonas de alto riesgo, de identificar los intereses, preocupaciones o necesidades de los ciudadanos, entre otras cosas. Estas dos tecnologías permiten un sinnúmero de opciones permitiendo una toma de decisiones sólida.

ii. Mejoras en el transporte

La conectividad de los sistemas de transporte tiene uno de los mayores potenciales para mejorar, de manera drástica, la eficiencia en las ciudades. Hablamos de tecnologías como señales de tráfico inteligentes, capaces de optimizar el flujo del tráfico o aliviar las congestiones durante horas pico; o como gestión de aparcamientos inteligente. Todas estas tecnologías permiten a las ciudades servir mejor a sus ciudadanos a pesar del rápido crecimiento en la población.

iii. Servicios públicos eficientes

Teniendo en cuenta las limitaciones que tenemos en cuanto a recursos naturales, el uso de las tecnologías inteligentes brinda a las ciudades las herramientas necesarias para poder conservar y reducir, de una manera efectiva, desperdicios involuntarios de agua y electricidad. Por ejemplo, los sensores inteligentes ahora son capaces de identificar cuando se ha producido una fuga en las tuberías, permitiendo una mayor rápida acción, reparando los daños en tiempos más breves, y reduciendo las cantidades de agua perdidas.

iv. Mejora en las infraestructuras

Con los años, carreteras, puentes y edificios, a menudo, empiezan a requerir una inversión gigante para poder llegar a mantenerlos en un buen estado. Las tecnologías actuales pueden proveer de información valiosa a las ciudades, realizando análisis predictivos sobre las estructuras antes de que haya un fallo en las mismas. También, los sensores inteligentes pueden transmitir datos identificando posibles inclinaciones o grietas en las diversas estructuras, notificando al personal sobre la necesidad de realizar una inspección o mantenimiento.

Características de una *Smart City*

Al igual que ha ocurrido a la hora de definir el concepto de *Smart City*, tampoco hay mucha información sobre las características que definirían una *Smart City*. Sin embargo, una consulta líder, llamada *Frost and Sullivan*, ha aportado una imagen que muestra características coherentes que debería tener una *Smart City*.

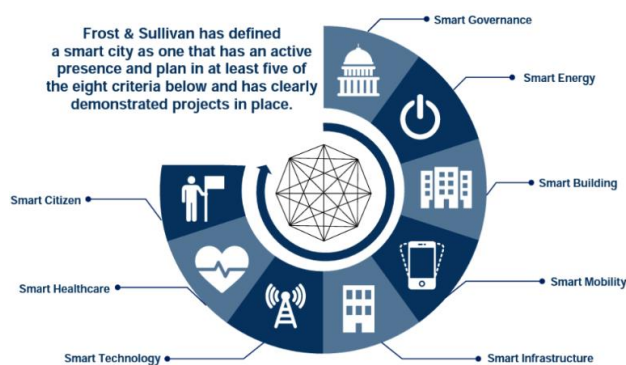


Figura 2: Características de una *Smart City*

Fuente: <https://itchronicles.com/smart-city/what-are-the-features-of-a-smart-city-to-look-for/>

Es por ello por lo que, a continuación, vamos a describir, brevemente, las 8 características que debería tener una *Smart City*:

i. Smart Governance

Las *Smart Cities* deben tener una presencia activa y un plan de gobernanza que permita mejorar la participación pública mediante el uso de diferentes herramientas e iniciativas. Esto permite la mejora en la comunicación con los ciudadanos a través de, por ejemplo, teléfonos o aplicaciones móviles. Además de hacer que los servicios sean más accesibles mediante estas herramientas, también permite que los servicios públicos sean más asequibles, responsables y transparentes.

ii. Smart Energy

La contaminación es uno de los mayores problemas que está creando la rápida urbanización. Es por ello por lo que gestionar correctamente la energía es una de las maneras de ser capaces de reducir la huella de carbono de la ciudad y descarbonizar el consumo de energía.

La *Smart Energy* consistiría en la correcta gestión digital de la energía en las ciudades, incluyendo el uso de redes inteligentes, medidores e inteligentes almacenamientos de energía. Esto daría como ventaja un gran ahorro a la hora de realizar operaciones en infraestructuras públicas y privadas. Además, también permitiría a las ciudades comprender su demanda de energía, pudiendo comprender las cargas dominantes y las fluctuaciones diarias para priorizar un consumo reducido de energía.

iii. Smart Building

Cuando hablamos de *Smart Buildings*, hacemos referencia a edificios automatizados e inteligentes, en los que se incluye la integración de energía renovable. Estos deben estar diseñados para maximizar la productividad y la eficiencia de la misma, haciendo uso de nuevas tecnologías que ofrezcan un lugar seguro y conectado para las personas trabajando dentro del edificio.

Para ello, se usan diferentes tecnologías, como lo pueden ser los de acceso y autenticación de personas. La tecnología IoT es el corazón de los *Smart Buildings*, permitiendo ayudar a mejorar la conectividad del edificio, al mismo tiempo que su eficiencia.

iv. *Smart Mobility*

La *Smart Mobility* tiene como objetivo mejorar la movilidad urbana de los ciudadanos, aportando nuevas innovaciones como los coches autónomos, autobuses eléctricos, bicis eléctricas, etc. Todo esto pondrá a la industria del transporte al límite de la disrupción masiva, impulsando la necesidad de crear redes de transporte más sólidos. Además de todo esto, también se incluirán reservas y pago de viajes utilizando dinero digital y tecnología que conecta las infraestructuras con los ciudadanos.

v. *Smart Infrastructure*

El ecosistema de las ciudades está cambiando a un ritmo vertiginoso, por lo que es importante que las infraestructuras de la ciudad, también, puedan crecer y adaptarse al mismo ritmo. Las *Smart Infrastructures* permiten la gestión digital de las infraestructuras, haciendo uso de sensores IoT en áreas clave como lo son la gestión digital del agua y de los desechos. Las *Smart Cities* tienen como objetivo mejorar la infraestructura y la prestación de servicios aprovechándose de la tecnología, la información y los datos. Esto abarca todos los aspectos de la infraestructura de una ciudad, desde el agua hasta los desechos, desde la electricidad hasta la educación, desde la atención médica hasta la conectividad IT.

vi. *Smart Technology*

La *Smart Technology*, por razones que son obvias, es una característica fundamental dentro de una *Smart City*. Sin embargo, *Frost and Sullivan* destacan la conectividad perfecta como la clave de la *Smart Technology*, ya que sin conectividad y sin la capacidad de vincular tecnologías, la *Smart City* se derrumbaría.

Este tipo de tecnología inteligente se puede instaurar en, prácticamente, casi todos los sectores de una ciudad: desde la generación de energía hasta el uso de desechos para crear un nuevo fertilizante. Claro está, todo haciendo uso de datos. Ya que, si no recopilásemos datos, sería imposible medir los impactos que han tenido los cambios que se han realizado.

vii. *Smart Healthcare*

Uno de los sectores más grandes en adoptar las tecnologías son el sector de la salud. Las tecnologías IoT han permitido crear un ecosistema en el que los pacientes pueden ser rastreados y monitorizados de forma remota, reduciendo en gran medida la presión sobre los hospitales y proveedores médicos al limitar la cantidad de tiempo que las personas pasan yendo a las citas para ver a un profesional.

Aunque, en la actualidad, solamente estamos usando, prácticamente, la superficie de todo el potencial que tenemos con la tecnología, cada vez más se irán añadiendo diversas tecnologías que aportarán diversas ventajas a varias situaciones de alto peligro, como lo pueden ser salas en las que hay enfermedades infecciosas dentro de un hospital.

viii. *Smart Citizens*

La última característica que define una *Smart City* son los propios ciudadanos, ya que son esenciales para el éxito general de una *Smart City*. No tiene sentido proporcionar a las personas acceso a la tecnología si no tienen idea de cómo usarla o cómo puede beneficiarlos. Es por ello que es importante que se quiera instruir a la población sobre las ventajas o beneficios que puede llegar a aportar convertirse en una *Smart City*. Los ciudadanos están empoderados a aprender, desarrollarse y crecer con la ciudad que los rodea, lo cual ayuda a impulsar un ciclo de mejora continua.

Ejemplos de *Smart Cities*

- Singapur, República de Singapur

Esta *Smart City* es la piedra angular del mundo. Singapur ha logrado mejorar la seguridad, fortaleciendo las instalaciones públicas y reduciendo los tiempos de espera para los servicios esenciales. Todo en esta ciudad está diseñado para mejorar la calidad de vida. Gracias al uso de tecnologías disruptivas y al uso del IoT, los ciudadanos pueden acceder a la información desde cualquier lugar y realizar casi todas las transacciones en línea.

- Zúrich, Suiza

Zúrich es una de las pocas ciudades que pone a los ciudadanos en el centro de su estrategia, encontrando un equilibrio entre el ecosistema y el desarrollo urbano. Se destaca por su inversión en redes e infraestructuras de fibra óptica, así como una población que cuenta con altas habilidades digitales. Otras áreas creativas que la convierten en una ciudad inteligente son la gestión de residuos y el uso que tiene de la energía, la seguridad digital, la reducción de carbono, y la inversión en salud y educación para cerrar la brecha digital en la población.

- Oslo, Noruega

Uno de los logros de Oslo es que, junto con muchas asociaciones públicas y privadas, han aparecido numerosas iniciativas de empresarios para ofrecer planes de *Smart Cities*, con el objetivo de mejorar la vida de los ciudadanos. La capital noruega es conocida por su educada población, su buena gestión de residuos y una cantidad enorme de espacios verdes. Como *Smart City*, tiene políticas dedicadas a la reducción del consumo de energía y la descongestión del tráfico urbano. Además, está muy comprometida con el desarrollo ecológico y sostenible, ya que se ha demostrado que beneficia el desarrollo comercial y social.

- Taipéi, Taiwán

El uso de la tecnología y la promoción de diversas formas de innovación ha tenido un impacto positivo en muchos aspectos: salud, seguridad, movilidad, gobernanza, entre otras. Esta metrópoli hace combinar tecnología, cultura y liderazgo para afrontar los retos que plantean las *Smart Cities*. La estrategia que siguen es promover el transporte público para expandir el desarrollo de la economía, mejorar la seguridad pública, y controlar y reducir la contaminación. Por ejemplo, con la tecnología *Smart City*, los ciudadanos pueden obtener información en tiempo real sobre la temperatura, la humedad y la calidad del aire.

- Lausanne, Suiza

Lausana mejora la calidad de vida de sus ciudadanos a través de cuatro iniciativas estratégicas: energía, movilidad, edificios y comercio. Las altas competencias digitales de la población, junto a la cultura tecnológica de la misma, ayudan a recopilar datos y patrones de comportamiento para crear respuestas eficaces a sus necesidades. La ciudad también es conocida por ser pionera en la creación de eco vecindarios, es decir, viviendas que cumplen desde el inicio de la obra todos los requisitos ambientales y energéticos.

1.3. GESTIÓN INTELIGENTE DE RESIDUOS

Como ya hemos hablado de ello, el crecimiento de la población en las zonas urbanas está siendo enorme, provocando que los residuos desechados también hayan aumentado debido a ello. Al no haber un sistema encargado de rastrear ni monitorizar todos estos desechos que son desechados en los contenedores, en numerosas ocasiones, nos encontramos con zonas en las que los contenedores no han sido capaces de almacenar todos los residuos.

Estas situaciones provocan malestar a los ciudadanos debido al mal olor, y provoca, además, problemas de salud en ellos. Es cierto que las ciudades se están volviendo, cada vez, más inteligentes, pero éstas no están adoptando ningún tipo de medidas para contrarrestar este problema que está surgiendo. Es debido a todo esto que, en este proyecto, vamos a aportar diversas soluciones que se podrían adoptar de cara a frenar estos problemas que están apareciendo en las ciudades.

i. Contenedores soterrados

Una de las primeras soluciones que se propone es el uso de contenedores soterrados, o, también conocidos como contenedores subterráneos. Este tipo de contenedores se utiliza para que los usuarios depositen los residuos que van a desechar y tienen como característica principal, al contrario que los contenedores tradicionales que conocemos, de quedar bajo tierra bajo una tapadera, generalmente, metálica.

Estos contenedores son capaces de almacenar hasta un 60% más de basura, ya que son capaces de expandir la capacidad de almacenamiento que tienen en el suelo. Además, otra ventaja que ofrecen estos contenedores es que son capaces de recoger todos los residuos en un espacio mínimo. Esto se vuelve muy importante hoy en día, ya que en las zonas urbanas hay cada vez menos espacio, y las filas de contenedores tradicionales va en aumento debido a los problemas que hemos mencionado anteriormente.

Otra cosa muy común en la mayoría de los países es que se suelen prender de fuego los contenedores tradicionales en las protestas que suelen ocurrir. Al ser elementos inmóviles los contenedores soterrados, eso provocaría una importante reducción del daño en las vías públicas, ya que no tendrían mucho interés prenderles fuego a esos contenedores.

Tipos de contenedores soterrados

- Contenedores soterrados con plataforma

En este tipo de contenedores se sitúan dentro de un foso que está cubierto con una plataforma dentro, la cual ayuda a sacar los residuos almacenados. Además, los residuos son depositados por los ciudadanos a través de un buzón. Sin embargo, a la hora de realizar el vaciado del contenedor, existen dos formas de sacar los residuos:

- Contenedores elevados con mecanismos hidráulicos: Este tipo de mecanismos están hechos, principalmente, para camiones que tengan carga trasera o lateral. De tal manera que funcionan como un cajón que se esconde bajo tierra, teniendo un mecanismo en el interior que lo eleva a la superficie y hace que sea fácil el vaciado del contenedor.

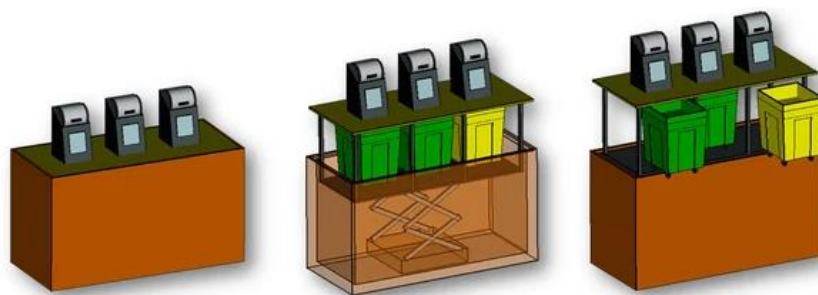


Figura 3: Contenedores elevados con mecanismos hidráulicos

Fuente: <https://www.molok.com/es/blog/contenedores-soterrados>

- Contenedores elevados con grúa: Este tipo de contenedores están pensados para ser capaces de sustituir iglús u otro tipo de contenedores de carga superior. Para ello, se necesitarán camiones capaces de elevar estos contenedores.

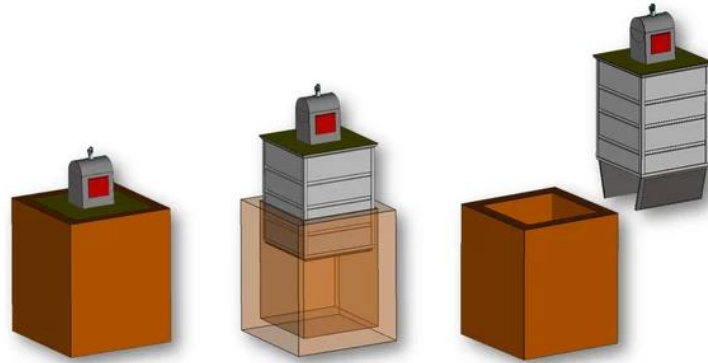


Figura 4: Contenedores elevados con grúas

Fuente: <https://www.molok.com/es/blog/contenedores-soterrados>

- Contenedores soterrados sin plataforma

Este tipo de contenedores, al contrario que los anteriores, no hacen uso de una plataforma. Por lo que, este contenedor se entierra de forma parcial, quedando a la vista un tercio de este. También son denominados contenedores semi soterrados, debido a que no están del todo enterrados en la superficie.

Para poder apreciar una diferencia entre los dos tipos de contenedores soterrados, haremos uso de la siguiente figura:

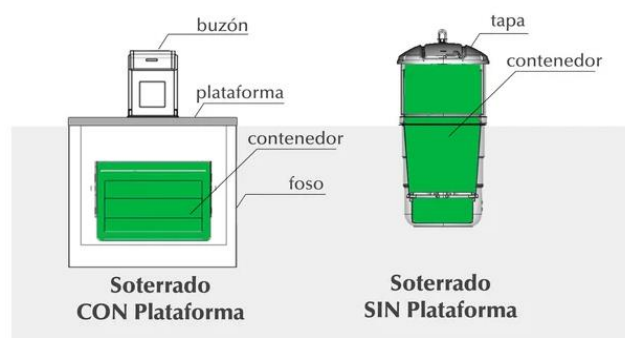


Figura 5: Diferencia entre contenedores soterrados con y sin plataforma

Fuente: <https://www.molok.com/es/blog/contenedores-soterrados>

ii. Recogida neumática

Durante muchísimo tiempo, el único método que se ha utilizado para recoger y transportar residuos han sido los camiones de basura de carga trasera o lateral, técnicamente conocidos como RCV (*Refuse Collection Vehicles*), con mano de obra que apoya el movimiento de los contenedores hacia los vehículos. Durante todo este tiempo, el método de transporte no se ha cambiado, implicando un gran trabajo para aquellos que lo realizan y contaminando debido al uso de la gasolina en los camiones.

Si hablamos desde una perspectiva medioambiental, son muchos los daños que se ocasionan debido a los volúmenes de los residuos generados por el aumento de la población y por el uso, mayoritario, de los camiones de basura. Es por ello por lo que, hoy en día, muchas ciudades han empezado a implantar una solución nueva y única a la recogida de residuos en aquellas áreas densamente pobladas. Es el llamado sistema de recogida neumática, o popularmente conocido como AWCS (*Automated Waste Collection System*).

Los sistemas de recogida neumática de residuos se basan en una red de buzones de vertido que están conectado, mediante tuberías subterráneas, al punto de captura en el que se realiza una aspiración del circuito. Estos buzones se pueden poner tanto en el interior de las viviendas como en áreas que son comunitarias como en áreas públicas. Los residuos caen, debido a la gravedad, hasta las válvulas instaladas subterráneamente, donde se acumulan temporalmente hasta que se realiza el proceso de aspiración. Existen dos sistemas para poder realizar este proceso:

- Sistema estático: En este tipo de sistema, se cuenta con una central fija que puede ser ubicada tanto en la vía pública como en el subterráneo. Esta central cuenta con un ordenador que coordina los programas de vaciado de manera periódica o el vaciado por niveles de llenado de las válvulas. El mecanismo de aspiración funciona con la creación de depresión en la red de cañerías, introduciendo aire que permite aspirar los residuos hasta la central, donde se almacenan y se destinan a sus correspondientes plantas de tratamiento.

- Sistema móvil: Este tipo de sistema cuenta con varias bajantes, que están puestas verticalmente, las cuales están conectadas a unos contenedores que, a su vez, están conectados, mediante tuberías, entre ellos por grupos. Cada uno de estos grupos tiene un punto de succión por el que pasa un vehículo encargado de aspirar los residuos que se desechan de forma periódica.



Figura 6: Sistema móvil de recogida neumática

Fuente: <https://www.esmartcity.es/comunicaciones/comunicacion-sistemas-recogida-neumatica-residuos-eliminaron-emisiones-co2-atmosfera-equivalentes-436-piscinas-olimpicas-ano>

iii. Monitorización de la recogida de residuos

En los anteriores apartados, hemos descrito diferentes soluciones que pueden aportar mejoras a los problemas que están surgiendo hoy en día. Empezamos el apartado explicando la importancia que podrían tener los contenedores soterrados, enseñando los diferentes tipos y sus tipologías. A continuación, se sigue el apartado dando paso a la recogida neumática, una nueva forma de recoger los residuos tirados en los contenedores de una manera silenciosa y segura para el ciudadano. Sin embargo, aún queda una solución más de la que hablar, y esta será en la que nos centremos el resto del proyecto. Esta solución consiste en la aplicación de tecnologías, que existen en la actualidad, para mejorar la eficiencia con la que se recogen los residuos reciclables de los contenedores.

En la actualidad, existe una tradicionalidad en cuanto al tema de la recogida de residuos, ya que a la hora de decidir sobre las rutas que deben seguir los camiones encargados de la recogida, normalmente son rutas basadas en la experiencia del personal contratado, o rutas muy fijas que, rara vez, tienen una modificación en cuanto a la ruta a seguir. De tal manera que suele ser una situación muy común el encontrarse con contenedores que no están del todo llenos, otros que están desbordados de basura, o incluso colapsados. Además, a todo esto, le añadimos el problema de que la longitud de las rutas se va incrementando debido al aumento de la población, ya que se deben poner una mayor cantidad de contenedores en diferentes lugares para poder abarcar toda la población. Todo esto provoca una grandísima ineficiencia, que se podría reducir considerablemente si se hiciese un uso correcto de las tecnologías.

La solución que se pretende implantar se va a dividir en diversos puntos. En el primero de todos, hablaremos sobre los sensores. Estos sensores van a tener la función de recoger datos sobre la capacidad del contenedor, realizando mediciones sobre la cantidad de residuos que hay dentro de los contenedores. En el segundo punto, hablaremos sobre la transmisión de los datos a una central. Para ello, haremos uso de tecnologías IoT, las cuales se encargarán de recoger los datos de los sensores y de enviarlos, por Internet, hacia la central encargada de recogerlos.

En el tercer y último punto, hablaremos sobre las bases de datos, las cuales se encargarán de recoger y guardar todos los datos que se han enviado desde los sensores. Además, gracias al uso del *Big Data*, podremos guardar los datos que vamos recaudando de los sensores, pudiendo hacer futuros estudios sobre las épocas que mayor volumen de residuos es recogido, los días de la semana que más o menos residuo se deja en los contenedores, y mucho más. Con todos estos datos, se usará un programa, el cual se desarrollará en profundidad más adelante, el cual se encargará de optimizar las rutas que siguen los trabajadores encargados de recoger los contenedores en base a los contenedores que interesen en el momento.

Este tipo de sistema estará enfocado en la recogida de basura reciclable, como, por ejemplo, vidrio, cartón, etc. Esto es debido a que, en el caso de basura orgánica, no sería posible aplicar esto ya que, si dejásemos ese tipo de residuo durante mucho tiempo en el contenedor, sería insalubre y acumularía malos olores, además de otros efectos negativos.

Sensores en los contenedores

Como ya hemos explicado, necesitamos sensores que sean capaces de detectar la capacidad de los contenedores. Dicho de otra manera, podrías decir que necesitamos sensores volumétricos, sensores capaces de determinar el volumen de un objeto. Puesto que debemos tener en cuenta que los sensores nunca deben ser dañados, la mejor opción que hay sería ponerlos dentro y en la parte alta del contenedor. A continuación, se desarrollan diversos sensores que pueden ser útiles para nuestro propósito:

- Sensor láser

Este tipo de sensores miden la distancia a la que se encuentra el objeto en función de la intensidad de la luz que reciben. La forma en la que funcionan es emitiendo una luz en línea recta desde un emisor, y recibiendo el haz de luz en el receptor, en el cual se mide la intensidad de la luz que ha recibido. Dentro de los tipos de sensores láser que hay, los que nos interesarían serían aquellos con transmisor y receptor en el mismo módulo, ya que la idea es instalar el sensor en la parte superior del contenedor para que no resulte dañado.

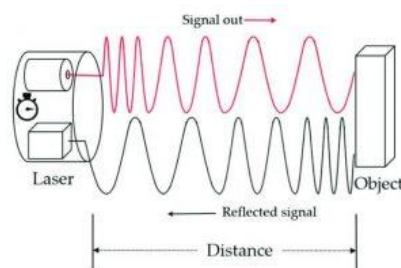


Figura 7: Funcionamiento sensor láser

Fuente: <https://www.omch.co/es/laser-distance-sensor/>

Modelos de sensores como VL53L1X o VL53L0X pueden ser de interés, ya que hacen uso de un sistema llamado Time of Flight (ToF). Este sistema mide el tiempo de vuelo a través de la emisión de pulsos de luz infrarroja en la frecuencia de los 20 MHz. Se diferencian de los sensores convencionales ya que estos se basan en la cantidad de intensidad de luz que se refleja para evaluar la distancia a la que está el objeto.



Figura 8: Modelo VL53L1X

Fuente: <https://www.pishop.ca/product/vl53l1x-time-of-flight-tof-sensor-breakout/>

- Sensor de ultrasonidos

Este tipo de sensores ultrasónicos miden la distancia respecto al objeto mediante ondas ultrasónicas. La forma en la que calculan la distancia es a través del tiempo que transcurre, desde el momento en el que se envía la onda hasta el momento en el que es recibida por el propio sensor, después de que haya sido reflejada por el objeto. Podemos decir que trabajan con el mismo principio que el sensor láser introducido anteriormente, con el sistema ToF, pero haciendo uso de ondas ultrasónicas en vez de láser infrarrojo.

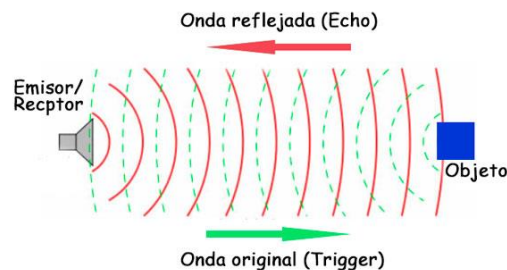


Figura 9: Funcionamiento sensor ultrasónico

Fuente: <https://elctavobit.com/arduino/sensor-ultrasonico-hc-sr04-y-arduino/>

Uno de los modelos de interés puede ser el sensor HC-SR04, ya que es un ejemplo muy usado en muchos sectores como sensor.



Figura 10: Modelo HC-SR04

Fuente: <https://es.geekbuying.com/item/HC-SR04-Ultrasonic-Sensor-Distance-Measuring-Module-3-3V-5V-Compatible-for-Arduino-NodeMCU-360876.html>

- Sensor inductivo

A diferencia de los sensores que hemos descrito anteriormente, este tipo de sensores tienen una particularidad y es que necesitan una superficie metálica para ser capaces de realizar las medidas. En este caso, el sensor contiene un devanado interno por el cual circula la corriente, generando un campo magnético en el momento en el que se le introduce corriente al material. A medida que se acerca al objeto, el flujo de corriente inducido aumenta, disminuyendo de esa manera la oscilación. Esa variación que se produce emite una señal indicando que se ha detectado un objeto.

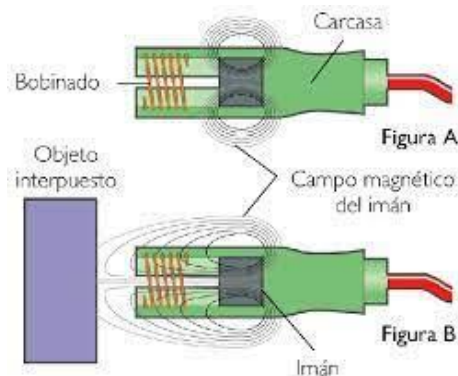


Figura 11: Funcionamiento sensor inductivo

Fuente: https://www.ecured.cu/Sensor_inductivo

- Sensor capacitivo

El sensor capacitivo es un sensor cuya función consiste en detectar la distancia que existe entre un objeto y el propio sensor. Para ello, se basa en la variación que tiene la corriente cuando el objeto varía la distancia respecto al sensor, siendo capaz de medirlo mediante la capacidad eléctrica.

La capacidad eléctrica es la relación que existe entre la capacidad de carga que es capaz de acumular un material conductor y la tensión eléctrica, es decir, la diferencia de potencial entre dos conductores. Cuando se produce esta diferencia de potencial entre las placas, se crea un campo eléctrico que provoca un almacenamiento de cargas eléctricas. La variación de esa carga es la que le indica al sensor la distancia a la que se encuentra el objeto.

- Sensor microondas

Este tipo de sensores se basan en el principio del efecto Doppler. Como si hablásemos de un radar, realizan la detección de los objetos mediante la emisión de señales del rango de frecuencia del microondas y midiendo el tiempo que tarda en recibirse el eco. Si el objeto se encuentra en movimiento, la señal rebotada recibida estará modificada en frecuencia, generando una señal de salida digital.



Figura 12: Funcionamiento sensor microondas

Fuente: <https://mkelectronica.com/producto/sensor-movimiento-microondas/>

Tipo de sensor	Ventajas	Desventajas
Láser	Radio de detección → 1'7 – 3'6 m Bajo consumo → 16 mA FOV → 15° – 27° Detección de materiales tanto metálicos como no metálicos	Resolución → ± 20 mm
Ultrasonido	Radio de detección → 2 – 400 cm Bajo consumo → 15 mA FOV → 30° Detección de materiales tanto metálicos como no metálicos	Resolución → ± 0'3 cm
Inductivo	Precisión → ≤ 2% de escala completa	Radio de detección → 15 – 40 mm
Capacitivo	Sensibilidad → 1000 ± 0'2% μm/V	Radio de detección → ≤ 10 mm
Microondas	Radio de detección → 8 m FOV → 60° - 70°	Alta vulnerabilidad ante falsas alarmas

Tabla 1: Ventajas y desventajas de los diversos sensores

Fuente: Elaboración propia

Observando la tabla, podemos apreciar que los sensores láser y ultrasonido son los únicos que nos serían útiles en proyecto, ya que los sensores inductivos y capacitivos disponen de un rango de detección muy bajo para la idea que se está planteando. Además, a pesar de que el sensor microondas tiene un rango muy alto de detección, presenta un problema con las falsas alarmas, lo cual podría afectar a nuestra idea ya que queremos obtener datos fiables sobre los contenedores.

Comunicación inalámbrica

Hemos hablado sobre los sensores encargados de detectar la cantidad de basura recogida en los contenedores. Ahora vamos a hablar de la tecnología inalámbrica que usaremos para enviar esos datos que recoge el sensor para enviarlos directamente a la supuesta central encargada de recibir esos datos.

Para realizar esta parte del capítulo nos apoyaremos en las tecnologías IoT. Básicamente, el IoT es un proceso que permite la conexión de elementos físicos a Internet. La forma en la que influye esto en nuestro caso es que vamos a usar una serie de tecnologías inalámbricas que deben ser capaces de conectarse a internet para poder ser capaces de enviar esa información que requerimos a la central de datos.

Al tener que usar un rango de cobertura grande, ya de por sí descartaremos posibles opciones como lo son Bluetooth o Zigbee, ya que el rango que proporcionan es de 10 y 75 metros, respectivamente. Por lo que, a continuación, desarrollaremos los diversos métodos de comunicación inalámbrica que podrían ser útiles para nuestro proyecto:

- Sigfox

Sigfox es una red tecnológica que hace uso de la banda UNB (Ultra Narrow Band). Esta banda permite a los dispositivos funcionar a bajas velocidades, con transferencia de datos de 10 a 1.000 bits por segundo. Asimismo, esta tecnología hace uso de un canal de espectro muy estrecho, aproximadamente de 1 KHz de anchura. Esto permite a los dispositivos tener un poder de penetración muy alto, consiguiendo un gran alcance inalámbrico de hasta 20 km en zonas de campo abierto, y de hasta 1,5 km en zonas urbanas.

Con todo esto, los sensores podrán enviar datos a unas antenas Sigfox que serán las encargadas de recibir las señales enviadas por los dispositivos para, posteriormente, enviar la información a un servidor Sigfox, y de ahí enviarlo al lugar que nos interese. La figura siguiente resume lo que acabo de explicar:

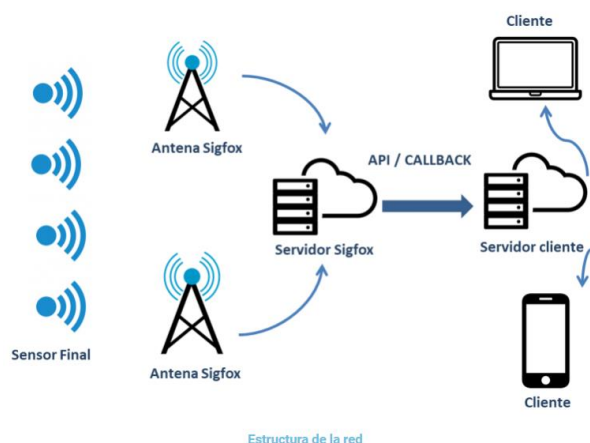


Figura 13: Funcionamiento de Sigfox

Fuente: <http://productos-iot.com/sigfox-3/>

- LoRa

LoRa, de sus siglas en inglés *Long Range*, es una tecnología estandarizada por LoRa Alliance que permite el desarrollo de redes IoT. Esta tecnología hace uso de modulación del tipo *spread spectrum* (amplio espectro).

Este tipo de modulación permite a la señal tolerar el ruido en la comunicación, permite caminos múltiples a la señal y hace uso del efecto Doppler, por lo que mantiene un consumo de energía muy bajo. Las frecuencias que usa esta tecnología varían dependiendo del país, pero suelen ser usadas las bandas de 433 MHz, 868 MHz y 915 MHz. También, tienen una velocidad de transmisión de datos muy baja, entre 0,3 y 5,5 Kbps. Esto permite que el alcance que tienen este tipo de comunicaciones pueda llegar a ser de hasta 15 km en caso de ser un área abierta y de 2 km en zonas urbanas.

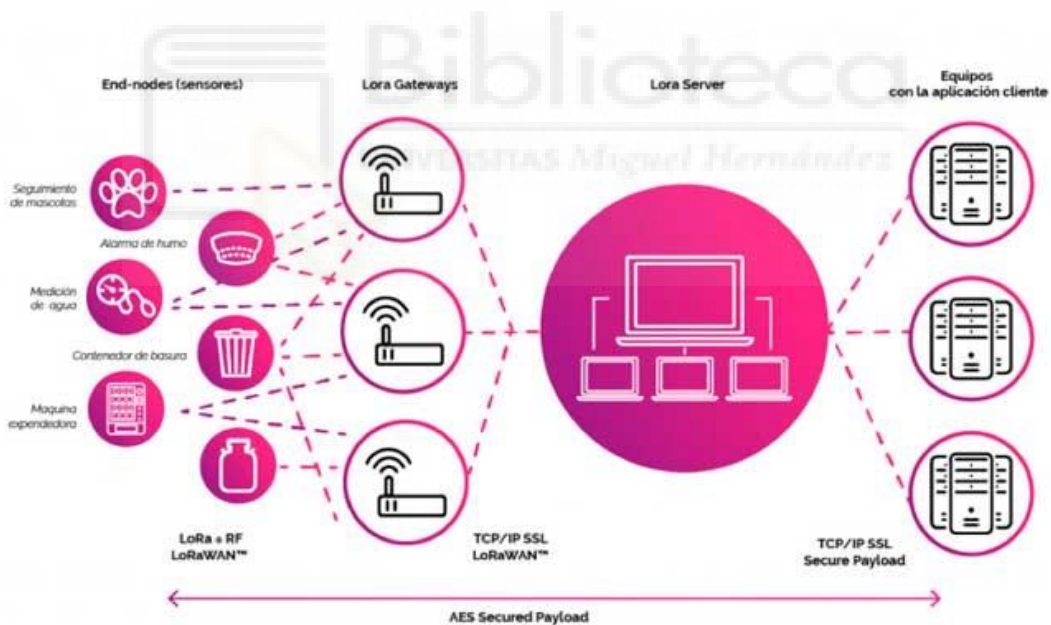


Figura 14: LoRa y LoRaWAN

Fuente: <https://www.iagua.es/blogs/carles-bianciotto/conectar-sensor-lorawan-cero>

A la hora de usar esta tecnología, también es importante definir LoRaWAN. LA diferencia que existe entre ambas tecnologías es que LoRa es la tecnología encargada de hacer que se produzca el envío de señales a larga distancia para que lleguen a las antenas. Mientras que LoRaWAN es el protocolo de red que usa la tecnología LoRa para comunicar y administrar dispositivos LoRa.

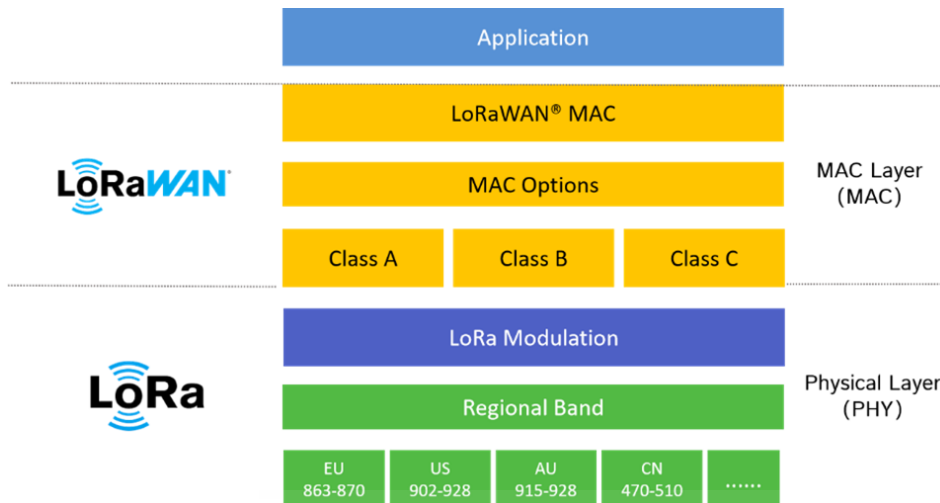


Figura 15: Estructura del LoRa

Fuente: <https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>

- NB-IoT

La tecnología NB-IoT, de las siglas en inglés *Narrow Band Internet of Things*, es una tecnología LPWAN de bajo coste que puede ofrecer una duración de sus baterías de hasta 10 años, además de una comunicación de largo alcance una penetración superior a la 2G, lo que permite usarse en dispositivos en los que otras tecnologías tendrían mayor dificultad para hacer llegar a la señal.

Las velocidades de envío de datos son relativamente bajas, alrededor de las decenas de KBps con altas latencias que podrían ser de hasta 20 segundos. Lo cual hace preferible hacer uso de UDP en vez de TCP, como se suele usar. Hace uso de un ancho de banda muy estrecho, aproximadamente de 200 KHz, y permite desplegarse en varias bandas de frecuencia. Además, al ser un estándar creado por 3GPP, se puede utilizar el espectro tanto licenciado como no licenciado. Por último, permite el despliegue masivo de IoT ya que puede proporcionar hasta 10.000 conexiones.

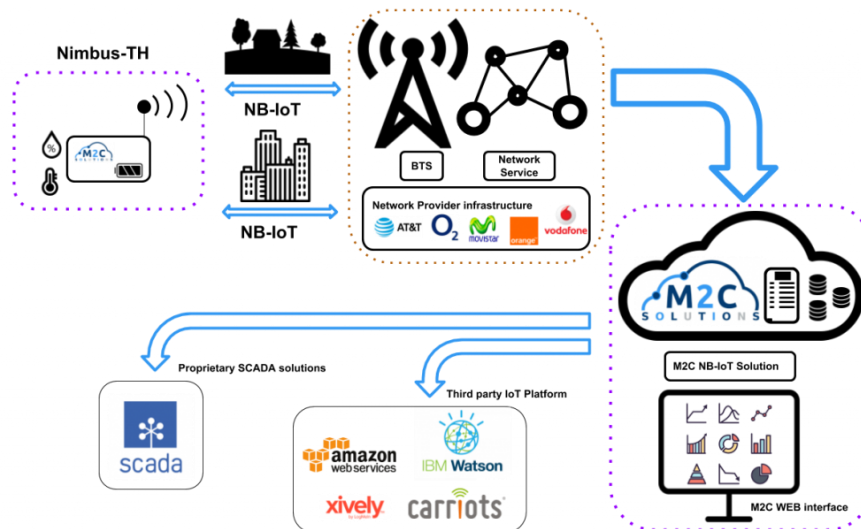


Figura 16: NB-IoT

Fuente: <https://www.m2csolutions.com/blog/autonomia-de-los-dispositivos-iot-temperatura-y-humedad-sobre-nbiot/>

- LTE-M

Por último, describimos esta tecnología, de sus siglas en inglés *Long Term Evolution for Machines*. Esta tecnología ha sido estandarizada por 3GPP, también, y la idea es que coexista con las ya existentes redes móviles 2G, 3G y 4G. Además, se pretende que también se beneficie de todas las características relacionadas con la seguridad y la privacidad de las redes móviles.

La tecnología hace uso de un ancho de banda de alrededor de 1,4 MHz, lo cual es mucho mayor al resto de tecnologías que hemos mencionado. Las velocidades de envío de datos giran en torno a 1 Mbit/s, junto a una duración de 10 años en las baterías. Tienen una potencia de transmisión de 20 a 23 dBm, y permiten el envío de datos por voz, pero aumentando el coste en los dispositivos que sean compatibles con LTE-M.

Una vez descrito todas las tecnologías de interés, las más favorables a poder llegar a usarse son tanto Sigfox como LoRa, ya que las otras dos restantes hacen uso de bandas de frecuencia con licencia y el coste total aumentaría. Aunque tampoco son malas opciones si no importa o no se tiene en cuenta el coste adicional que conllevaría usar esas tecnologías.

Tecnología	Velocidad de datos	Ancho de banda	Banda de frecuencia	Alcance
Sigfox	10 – 1000 bps	< 1 KHz	Banda ISM sin licencia 868 MHz EU 902 – 923 MHz resto	15 – 30 km → Zonas rurales 1 – 5 km → Zonas urbanas
LoRa	0'3 – 50 Kbps	125 – 500 KHz	Banda ISM sin licencia 433 – 868 MHz EU 915 MHz USA 433 MHz Asia	20 km → Zonas rurales 3 km → Zonas urbanas
NB-IoT	< 100 Kbps	180 KHz	Banda con licencia 3GPP	5 km → Zonas rurales 1 km → Zonas urbanas
LTE-M	Hasta 1 Mbps	1'4 MHz	Banda con licencia 3GPP	5 km → Zonas rurales 1 km → Zonas urbanas

Tabla 2: Características de las diferentes comunicaciones inalámbricas

Fuente: Elaboración propia

Bases de datos

La base de datos será el destino final a dónde irán los datos que enviemos desde los sensores implantados en los contenedores haciendo uso de los módulos de comunicación que hayamos decidido usar para la transmisión. La elección de la base de datos dependerá de los intereses de aquellos que adopten el proyecto, así que, a continuación, se mostrará una tabla comparativa con posibles bases de datos que podrían ser de interés:

Base de datos	Prestaciones	Limitaciones
MariaDB	<p>API disponible en numerosos lenguajes de programación</p> <p>Tiene licencia GPL, LGPL o BSD.</p> <p>Su administración se basa en usuarios y privilegios.</p> <p>Base de datos NoSQL.</p> <p>Ofrece numerosos comandos para eliminar características que afectan al rendimiento.</p>	<p>No es posible restaurar un servidor eliminado</p>
MySQL	<p>API disponible en numerosos lenguajes de programación como C, C++ y Java, entre otros.</p> <p>Destacable su velocidad de respuesta.</p> <p>Su administración se basa en usuarios y privilegios.</p> <p>Altamente estable.</p>	<p>No incluye características de objetos como tipos de datos estructurados definidos por el usuario.</p>
PostgreSQL	<p>API disponible en numerosos lenguajes de programación como C, C++ y Java, entre otros.</p> <p>Su administración se basa en usuarios y privilegios.</p> <p>Altamente estable.</p> <p>Control de concurrencia multi-versión.</p> <p>Implementación de algunas extensiones de orientación a objetos.</p>	<p>Las transacciones abortan si encuentran un fallo durante su ejecución.</p> <p>No soporta <i>tablespaces</i> para definir dónde almacenar la base de datos.</p>

Tabla 3: Características de las diferentes bases de datos

Fuente: Elaboración propia



2. MATERIAL Y MÉTODOS

2.1. INTRODUCCIÓN

Se ha estado hablando de qué posibles soluciones se pueden implantar a la hora de realizar mejoras a la gestión de residuos. Entre todas esas soluciones, hemos puesto especial interés en el uso de la tecnología para la implantación de sensores capaces de enviar información a una central encargada de recogerlos. Pero nos falta una parte por explicar, y es saber qué vamos a hacer con esa información.

La idea final de la solución que se ha presentado en esta parte del proyecto es mejorar la eficiencia con la que los camiones de residuos hacen sus rutas. Para ello, la central encargada deberá analizar los datos obtenidos de los contenedores y decidir cuáles son los contenedores de interés para recoger en ese momento y cuáles no, haciendo un descarte y evitando ir a contenedores que en ese momento no es necesario recoger.

Para ello, he creado un programa en Python que es capaz de calcular cuál es la ruta óptima para recorrer todos los contenedores y volver al punto de origen que es desde donde sale el camión de residuos a hacer el trabajo. Para ello, nos apoyaremos en un problema de optimización muy usado en Investigación Operativa y que nos ayudará a realizar los cálculos necesarios para obtener dicha ruta: el TSP. En este punto nos dedicaremos a introducir el TSP, además de la formulación que deberemos hacer para poder calcular la solución a la que aspiramos obtener.

Además, también hablaremos de una librería que ha sido muy importante a la hora de desarrollar el programa en Python y que usa el TSP para resolver los ejemplos que hemos resuelto en este proyecto. Y, para finalizar, explicaremos el algoritmo que se ha seguido en el programa para tener una idea aproximada de cuál es el funcionamiento del programa a la hora de obtener las soluciones que nos da y cómo las obtiene.

2.2. TRAVELLING SALESMAN PROBLEM

El problema del viajante de comercio, o más bien conocido como TSP debido a sus siglas en inglés *Travelling Salesman Problem*, es un problema de optimización combinatoria muy importante dentro de la investigación operativa y de la ciencia de la computación, la cual intenta responder a la siguiente pregunta: “Dada una lista de ciudades, y las distancias que existen entre cada par de ellas, ¿cuál es la ruta más corta que visita cada ciudad, exactamente, una vez y vuelve a la ciudad de partida?”.

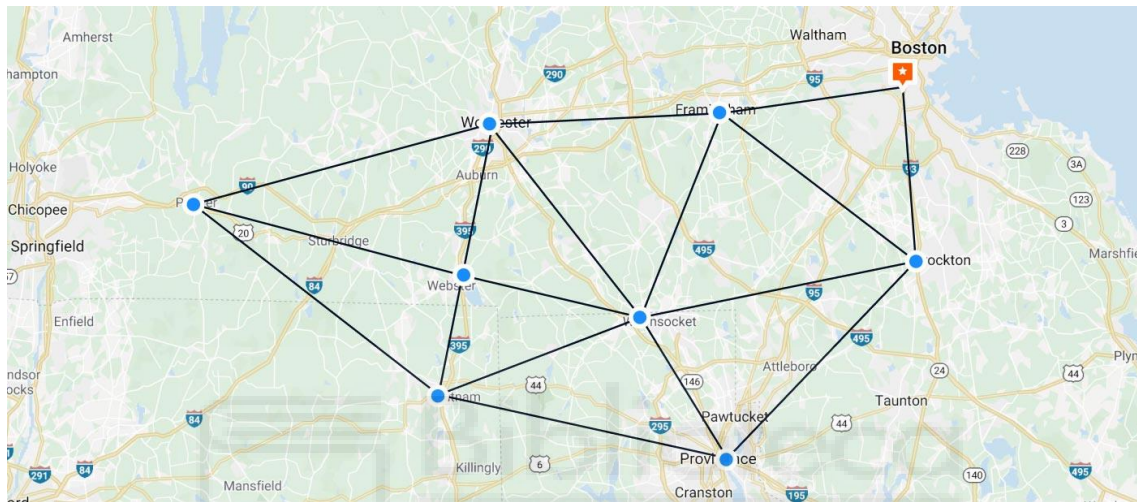


Figura 17: TSP

Fuente: <https://optimoroute.com/travelling-salesman-problem/>

Si consideramos que el conjunto de ciudades es similar a una red de nodos, donde la distancia que existe entre cada par de ellas podría ser representada como c_{ij} , el problema consiste en tratar de averiguar cuál sería el orden que debería seguirse para poder visitar todos los nodos de la red, minimizando la distancia total recorrida.

2.3. FORMULACIÓN DEL TSP

Nuestra idea original, como ya hemos comentado, es ser capaces de obtener la ruta óptima a un número de contenedores de basura, según los datos que nos den los sensores. Para ello, se hace la extrapolación del problema que tenemos al TSP, para así poder hacer una similitud entre los problemas y poder aplicar un método matemático para su resolución. Es por ello por lo que introduciremos brevemente una introducción a la teoría de grafos. Posteriormente, se explicará y desarrollará la formulación matemática que se deberá seguir para plantear un problema relacionado con el TSP y, finalmente, se explicará formas de destruir posibles bucles que puedan aparecer en las soluciones.

Un grafo es un par $G = (V, A)$, donde V es un conjunto de elementos ($V = \{1, 2, \dots, n\}$), que se denominan vértices o nodos, y A es un conjunto de pares de nodos denominamos arcos o aristas, los cuales representan una unión entre dos nodos.



Figura 18: Ejemplo TSP en España

Fuente: <https://www.mundodeportivo.com/urbantecno/android/grafos>

Durante los anteriores apartados, hemos comentado que el TSP puede ser descrito con los siguientes dos aspectos: las ciudades y las distancias entre cada par de ellas. Por lo que, dándole un aspecto más matemático, podríamos definir lo siguiente: Las ciudades podríamos denotarlas por los, anteriormente definidos, vértices, donde cada vértice estaría representado por $i = 2, \dots, n$. Y donde el vértice 1 representaría el vértice origen y destino del TSP. En cambio, las distancias entre cada par de ciudades podrían estar denotadas por los arcos (i, j) . Estos arcos estarían representados por un coeficiente c_{ij} , que sería un valor no negativo, o valor entero positivo, y representaría el coste que tendría de pasar de un vértice i a otro vértice j .

Para estudiar un grafo, ya no sólo es necesario una representación gráfica del mismo, si no, también, una representación analítica con la que podamos trabajar. Es por ello, que los grafos pueden ser representados por matrices. Estas matrices se dividen en dos grupos:

- **Matriz de incidencia.** Las entradas de la matriz indican si un nodo forma parte, o no, de un arco o arista.
- **Matriz de adyacencia.** Las entradas de la matriz indican si existe un arco o arista entre dos nodos.

Un buen ejemplo de lo que hemos explicado, podría ser el siguiente, en el que se representa un grafo no dirigido junto a la matriz de incidencia que lo define:

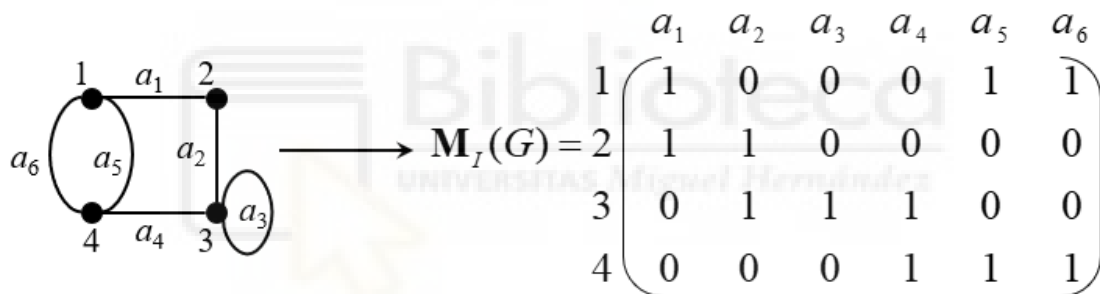


Figura 19: Matriz de incidencia del grafo

Fuente: Apuntes de la asignatura Ingeniería del Tráfico de la UMH

Explicado todo esto, existen diferentes tipos de TSP que varían en función de la matriz de costes, o distancias, a la que está asociado el problema y esto podemos dividirlo en los siguientes apartados:

- **Simétrico.** La matriz será simétrica cuando el grafo G sea un grafo no dirigido, es decir, la distancia entre cada par de ciudades es exactamente la misma, da igual el sentido que tome. Cogiendo un ejemplo, si, desde el vértice 1 hasta el 2, el coste es de 3; en el sentido contrario, el valor seguiría siendo el mismo en este caso. Por lo que, como hemos comentado antes, la matriz de costes será simétrica y el problema se pasará a denominar STSP.

- **Asimétrico.** La matriz será asimétrica cuando el grafo G sea un grafo dirigido, es decir, la distancia puede ser diferente en la ida como en la vuelta. Aunque también existe la posibilidad de que no exista ninguna de ellas. Esto haría, como hemos comentado antes, que la matriz sea asimétrica y, por lo tanto, el problema pase a ser ATSP.

Cada ciudad va a estar representada por un vértice i , enumerando las ciudades desde 1 hasta n , que será la última ciudad por visitar; mientras que la distancia entre cada par de ellas estará representada por el peso de la arista, c_{ij} . Además, consideraremos que x_{ij} será la representación de la arista que enlaza del vértice i al j , de tal forma que, si la arista está en la solución, adoptará el valor de 1, mientras que, si no lo está, tendrá el valor de 0. Dicho esto, definimos la función objetivo, que quedaría de la siguiente manera:

$$\text{minimizar } \sum_{i,j=0}^n c_{ij}x_{ij}$$

Figura 20: Función objetivo

Fuente: <https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

La función objetivo representa el sumatorio de las distancias entre cada par de nodos, siendo el objetivo minimizar el valor que se obtiene en la función. Para ello, nos apoyaremos en las siguientes restricciones:

$$\begin{aligned} (1) \quad & \sum_{i=0}^n x_{ij} = 1 & j = 0, \dots, n \\ (2) \quad & \sum_{j=0}^n x_{ij} = 1 & i = 0, \dots, n \end{aligned}$$

Figura 21: Restricciones

Fuente: <https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

En la definición del TSP que hemos introducido anteriormente, hemos realizado la pregunta de encontrar la ruta óptima pasando solamente una vez por cada ciudad y volviendo al origen. Las restricciones, que acabamos de mencionar, representan esto que acabamos de decir. La primera restricción indica que en cada nodo sólo puede haber una arista que llegue al nodo. Y en la segunda restricción, se indica que, de cada nodo, puede salir, solamente, una arista.

Estas restricciones son sumamente necesarias para obtener la solución del problema. Pero nos surge un problema muy importante a la hora de obtener las soluciones, y son los llamados subtours o subcircuitos. Un ejemplo es el expuesto en la siguiente figura:

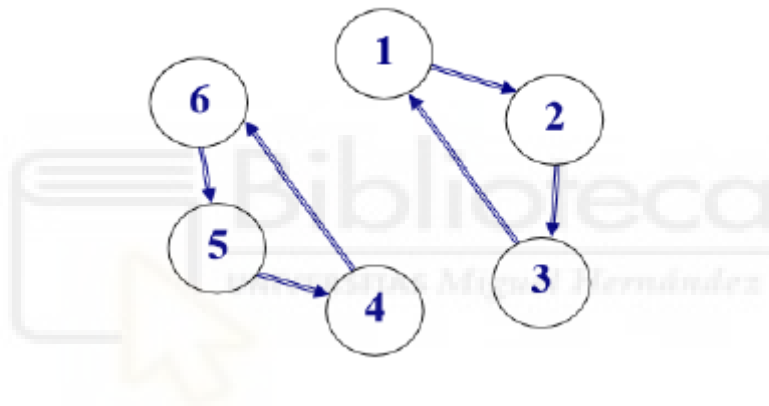


Figura 22: Bucles / Subtours

Fuente: <https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

Podemos observar que, en el ejemplo, la solución que se ha obtenido no responde correctamente a la pregunta que lanza el TSP, ya que no existe una ruta que esté conectada y pase por todos los nodos del problema. Por lo que no se le considera una solución válida para el TSP. Este tipo de bucles que se forman, cuando obtienes una solución, se llaman subtours, o ciclos.

Existen actualmente dos formas para poder deshacerse de estos ciclos que se forman. La primera que vamos a explicar es principalmente usada en los problemas de TSP para eliminar ciclos que se formen durante las resoluciones del problema.

$$(3) \quad \sum_{i,j}^n x_{ij} \leq |K| - 1 \quad 1 \leq i \neq j \leq n, K \subsetneq N$$

$$0 \leq x_{ij} \leq 1, \quad x_{ij} \in Z, \forall i, j$$

Figura 23: Primera manera de romper un bucle

Fuente: <https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

La $|K|$ representa el número total de nodos que influyen en el ciclo. Si nos damos cuenta, en los dos bucles, el número de aristas total siempre es igual al número de nodos total que hay en el ciclo. Por lo que, lo que hacemos es restarle 1 al número de aristas para que, de esa forma, no se vuelva a dar ese bucle nunca. Cogiendo el ejemplo anterior, la restricción quedaría así:

$$x_{46} + x_{65} + x_{54} \leq 2$$

La segunda forma de deshacernos del bucle es una formulación alternativa, planteada por Miller, Tucker y Zemlin, en la que se usan nuevas variables de decisión u_i con $i = 1, \dots, n$, llamadas condiciones de Miller-Trucker.

$$(3) \quad u_i - u_j + nx_{ij} \leq n - 1 \quad 1 \leq i \neq j \leq n$$

$$0 \leq x_{ij} \leq 1, \quad x_{ij} \in Z, \forall i, j$$

Figura 24: Segunda manera de romper un bucle

Fuente: <https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

Para entender cómo funciona esta forma alternativa, véase el trabajo realizado por Francisco José Veiga Losada, en *El problema del viajante* (2013). La diferencia entre estas dos formas de romper los ciclos es la siguiente: en la primera forma, se generan muchas restricciones, pero el número de variables no varía. En cambio, la segunda forma hace uso de más variables, pero, por el contrario, tiene menos restricciones que en la anterior formulación. Por lo que el uso de ambas formas es totalmente válido, pero su uso dependerá en función de lo que nos interese.

2.4. LIBRERÍA *LPSOLVE*

A la hora de hacer el programa, se ha hecho uso de una librería que ha tomado mucho protagonismo, ya que, prácticamente, todo el programa gira en torno a ella, y esa es la librería *Lpsolve*, la cual nos vamos a encargar de explicar en qué consiste la librería y porqué es tan importante.

Es una librería, disponible en Python, la cual nos permite resolver problemas de programación lineal y entera, como lo es el TSP. Para ello, la librería se apoya en dos métodos muy conocidos que son el simplex revisado, que, a diferencia del simplex, el simplex revisado está especialmente hecho para los ordenadores; y el método de ramificación y acotación.

Al ser esta librería la encargada de encontrarnos las soluciones que estamos buscando a la eficiencia de las rutas de los camiones, se ha hecho todo el programa entorno a la inserción de los datos en la librería y a la traducción de la solución que nos ofrece la librería para poder obtener la solución que estamos buscando.

2.5. ALGORITMO

Después de haber explicado la importancia que toma la librería que se ha mencionado antes. Vamos, ahora, a explicar en qué consiste el algoritmo que se ha desarrollado para este programa. Para ello, nos apoyaremos en un diagrama de flujo que resume el funcionamiento del programa:



Figura 25: Algoritmo del programa

Fuente: Elaboración propia

Lo primero de todo que se hace en el programa, es crear un bloc de notas, hecha de manera manual por el usuario, en el que se introduce la formulación oportuna,

apoyándonos en el TSP, para describir el problema al que nos enfrentamos, y que la librería sea capaz de leer todos esos datos.

Posteriormente, el programa entrará en el bloc de notas, y leerá y dividirá la información que se ha introducido en el bloc de notas de tal manera que la librería sea capaz de entender esos datos, ya que será la encargada de realizar los cálculos oportunos para obtener la solución al problema que le hemos introducido.

Una vez calculado el problema, como ya comentamos en el capítulo 3 de este proyecto, existe la posibilidad de que se dé el caso de que haya bucles en la solución que hemos obtenido. Esto es no es algo que queramos obtener en la solución del programa, por lo que, en el caso de que exista un bucle, usaremos el primer método mencionado para romper los bucles. Es por ello que se identificará el bucle que se ha formado, se introducirá la restricción oportuna que hará que ese bucle no vuelva a aparecer, y se volverá a realizar los pasos previos para que la librería vuelva a calcularnos la solución.

Este proceso se repetirá las veces que sea necesario hasta que el programa detecte que ya no hay más bucles en la solución. Será en ese entonces cuando se finalice el programa traduciendo la solución que se ha obtenido y enseñando gráficamente cuál sería la ruta óptima que se ha obtenido por el programa.



3. RESULTADOS Y DISCUSIÓN

Ya hemos explicado todos los métodos que se usan en el programa para poder obtener la ruta óptima. Por lo que vamos a introducir, ahora, un caso real relacionado con una ruta que se sigue en Santa Pola, aplicando una suposición en la cual los sensores nos dicen que debemos recoger una serie de contenedores específicos de la ruta.

Urbaser es una empresa de residuos que tiene encargado recoger todos los residuos que hay en Santa Pola. Es por ello que tienen una serie de rutas fijas creadas que siguen prácticamente todos los días del año. He logrado obtener una de las rutas que siguen los trabajadores de esta empresa en la que pasan por más de 20 contenedores durante la ruta. Para ofrecer un ejemplo real de cómo sería una recogida inteligente de contenedores, supondremos que, los sensores que hemos implementado nos han enviado la información sobre los contenedores que se deben recoger hoy. Dando lugar a la recogida de los 9 contenedores representados en la siguiente figura con sus localizaciones aproximadas:

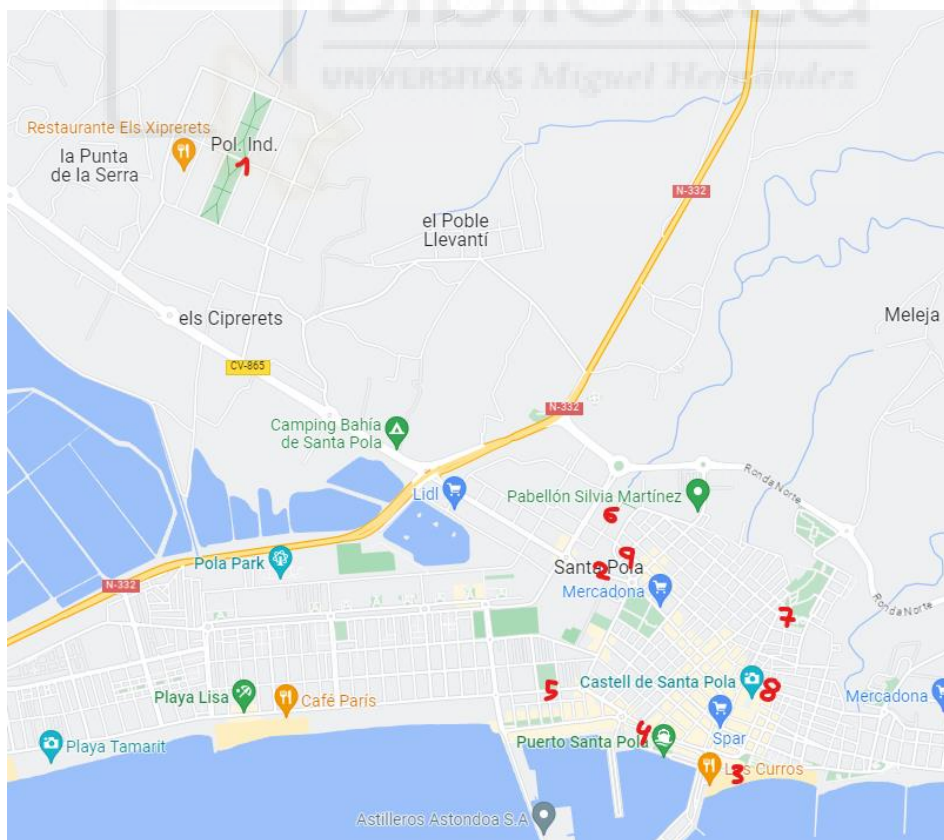


Figura 26: Contenedores elegidos para el ejemplo

Fuente: Google Maps

Los números marcados en el mapa serán aquellas localizaciones por las que deberá pasar el camión de basura para recoger los residuos. El lugar desde el que se parte sería desde el Polígono Industrial. Desde ahí, se tendría que pasar por todos los contenedores de basura y se volvería allí para dejar el camión. Por lo que, una vez introducido los datos necesarios para poder averiguar la ruta óptima dentro del programa, la solución que ha salido ha sido la siguiente:

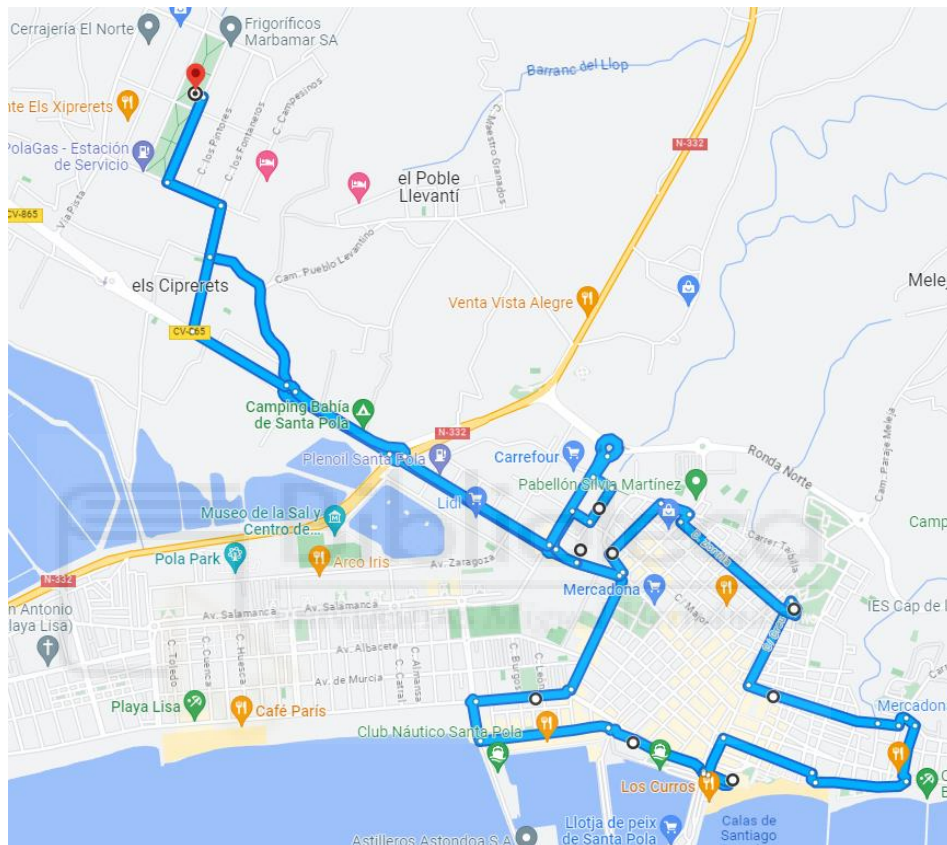


Figura 27: Solución del ejemplo

Fuente: Google Maps

Los nodos del ejemplo están especificados en el Anexo 1. Por lo que, viendo el mapa que representa la solución que ha obtenido el programa, esta sería la ruta que seguiría el camión de basura para esos respectivos contenedores: Polígono industrial – Farmacia Bonmati – Restaurante Batiste – Ayuntamiento – Carrer Dean Llopez, 10 – Calle Sant Paqual, 33 – Calle Azorín, 2 – Ctra. Elche-Santa Pola – Calle Lina Baño – Polígono Industrial.



4. CONCLUSIONES

Todo el trabajo realizado en este proyecto se puede resumir en los siguientes puntos descritos a continuación:

- Se ha desarrollado un programa que se encarga de encontrar la ruta más corta partiendo desde un punto inicial y volviendo al mismo.
- Se ha descrito el concepto de *Smart City*, aportando los beneficios que aporta, junto a sus características, que sirven para diferenciar las *Smart Cities* de las ciudades convencionales. Además de la aportación de ejemplos de *Smart Cities* existentes en la actualidad.
- Se ha propuesto diversas soluciones a la gestión de residuos en las que se incluyen los contenedores soterrados, la recogida neumática y la aplicación de la tecnología para la mejora en la eficiencia de la recogida de residuos.
- Se ha profundizado en la aplicación de la tecnología, proponiendo el uso de diversos sensores, diferentes comunicaciones inalámbricas y bases de datos para hacer más eficiente y sostenible la recogida de basuras
- Se ha introducido el TSP, explicando su utilidad dentro del proyecto y mostrando la formulación requerida para poder usarlo en el programa desarrollado.
- Se ha introducido el algoritmo que se ha usado para crear el programa, junto a detalles importantes sobre la librería que hace uso del TSP para poder hacer que funcione el programa.
- Se hace uso de un ejemplo real en el que se calcula la ruta óptima en un caso en el que los sensores nos indican que hay 9 contenedores para recoger.

En cuanto al programa que se ha creado, podría utilizarse como librería en un sistema experto que calculase de manera automática, la ruta a seguir en función de los datos proporcionados por los sensores y contenedores inteligentes que indiquen que han de ser recogidos.



5. ANEXOS

ANEXO 1: MATRIZ NO SIMÉTRICA DEL EJEMPLO REAL (TIEMPO EN MINUTOS)

Nodos	1	2	3	4	5	6	7	8	9
1	-	5	9	7	7	6	9	10	7
2	5	-	4	3	2	1	4	6	2
3	8	4	-	1	3	5	5	6	4
4	7	2	1	-	1	3	4	6	3
5	7	3	4	3	-	3	6	8	4
6	5	2	6	5	4	-	6	6	3
7	8	4	7	5	5	5	-	6	3
8	9	5	8	6	6	6	2	-	4
9	5	1	4	3	2	2	4	7	-

Tabla 4: Matriz no simétrica

Fuente: Elaboración propia

Polígono industrial – 1

Ctra. Elche-Santa Pola – 2

Ayuntamiento Santa Pola – 3

Restaurante Batiste – 4

Farmacia Bonmati – 5

Carrer Lina Baño – 6

Calle Sant Pasqual, 33 – 7

Carrer Dean Llopez, 10 – 8

Calle Azorín, 2 – 9

ANEXO 2: PROGRAMA EN PYTHON

```
from lpsolve55 import *
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import time

inicio = time.time()

#-----
-----

def grafo_no_dirigido(todos_nodos):
    nodos = []
    ejes_nodos = []
    contenedor = ''

    for x in todos_nodos:
        try:
            err1 = nodos.index(x[0])

        except:
            err1 = -1

        if err1 == -1:
            nodos.append(x[0])

        contenedor = x[1] + x[0]

        try:
            err1 = ejes_nodos.index(contenedor)

        except:
            err1 = -1

        if err1 == -1:
            ejes_nodos.append(x)

    G = nx.Graph()
    G.add_nodes_from(nodos)

    for x in ejes_nodos:
        G.add_edge(x[0], x[1])

    fig, ax = plt.subplots(figsize=(10, 10))
    nx.draw(G, pos = nx.circular_layout(G), with_labels=True, ax=ax)

#-----
-----

def grafo_dirigido(ruta_mas_corta):
    nodos = []

    for x in ruta_mas_corta:
```

```

    try:
        err1 = nodos.index(x[0])

    except:
        err1 = -1

    if err1 == -1:
        nodos.append(x[0])

G = nx.DiGraph()
G.add_nodes_from(nodos)

for x in ruta_mas_corta:
    G.add_edge(x[0], x[1])

fig, ax = plt.subplots(figsize=(10, 10))
nx.draw(G, pos = nx.circular_layout(G), with_labels=True, ax=ax)

#-----
-----

#Esta definición se encarga de ordenar la ruta definitiva para que
tenga sentido
def ordenar_ruta(ruta_obtenida):
    ruta_ordenada = []
    ordenado = 'no'
    #Funciona de la misma manera que los sub-ciclos, pero ordenando
    contenedor = ruta_obtenida[0]
    ruta_ordenada.append(contenedor)

    while True:
        for x in ruta_obtenida:
            if contenedor[1] == x[0]:
                contenedor = x

                try:
                    err1 = ruta_ordenada.index(x)

                except:
                    err1 = -1

                if err1 == -1:
                    ruta_ordenada.append(x)

                if len(ruta_ordenada) == len(ruta_obtenida):
                    ordenado = 'si'
                    break

                else:
                    break

        if ordenado == 'si':
            break

    return ruta_ordenada

```

```

#-----
#-----

#Esta definición se usa para crear un array de ceros en función del
número de nodos que haya
def array_ceros(nodos):
    lista = []
    i = 0

    while i < len(nodos):
        lista.append(int(0))
        i += 1

    array = np.array(lista)

    return array

#-----
#-----

#Esta definición simplemente nos ayuda a identificar los nodos que
forman la "supuesta" ruta
def ruta(nodos_usados, solucion):
    ruta_mas_corta = []
    i = 0

    while i < len(solucion):
        if solucion[i] == 1.0:
            ruta_mas_corta.append(nodos_usados[i])

        i += 1

    return ruta_mas_corta

#-----
#-----

def encontrar_sub_ciclos(ruta, ruta_trabajo):
    bucle = []
    contenedor = ruta[0]
    contenedor2 = ruta[0]
    contenedor3 = "00"
    contenedor4 = "00"
    ciclo = ''
    i = 0

    bucle.append(contenedor) #Guardamos los valores para saber si se
ha formado un ciclo o no

    while True: #Entramos en este bucle hasta que se cumplan las
condiciones para salir de él
        for x in ruta:
            if contenedor[1] == x[0]: #Sirve para ir guardando la
unión de los nodos
                contenedor = x

```

```

    try:
        err1 = bucle.index(x) #Buscamos esos nodos en el
bucle

    except:
        err1 = -1

    if err1 == -1: #En caso de no estar, lo añadimos
        bucle.append(x)

    #Si se da el caso de que se llega al nodo origen sin
haber pasado por todos los nodos, es un ciclo
    if len(bucle) != len(ruta) and contenedor2[0] == x[1]:
        for x in bucle:
            if x[0] == "1":
                contenedor3 = x

            if x[1] == "1":
                contenedor4 = x

    if contenedor3[0] == "1" and contenedor4[1] ==
"1":

        while True:
            if len(bucle) == 0:
                break

            bucle.pop(0)

            i += 1
            contenedor = ruta[i]
            contenedor2 = ruta[i]
            bucle.append(contenedor)

    if contenedor3[0] != "1" and contenedor4[1] !=
"1":

        ciclo = 'si'
        break

    else:
        contenedor3 = "00"
        contenedor4 = "00"

    elif len(bucle) == len(ruta):
        ciclo = 'ruta encontrada'
        break

    else:
        break

    #Si hemos encontrado un ciclo, escribimos ese ciclo en el bloc
de notas para volver a calcularlo
    if ciclo == 'si':
        fichero_escribir = open(ruta_trabajo, "a")
        fichero_escribir.write("\n")

z = 0

```

```

        while z < len(bucle):
            if z == (len(bucle) - 1):
                fichero_escribir.write("x" + str(bucle[z]) + " <="
" + str(len(bucle) - 1))

            else:
                fichero_escribir.write("x" + str(bucle[z]) + " +
")

                z += 1

        fichero_escribir.close()

        break

        #Si hemos encontrado la ruta, simplemente salimos del bucle
        elif ciclo == 'ruta encontrada':
            break

        else:
            continue

    return ciclo

#-----
-----

def funcion_objetivo(lista_funcion):
    fn_obj = []
    nodos = []

    split_funcion = lista_funcion.split() #Realizamos el split para
cada espacio que se encuentre

    for x in split_funcion:
        i = 0
        funcion_obj = ''

        while True:
            if x[i] == 'x':
                #Si el valor encontrado es una x y es la primera
                #posición, entonces el valor es 1 y añadimos el nodo correspondiente
                if i == 0:
                    fn_obj.append(int(1))
                    nodos.append(x[i+1] + x[i+2])
                    break

                #Si el valor encontrado es una x y es una posición
                #diferente, entonces el valor es el guardado y añadimos el nodo
                #correspondiente
            else:
                fn_obj.append(int(funcion_obj))
                nodos.append(x[i+1] + x[i+2])
                break

        #En caso de encontrar una suma, no nos interesa guardar el
valor

```

```

        elif x[i] == '+':
            break

        #En caso de que se encuentre algo diferente a lo anterior
        (que deberá ser un n°), se guardará el valor y se añadirá un valor de
        1 a la i
    else:
        funcion_obj += x[i]
        i += 1

    return fn_obj, nodos #Devolvemos el valor

#-----
-----

def restricciones(lista, nodos):
    i = 0
    igual = 0
    restriccion_derecha = 0

    split_lista = lista.split()
    array = array_ceros(nodos) #Creamos un array de ceros

    for x in split_lista:
        restriccion = ''

        if x[i] == 'x': #En caso de encontrar una x, guardamos la
        posición en la que se encuentra el nodo y le añadimos un 1 al array de
        ceros.
            restriccion += x[i+1] + x[i+2]
            index = nodos.index(restriccion)
            array[index] = int(1)

        elif x[i] == '=' or x[i] == '<': #En caso de encontrar un
        igual o un <, cambiamos el valor de igual
            igual = 1

        elif igual == 1: #Cuando se cambie el valor de igual,
        guardamos el valor de esa iteración
            restriccion_derecha = int(x[i])

        else: #Cualquier otra cosa, se ignora y se continua con el
        bucle
            continue

    return array, restriccion_derecha

#-----
-----

contador = 0

def problema_del_viajero(lista_fichero):
    fn_obj = []
    nodos = []
    ruta_mas_corta = []
    i = 1

```

```

global contador

del fn_obj #Eliminamos posibles elementos guardados anteriormente
del nodos #Eliminamos posibles elementos guardados anteriormente
del ruta_mas_corta #Eliminamos posibles elementos guardados
anteriormente

#Conseguimos los valores de la función objetivo junto a sus
respectivos nodos
fn_obj, nodos = funcion_objetivo(lista_fichero[0])

if contador == 0: #Se encarga de enseñar el grafo creado mediante
un plot
    grafo_no_dirigido(nodos)
    contador += 1

#Inicializamos la librería
lp = lpsolve('make_lp', 0, len(fn_obj))
lpsolve('set_verbos', lp, 'IMPORTANT')
lpsolve('set_obj_fn', lp, fn_obj)

while i < len(lista_fichero): #Añadimos todas las restricciones al
programa
    array, restriccion_derecha = restricciones(lista_fichero[i],
nodos) #Conseguimos un array para añadir una restricción a la librería

    if restriccion_derecha == 1: #En caso de que el valor sea 1,
quiere decir que la restricción tiene un igual
        lpsolve('add_constraint', lp, array, 'EQ', 1)
    else: #En caso de que no tenga el valor de 1, querrá decir que
la restricción es un ciclo.
        lpsolve('add_constraint', lp, array, 'LE',
restriccion_derecha)

    i += 1

i = 1

while i <= len(fn_obj): #Establecemos todas las variables en
binario para que solo acepten valores 0 y 1
    lpsolve('set_binary', lp, i, True) #Establecemos que todas las
variables de nuestro lp son binarias
    i += 1

#Resolvemos el problema de optimización y obtenemos una ruta
solucion = lpsolve('solve', lp)

obj = lpsolve('get_objective', lp)

[variables, x] = lpsolve('get_variables', lp)

ruta_mas_corta = ruta(nodos, variables)

lpsolve('delete_lp', lp)

return ruta_mas_corta

```



```

#-----
-----

while True:
    #Abrimos y leemos el fichero
    ruta_trabajo = "/Users/mrbot/Desktop/Prácticas posible
TFG/Trabajos/4º trabajo/Optimización.txt"
    fichero_abierto = open(ruta_trabajo, 'r')
    fichero_leido = fichero_abierto.read()

    #Aplicamos el .split() al fichero para separarlo por cada salto de
línea
    lista_fichero = fichero_leido.split('\n')

    #Aplicamos un bucle para ir encontrando ciclos en el grafo hasta
encontrar la ruta
    ruta_obtenida = problema_del_viajero(lista_fichero) #Obtenemos una
ruta con el lpsolve

    ciclo = encontrar_sub_ciclos(ruta_obtenida, ruta_trabajo)
    #Comprobamos si hay ciclos

    if ciclo == 'ruta encontrada': #En caso de haber encontrado la
ruta, la ordenamos y la mostramos
        grafo_dirigido(ruta_obtenida)
        fichero_abierto.close()
        break
    else:
        fichero_abierto.close()
        continue

fin = time.time()
print(fin-inicio)

```



6. BIBLIOGRAFÍA

[1] Fernando de Querol Cumbreira, “Contaminación del aire urbano: riesgos y alternativas”, Instituto Español de Estudios Estratégicos.

https://www.ieee.es/Galerias/fichero/docs_opinion/2018/DIEEEO63-2018_Contaminacion_AireUrbano_QuerolCumbreira.pdf

[2] Autor desconocido, “Smart Cities: the digital transformation of the cities”, Iberdrola.

<https://www.iberdrola.com/innovation/smart-cities>

[3] Autor desconocido, “Why do we need Smart Cities?”, Telefónica.

<https://www.telefonica.com/en/communication-room/blog/why-do-we-need-smart-cities/>

[4] Mark Warner, Stacey Mansker-Young, “Thinking about becoming a Smart City? 10 benefits of Smart Cities”, Plante Moran.

<https://www.plantemoran.com/explore-our-thinking/insight/2018/04/thinking-about-becoming-a-smart-city-10-benefits-of-smart-cities>

[5] William Goddard, “What are the features of a Smart City to look for?”, Chronicles.

<https://itchronicles.com/smart-city/what-are-the-features-of-a-smart-city-to-look-for/>

[6] Autor desconocido, “What are the features of a smart city?”, NEC.

<https://www.nec.co.nz/market-leadership/publications-media/what-are-the-features-of-a-smart-city/>

[7] Autor desconocido, “¿Qué son los contenedores soterrados de basura?”, Ecoembes.

<https://ecoembesdudasreciclaje.es/contenedores-soterranos/>

[8] Autor desconocido, “Sistema de recogida: Neumática”, Ministerio para la transición ecológica y el reto demográfico.

<https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/prevencion-y-gestion-residuos/flujos/domesticos/gestion/sistema-recogida/neumatica.html>

[9] Autor desconocido, “A new generation, long distance ranging Time-of-Flight sensor based on ST’s FlightSense™ technology”, ST.

<https://www.pololu.com/file/0J1506/VL53L1X.pdf>

[10] Autor desconocido, “NI30U-EM30WD-AP6X-H1141/3GD Sensor inductivo – para la industria alimenticia”, Turck.

https://www.turck.de/datasheet/es/edb_1634861_esl_es.pdf

[11] Autor desconocido, “BC10-P30SR-VP4X2/3GD Sensor capacitivo”, Turck.

https://www.turck.de/datasheet/es/edb_2505006_esl_es.pdf

[12] Autor desconocido, “Capacitive sensors”, KS Series.

<https://www.waycon-sensor.it/fileadmin/capacitive-sensors/Capacitive-Sensor.pdf>

[13] Autor desconocido, “Inductive Sensor With Extended Switching Distance BI4U-M12-VP44X-H1141 L100”, Turck.

https://www.turck.de/datasheet/en/edb_1634917_gbr_en.pdf

[14] Benjimflores, “Sigfox: La red del IoT”, AG Electrónica

<https://agelectronica.blog/2019/09/18/sigfox-la-red-del-iot/>

[15] Autor desconocido, “Qué es LoRa, cómo funciona y características principales”, Venco.

<https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>

[16] Autor desconocido, “Diferencias entre NB-IOT y LTE-M”, Accent Systems.

<https://accent-systems.com/es/diferencias-nb-iot-lte-m/>

[17] Marc Gibert Ginestà, Oscar Pérez Mora, “Bases de datos en PostgreSQL”, UOC.

https://openaccess.uoc.edu/bitstream/10609/200/4/Bases%20de%20datos_Módulo6_Bases%20de%20datos%20en%20PostgreSQL.pdf

[18] Luis Alberto Casillas Santillán, Marc Gibert Ginestà, Óscar Pérez Mora, “Bases de datos en MySQL”, UOC.

<http://ual.dyndns.org/biblioteca/Bases%20de%20Datos%20Avanzado/Pdf/05%20Bases%20de%20datos%20en%20MySQL.pdf>

[19] Autor desconocido, “Features of MariaDB”, Javapoint.

<https://www.javatpoint.com/mariadb-features>

[20] Ariel, “Smart Cities: un primer paso hacia la internet de las cosas”, Fundación Telefónica

<https://www.aeiciberseguridad.es/descargas/categoria6/9704170.pdf>

[21] Álvaro Cárdenas, “Gestión inteligente de residuos urbanos en tu Smart City”, FIWOO

<https://secmotic.com/gestion-inteligente-de-residuos-urbanos-en-tu-smart-city/#gref>

[22] Anonymous Contributors, "guia_lp_solve", Apuntes de Investigación Operativa

https://knuth.uca.es/io/doku.php?id=guia_lp_solve&rev=1263471882

[23] Andrés Pérez Montilla, Juan de Dios Díaz Ramírez, Victoria Franco Otero, Pilar López Ramos, Azahara Carpintero Nieto, “El problema del viajante de comercio o TSP”, Departamento de Estadística e Investigación Operativa de la Universidad de Cádiz

<https://knuth.uca.es/moodle/course/view.php?id=72>

[24] José Salvador Cánovas Peña, “Teoría de grafos”, Dpto. de Matemática Aplicada y Estadística

<https://www.dmae.upct.es/~jose/discreta/grafos.pdf>

[25] Laura Montoya Torres, “Una aplicación del problema del viajante de comercio a la distribución del dinero en efectivo en la región de Murcia”, Facultad de ciencias de la empresa, Universidad Politécnica de Cartagena

<https://repositorio.upct.es/bitstream/handle/10317/9112/tfg-mon-apl.pdf?sequence=1>

[26] Diego García Rubiera, “Contenedores soterrados”, MOIOK

<https://www.molok.com/es/blog/contenedores-soterrados>

[27] Xavier Saura Mas, “Sistema de monitorización del estado de un contenedor de reciclaje”, Universitat Oberta de Catalunya, Máster en Ingeniería de Telecomunicación

<https://openaccess.uoc.edu/bitstream/10609/117766/8/dalonsorodrTFM0620memoria.pdf>

[28] Autor desconocido, “Tecnología LoRa y LoRaWAN”, CatSensors

<https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>

[29] Rodrigo Hernández, “¿Qué es la tecnología LoRa y por qué es importante para IoT?”, The Things Industries

<https://www.thethingsnetwork.org/community/santa-rosa/post/que-es-la-tecnologia-lora-y-por-que-es-importante-para-iot>

[30] Autor/a desconocido/a, “Qué es LoRa, cómo funciona y características principales”, VencoEl

<https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>

[31] Autor/a desconocido/a, “Narrowband – Internet of Things (NB-IoT)”, GSMA

<https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/>

[32] Matt Thomas, “Una introducción a LPWAN y NB-IoT”, Wireless Logic

<https://www.wirelesslogic.com/es/articulo/nb-iot/>

[33] Autor/a desconocido/a, “NB-IoT Vs LTE-M, El Nuevo Rey IoT En El Mundo 4G”, PickData by CIRCUTOR

<https://www.pickdata.net/es/noticias/nb-iot-vs-lte-m-nuevo-rey-iot-mundo-4g>

[34] Autor/a desconocido/a, “Tecnología Sigfox: Internet de las cosas”, dset ENERGY

<http://productos-iot.com/sigfox-3/>

[35] Profesor José Luis Sainz-Pardo Auñón, apuntes de la asignatura Ingeniería del Tráfico de la UMH