

Hacia una ciudad sostenible: el caso de un sistema de gestión del tráfico en Puerto Real

Rubén Expósito Marín

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Aplicaciones multimedia de nueva generación

Consultor: David García Solórzano

Profesor responsable de la asignatura: David García Solórzano

5 de Enero de 2022



Esta obra está sujeta a una licencia de [Atribución-
NoComercial-SinDerivadas 3.0 España de Creative
Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Hacia una ciudad sostenible: el caso de un sistema de gestión del tráfico en Puerto Real</i>
Nombre del autor:	<i>Rubén Expósito Marín</i>
Nombre del consultor/a:	<i>David García Solórzano</i>
Nombre del PRA:	<i>David García Solórzano</i>
Fecha de entrega (mm/aaaa):	<i>01/2022</i>
Titulación o programa:	<i>Grado en Ingeniería de Tecnología y Servicios de Telecomunicación</i>
Área del Trabajo Final:	<i>Aplicaciones multimedia de nueva generación</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Sistema, Arduino, tráfico</i>
Resumen del Trabajo (máximo 250 palabras):	
<p><i>En este documento se presenta un sistema a implementar en la ciudad de Puerto Real (Cádiz) para la gestión del tráfico mediante paneles informativos y apps para dispositivos móviles. El fin de dicho sistema es informar sobre la disponibilidad de plazas en los distintos aparcamientos disuasorios distribuidos por la ciudad y a la vez incentivar el uso de las zonas peatonales y la bicicleta. Se pretende paliar la posible reducción de visitantes a los comercios del centro tras su peatonalización, a la vez que se potencia la imagen de Puerto Real como población eco-friendly, se aumenta la concienciación y percepción como algo propio de la conciencia ecológica en la población y se distribuye el tráfico de un modo más armónico, dinámico e inteligente.</i></p>	

Abstract (in English, 250 words or less):

This document presents a system to be implemented in the city of Puerto Real (Cádiz). The system is founded on the idea of improving the traffic management with information panels and apps. The goal of the system is informing the users about the availability of space in dissuasive car parks spread around the surroundings of the city. The topmost idea is inviting the users to walk or ride a bike. The desired consequence is avoid the reduction in visitors to the city center shops after it has been completely pedestrianized. A side consequence could be also empowering the city as an eco-friendly place, increasing the self-perception and awareness of being an ecologic city. With this system, traffic in the city will be more dynamic, organized and intelligent.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	7
1.3 Enfoque y método seguido.....	7
1.4 Planificación del Trabajo	9
1.4.1 Alcance.....	9
1.4.2 Entregables	10
1.4.3 Calendario de Trabajo	10
1.4.4 Tareas y Diagrama de Gantt.....	11
1.4.5 Riesgos e incidencias	13
1.5 Breve resumen de productos obtenidos	13
1.6 Breve descripción de los otros capítulos de la memoria	13
2. Estado del Arte	15
2.1 Estado del arte de las soluciones del problema a nivel global	15
2.2 Estado del arte de las soluciones comerciales y tecnologías empleadas.....	17
2.3 Soluciones para reducir las emisiones de co2 en el centro urbano de puerto real	22
3. Arduino	23
3.1: El microcontrolador Arduino	23
3.1.1: Qué es Arduino	23
3.1.2: Tipos de placas Arduino.....	23
3.1.3: Software	25
3.2: Primera aproximación: La placa Arduino UNO	27
3.3: Selección final: La placa Arduino MKR GSM 1400	28
4. Sensores y Actuadores Empleados	30
4.1: Servomotor.....	30
4.2: Sensor de co2	32
4.3: Sensor de ultrasonidos.....	34
5. Módulos de Sensores	35
5.1: Hardware	35
5.1.1: Placa Arduino MKR GSM 1400	35
5.1.2: Circuito final del módulo de sensores.....	36
5.2: Software	37
5.2.1: Código de la placa.....	37
5.2.2: La plataforma Arduino IoT Cloud	40
6. Conclusiones.....	42
7. Glosario	44
8. Bibliografía.....	47

Lista de figuras

Figura 1 - Panel informativo situado en la calle y alimentado con los datos de los sensores que contabilizan los vehículos	2
Figura 2 - Sensores magnéticos instalados en el techo de un parking subterráneo.	2
Figura 3 - Mapa de aparcamientos disuasorios existentes en la ciudad de Puerto Real	4
Figura 4 - Diagrama del módulo sensor	5
Figura 5 - Interfaz de usuario de la app para dispositivos móviles	6
Figura 6 - Diagrama de bloques	8
Figura 7 - Diagrama de Gantt	12
Figura 8 - Sensor magnético de Libelium	18
Figura 9 - Mapa de sensores de SFPark	18
Figura 10 - Dispositivo Sony IMX500	19
Figura 11 - Placas de iniciación Arduino	24
Figura 12 - Arduino shields	24
Figura 13 - Placas Arduino IoT	25
Figura 14 - IDE Arduino	26
Figura 15 - Placa Arduino UNO	27
Figura 16 - Placa Arduino MKR GSM 1400	28
Figura 17 - Antena dipolo cinco bandas	29
Figura 18 - Circuito de calibrado del servomotor	30
Figura 19 - Montaje de circuito de prueba del servomotor	31
Figura 20 - Circuito CO ₂	32
Figura 21 - Montaje de circuito de calibrado del sensor de CO ₂	33
Figura 22 - Circuito detector ultrasonidos	35
Figura 23 - Circuito inicial Arduino MKR GSM 1400	36
Figura 24 - Circuito final del módulo de sensores	36
Figura 25 - Circuito final del módulo de sensores	37
Figura 26 - Variables de Arduino IoT Cloud creadas	40
Figura 27 - Widgets de Arduino IoT Cloud	41
Figura 28 - Prototipo final	41

Lista de tablas

<i>TABLA 1 – ENTREGABLES</i>	10
<i>TABLA 2 – CALENDARIO DE DEDICACIÓN</i>	11

1. INTRODUCCIÓN

Tras cursar satisfactoriamente todas las asignaturas del plan del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, se presenta este proyecto que reúne todos los conocimientos adquiridos en el transcurso del mismo.

En el presente documento se presenta el diseño y desarrollo de un sistema de automatización de la gestión del tráfico en una ciudad. La idea nuclear en torno a la que gira el proyecto y el hilo conductor de este documento es la reducción de los niveles de contaminantes en el centro urbano. Para ello, en el sistema se toman como puntos de actuación los distintos aparcamientos disuasorios que rodean el centro urbano de la ciudad. Mediante el empleo de módulos detectores en las entradas de los mismos, el sistema conoce la ocupación de cada uno y redirige el flujo de tráfico hacia los distintos aparcamientos. Los módulos también implementan un sensor y un algoritmo de detección del nivel de CO₂ que nos permite además basar ese control del flujo de tráfico en función de la contaminación de cada zona de la ciudad. Este módulo se basa en el microcontrolador Arduino, como se detallará en los siguientes apartados.

1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO

Existen bastantes **sistemas de aparcamiento** tanto para espacios cerrados (en superficie o subterráneos) como en la vía pública. También distintos tipos de tecnologías. En ese sentido, podemos diferenciar dos soluciones distintas^{[1][2]}:

1. **Sistemas que contabilizan** los coches que entran y salen a través de accesos controlados a recintos cerrados (figura 1). Calculan las plazas de aparcamiento libres en base a ello:



FIGURA 1 - PANEL INFORMATIVO SITUADO EN LA CALLE Y ALIMENTADO CON LOS DATOS DE LOS SENSORES QUE CONTABILIZAN LOS VEHÍCULOS

2. Sistemas que mediante **sensores magnéticos** (figura 2) o infrarrojos (IR) detectan la presencia de un coche aparcado en una plaza concreta:



FIGURA 2 - SENSORES MAGNÉTICOS INSTALADOS EN EL TECHO DE UN PARKING SUBTERRÁNEO.

En muchas ocasiones se emplean ambos sistemas combinados, como por ejemplo en algunos parkings que además tienen **paneles informativos** en las calles de acceso.

Todas las soluciones actuales del mercado están principalmente orientadas a su uso en centros comerciales, aparcamientos o centros de grandes smart

cities^[3]. Toda una industria se ha desarrollado para cubrir esta demanda, siendo los principales exponentes a nivel estatal **Libelium^[4]**, **Urbiotica^[5]** o **Dinycon^[6]**. Esta última empresa ha desarrollado los sistemas de aparcamiento inteligente en la ciudad de **Santander**. A nivel internacional la empresa **ParkHelp^[7]**, con amplia experiencia, fue seleccionada para desarrollar una experiencia piloto en **Málaga**.

En los últimos tiempos, y dentro de los esfuerzos para luchar contra el cambio climático, se incide en el ahorro que supone reducir el tiempo de búsqueda de plaza. Así, este mercado está centrado en el concepto smart city y en la integración con otros sistemas para conseguir una optimización económico-temporal para los usuarios. Es decir, **el enfoque de la industria actual es hacia la mejora del ahorro y la productividad**.

La propuesta del presente proyecto pretende ir un paso más allá. El proyecto se focalizará en la **reducción de los gases de efecto invernadero en los entornos urbanos**. Esto irá en línea con el cumplimiento de las directivas que pondrá en marcha la Unión Europea en la próxima década.

La solución propuesta parte del estado actual de la tecnología y se enfoca en el caso particular de la localidad de Puerto Real y en la **necesidad de reducir las emisiones de CO2 a la atmósfera**. Como precedente se tiene en cuenta que la ciudad tiene una disposición en estrella, con un centro urbano que estará cerrado al tráfico y donde se concentra la actividad económica. Solo se contemplan aparcamientos con entradas y salidas controladas.

El sistema **permitirá a los gestores del mismo decidir** qué número de plazas hay disponibles en cada aparcamiento, así como reducir a cero el número de plazas si es necesario (por ejemplo para eventos, fiestas, manifestaciones). De este modo el sistema **atiende las necesidades de la ciudad** (minimizar el tráfico y contaminación por zonas) y las de los ciudadanos y los comercios.

Se crean **7 aparcamientos públicos** (figura 3). En cada uno de ellos se situará un **contador de vehículos** en su acceso, así como un panel indicando el

número de plazas libres, un módulo de comunicaciones 5G y una **barrera automática** que controlará el acceso al aparcamiento.

Además, se ubica módulo de sensores con **panel informativo**, contador de vehículos y módulo de comunicaciones en cada uno de los cinco accesos a la ciudad. Su fin es cotejar los datos de vehículos que entran o salen de la ciudad con los que hacen uso de las bolsas de aparcamiento del sistema.



FIGURA 3 - MAPA DE APARCAMIENTOS DISUASORIOS EXISTENTES EN LA CIUDAD DE PUERTO REAL

El **módulo de sensores** (figura 4) emplea el microcontrolador Arduino. Se conectará un módulo **sensor magnético** en los pines de entrada. El **display de LEDs** para informar a los conductores se conectará en los pines de salida digitales. En los pines de salida se conectará también la **barrera de acceso** a través de un relé. Para el piloto desarrollado dentro de este TFG se podrá

emplear una barrera a escala movida por un servomotor a modo de demostración. En una fase posterior deberá considerarse si es conveniente prescindir de las barreras físicas y en su lugar informar por los paneles sobre las sanciones si se accede a la zona no permitida en ese momento.

Finalmente, se añade un **módulo 5G** para las comunicaciones con el servidor. El bloque es alimentado mediante una **batería** de capacidad suficiente y un panel solar.

Se dispone también de un **servidor** que almacena la información de recuento de vehículos. La comunicación se realiza mediante una **API REST**.

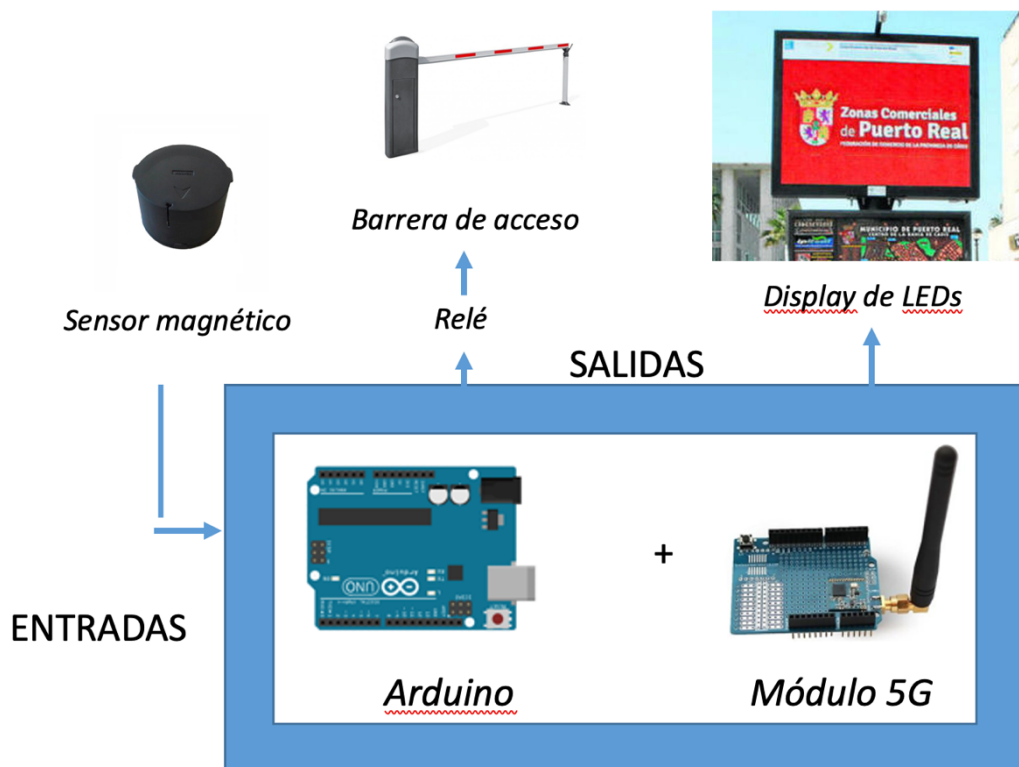


FIGURA 4 - DIAGRAMA DEL MÓDULO DE SENSORES

Se crea una **app para iOS y Android** (figura 5). En ella se muestra un mapa con los aparcamientos y las plazas disponibles. Se añade información relativa al ahorro de combustible y de **emisiones** al emplear el aparcamiento en lugar de acceder al centro en coche. También se incluye una **calculadora de Kg de**

CO2 y ahorro económico si en lugar de usar el coche se sustituye por transporte público, caminar o la bicicleta.

La app obtiene la información de una **API REST** puesta a disposición del servidor, que recibe la información de los módulos situados en los aparcamientos. Este **servidor** también cuenta con un servicio web que aloja el **portal de administración del sistema**. En él se puede dar de alta, modificar o eliminar zonas, así como gestionar el número de plazas en cada zona.



FIGURA 5 - INTERFAZ DE USUARIO DE LA APP PARA DISPOSITIVOS MÓVILES

1.2 OBJETIVOS DEL TRABAJO

- Crear un **sistema inteligente** que permita distribuir el tráfico en tiempo real con el objetivo principal de reducir las emisiones de gases de efecto invernadero en determinadas zonas congestionadas.
- Crear un piloto del **módulo de sensores**. Debe permitir detectar y contabilizar vehículos, detectar el nivel de CO₂, comunicarse mediante red móvil y controlar una barrera de acceso.
- A modo de consecuencia del punto anterior, investigar y aprender el funcionamiento del microcontrolador Arduino, su entorno y los distintos componentes electrónicos y eléctricos a emplear.
- Crear una interfaz de administración que permita a los gestores públicos **redirigir el tráfico** según situaciones de emergencia, festividades, concentraciones, etc.
- Crear una **app para iOS y Android**. Mostrará un mapa con información sobre los aparcamientos y el ahorro de combustible y de **emisiones** al emplearlos.
- Conseguir que la ciudad **se adelante** a las exigencias de las nuevas directivas europeas en cuanto a restricción de tráfico en las ciudades y **peatonalización**.
- Compaginar estos objetivos con la **actividad comercial** del centro de la ciudad.

1.3 ENFOQUE Y MÉTODO SEGUIDO

Dado que se trata de un proyecto cuya finalidad es el aprendizaje y la demostración de conocimientos adquiridos en la carrera, se opta por implementar un sistema que combine hardware y software. En ese sentido, disponemos de distintas opciones como Raspberry o Arduino, siendo finalmente seleccionada la última. El sistema a implementar trata de dar solución al problema de la polución, el tráfico y el aparcamiento en los centros urbanos congestionados. Para ello, necesitaremos una red de sensores en distintos puntos de la ciudad. La tecnología nuclear en esos módulos de

sensores será el microcontrolador Arduino, ya que reúne las características técnicas y económicas requeridas para un proyecto académico de este tipo. Raspberry Pi es un sistema de propósito general y excesivo para las funciones a ejecutar por el módulo. Si el proyecto se desarrollara con la intención de lanzarlo al mercado, habría que realizar cambios, optando quizás por desarrollar un microcontrolador más específico y dedicado, a la vez que más económico al producirlo en escala.

En cuanto a las tecnologías presentes en el mercado, existe un amplio abanico de empresas que desarrollan sistemas de aparcamiento y control de tráfico. Se opta por desarrollar un sistema similar a los existentes, simplificando la tecnología de detección para reducir costes del dispositivo. El elemento diferenciador es la adición de un módulo de detección de CO₂, que permitirá aportar información de contaminación ambiental al sistema y que éste decida en qué momento cerrar ciertas áreas de aparcamiento.

Se puede observar el diagrama de bloques del sistema en la figura 6.

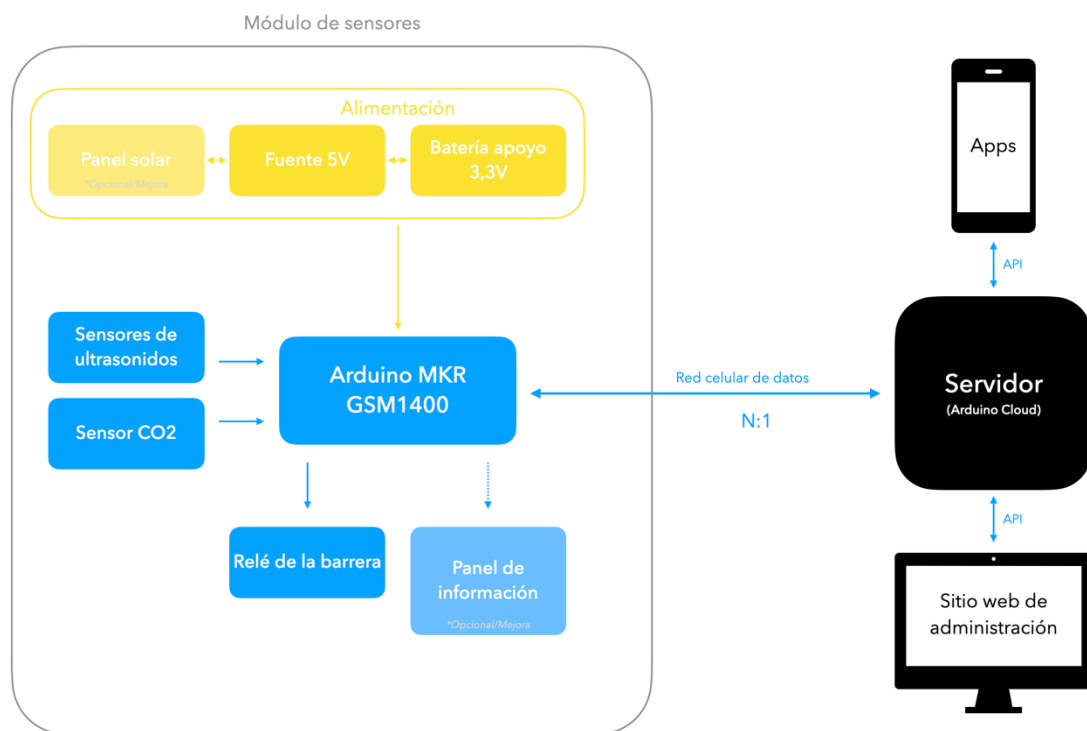


FIGURA 6 - DIAGRAMA DE BLOQUES

1.4 PLANIFICACIÓN DEL TRABAJO

La metodología de trabajo a emplear será **Scrum**. Se realizarán **sprints semanales** donde se evaluará el grado de cumplimiento de los objetivos de la semana anterior y en función del progreso se establecerán nuevos objetivos para la semana siguiente. El **daily standup** se hará a modo interno sin necesidad de coordinar con el profesor.

1.4.1 ALCANCE

Los siguientes puntos forman parte del alcance del proyecto:

- Planificación del proyecto.
- Investigar el estado del arte de los sistemas de regulación de tráfico y aparcamiento.
- Investigar el estado del arte de las tecnologías de sensores de detección y CO₂.
- Formación e investigación de tecnologías de microcontroladores disponibles.
- Selección de microcontrolador y componentes hardware a emplear.
- Investigación sobre tecnologías de comunicación inalámbrica disponibles para el microcontrolador seleccionado.
- Investigación y formación en tecnologías o plataformas disponibles para el servidor.
- Diseño del circuito electrónico del módulo de sensores.
- Desarrollo del software del módulo de sensores.
- Creación de la interfaz para configurar y controlar el módulo de sensores.
- Creación física del proyecto piloto del módulo de sensores diseñado.
- Realización de la memoria del proyecto.
- Realización de la presentación del proyecto.

No incluido en el alcance, aunque deseable si el tiempo disponible lo permitiera:

- Desarrollo de la lógica de negocio del sistema en el servidor y el panel de administración.
- Desarrollo de las apps para dispositivos móviles (para usuarios finales).
- Desarrollo del panel informativo del módulo de sensores.
- Empleo de un sistema de detección de vehículos con las últimas innovaciones disponibles y plenamente fiable (excede el alcance temporal y económico de este proyecto).

1.4.2 ENTREGABLES

Los hitos o entregables del proyecto van en consonancia con las entregas o PACs de la asignatura, la memoria y la presentación. Así, obtenemos la tabla 1:

Entrega	Descripción	Fecha
PAC1	Elaboración de la propuesta	16/09/2021
PAC2	Investigación del estado del arte	30/09/2021
PAC3	Diseño e implementación	28/10/2021
PAC4	Redacción de la memoria	16/12/2021
PAC5a	Elaboración de la presentación	06/01/2022
PAC5b	Defensa del proyecto ante tribunal	17/01/2022

TABLA 1 - ENTREGABLES

1.4.3 CALENDARIO DE TRABAJO

La asignatura del TFG contiene una carga lectiva de 12 créditos. Estos corresponden a 300 horas (a 25 créditos/hora). Hay que distribuir dicha carga lectiva a lo largo de las 18 semanas de las que se dispone para su elaboración. Se deriva una carga de unas 17 horas por cada semana. Se destinarán 2 horas diarias entre semana y 3,5 horas sábados y domingos. Se emplearán también días festivos y puentes para reforzar el desarrollo. Es un compromiso importante a asumir fuera de la horas laborales. En la tabla 2 se puede observar el calendario de dedicación del proyecto:

MES	Horas L-V	Horas S-D	Total mensual
Septiembre	22	14	36
Octubre	42	35	77
Noviembre	44	28	72
Diciembre	46	28	74
Enero	42	28	70

TABLA 2 – CALENDARIO DE DEDICACIÓN

Se obtienen 329 horas en total, suficiente para completar las 300 horas correspondientes a la carga lectiva del TFG y además disponer de 29 horas extra para cubrir posibles mejoras o contratiempos.

1.4.4 TAREAS Y DIAGRAMA DE GANTT

Se definen a continuación las distintas tareas que componen cada entregable o hito descrito en el apartado 1.4.2:

1. PAC 1: Propuesta (16/09/2021 - 29/09/2021)
 - a. Realización de la propuesta – idea y concreción.
 - b. Investigación preliminar y documentación.
 - c. Elaboración de la propuesta (PAC1).
 - d. Revisión de la respuesta y aplicación de correcciones.
2. PAC 2: Estado del arte (30/09/2021 – 27/10/2021)
 - a. Investigación del estado general del problema a solucionar.
 - b. Investigación de tecnologías concretas a emplear.
 - c. Investigación de empresas y soluciones en el mercado.
 - d. Investigación del microcontrolador Arduino.
 - e. Investigación de sistemas inalámbricos de comunicación para Arduino.
 - f. Investigación de sensores a emplear en el sistema.
 - g. Adquisición de material a emplear en el sistema.
 - h. Elaboración del documento PAC2.
3. PAC3: Diseño e implementación (28/10/2021 – 15/12/2021)
 - a. Formación y pruebas con placa de aprendizaje Arduino UNO.
 - b. Elaboración de circuitos parciales para sensores y servomotor.

- c. Elaboración de circuito parcial de comunicaciones.
- d. Diseño del circuito final a implementar.
- e. Creación del circuito final sobre placa Arduino MKR GSM1400.
- f. Desarrollo del software a inyectar en el módulo Arduino.
- g. Desarrollo del panel de administración del módulo en Arduino IoT Cloud.
- h. Pruebas del proyecto piloto.
- i. Elaboración del documento PAC3, vídeos y fotos de demostración.

4. PAC4: Memoria (16/12/2021 – 05/01/2022)

- a. Aplicación de correcciones recibidas para PACs anteriores.
- b. Reedición de todo el contenido existente de forma estructurada en el nuevo documento.
- c. Adición de contenido restante para completar la memoria.

5. PAC5a: Presentación (06/01/2022 – 16/01/2022)

- a. Elaboración de la presentación como compendio resumido de la memoria.

6. PAC5b: Defensa (17/01/2022 – 23/01/2022)

- a. Respuesta a las preguntas del tribunal de evaluación.

Se puede observar el diagrama de Gantt resultante en la figura 7:

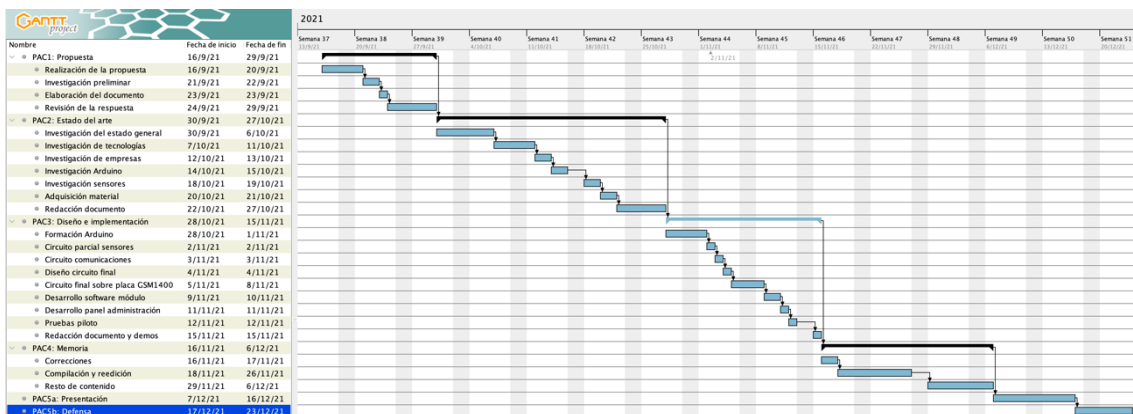


FIGURA 7 - DIAGRAMA DE GANTT

1.4.5 RIESGOS E INCIDENCIAS

Se observan distintos riesgos:

- El proyecto se ha desarrollado en el contexto de unas circunstancias especiales debidas a la pandemia provocada por el coronavirus y su enfermedad asociada covid-19. Como consecuencia, las cadenas mundiales de suministros se han visto muy afectadas y, en especial, la producción de microchips. Era por ello el principal riesgo para la consecución de los objetivos de este proyecto, dada la pretensión de crear un proyecto piloto físico. Como contramedida, se realizó el pedido de material en una fase temprana (al comienzo de la PAC2). Así, y habiendo sido finalmente posible la conclusión del proyecto piloto, el proyecto se ha visto afectado por esta incidencia y no se han podido conseguir todos los objetivos propuestos.
- Al desconocer a priori la tecnología Arduino y sus posibles debilidades, se corre el riesgo de no poder absorber toda la información necesaria para la elaboración del prototipo a tiempo.

1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS

El producto a obtener es un sistema que permita regular el tráfico según la contaminación del centro urbano. En ese sentido los elementos obtenidos son:

- Prototipo hardware del módulo con todos los sensores necesarios, actuador para la barrera y módulo de comunicaciones móviles.
- Software del módulo de sensores.
- Panel de administración del módulo.
- Memoria del proyecto.
- Presentación del proyecto.

1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA

Se han estructurado los contenidos de los siguientes capítulos teniendo en cuenta que se trata de un proyecto en evolución constante desde su comienzo.

Por tanto, en primer lugar se analiza el estado del arte de las soluciones propuestas a nivel mundial para el problema y las tecnologías implicadas. Luego se incluye la información del sistema Arduino empleado. Posteriormente, se analizan uno por uno los sensores empleados, sus circuitos aislados y sus posibles alternativas (y el motivo de su elección). Finalmente se muestra el software implementado en el sistema Arduino para finalmente analizar la herramienta Arduino IoT Cloud empleada para crear el panel de administración.

- **Capítulo 2:** Estado del arte
 - 2.1 Estado del arte de las soluciones al problema a nivel global.
 - 2.2 Estado del arte de las soluciones comerciales y tecnologías empleadas.
- **Capítulo 3:** Arduino: se expone una explicación sobre el microcontrolador Arduino empleado para el desarrollo inicial del proyecto. También las características de la placa empleada finalmente y el motivo de hacerlo.
 - 3.1 El microcontrolador Arduino.
 - 3.2 Primera aproximación: La placa Arduino UNO.
 - 3.3 Selección final: La placa Arduino MKR GSM 1400.
- **Capítulo 4:** Sensores y actuadores empleados: se detallan los distintos sensores empleados en el proyecto. También se muestran las alternativas y se describe el motivo de la elección. Para algunos, se muestra el circuito parcial desarrollado para aprendizaje y pruebas.
 - 4.1 Servomotor.
 - 4.2 Sensor de CO2.
 - 4.3 Sensor de ultrasonidos.
- **Capítulo 5:** Módulo de sensores: se presenta la versión final del hardware y software desarrollado para el módulo de sensores junto con una explicación detallada del código y las decisiones tomadas.
 - 5.1 Hardware.
 - 5.1.1 Placa Arduino MKR GSM 1400.
 - 5.1.2 Circuito final del módulo de sensores.
 - 5.2 Software.

- 5.2.1 Código de la placa.
- **Capítulo 6:** Conclusiones
- **Capítulo 7:** Glosario
- **Capítulo 8:** Bibliografía

2. ESTADO DEL ARTE

2.1 ESTADO DEL ARTE DE LAS SOLUCIONES DEL PROBLEMA A NIVEL GLOBAL

Existen iniciativas tanto públicas como privadas encaminadas a paliar dos de los principales problemas de nuestro tiempo: la movilidad en zonas urbanas y la lucha contra el cambio climático. En ese sentido, se necesita focalizar el problema y acudir a soluciones ya existentes, mejorándolas a nivel conceptual y técnico.

Poniendo el foco en las soluciones aportadas a lo largo de la historia para facilitar la movilidad en centros urbanos se encuentran buenos ejemplos y recursos que serán útiles en la consecución del objetivo de este proyecto de reducir las emisiones de CO₂ derivadas del transporte de personas.

Una de estas soluciones es el aparcamiento disuasorio. Fueron inventados a principios del siglo XX en las principales ciudades de EEUU, donde se situaban junto a las estaciones de tren para fomentar el uso de éste. Posteriormente se aplicaron en otros espacios de las ciudades, como centros comerciales. Pronto se dieron cuenta^[8] de que empleando aparcamientos disuasorios podían solucionar un problema de diseño de las ciudades americanas: centros compactos y bien comunicados con transporte público por un lado y áreas de diseminados en las afueras de la ciudad por otro. Este problema se amplió a Europa conforme se imitaba el modelo urbano americano.

Existieron también experimentos fallidos. Se puede mencionar el de los años 60 en París, donde se ubicaron macro aparcamientos en el centro de la

ciudad^[8]. Pero estos terminaron siendo utilizados como aparcamientos regulares. La solución estaba en buscar el límite entre el centro compacto y las áreas de viviendas diseminadas y sin buen transporte público. Aplicando allí estos aparcamientos disuasorios se conseguirían mejores resultados. Los habitantes de los diseminados, acostumbrados a moverse siempre en vehículo privado, podían aparcar en el aparcamiento disuasorio más cercano a su destino. Desde él partirían líneas de bus, metro o tranvía creadas específicamente para dar servicio al aparcamiento. Este concepto se hizo pronto muy popular especialmente en Centro Europa^[8], reduciendo fuertemente el tráfico en el centro de las ciudades y recuperándolas para la ciudadanía. En el caso de España, es en los últimos años cuando se ha comenzado a andar en esa dirección.

Muchas corporaciones municipales implementan distintas medidas:

- Intentar sustituir viajes en vehículo privado por viajes en vehículo público.
- Sustituir viajes motorizados por viajes no motorizados (o levemente), es decir a pie, en bicicleta o en patinete eléctrico.
- Minimizar el número de desplazamientos a realizar.

Se analizan a continuación las soluciones que ha ofrecido el mercado para lograr estos objetivos. En primer lugar, y siguiendo con los aparcamientos disuasorios, existen distintas tipologías^[8]:

- Según el modo de transporte público asociado: tren, metro, bus, bicicletas de alquiler o patinetes eléctricos. Típicos de las grandes urbes.
- Según el horario de funcionamiento, que puede ser a diario, solo fines de semana o solo en periodos especiales (vacaciones, ferias, eventos, etc).
- Según el tamaño de las instalaciones y las plazas disponibles:
 - Menos de 200 plazas: Se emplean en entornos muy urbanizados y con poco espacio, creando una malla tupida de aparcamientos. Si no son bien dimensionados pueden saturarse fácilmente.
 - Entre 200 y 500 plazas: tamaño perfecto para ser atendidos por autobuses de alta eficacia. Puede ser poco rentable

económicamente si se quieren ofrecer todos los servicios de seguridad, iluminación, WC, cafetería, etc.

- Entre 500 y 1000 plazas: para ser servido por ferrocarril o metro. Es el tamaño óptimo para maximizar la inversión.
- Más de 1000 plazas: servidos por varios sistemas distintos de alta capacidad. Presentan el problema de que dado su tamaño requieren soluciones para movilidad interior en el propio aparcamiento.
- Según el motivo del desplazamiento del usuario a la ciudad destino:
 - Desplazamiento por trabajo.
 - Desplazamiento relacionado con el sector servicios: compras, centros educativos, visitas al médico, gestiones administrativas, playas, parques naturales, zonas de ocio nocturno, etc.
 - Desplazamiento relacionado con el turismo.

2.2 ESTADO DEL ARTE DE LAS SOLUCIONES COMERCIALES Y TECNOLOGÍAS EMPLEADAS

Una vez analizada la cuestión en cuanto al impulso público, se analizan las tecnologías que se están implementando para conseguir esos objetivos.

Se pueden diferenciar cuatro soluciones tecnológicas diferentes que cubren distintos casos de uso:

1. Redes de sensores en la vía pública para detectar si hay un vehículo estacionado en cada plaza: son las más habituales, ya presentes en la mayoría de aparcamientos subterráneos por su utilidad en el guiado de los usuarios a través de estos y la disminución del tráfico en su interior. La innovación en esta área ha venido por parte de la conexión de estos sensores a la nube, lo que permite emplearlos también en exteriores para las plazas públicas en la calle. La firma española Libelium^[9] fabrica este tipo de sensores que emplean tecnología radar y magnética (figura 8).



FIGURA 8 - SENSOR MAGNÉTICO DE LIBELIUM

Usan LoRaWAN para la comunicación, consiguiendo un consumo ajustado que permite que la batería dure 5 años o más. Uno de los primeros proyectos donde se empleó esta tecnología es el sistema SFPark^[10] de San Francisco, EEUU, que mediante el despliegue de miles de sensores en sus calles permite incluso ofrecer distintos precios por aparcar en cada calle.

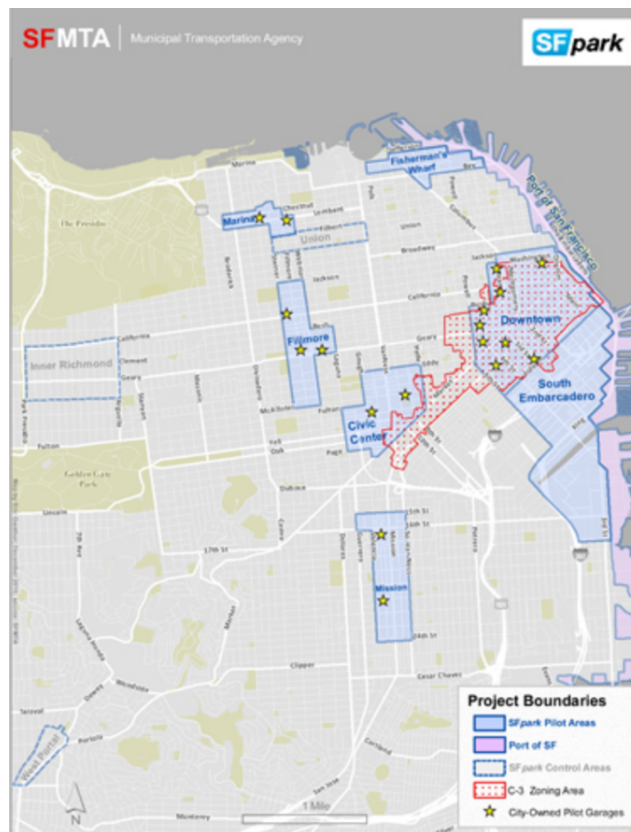


FIGURA 9 - MAPA DE SENSORES DE SFPARK

2. Redes basadas en cámaras para detectar plazas vacías en la calle: es una nueva tecnología que se está desplegando en los últimos años. Emplea sensores fotográficos que mediante inteligencia artificial determinan si hay o no plazas libres. Tiene la ventaja de que además permite controlar a los peatones y cualquier elemento móvil en la vía pública.

Hay dos proyectos principales en marcha relacionados con esta tecnología:

- Por un lado, **Sony** está desplegando en Roma un proyecto piloto empleando su cámara IMX500^[11]. Su sistema tiene un coste elevado ya que requiere este modelo de cámara dedicada, pero tiene la ventaja de que respeta la privacidad (la IA se realiza dentro de la propia cámara, por lo que las imágenes nunca salen de ella). Además, puede emplearse para controlar la saturación de peatones en la vía pública o en autobuses.

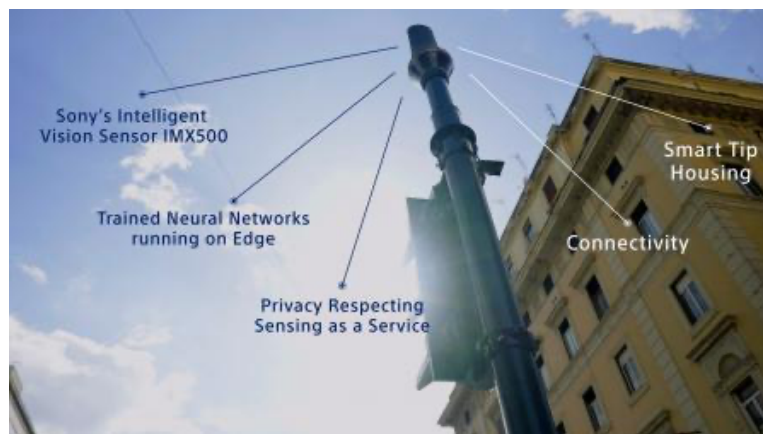


FIGURA 10 - DISPOSITIVO SONY IMX500

- Por otro lado, existe otro proyecto piloto en Israel llamado **ParKam**^[12]. Este sistema emplea las propias cámaras de vigilancia de la ciudad para mediante inteligencia artificial determinar dónde hay plazas libres. Es mucho más invasivo en cuanto a la privacidad que el sistema de Sony, ya que emplea

estas imágenes. Aunque eran ya tomadas igualmente con motivos de seguridad.

3. Redes basadas en 5G^[13]: las redes de aparcamiento inteligente que posibilita la tecnología móvil 5G están en una fase aún embrionaria, pero prometen solucionar muchos de los problemas actuales en un futuro. Se basan en el hecho de que cada vehículo facilitará información sobre su ubicación, estado y demás parámetros para permitir la conducción autónoma. Esta información permitirá también saber si el vehículo está aparcado y dónde. Es la mejor opción para outdoor, ya que evita el enorme coste que conlleva instalar una red de sensores en la vía pública. Sin embargo, para que sea efectiva esta tecnología necesita que antes se extienda su uso a la mayoría de vehículos, algo aún muy lejano en el tiempo.

4. Sistemas para aparcamientos disuasorios basados en sensores en los accesos: se trata de una tecnología ampliamente implementada en aparcamientos subterráneos y grandes superficies. Mediante sensores en las entradas y salidas del aparcamiento se controla cuántos vehículos entran y salen y por tanto el número de plazas libres. Es un sistema menos preciso, que no guía hasta la plaza en concreto, pero mucho más barato y de fácil mantenimiento. La firma Urbiotica^[14] comercializa un sistema basado en sus sensores U-Flow y U-Spot. El sensor magnético de conteo U-Flow se ubica en entradas y salidas para detectar el paso de un vehículo y sumarlo o restarlo en la base de datos del sistema. El sensor U-Spot por su lado se coloca en plazas PMR (personas con movilidad reducida) o reservadas y se encarga de contar esos vehículos específicos para poder tenerlos en cuenta en el registro general de plazas del aparcamiento.

Una vez comentadas las distintas tecnologías existentes en el mercado, se puede comprobar que el presente proyecto trata de aportar soluciones innovadoras en cuanto a los sistemas para aparcamientos disuasorios.

Aunque ya existen empresas asentadas en el mercado que ofrecen soluciones completas, como Urbiotica, también es cierto que el enfoque es diferente. En el presente proyecto se trata de dar una solución a la contaminación en el centro urbano de la ciudad. Concretamente, se pone el foco en el CO₂ emitido por los vehículos que circulan por la ciudad buscando aparcamiento. Se estima que el 30% de los vehículos que en un instante dado circulan por la ciudad lo hacen buscando una plaza de aparcamiento^[15]. Se genera en torno a 1,3Kg de CO₂ por cada vez que se aparca un vehículo^[16].

Para reducir estas emisiones drásticamente, el presente proyecto trata de dar respuesta mediante una experiencia piloto en la ciudad de Puerto Real. Se seleccionarán distintos aparcamientos disuasorios ya existentes en la localidad y se propondrá la creación de otros nuevos a fin de dimensionar la solución para dar respuesta a las necesidades de aparcamiento una vez se eliminen todas las plazas públicas presentes en las calles del centro. En cada aparcamiento se instalarán sensores de entrada y salida en sus accesos. El sensor deberá ser capaz de detectar de forma fiable la presencia de un vehículo y su dirección (si está entrando o saliendo). Además, el sensor dispondrá de un detector del nivel de CO₂. Este enviará la información recogida a una base de datos central. Esta base de datos alimentará un panel de control desde donde el personal municipal podrá controlar el flujo del tráfico y tomar decisiones en base a ello. Se dispondrá también de paneles informativos y aplicaciones móviles para informar a los usuarios de la disponibilidad de plazas.

Y en este punto es donde se marcará la diferencia con respecto a los sistemas convencionales. Además de mostrarle a los usuarios el número de plazas libres en cada aparcamiento, se les mostrará el nivel de CO₂ en la población y la cantidad de emisión de CO₂ que ahorrarían si hicieran el mismo trayecto en bicicleta o a pie. En los accesos a los aparcamientos se instalará una barrera conectada al sistema. Esta barrera podrá ser cerrada de forma automática si el nivel de CO₂ detectado sube de determinado umbral (se mostraría un mensaje de advertencia en los paneles instalados en la ciudad y en las apps). La barrera podrá ser cerrada también desde el panel de control por parte del personal de

movilidad del ayuntamiento si fuera necesario (para redirigir el tráfico a otras zonas de la ciudad, por ferias, obras, etc).

2.3 SOLUCIONES PARA REDUCIR LAS EMISIONES DE CO2 EN EL CENTRO URBANO DE PUERTO REAL

En nuestro caso, al tratarse de una población de 41.700 habitantes ^[17], no se dispone de servicios de transporte de alta densidad. En este caso se pretende apoyar la peatonalización del centro urbano, con lo cual el transporte será a pie, en bicicleta, en patinete eléctrico o en scooter. La orientación será para un uso diario, aunque se podrá aplicar en grandes eventos (como ferias) para facilitar la organización de las plazas y distribución de los coches. Los aparcamientos disuasorios están dimensionados a menos de 200 plazas y cubren todo tipo de desplazamientos, tanto trabajo como ocio o compras. Debemos tener en cuenta que se trata de hacer el centro urbano más amigable y descarbonizado. Esto supone un factor diferencial de este proyecto con respecto a los que se encuentran en el mercado ya en producción, que se suelen enfocar hacia grandes urbes con transporte público de masas y el foco se pone en su potenciación.

En el mismo sentido podemos comprobar que desde los organismos públicos se está potenciando la implantación de zonas de bajas emisiones ^[18]. Estas se enmarcan en un contexto de concienciación creciente ante el reto del cambio climático. Para combatirlo, la Unión Europea, los gobiernos que la componen y los ayuntamiento están desplegando una serie de ayudas y normativas para forzar un cambio radical en los hábitos de movilidad y consumo.

Dentro de las principales actuaciones solicitadas en el programa de ayudas a municipios para la implantación de zonas de bajas emisiones podemos ver que se encuentran la peatonalización del centro de las ciudades, el impulso de la bicicleta, la descarbonización del transporte urbano y la creación de aparcamientos disuasorios. Esto encaja a la perfección con el objetivo de este

proyecto y, además, es algo que el propio Ayuntamiento de Puerto Real está impulsando a través de su proyecto +Ciudad ^[19].

3. ARDUINO

3.1: EL MICROCONTROLADOR ARDUINO

3.1.1: QUÉ ES ARDUINO

El microcontrolador Arduino, según la definición de la propia compañía, es “ una plataforma open-source basada en un hardware y software fácil de usar. Las placas Arduino son capaces de leer entradas – la luz que incide en un sensor, un dedo pulsando un botón o un mensaje de Twitter – y convertirlas en salidas – activar un motor, encender un LED, publicar algo online. Puedes decirle a tu placa qué hacer enviándole un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo se utiliza el lenguaje de programación Arduino (basado en Wiring) y el Arduino Software (IDE), basado en Processing”.^[20]

Es por tanto una plataforma multidisciplinar ideal para prototipar y validar ideas. Es ideal para sistemas con sensores y actuadores para automatizar tareas rutinarias. Es ampliamente utilizado en domótica, industria, robótica, agricultura, IoT, etc.^[21]

3.1.2: TIPOS DE PLACAS ARDUINO

Existen distintos tipos de placas Arduino (figura 11). Algunas destinadas a aprendizaje inicial, como Arduino UNO. Otras, añaden algunas funcionalidades específicas relativas a conectividad, sensores, forma, tamaño, precio, etc.^[22]

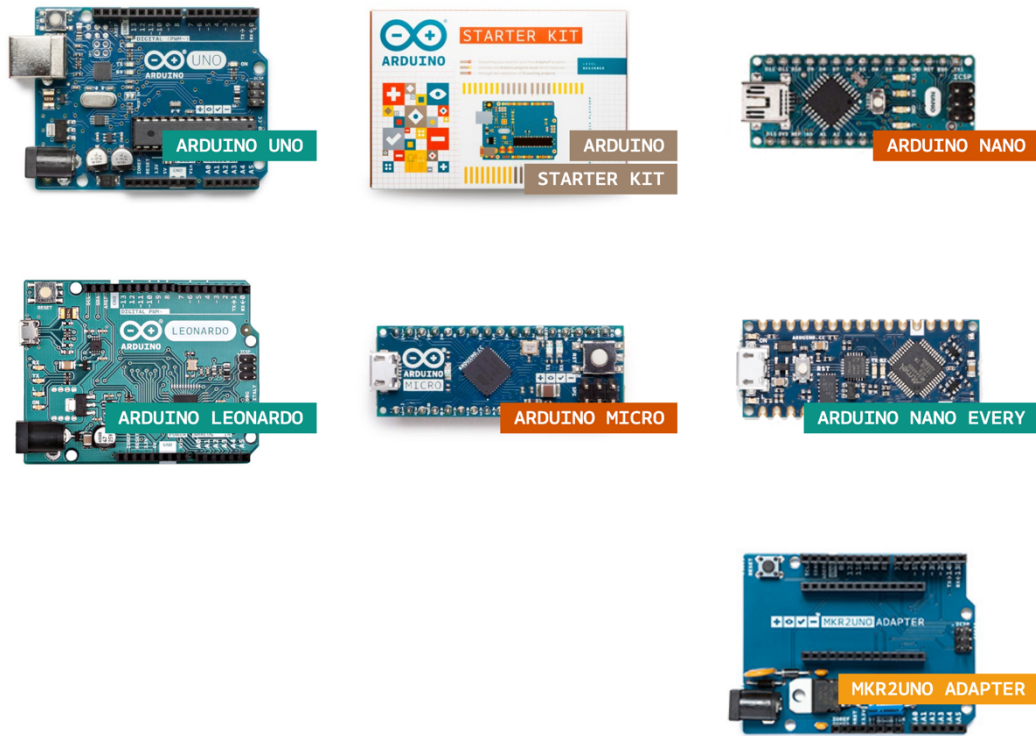


FIGURA 11 - PLACAS DE INICIACIÓN ARDUINO

También existen las llamadas “shields” (figura 12), que son placas que añaden alguna función específica (como conectividad Wifi) y se conectan sobre la placa Arduino de un modo muy sencillo.

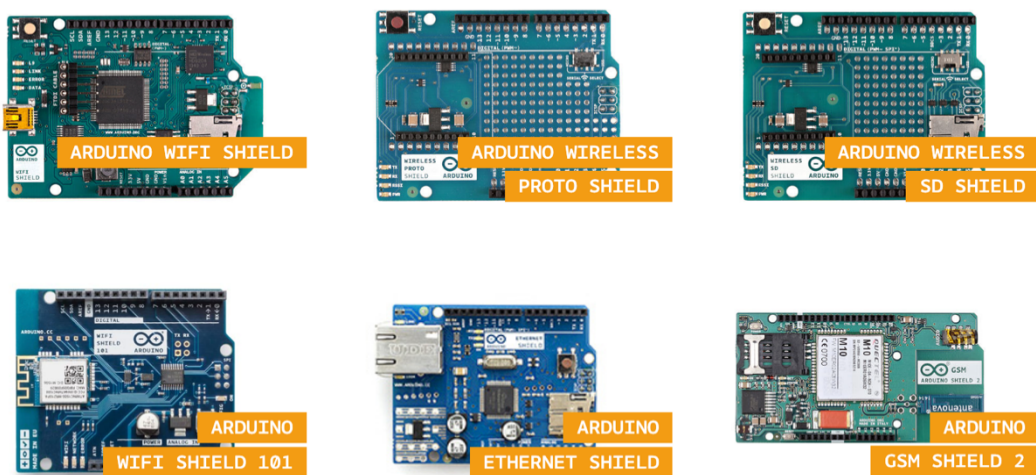


FIGURA 12 - ARDUINO SHIELDS

Por último, existe un nuevo tipo de categoría de placas Arduino denominada IoT (Internet of Things) como se puede ver en la figura 13. Su finalidad es crear productos con conectividad inalámbrica de un modo sencillo. Ofrece también la posibilidad de conectarse a Arduino Cloud, que permite recolectar la información de los dispositivos y ofrecerla a través de una API para que sea consumida en clientes web o móvil.



FIGURA 13 - PLACAS ARDUINO IOT

3.1.3: SOFTWARE

El sistema Arduino dispone de un lenguaje de programación propio y también un IDE dedicado, ambos también llamados “Arduino”^[23].

El lenguaje de programación Arduino está basado en el lenguaje de programación C, de bajo nivel e imperativo y por tanto muy apropiado para dispositivos con poca memoria y que ejecutan un conjunto de código reducido. Los programas creados con este lenguaje se almacenan en ficheros denominados “Sketch”, que siempre tienen una misma estructura (ver figura 14):

- Función **setup()**: Esta función se emplea para inicializar las variables y configurar las distintas entradas y salidas analógicas y digitales de la placa.
- Función **loop()**: se trata de un bucle, que se ejecuta recurrentemente solo limitado por la velocidad de proceso del microcontrolador. Contiene la lógica principal del sistema. Normalmente se añaden pequeños retardos para permitir obtener la información de los distintos sensores o realizar las acciones en los actuadores antes de la siguiente ejecución del bucle.



```
sketch_jan11a Arduino 1.8.16
sketch_jan11a
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
1
Arduino Uno en /dev/cu.usbmodem14101
```

FIGURA 14 - IDE ARDUINO

Además del IDE para escritorio, Arduino ha creado el Arduino Web Editor, que mediante un proceso *demon* instalado en el equipo del usuario permite rastrear posibles placas Arduino conectadas e inyectar el código desde la web, evitando tener que instalar un programa dedicado. Por último, Arduino Cloud permite también programar la placa desde un entorno Web. En este caso orientado a la programación de placas Arduino IoT.

3.2: PRIMERA APROXIMACIÓN: LA PLACA ARDUINO UNO

Tras comprobar lo distintos tipos de placas Arduino y shields disponibles, se decide empezar haciendo una prospección de posibilidades del sistema de aprendizaje UNO^[24] (figura 15). Como puede observarse en la figura 11, existen más placas de iniciación como la micro o la nano. Sin embargo, son placas para fines específicos (ahorro de tamaño o de coste) que no son válidos para la consecución del objetivo inicial de aprender sobre el sistema Arduino y detectar sus potencialidades de cara al proyecto. La placa UNO es perfecta para este cometido, ya que es compatible con la mayoría de shields, es sencilla y amplia para trabajar con los pines y es económica.

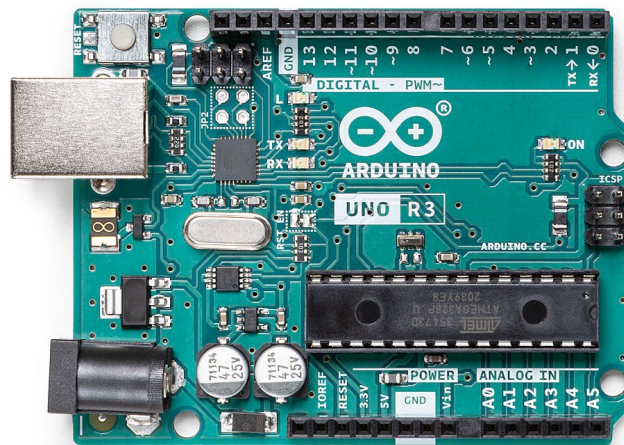


FIGURA 15 - PLACA ARDUINO UNO

Dispone de un microcontrolador ATmega328, 14 pines digitales, 6 entradas analógicas, 32KB de memoria y una velocidad de 16MHz. La alimentación puede realizarse mediante el puerto USB (5 voltios) y también a través del pin Vin (donde por ejemplo se puede conectar una batería).

Se comienza realizando una introducción al microcontrolador Arduino. Para ello, se realizan algunos de los proyectos contenidos en el Arduino Starter Kit. Se seleccionan los que están relacionados con sensores y servomotores, ya que serán los elementos empleados en el proyecto. Se completa también un breve curso de introducción a Arduino^[25]. En el capítulo 3 se exponen los

distintos sensores seleccionados para el proyecto, su motivación y un breve circuito de demostración sobre Arduino UNO.

3.3: SELECCIÓN FINAL: LA PLACA ARDUINO MKR GSM 1400

Existen diversas opciones para ampliar la funcionalidad de Arduino UNO. La más conocida es el empleo de “shields”, como se ha mencionado en el apartado 2.1.2. En este caso se emplearía un shield que permita enviar la información mediante tecnología móvil GSM, ya que los módulos estarán instalados en la vía pública y no está garantizada la conexión mediante otros medios inalámbricos o cableados.

Sin embargo, tras una breve investigación los shields GSM resultan ser escasos y costosos^[26]. La solución viene finalmente por modificar el planteamiento inicial y emplear un módulo Arduino que ya integre el módulo GSM. Como podemos ver en la figura 13, existen varias placas Arduino IoT que integran algún chip para comunicaciones inalámbricas. También se pueden encontrar clones de fabricantes distintos a Arduino en el mercado, pero no ofrecen la integración directa con el sistema Arduino IoT. Finalmente y debido a esto, el modelo seleccionado, por coste, tamaño y calidad es el Arduino MKR GSM 1400 (figura 16).



FIGURA 16 - PLACA ARDUINO MKR GSM 1400

La placa MKR GSM 1400 incorpora un procesador Arm Cortex M0 32-bit SAMD21, el chip GSM u-blox SARA-U201 y el chip criptográfico ATECC508 que puede almacenar hasta 16 claves en su array EEPROM.

Siguiendo la documentación oficial de Arduino se realizan los tutoriales y prácticas relacionados con el módulo MKR GSM 1400^[27]. En el desarrollo de estos tutoriales, se descubre que la placa necesita una alimentación de unos 500mA normalmente, aunque en los GSM handshake (cuando la placa se inicia y se registra en un proveedor de red) puede alcanzar los 2A. Por lo tanto, se necesita conectar una batería LiPo que suplemente la alimentación USB. La alimentación USB proporciona 5V y una corriente suficiente para programar y activar alguna luz o actuador, pero insuficiente para los GSM handshakes. Se selecciona una de 2000mA y 3,7V, que debería ser suficiente para este cometido. La propia placa carga la batería cuando se está operando normalmente y cambia automáticamente entre alimentación general o batería cuando es necesario, con lo que el aporte de corriente está asegurado.

Se emplea además una antena tipo dipolo cinco bandas resistente al agua (figura 17). Esta antena soporta las bandas GSM 850 / 900 / 1800 / 1900 MHz y UMTS. Esta antena será más que suficiente para cubrir las necesidades de conectividad del módulo de sensores.



FIGURA 17 - ANTENA DIPOLO CINCO BANDAS

4. SENSORES Y ACTUADORES EMPLEADOS

En este capítulo se comentan los dispositivos de entrada y salida empleados en el módulo de sensores. Para cada uno, se muestran las distintas alternativas, el motivo de la elección final y un circuito de demostración del elemento aislado.

4.1: SERVOMOTOR

Como una primera aproximación, se realiza la simulación de la barrera que impedirá la entrada de vehículos al aparcamiento. Por motivos prácticos y de alcance del proyecto no se estimará el empleo de una barrera real. En este caso, sería necesario crear un circuito de adaptación con relé, ya que el motor de la barrera tiene obviamente una alimentación independiente mucho más potente.

Para simular la barrera, se emplea el módulo Arduino UNO, conectando un servo regulado por un potenciómetro. Añadimos un condensador porque al arrancar el motor necesita un extra de corriente. El condensador almacena ese extra puntual de corriente y lo libera en el arranque del servo. Evitamos así dejar debilitado el circuito esos milisegundos. Posteriormente el potenciómetro se retirará del circuito.

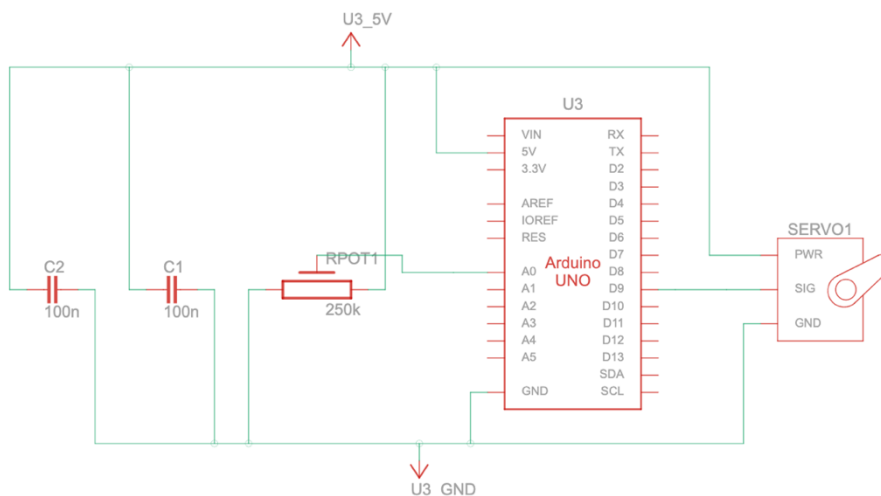


FIGURA 18 - CIRCUITO DE CALIBRADO DEL SERVOMOTOR

Código:

```
#include <Servo.h>
Servo myServo;

int const potPin = A0;
int potVal;
int angle;

void setup() {
  // put your setup code here, to run once:
  myServo.attach(9);

  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  potVal = analogRead(potPin);
  Serial.print("potVal: ");
  Serial.print(potVal);

  angle = map(potVal, 0, 1023, 0, 179);
  Serial.print(", angle: ");
  Serial.print(angle);

  myServo.write(angle);
  delay(15);
}
```

En el código, se conecta el potenciómetro a la entrada analógica A0 y el servo en la salida digital 9. Se traduce el valor del potenciómetro a ángulo de giro del servo y se introduce un delay de 15ms para permitir que el servo actúe.

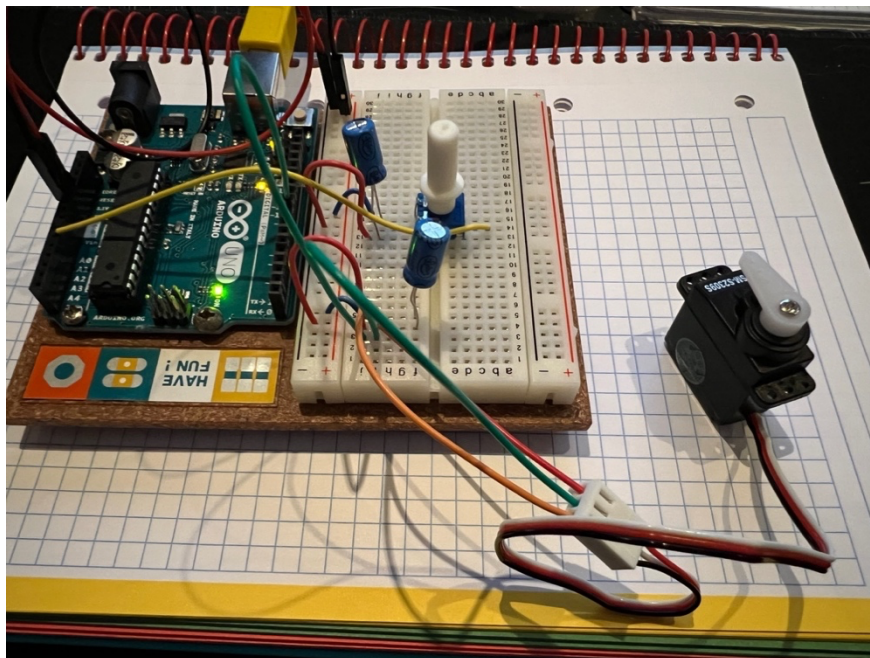


FIGURA 19 - MONTAJE DE CIRCUITO DE PRUEBA DEL SERVOMOTOR

Mediante este pequeño circuito, se calibra el servo de modo que para cada giro del potenciómetro se identifican los valores que corresponden a los grados que se aplican en el servo.

4.2: SENSOR DE CO₂

Posteriormente, se reutiliza el mismo circuito con la placa Arduino UNO (ver figuras 20 y 21) para incorporar y calibrar el sensor de CO₂. Existen varios modelos de medidores de CO₂ según las necesidades. Por ejemplo, el SprintIR-W de GSS^[28] permite medir niveles de CO₂ que cambian rápidamente. Luego, hay otros dispositivos como el CO₂ Probe GMP252 de Vaisala^[29], indicado para ser empleado en entornos húmedos (para agricultura principalmente). También existen sensores de nueva generación y muy reducido tamaño y consumo como el SCD4x de Sensirion^[30].

Finalmente y tras analizar las opciones, se selecciona el sensor MQ135^[31]. Es un sensor sencillo, estable, de larga duración, alta sensibilidad, rápida respuesta, amplio rango de detección y muy económico y por tanto indicado para este proyecto piloto. Tras investigarlo^[32], obtenemos que los niveles esperables en el centro de una ciudad (valores altos) serían de unos 700 PPM de CO₂. Para calibrar el sensor, se muestran por consola unos valores para el rango bajo, medio y alto de CO₂ que servirán de guía. Una vez anotados los valores, podemos establecer los rangos de niveles de CO₂ para enviar la señal correspondiente al servidor.

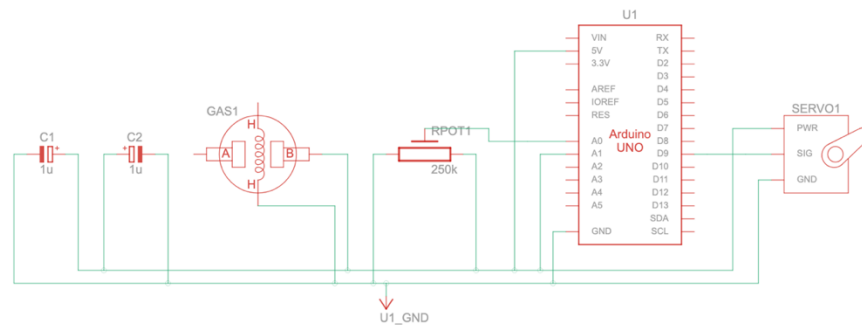


FIGURA 20 - CIRCUITO CO₂

Código:

```
#include <Servo.h>
Servo myServo;
int const potPin = A0;
int const co2Pin = A1;
int potVal;
int angle;
int valorCO2; // Valor de CO2 medido

void setup() {
  // put your setup code here, to run once:
  myServo.attach(9);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  potVal = analogRead(potPin);
  Serial.print("potVal: ");
  Serial.print(potVal);

  angle = map(potVal, 0, 1023, 0, 179);
  Serial.print(" angle: ");
  Serial.print(angle);
  myServo.write(angle); // Escribimos el ángulo de rotación en grados a efectuar en el servo

  valorCO2 = analogRead(A0); // lee el valor que llega por la entrada A0 y lo asigna a la variable valorCO2
  Serial.print(" CO2: ");
  Serial.print(valorCO2);
  Serial.print(" PPM");
  if(valorCO2 < 300) {
    Serial.println(" BAJO");
  } else if(valorCO2 >= 300 && valorCO2 < 600) {
    Serial.println(" MEDIO");
  } else if(valorCO2 >= 600) {
    Serial.println(" ALTO");
  }
  delay(500); // Insertamos un retardo para permitir que el salto en el servo sea pequeño y el sensor CO2 no lea demasiado a menudo
}
```

Añadimos la conexión de señal del sensor de CO₂ a la entrada analógica A1. Posteriormente, en el loop se muestran los valores recibidos por el pin y se asignan a intervalos BAJO, MEDIO y ALTO según el valor de CO₂.

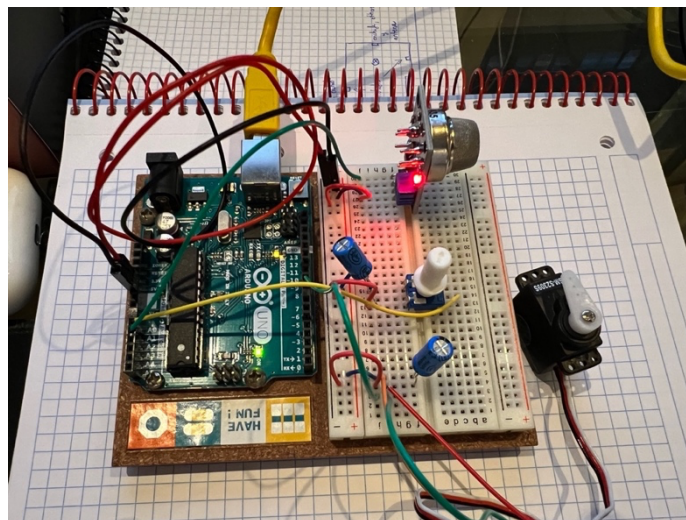


FIGURA 21 - MONTAJE DE CIRCUITO DE CALIBRADO DEL SENSOR DE CO₂

4.3: SENSOR DE ULTRASONIDOS

Para simular la detección del vehículo, se hace un estudio de los distintos sistemas existentes en el mercado^{[4][33][34]}. Los dispositivos comerciales más complejos son los basados en detección magnética. Estos aprovechan que los vehículos a detectar siempre contendrán hierro. Se trata de elementos demasiado complejos y costosos como para incluirlos en el alcance de este proyecto piloto. Además, el núcleo de este proyecto no es la tecnología de detección de vehículos en sí. Por lo tanto, la decisión final es emplear cualquier dispositivo de detección simple que permita simular la detección de un vehículo real y que ofrezca una señal digital o analógica apropiada para el módulo Arduino.

En ese sentido, se decide emplear detectores de ultrasonidos. Estos son los más sencillos y permitirán realizar la simulación del sistema con solvencia y evitando complicaciones fuera del alcance de este proyecto.

Se decide emplear dos sensores de ultrasonidos HC-SR04. Su rango de medición teórico es de 2 a 400cm con una resolución de 0.3cm^[35]. Sería suficiente en teoría para el objetivo de detectar vehículos entrando o saliendo del aparcamiento. Sin embargo, su rango real es menor y además son muy imprecisos en exteriores. Esto no supone un problema puesto que en el presente proyecto se va a trabajar con objetos simulados y a escala. Para la implementación real serían necesarios sensores magnéticos mucho más complejos y caros, lo cual se indica como mejora.

La disposición de los sensores será en línea con el carril, empleando dos sensores de ultrasonidos. Con ello, se consigue reconocer si el vehículo está entrando o saliendo según qué sensor de los dos detecta primero (ver figura 22).

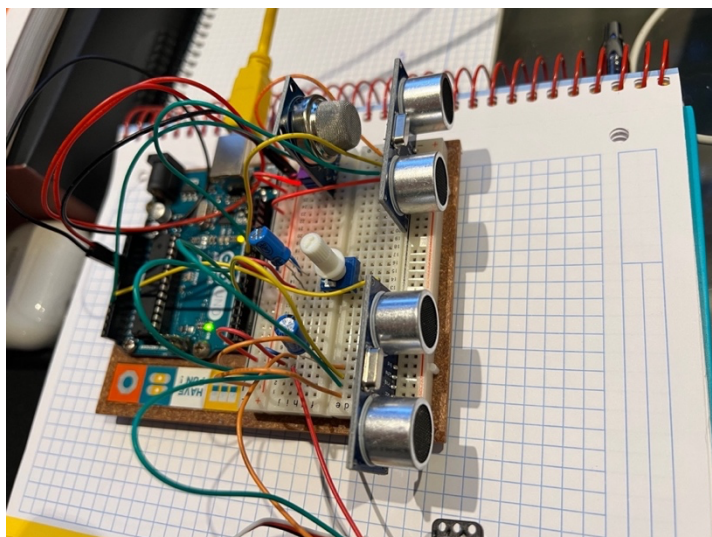


FIGURA 22 - CIRCUITO DETECTOR ULTRASONIDOS

5. MÓDULOS DE SENSORES

5.1: HARDWARE

5.1.1: PLACA ARDUINO MKR GSM 1400

Para el diseño del circuito final del módulo de sensores se emplea la placa Arduino MKR GSM 1400. Esta placa posteriormente se conecta a Arduino IoT Cloud a través de una SIM específica de Arduino que ofrece una tarifa gratuita con una capacidad limitada (y con pago por uso si se desea revasar ese límite). Dado lo limitado del tiempo finalmente disponible para ejecutar el proyecto, la nube de Arduino hará las funciones de backend y web de administración (simplificados). Para ello, se siguen los tutoriales correspondientes^[36]. Una vez dada de alta la tarjeta SIM en el portal de Arduino, se instala y configura con los datos recibidos (APN/PIN/Usuario/Contraseña).

Se conectan a continuación la antena GSM, la batería LiPo de 3,3V y el cable USB. Finalmente, se carga el código de prueba (Un “Hola mundo” que se repite cada 0,5s) en la placa desde la plataforma Arduino IoT para comprobar que funciona correctamente (ver figura 23).

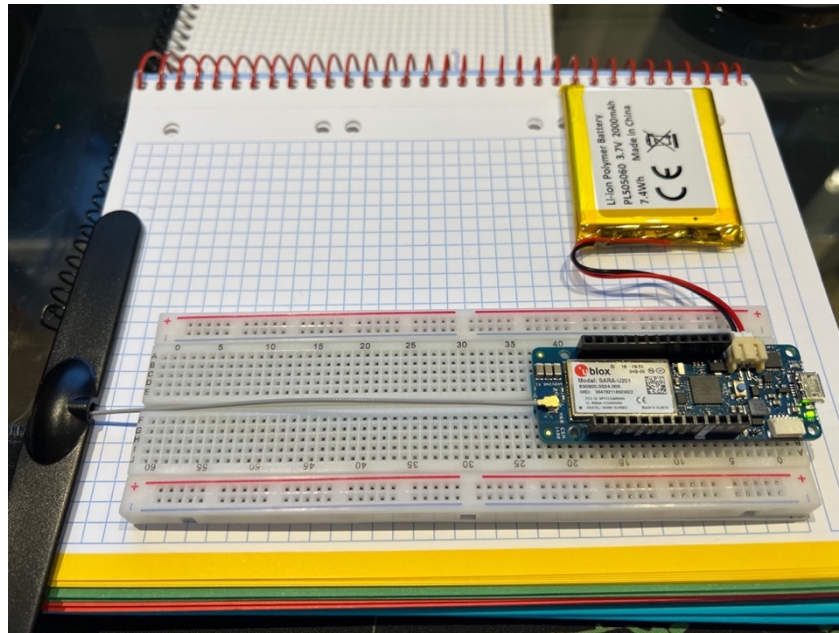


FIGURA 23 – CIRCUITO INICIAL ARDUINO MKR GSM 1400

5.1.2: CIRCUITO FINAL DEL MÓDULO DE SENSORES

Finalmente, se traslada el circuito desarrollado para cada sensor a la placa Arduino MKR GSM 1400, adaptando los pines necesarios para dar cabida a todos los elementos (ver figuras 24 y 25).

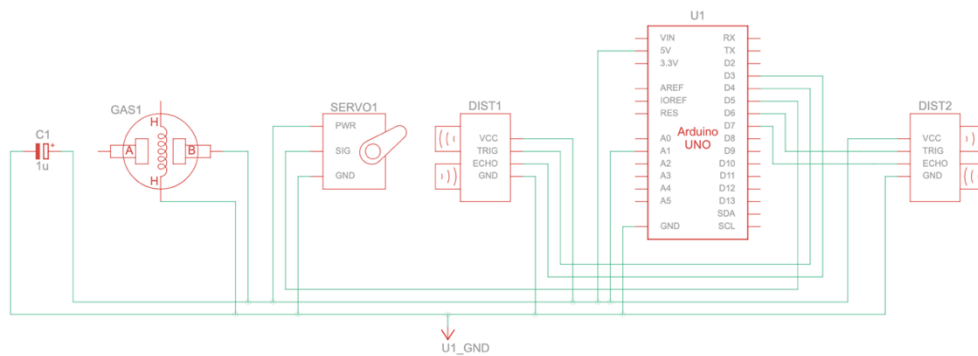


FIGURA 24 - CIRCUITO FINAL DEL MODULO DE SENSORES

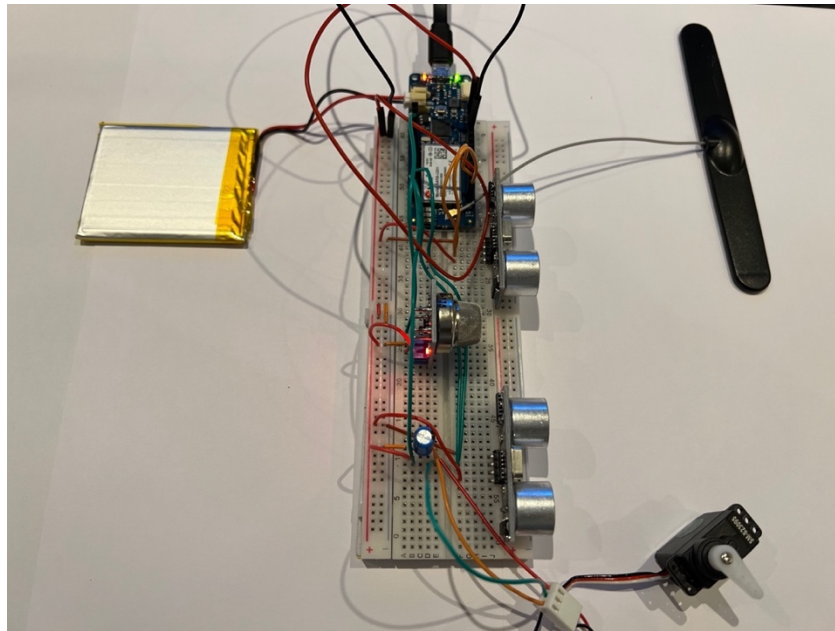


FIGURA 25 - CIRCUITO FINAL DEL MÓDULO DE SENSORES

5.2: SOFTWARE

5.2.1: CÓDIGO DE LA PLACA

A continuación, se muestra el código final de la placa. Contiene comentarios por cada bloque o línea de código significativa:

Código (versión final):

```
/*  
Sketch generated by the Arduino IoT Cloud Thing "Untitled"  
https://create.arduino.cc/cloud/things/7b9edcdf-20ff-4388-850e-11ceb72437c5  
  
Arduino IoT Cloud Variables description  
  
The following variables are automatically generated and updated when changes are made to the Thing  
  
int nivelCO2;  
bool vehiculoEntrando;  
bool vehiculoSaliendo;  
bool barreraAbierta;  
int vehiculosEnAparcamiento;  
  
Variables which are marked as READ/WRITE in the Cloud Thing will also have functions  
which are called when their values are changed from the Dashboard.  
These functions are generated with the Thing and added at the end of this sketch.  
*/  
  
#include "thingProperties.h"  
#include <Servo.h>
```

```

Servo barrera;

const int barreraPin = 5;
const int co2Pin = A1;
const int EchoPin1 = 3;
const int TriggerPin1 = 4;
const int EchoPin2 = 7;
const int TriggerPin2 = 6;

const int distanciaDeteccion = 15;
const int tiempoDeteccionDesactivada = 10000;

unsigned long deteccionDesactivadaHasta = 0;
bool posibleVehiculoEntrando = false;
bool posibleVehiculoSaliendo = false;

void setup() {
  // Conecta el servo al pin digital 5
  barrera.attach(barreraPin);

  // Configura los pins de trigger y echo de los dos sensores de ultrasonidos
  pinMode(TriggerPin1, OUTPUT);
  pinMode(EchoPin1, INPUT);
  pinMode(TriggerPin2, OUTPUT);
  pinMode(EchoPin2, INPUT);

  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
   * The following function allows you to obtain more information
   * related to the state of network and IoT Cloud connection and errors
   * the higher number the more granular information you'll get.
   * The default is 0 (only errors).
   * Maximum is 4
   */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();

  // CO2
  nivelCO2 = analogRead(co2Pin); // lee el valor que llega por la entrada analogica y lo asigna a la variable valorCO2
  Serial.print("CO2: ");
  Serial.print(nivelCO2);
  Serial.print(" PPM");
  if(nivelCO2 < 300) {
    Serial.println(" BAJO");
  } else if(nivelCO2 >= 300 && nivelCO2 < 600) {
    Serial.println(" MEDIO");
  } else if(nivelCO2 >= 600) {
    Serial.println(" ALTO");
  }
}

// DISTANCIA
// Mostramos ambas distancias por el Serial Monitor
int distancia1 = ping(TriggerPin1, EchoPin1);
Serial.print("Distancia 1: ");
Serial.print(distancia1);
Serial.println(" cm");
int distancia2 = ping(TriggerPin2, EchoPin2);
Serial.print("Distancia 2: ");
Serial.print(distancia2);
Serial.println(" cm");

// Cuando se ha detectado un vehiculo entrando o saliendo, se desactiva la deteccion unos

```

```

//segundos para evitar falsos positivos (doble deteccion del mismo vehiculo)
if(millis() > deteccionDesactivadaHasta){
  if(posibleVehiculoEntrando){
    if(distancia2 < distanciaDeteccion){
      // Si detectamos vehiculo en el sensor 2, confirmamos su entrada al aparcamiento
      Serial.println("Vehiculo entrando");
      posibleVehiculoEntrando = false;
      vehiculoEntrando = true;
      vehiculosEnAparcamiento++; // Contabilizamos el vehiculo
      // Deshabilitamos la deteccion durante unos segundos para evitar falsos positivos
      deteccionDesactivadaHasta = millis() + tiempoDeteccionDesactivada;
    }
  } else if(posibleVehiculoSaliendo){
    if(distancia1 < distanciaDeteccion){
      Serial.println("Vehiculo saliendo");
      posibleVehiculoSaliendo = false;
      vehiculoSaliendo = true;
      if(vehiculosEnAparcamiento > 0){
        vehiculosEnAparcamiento--; // Restamos el vehiculo
      }
      deteccionDesactivadaHasta = millis() + tiempoDeteccionDesactivada;
    }
  } else {
    if(distancia1 < distanciaDeteccion){
      // Si detectamos un vehiculo entrando (sensor 1), lo marcamos y esperamos a detectar en sensor2
      Serial.println("Posible vehiculo entrando");
      posibleVehiculoEntrando = true;
    } else if(distancia2 < distanciaDeteccion){
      // Si detectamos un vehiculo saliendo (sensor 2), lo marcamos y esperamos a detectar en sensor1
      Serial.println("Posible vehiculo saliendo");
      posibleVehiculoSaliendo = true;
    }
  }
} else {
  Serial.println("Deteccion desactivada");
  vehiculoEntrando = false;
  vehiculoSaliendo = false;
}
}

delay(500); // Insertamos un retardo para permitir que el salto en el servo sea pequeño
}

// Funcion para calcular la distancia en función de las señales trigger y echo
int ping(int TriggerPin, int EchoPin) {
  long duration, distanceCm;

  digitalWrite(TriggerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds(4);
  digitalWrite(TriggerPin, HIGH); //generamos Trigger (disparo) de 10us
  delayMicroseconds(10);
  digitalWrite(TriggerPin, LOW);

  duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos, en microsegundos

  distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
  return distanceCm;
}

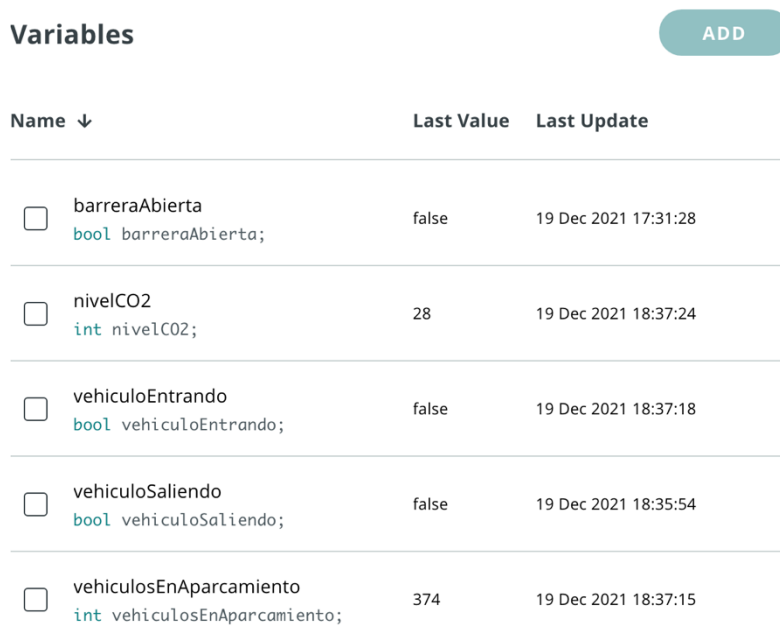
/*
  Since BarreraAbierta is READ_WRITE variable, onBarreraAbiertaChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onBarreraAbiertaChange() {
  // Add your code here to act upon BarreraAbierta change
  if(barreraAbierta) {
    Serial.println("Abriendo barrera");
    barrera.write(90); // Escribimos el ángulo de rotación en grados a efectuar en el servo
  } else {
    Serial.println("Cerrando barrera");
    barrera.write(170);
  }
}
}

```

5.2.2: LA PLATAFORMA ARDUINO IOT CLOUD

En la plataforma Arduino IoT Cloud el código se crea en forma de un fichero Sketch. Se crea un sketch para la placa MKR GSM 1400.

En este caso, se aprovechará también una de las características de Arduino IoT Cloud. En lugar de simplemente almacenar la información de los sensores en variables locales y mostrarlas por el monitor serie, se enviarán a variables de Arduino IoT (ver figura 26). Estas variables son una especie de *triggers* o disparadores que ejecutan cierto código cuando son actualizadas. Se define una de solo lectura para el nivel de CO₂, otra de lectura para controlar la entrada de un vehículo en el aparcamiento y otra de lectura también para controlar la salida de un vehículo del aparcamiento. Por último, se añade una de lectura/escritura para controlar el estado de la barrera de paso (abierta o cerrada) y una última variable para almacenar el total de vehículos estacionados en el aparcamiento. Toda esta información la proveerá el módulo a través de su conexión GPRS al servidor, donde se analizarán en conjunto con la de otros módulos y se integrarán en la lógica de negocio del sistema (propuesto como mejora).



Name ↓	Last Value	Last Update
<input type="checkbox"/> barreraAbierta <code>bool barreraAbierta;</code>	false	19 Dec 2021 17:31:28
<input type="checkbox"/> nivelCO2 <code>int nivelCO2;</code>	28	19 Dec 2021 18:37:24
<input type="checkbox"/> vehiculoEntrando <code>bool vehiculoEntrando;</code>	false	19 Dec 2021 18:37:18
<input type="checkbox"/> vehiculoSaliendo <code>bool vehiculoSaliendo;</code>	false	19 Dec 2021 18:35:54
<input type="checkbox"/> vehiculosEnAparcamiento <code>int vehiculosEnAparcamiento;</code>	374	19 Dec 2021 18:37:15

FIGURA 26 - VARIABLES DE ARDUINO IOT CLOUD CREADAS

A continuación, se añaden algunos widgets al Dashboard de Arduino IoT Cloud para visualizar y controlar de un modo sencillo e interactivo las variables anteriormente creadas en la nube (ver figura 27).

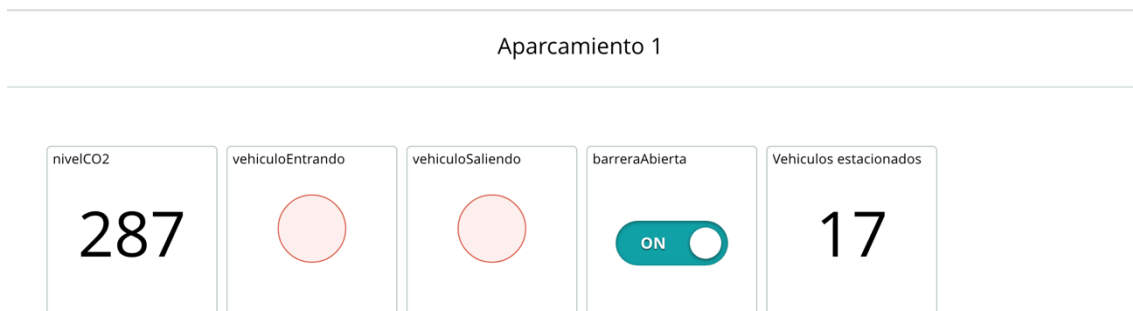


FIGURA 27 - WIDGETS DE ARDUINO IOT CLOUD

Finalmente, y siguiendo el espíritu ecologista del proyecto, se realizan dos cajas reutilizando sobres de cartón de paquetería. En ellas se instalan la placa desarrollada y el servo. Para completar el pequeño diorama representativo de la ubicación del módulo, se crea una plantilla simulando un aparcamiento. Se ubican sobre ella las dos cajas como se puede apreciar en la figura 28.

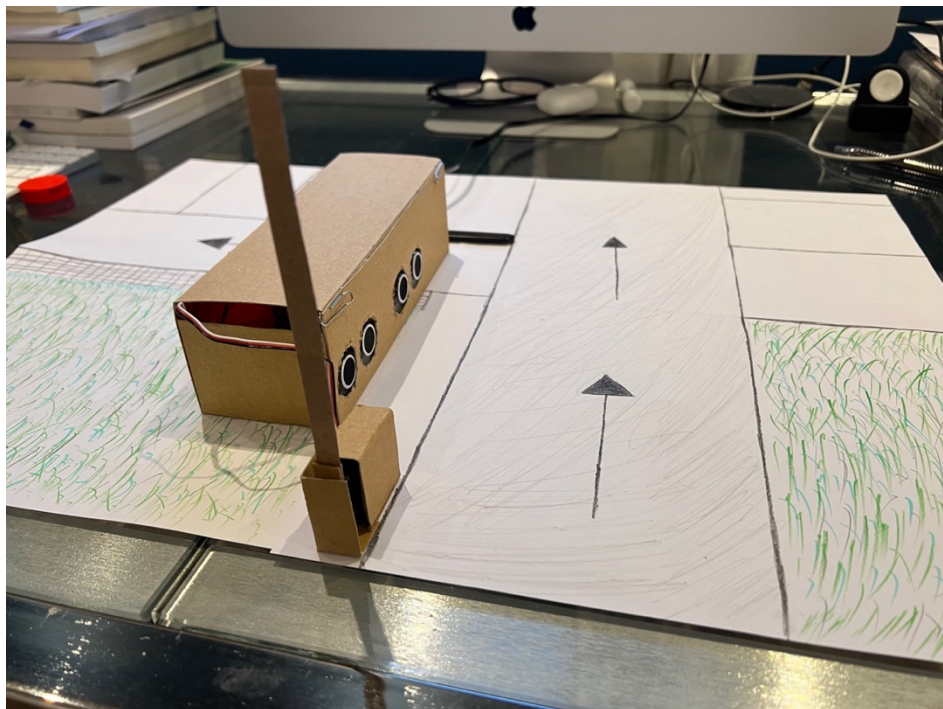


FIGURA 28 – PROTOTIPO FINAL

6. CONCLUSIONES

Tras completar todas las tareas necesarias para la ejecución del proyecto, se han investigado documentos y artículos sobre transporte, ecología, nuevas tendencias en sistemas de aparcamiento y sensores. Son campos tremendamente interesantes y actuales, que aportan una visión fresca y renovada sobre los problemas más acuciantes de nuestro entorno.

Es importante remarcar las enormes posibilidades que ofrece el universo Arduino y todas sus posibles expansiones o clones. Es un sistema que ha puesto en nuestras manos un modo sencillo de llevar a efecto proyectos que antes requerían de largos meses de desarrollo. Su aplicación en este caso ha permitido que el proyecto se centrara en sus objetivos sin perder el tiempo en elementos del diseño del circuito que ya están resueltos en Arduino.

No se han conseguido todos los objetivos propuestos en un principio. Hay diversos motivos. Probablemente el alcance del proyecto era demasiado ambicioso para pertenecer a un TFG y teniendo que compatibilizarlo con un trabajo en el día a día. Además, se encontró el problema de la escasez de algunos componentes necesarios para avanzar (como la placa MKR GSM o la tarjeta SIM). Estos problemas de suministro, derivados de los efectos de la pandemia ocasionada por el coronavirus, ocasionaron un retraso insalvable que limitó a posteriori los objetivos realizables.

Por los mismos motivos mencionados anteriormente, el seguimiento del proyecto según la metodología y diagrama de Gantt propuestos sufrió modificaciones y retrasos. Esto no ha supuesto sin embargo el fracaso del proyecto, ya que se pudieron adaptar los recursos y la dedicación en fases posteriores para cumplir varios objetivos del proyecto. Finalmente, y también importante, se consiguió obtener un producto físico final completo. El módulo de sensores.

Para finalizar, se plantean algunas posibles mejoras futuras del proyecto:

- Alimentación extra mediante placa solar.

- Implementación de las apps-clientes iOS y Android para mostrar a los usuarios finales la información sobre parámetros medioambientales, dónde aparcar, etc.
- Implementación de paneles de LEDs de exterior para mostrar la misma información en los principales puntos de acceso a la zona controlada por el sistema.
- Emplear sensores magnéticos profesionales para la detección y el conteo de vehículos entrantes y salientes en el aparcamiento.
- Emplear un sensor más completo que el MQ135, que solo percibe el nivel de CO2.
- El sensor de ultrasonidos HC-SR04 pierde precisión con el tiempo y necesita ser reemplazado. Sería necesaria una rutina de mantenimiento implementada en el servidor que avisara de algún modo sobre esta necesidad periódicamente.

7. GLOSARIO

Actuador: Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

Aparcamiento disuasorio: Estacionamientos para automóviles situados en la periferia de ciudades generalmente grandes, cuyo fin es alentar a los conductores a aparcar su vehículo privado y acceder al centro de las ciudades mediante el transporte público. Este tipo de aparcamientos suelen ubicarse próximos a estaciones de autobús o ferrocarril para facilitar el transbordo, constituyéndose en intercambiadores que fomentan la intermodalidad entre el transporte privado y el transporte colectivo. Por lo general suelen ser aparcamientos gratuitos.

API REST: Estilo de arquitectura *software* para sistemas hipermedia distribuidos como la World Wide Web. Se basa en operaciones POST, GET, PUT y DELETE.

Arduino: Plataforma open-source basada en un hardware y software fácil de usar. Las placas Arduino son capaces de leer entradas - la luz que incide en un sensor, un dedo pulsando un botón o un mensaje de Twitter – y convertirlas en salidas – activar un motor, encender un LED, publicar algo online. Puedes decirle a tu placa qué hacer enviándole un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo se utiliza el lenguaje de programación Arduino (basado en Wiring) y el Arduino Software (IDE), basado en Processing.

Cloud: La computación en la nube (del inglés *cloud computing*), conocida también como servicios en la nube, informática en la nube, nube de cómputo o simplemente «la nube», es el uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, servidores, bases de datos, redes y software. En lugar de depender de un servicio físico

instalado, se tiene acceso a una estructura donde el software y el hardware están virtualmente integrado.

Daily standup: Dentro de la metodología Scrum, reunión diaria en la que el equipo informa de su progreso para la consecución de los objetivos del sprint.

Descarbonización: proceso de reducción de emisiones de carbono, sobre todo de dióxido de carbono (CO₂), a la atmósfera. Su objetivo es lograr una economía global con bajas emisiones que consiga la neutralidad climática a través de la transición energética.

Experiencia piloto: Estudio preliminar a pequeña escala realizado para evaluar la viabilidad, duración, coste, adversidades, y mejorar el diseño de estudio antes del desarrollo de un proyecto de investigación a gran escala. Los estudios pilotos, por lo tanto, no pueden ser apropiados para los estudios de caso.

IOT: Internet de las cosas (en inglés, *Internet of things*, abreviado *IoT*; IdC, por sus siglas en español) es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet.

LoRaWAN: especificación para redes de baja potencia y área amplia, LPWAN (en inglés, *Low Power Wide Area Network*), diseñada específicamente para dispositivos de bajo consumo de alimentación, que operan en redes de alcance local, regional, nacionales o globales.

Microcontrolador: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Peatonalización: Conversión de una calle o un área solo para el uso de peatones.

PMR: Persona con movilidad reducida. Persona cuya movilidad está limitada a efectos de la utilización de un medio de transporte debido a cualquier discapacidad física (sensorial o de locomoción, permanente o temporal) o mental, a su edad o a cualquier otra causa que necesite una atención especial y la adaptación a sus necesidades de los servicios que se ponen a disposición de todos los pasajeros.

Raspberry Pi: Es una serie de ordenadores de placa reducida, ordenadores de placa única u ordenadores de placa simple (SBC) de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital.

Scrum: Scrum es un marco de trabajo que define un conjunto de eventos, prácticas y roles, y que puede tomarse como conjunto base para definir el proceso de producción que usará un equipo de trabajo o dentro de un proyecto.

Sensor: Objeto capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas.

Shield: Placas que añaden alguna función específica (como conectividad Wifi) y se conectan sobre la placa Arduino de un modo muy sencillo.

Sketch: Nombre de los ficheros de código fuente de Arduino.

Smart City: La ciudad inteligente, a veces también llamada ciudad eficiente o ciudad súper-eficiente, se refiere a un tipo de desarrollo urbano basado en la sostenibilidad que es capaz de responder adecuadamente a las necesidades básicas de instituciones, empresas, y de los propios habitantes, tanto en el plano económico, como en los aspectos operativos, sociales y ambientales.

Sprint: Es el núcleo central de la metodología de trabajo Scrum. Se trata de un miniproyecto de no más de un mes cuyo objetivo es conseguir un incremento de valor en el producto que estamos construyendo. Todo 'sprint' cuenta con una definición y una planificación que ayudará a lograr las metas marcadas.

8. BIBLIOGRAFÍA

- [1]. En bici, hacia ciudades sin malos humos. Serie Programa Marco Ambiental nº6, Abril 2002. Comisión Europea.
- [2]. Abdul Ahad, Zishan Khan, Syed Aqeel Ahmad. World Journal of Engineering and Technology.
https://www.researchgate.net/publication/301740727_Intelligent_Parking_System
- [3]. Elsevier. Faheem, S.A. Mahmud, G.M. Khan, M. Rahman, H. Zafar. A Survey of Intelligent Car Parking System. <https://www.elsevier.es/en-revista-journal-applied-research-technology-jart-81-articulo-a-survey-intelligent-car-parking-S1665642313715803>
- [4]. Libelium. IoT products. Smart Parking.
http://www.libelium.com/products/smart-parking/?utm_source=banner_home&utm_medium=banner
- [5]. Urbiotica. Paneles de Señalización Variable.
<https://www.urbiotica.com/producto/paneles-digitales-informativos/>
- [6]. Dinycon. Plazas libres de aparcamiento en superficie.
<https://www.dinycon.com/component/content/article/27-proyectos/39-paneles-informativos-de-plazas-libres-de-aparcamiento-en-superficie-en-santander>
- [7]. ParkHelp. <http://www.parkhelp.com/>
- [8]. Guía para la implantación de aparcamientos disuasorios en Andalucía. Programa de Sostenibilidad Urbana CIUDAD 21. Consejería de Medio Ambiente. Junta de Andalucía. 2010.

- [9]. Libelium. Smart Parking. https://marketplace.libelium.com/smart-parking?gclid=Cj0KCQjwlOmLBhCHARIsAGiJg7mnpBrW2N7v0Y7DiYhoP4_2TQbZJ07n94zMPxi2YqwaqXmguKj9N7saAnkrEALw_wcB
- [10]. Business Impact: Una red de sensores para buscar aparcamiento. BBVA. Julio 2018.
- [11]. Francesc Bracero. Sony Estrena en Roma su sensor fotográfico capaz de entender lo que ve. La Vanguardia. <https://www.lavanguardia.com/tecnologia/20210527/7484391/sony-estrena-roma-sensor-fotografico-capaz-entender-ve.html> Barcelona 27/05/2021.
- [12]. Parkam. <https://www.parkam.com/>
- [13]. Richard Backhaus. Red 5G para coches inteligentes. Porsche Engineering Magazine, número 01/2019. https://newsroom.porsche.com/es_ES/tecnologia/2019/es-porsche-engineering-coches-inteligentes-red-5g-19510.html
- [14]. Urbiotica. Estacionamiento inteligente. <https://www.urbiotica.com/category/aparcamiento-inteligente-es/>
- [15]. Donald C. Shoup. Cruising for parking. 2006.
- [16]. Apcoa Parking Study: Search for parking space devours time and money <https://www.apcoa.com/press/press-releases/article/apcoa-parking-study-search-for-parking-space-devours-time-and-money/>. Stuttgart, 15/08/2013.
- [17]. Instituto de Estadística y Cartografía de Andalucía. Andalucía pueblo a pueblo. Fichas Municipales – Puerto Real. 2020.
- [18]. Programa de ayudas a municipios para la implantación de zonas de bajas emisiones y la transformación digital y sostenible del transporte urbano. Ministerio de Transportes, Movilidad y Agenda Urbana. 2021.
- [19]. Puerto Real + Ciudad: <http://www.masciudad.es/wp/>. 2020.
- [20]. Arduino. What is Arduino?. <https://www.arduino.cc/en/Guide/Introduction> 05/02/2018.
- [21]. Aprendiendo Arduino. Entornos de Aplicación Arduino. <https://aprendiendoarduino.wordpress.com/2016/06/26/entornos-de-aplicacion-arduino/>
- [22]. Arduino. Arduino Products. <https://www.arduino.cc/en/Main/Products>

- [23].Arduino. Software. <https://www.arduino.cc/en/software>
- [24].Arduino Store. Arduino UNO Rev3.
<https://store.arduino.cc/products/arduino-uno-rev3/>
- [25].Luis del Valle Hernández. Programarfácil. Curso de Arduino aprende a programar desde cero. [https://programarfacil.com/blog/arduino-blog/curso-de-arduino/#Mini-curso de programacion con Arduino GRATIS](https://programarfacil.com/blog/arduino-blog/curso-de-arduino/#Mini-curso-de-programacion-con-Arduino-GRATIS)
- [26].Arduino Store. SIM808 GPS/GPRS/GSM Shield.
<https://store.arduino.cc/products/sim808-gps-gprs-gsm-shield?queryID=undefined>
- [27].Arduino Docs. Arduino MKR GSM 1400.
<https://docs.arduino.cc/hardware/mkr-gsm-1400#use-your-arduino-mkr-gsm-1400-with-the-arduino-iot-cloud>
- [28].Digikey electronics. Sensor de CO2 SprintIR-W.
<https://www.digikey.es/es/product-highlight/g/gas-sensing-solutions/sprintir-w-co2-sensor> 12/06/2019.
- [29].Vaisala. CO2 Probe GMP252.
<https://www.vaisala.com/sites/default/files/documents/GMP252-Datasheet-B211567EN.pdf>
- [30].Sensorion. SCD4x Breaking the barrier in CO2 sensing.
https://www.mouser.es/datasheet/2/682/Sensirion_CO2_Sensors_SCD4x_Datasheet-2321195.pdf Abril 2021.
- [31].Eduardo Santos. Medidor de CO2 con Arduino.
<https://medidordeco2.info/medidor-de-co2-arduino/> 03/05/2021.
- [32].PCE Iberica. Definición de la calidad del aire y el CO2. <https://www.pce-iberica.es/medidor-detalles-tecnicos/definicion-calidad-aire-y-co2.htm>
- [33].ParkHelp. Outdoor Parking Sensors. <https://www.parkhelp.com/parking-guidance-technologies/outdoor-parking-sensors/>
- [34].Urbiotica. U-Spot – sensor de parking magnético.
<https://www.urbiotica.com/producto/u-spot/>
- [35].Lluis Llamas. Medir distancia con Arduino y sensor de ultrasonidos HC-SR04. <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/> 16/06/2015.

[36]. Karl Söderby. Getting Started With the Arduino IoT Cloud.

<https://docs.arduino.cc/cloud/iot-cloud/tutorials/iot-cloud-getting-started>

13/01/2022.