



UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Hack for **Alba**CITY

OPTIMIZACIÓN Y GESTIÓN DEL APARCAMIENTO EN EL CENTRO DE ALBACETE

Autores:

Hernan Indíbil de la Cruz Calvo.

Alejandro Moya Moya.

Jorge Valero Molina.

Alejandro Zornoza Martínez.

Abril, 2018



Índice de contenido

1. INTRODUCCIÓN	1
2. PLANTEAMIENTO DE LA SOLUCIÓN	2
2.1 OBJETIVOS	2
2.2 FUNDAMENTOS	3
2.3 POSIBLES LIMITACIONES	4
2.4 RESULTADOS ESPERADOS	5
2.5 TRABAJO FUTURO	5
3. DISEÑO DE LA SOLUCIÓN	7
3.1 ARQUITECTURA	7
3.2 DISEÑO DE PLAZA DE ESTACIONAMIENTO ESTÁNDAR	8
3.3 MEDIOS SOFTWARE	8
3.3.1 NODE-RED	8
3.3.2 BLUEMIX	10
3.3.3 CLOUDANT	10
3.3.4 NODE.JS	11
3.3.5 APACHE CORDOVA	11
3.4 MEDIOS HARDWARE	13
3.4.1 SENSORES	13
3.4.2 ELEMENTOS DE CONECTIVIDAD	15
4. EXPERIENCIA DE USUARIO	18
4.1 PRINCIPIOS DE DISEÑO	18
4.1.1 INTERFAZ DE LA APLICACIÓN	18
4.2 SOLUCIONES DE ACCESIBILIDAD	23
5. CONCLUSIONES	24
6. REFERENCIAS	25
ANEXOS	26
ANEXO I ENLACES DE CONTENIDO	26

1. INTRODUCCIÓN

En la actualidad los avances médicos y la mejora en la calidad de vida en los núcleos de las ciudades ha provocado un incremento notable del número de vehículos por unidad familiar. Puesto que las zonas céntricas de las ciudades debido a su construcción y disposición no pueden ser redefinidas, surge la necesidad de aportar soluciones que reduzcan el tiempo y número de vehículos en circulación en las zonas de mayor interés.

Estas zonas debido a los intereses culturales, laborales o de ocio de los particulares, son lugar especial de visita por los habitantes y visitantes de las ciudades. Por lo tanto, es normal que se experimente en determinadas horas y momentos del año un incremento notable en el uso de vehículos.

Nuestra línea de proyecto se centra principalmente en favorecer la gestión del aparcamiento en la zona centro de Albacete, aportando así una solución que permita de manera directa facilitar la localización y el proceso de aparcamiento. Así se reducen el número de vehículos en la zona centro y el tiempo en el que los mismos circulan buscando aparcamiento.

Puesto que el hecho de encontrar una solución para toda la población es algo complicado y la necesidad de redefinir los aparcamientos de la zona centro sin apenas pruebas de campo no es viable, nuestra propuesta inicialmente se centrará en un público más reducido, y ya que nuestro objetivo es favorecer la movilidad se quiere comenzar ideando la solución sobre las plazas destinadas para las personas con discapacidad.

El avance de la tecnologías de la información y la introducción de las mismas en las ciudades han dado a las Smart Cities un gran abanico de soluciones en muy diversas áreas. Es por esto por lo que consideramos que es un gran momento para que la ciudad de Albacete se introduzca entre las ciudades que dan un uso óptimo a la tecnología y con nuestra propuesta destacar en el panorama español de las Smarts Cities.

2. PLANTEAMIENTO DE LA SOLUCIÓN

2.1 OBJETIVOS

OBJ-1	<i>Gestionar de Manera Óptima el Aparcamiento en el Centro de Albacete. (1)</i>
Descripción	Diseñar un sistema que de manera sencilla permita a los usuarios satisfacer sus necesidades de aparcamiento.
Sub-objetivos	1.1. Gestionar de Manera Óptima el Aparcamiento en el Centro de Albacete para Personas con Discapacidad. 1.2 Georreferenciación de Aparcamientos del Centro de Albacete. 1.3 Estudio de la Casuística de los Diferentes Sensores y su Aplicación. 1.4 Conectividad de los Sensores y Otros Elementos.

Cuadro 1: OBJ-1. Gestionar de Manera Óptima el Aparcamiento en el Centro de Albacete.

OBJ-1.1	<i>Gestionar de Manera Óptima el Aparcamiento en el Centro de Albacete para Personas con Discapacidad. (2)</i>
Descripción	Inicialmente la solución se diseñará para aquellas personas con mayor necesidad de favorecer su movilidad.

Cuadro 2: OBJ-1.1. Gestionar de Manera Óptima el Aparcamiento en el Centro de Albacete para Personas con Discapacidad.

OBJ-1.2	<i>Georreferenciación de Aparcamientos del Centro de Albacete. (3)</i>
Descripción	Es necesario situar en el mapa, garantizando la precisión, la localización de los diferentes aparcamientos de los que se dispone.

Cuadro 3: OBJ-1.2. Georreferenciación de Aparcamientos del Centro de Albacete.

OBJ-1.3	<i>Estudio de la Casuística de los Diferentes Sensores y su Aplicación. (4)</i>
Descripción	Se deben seleccionar los sensores más adecuados, atendiendo a su precio y por supuesto, a sus ventajas de uso.

Cuadro 4: OBJ-1.3. Estudio de la Casuística de los Diferentes Sensores y su Aplicación.

OBJ-1.4	<i>Conectividad de los Sensores y Otros Elementos. (5)</i>
Descripción	Se debe aportar una solución para establecer la conectividad de los sensores de cada plaza de aparcamiento, con los servidores y servicios correspondientes, además del almacenamiento de información relevante.

Cuadro 5: OBJ-1.4. Conectividad de los Sensores y Otros Elementos.

2.2 FUNDAMENTOS

Supongamos una plaza de garaje perfectamente delimitada, la cuál cuenta con una serie de sensores y una conexión externa. Estos sensores aportan la información necesaria sobre si un vehículo se encuentra estacionado en la plaza o no. Mediante la conectividad externa se logra extraer dicha información permitiendo su tratamiento para así establecer conclusiones determinantes sobre el uso de la plaza y el futuro uso, en caso de quedar libre por parte de otros usuarios.

La problemática se sitúa en el centro de una ciudad, concurrida en tiempo y espacio, donde el uso de vehículos es constante e incremental. Surge obviamente una necesidad de gestionar la distribución de estos vehículos, para evitar desplazamientos innecesarios y reducir el tiempo medio de circulación de los vehículos a la hora de estacionar y de sus aspectos negativos de contaminación.

La solución que se propone reside en estos conceptos, el usuario mediante el uso de una aplicación podrá conocer en cualquier momento las plazas libres de la zona centro de Albacete, consiguiendo así reducir el tiempo que este tarda en buscar aparcamiento y obteniendo así el camino óptimo desde el lugar de la búsqueda hasta el lugar del estacionamiento. Esto permitirá eliminar el número de “vueltas” que el usuario realiza a la hora de aparcar con sus correspondientes beneficios tanto para el mismo como para el resto de ciudadanos.

También se considerará la posibilidad de que el usuario consulte antes de visitar algún lugar el estado de la saturación de la zona centro, permitiéndole observar el número de plazas disponibles y su distancia al lugar deseado. Este aspecto de la solución es de especial interés para ciudadanos no residentes en Albacete ya que desconocen la distribución de dichas plazas de aparcamiento.

Inicialmente esta propuesta se aplicará a las plazas de aparcamiento para personas con discapacidad, permitiendo un estudio sobre un reducido subconjunto de usuarios potenciales y así aportándole a estas personas el beneficio que necesitan. Este estudio proporcionará estadísticas de uso que permitirá extraer conclusiones determinantes sobre el uso que tiene la solución propuesta, para así poder ampliarla a todos los usuarios potenciales, es decir, los ciudadanos de Albacete.

En caso de extraer conclusiones no exitosas en el uso del sistema, se habrá aportado a las personas con discapacidad una herramienta muy útil pero que no se ha podido extrapolar al centro de Albacete.

Por tanto los beneficios que la solución propuesta en este documento aporta son:

- Reducción del tiempo medio de circulación para la búsqueda de aparcamiento.
- Reducción de la contaminación.
- Reducción del índice de siniestralidad.
- Reducción de la carga mental de los usuarios a la hora de encontrar aparcamiento.
- Aumento del posible turismo en la zona centro.
- Incremento de la imagen innovadora de la ciudad.
- La efectividad y éxito de la aplicación es independiente del uso completo de la misma por parte de toda la población.

2.3 POSIBLES LIMITACIONES

Durante el diseño de la solución se han encontrado estas limitaciones principales:

LIM-1	<i>Relativa al coste para la instalación en un determinado número de plazas.</i>
Causa	<i>Principalmente el coste de los elementos hardware o conectivos necesarios.</i>
Solución	Se ha realizado una búsqueda exhaustiva de los diferentes elementos para así proporcionar un presupuesto estimado.

Cuadro 6: LIM-1. Relativa al coste para la instalación en un determinado número de plazas.

LIM-2	<i>Relativa al tiempo medio de vida útil de los elementos hardware.</i>
Causa	<i>Los elementos hardware tienen un tiempo medio de vida útil, estimado entre 7 y 10 años.</i>
Solución	Se hará uso de sensores de fácil reemplazo y low batteries para evitar su excesivo mantenimiento.

Cuadro 7: LIM-2. Relativa al tiempo medio de vida útil de los elementos hardware.

LIM-3	<i>Adopción por parte de los usuarios.</i>
Causa	<i>El hecho de diseñar una solución no implica que sea la correcta para todos los usuarios.</i>
Solución	Los beneficios deben de ser latentes y visibles para los ciudadanos.

Cuadro 8: LIM-3. Adopción por parte de los usuarios.

LIM-4	<i>Daños sufridos por los sensores, intencionados o no.</i>
Causa	<i>Los sensores pueden ser vulnerables ante condiciones climatológicas o daños intencionados.</i>
Solución	Los sensores deben disponer de un soporte especial de instalación y una estructura adecuada ante fuerzas o inclemencias de otro tipo.

Cuadro 9: LIM-4. Daños sufridos por los sensores, intencionados o no.

LIM-5	<i>Reaparición y desaparición constante de las plazas.</i>
Causa	<i>Los ciudadanos pueden estacionar en cualquier momento, ocupando y desocupando las plazas.</i>
Solución	La actualización de las plazas se realizará en tiempo real y se establecerá un tiempo medio para determinar si un vehículo se encuentra estacionado.

Cuadro 10: LIM-5. Reaparición y desaparición constante de las plazas.

LIM-6	<i>Sobrecarga y exceso de peticiones sobre el servidor.</i>
Causa	<i>El uso de la aplicación puede ser alto y la respuesta en tiempo real puede incrementar el coste.</i>
Solución	La arquitectura que se proporcionará garantizará la escalabilidad y eficiencia del sistema a desarrollar.

Cuadro 11: LIM-6. Sobrecarga y exceso de peticiones sobre el servidor.

LIM-7	<i>La legislación de tráfico no permite el uso de teléfonos móviles durante la conducción.</i>
Causa	<i>Uno de los modos de uso de la aplicación se utiliza durante la realización de la conducción.</i>
Solución	La interfaz debe ser lo más simple posible y disponer de los medios para realizarse en modo de manos libres.

Cuadro 12: LIM-7. La legislación de tráfico no permite el uso de teléfonos móviles durante la conducción.

2.4 RESULTADOS ESPERADOS

Obviamente el éxito de la aplicación reside del uso que se haga de esta y la adopción que esta tenga por parte de los usuarios(para este segundo bastará con obtener las estadísticas relativas de número de descargas, inicios de sesión y nuevos registros). Por tanto es necesario diseñar un MVP(mínimo producto viable) que permita corroborar que esta solución sigue la línea correcta. Una estadística que permite observar si realmente el sistema cumple las expectativas es el cálculo del tiempo transcurrido desde el inicio de la aplicación hasta que el usuario aparca. Sí el periodo de aparcamiento supera el estimado se trataría de un caso negativo.

Puesto que el tiempo medio estimado no es una medición arbitraria y es necesario obtenerla con un cierto nivel de precisión. Se propone como solución al respecto la realización de una prueba de campo que permita en diferentes horarios y ante diferentes flujos de tráfico, estimar dicho periodo medio de aparcamiento que será variable atendiendo a la franja horaria o fecha (esto permitirá ser previsores ante posibles eventos que saturen el tráfico).

Ya que se dispondrá del estado de los sensores y la ubicación del usuario podemos aproximar el instante en el que un usuario ha aparcado determinando así su tiempo real de aparcamiento.

Para los casos desfavorables:

$$\text{Tiempo Real de Aparcamiento} - \text{Tiempo Medio Estimado} < 0 \quad (1)$$

Para los casos favorables:

$$\text{Tiempo Real de Aparcamiento} - \text{Tiempo Medio Estimado} \geq 0 \quad (2)$$

Obviamente estos valores se recogerán de manera acumulativa para facilitar su análisis en periodos de tiempo, como pueden ser de carácter mensual, trimestral o anual. El servidor recogerá un histórico de la base de datos de información relevante, a la cuál se facilitara su acceso a traves de una interfaz, para que el análisis sea cómodo, reproducible y no se requieran conocimientos adicionales para su obtención (por ejemplo por parte de un empleado de un ayuntamiento).

2.5 TRABAJO FUTURO

Tal y como se ha comentado en el OBJ-1, sería posible extrapolar la idea de la gestión de las plazas de aparcamiento para cualquier plaza del centro de Albacete o para toda la ciudad.

La idea consistiría en realizar un despliegue de un número de sensores a lo largo de las zonas de aparcamiento, estos sensores se encontrarán a una determinada distancia establecida. Tras esto, sería posible estimar cuantos de estos sensores son necesarios ocupar para aparcar el vehículo del usuario, indicando, por parte del usuario, la longitud de su vehículo.

Si se implementase, la aplicación debería de ser modificada, recogiendo información relativa del vehículo del usuario. Aún así no podríamos poner puntos de aparcamiento para todas las plazas, ya que el volumen de información almacenado sería mucho mayor. Es por ello que se podría incluir en los mapas líneas de color indicando los huecos disponibles (verde), ocupados (rojo) y posibles aparcamientos (amarillo) que hacen referencia donde el espacio disponible es escaso o la maniobra sea dificultosa, tal y como se ve en la siguiente imagen.

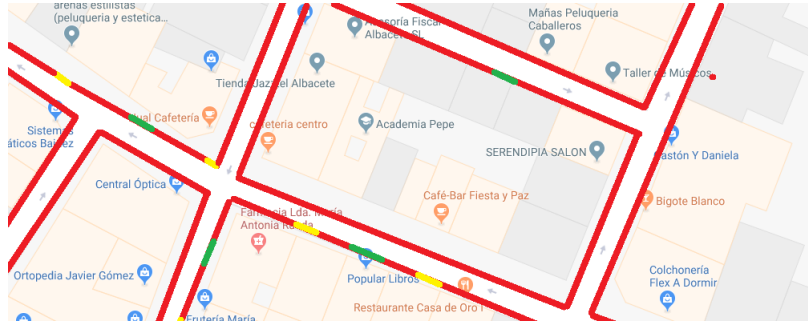


Figura 1: Imagen futura extrapolación de la solución sobre una porción de Albacete.

Al usuario se le indicará los posibles sitios disponibles para poder aparcar y él seleccionará el que prefiera, tal y como se ha planteado en la solución inicial.

Aquí se muestra un ejemplo para facilitar el entendimiento del sistema de colores planteado.

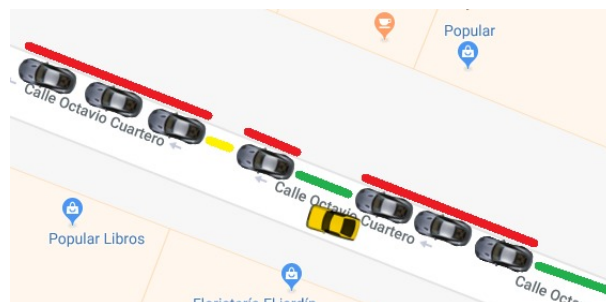


Figura 2: Imagen ejemplo de uso de la aplicación.

Donde,

- **Verde:** Representa el espacio disponible para aparcar.
- **Amarillo:** Representa el espacio en el cual es posible no aparcar (espacio reducido).
- **Rojo:** Espacio ocupado por otro vehículo.

Obviamente extrapolar esta solución a toda una ciudad incrementa notablemente la complejidad del problema, en términos de:

- Incremento del número de plazas a almacenar en la base de datos.
- Incremento notable del número de sensores a gestionar.
- Incremento del número de usuarios de la aplicación, por tanto el número de peticiones y respuestas del servidor incrementa.

Pero esta problemática tiene fácil solución, ya que la arquitectura definida es plenamente redefinible e incremental como el lector puede observar en la siguiente sección.

3. DISEÑO DE LA SOLUCIÓN

3.1 ARQUITECTURA

La arquitectura del sistema a desarrollar presenta el siguiente esquema:

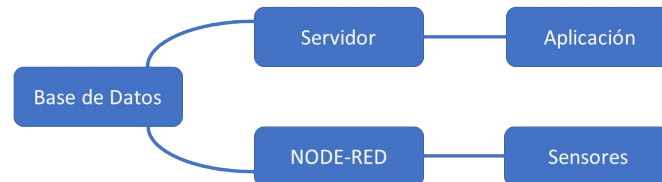


Figura 3: Esquema de la arquitectura del sistema

Como se observa en la imagen los elementos de nuestra arquitectura son:

- **Sensores:** Elementos hardware que captaran la información del entorno y que enviarán al entorno Node-RED para su procesamiento. En nuestra primera aproximación se realizará una simulación de los mismos en este mismo entorno.
- **Node-RED:** Establece el flujo de información entre la base de datos del sistema y los sensores y su correspondiente procesamiento.
- **Aplicación:** Establece conexión mediante un web-view con el servidor de la aplicación. Funciona como interfaz de comunicación con el usuario.
- **Servidor:** Es el alojamiento de la web-app y el encargado de gestionar y extraer la información de la base de datos para proporcionarla al usuario. También proporcionará datos relevantes para determinar conclusiones sobre el sistema, como por ejemplo, tiempo medio de estacionamiento, tiempo de uso de las plazas y los elementos correspondientes a la extracción de la información.
- **Base de Datos:** Almacena la información correspondiente a sensores y plazas, además de un histórico de información relevante. Elemento de unión entre la web-app y entorno Node-Red.

No existe ninguna ventaja latente en el uso de esta arquitectura, pero al no emplear una aplicación nativa y que esta no extraiga la información directamente de la base de datos permitimos que el sistema sea más escalable.

Si en un futuro se decide aumentar el número de clientes conectados al servidor no sería necesario rediseñar de nuevo la arquitectura para dar servicio a los mismos, simplemente de manera sencilla se debe establecer los medios de conectividad necesarios para estos nuevos clientes.

Mediante una capa de servicios de REST, se logrará disponer de diversos clientes heterogéneos conectados al servidor, como puede ser una aplicación Android, una aplicación de IOS o un cliente web.

Un ejemplo de posible beneficio podría ser una interfaz web que proporcionara al ayuntamiento la información relevante de uso o conclusiones obtenidas de análisis de grandes volúmenes de datos como mencionamos anteriormente.

Otra de las ventajas que esta arquitectura proporciona es relativa al coste de diseño, puesto que no es necesario realizar aplicaciones nativas, tanto para Android o IOS. Esto permite a las PYMES (Pequeñas y medianas empresas) realizar soluciones multiplataforma sin necesidad de tener que distinguir en el desarrollo de las aplicaciones, centrandose en la solución, ahorrando en tiempo y coste.

3.2 DISEÑO DE PLAZA DE ESTACIONAMIENTO ESTÁNDAR

Para representar de manera gráfica el concepto de plaza de estacionamiento que se propone, se aporta la siguiente imagen:



Figura 4: Imagen ejemplo plaza de estacionamiento

Como se puede observar cada plaza delimitada dispone de un único sensor, de tipo magnético, que funciona a modo de informador sobre si un vehículo ocupa la plaza o no. Este sensor tendrá definido un periodo de tiempo continuo, tras el cual el vehículo se da por estacionado. El valor de este periodo se establecerá siguiendo las directrices definidas por tráfico, este periodo será de 2 minutos, ya que legalmente se considera estacionamiento cuando un vehículo se encuentra inmóvil dicho tiempo. El sensor dara la plaza como ocupada una vez transcurrido dicho periodo de tiempo.

3.3 MEDIOS SOFTWARE

3.3.1 NODE-RED

Con Node-RED se realiza la gestión de:

1. Recoger la información de los sensores.
2. Tratar la información recibida de los sensores.
3. Almacenar estado de los sensores en cloudant.
4. Generar datos sobre estacionamientos (sensor y duración).

Consta de dos partes:

- **Recogida de información de sensores y tratamiento:**

Mediante peticiones get a los sensores se obtiene la información referente a si detectan o no un coche. Los sensores actualmente son simulados, pero es sencillo mediante un simple tratamiento convertir lo que los sensores devuelvan en lo que se requiere, que consiste en una afirmación o negación referente a si la plaza está ocupada o no. Actualmente en el tratamiento se añade la marca de tiempo (timestamp) del momento en el que la respuesta del sensor comienza a ser tratada.

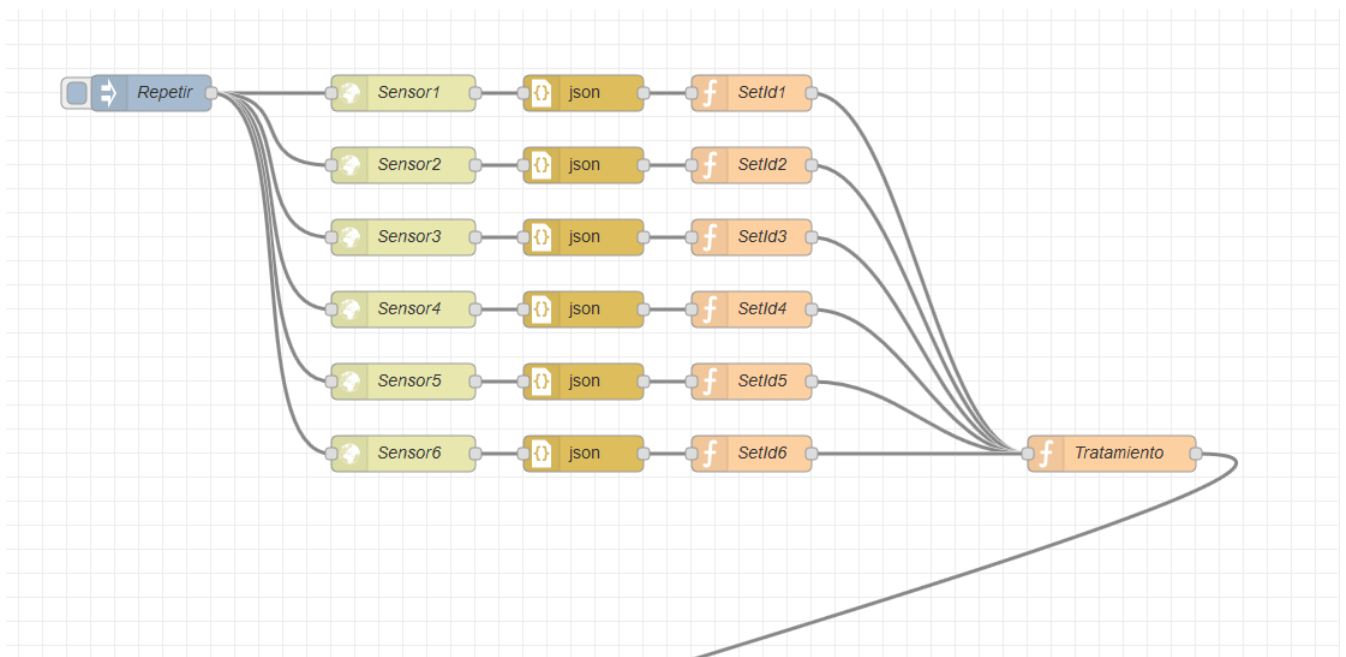


Figura 5: Imagen Node-RED disposición de sensores.

■ Generación de información y almacenamiento:

En esta etapa se realizan consultas a la base de datos para comprobar si el estado del sensor ha cambiado o no, de forma que solo se actualizara cuando haya pasado de no detectar un vehículo a detectarlo o viceversa, es decir, a modo de actuador. Así se dispone de la marca temporal (timestamp) en la que el vehículo estacionó/se marchó. Usando dicho tiempo, sólo para el caso de las salidas de vehículos, se puede obtener cuánto tiempo ha estado estacionado un vehículo ocupando cierto sensor (posición). Así se generan datos sobre estacionamientos y se actualiza la base de datos solo cuando es necesario, y no se hace nada en cualquier otro caso, reduciendo así el flujo de información o la repetición de la isma.

En esta etapa se realiza también la inclusión de información de cualquier sensor que no se encontrase previamente almacenado en la base de datos. Por ello con añadir un sensor a la parte anterior de recogida de información, esta parte se encarga de crearle una entrada nueva en la base de datos.

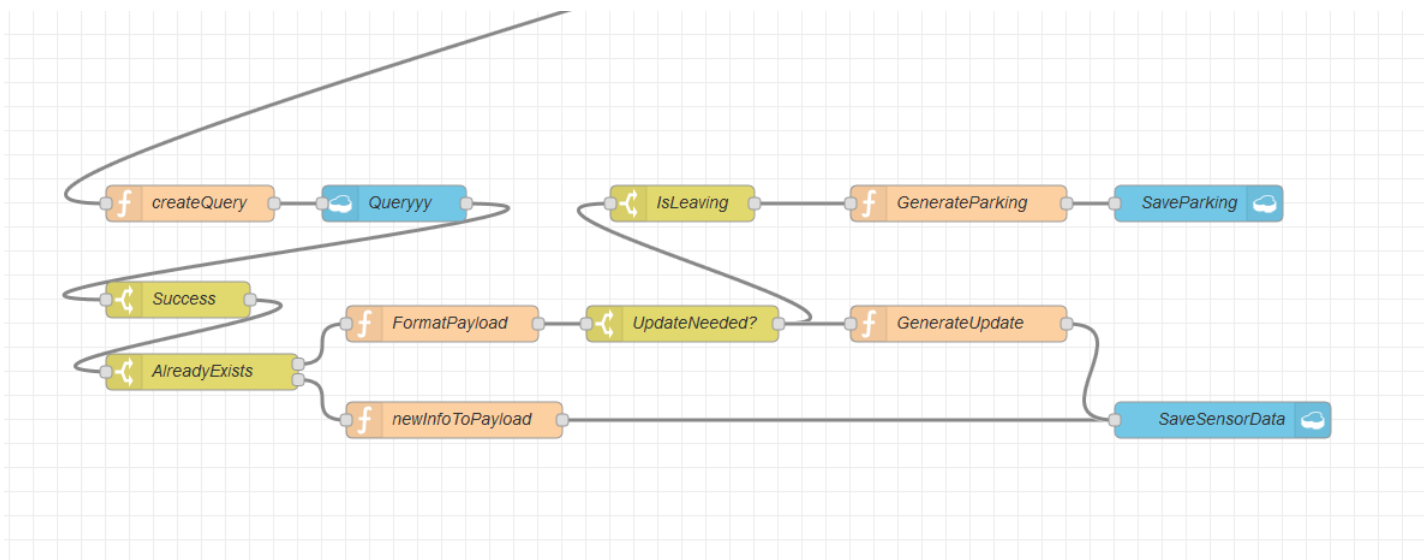


Figura 6: Imagen Node-RED recogida de información.

3.3.2 BLUEMIX

IBM Bluemix es un entorno PAAS (plataforma como un servicio) ofrecido por IBM, que permite el desarrollo de aplicaciones en diversos lenguajes, soportando Node.js, Python y Go entre otros. Con esto, es posible crear, ejecutar, desplegar y gestionar aplicaciones en la nube. Está basado en la tecnología libre Cloud Foundry.

El coste varía en función de lo que se consuma, pudiendo variar el precio según las necesidades. Por lo tanto, es fácilmente escalable.

3.3.3 CLOUDANT

A la hora de seleccionar el sistema de gestión de base de datos se opta por cloudant al tratarse de una base de datos NoSQL, lo cual aporta ventajas en cuanto a la escalabilidad (no requieren de un gran servidor sino que pueden ser distribuidas) y optimización de consultas sobre grandes volúmenes de datos entre otras. Cloudant es además gratuito para comunidades académicas y viene incluido en ciertos paquetes con herramientas software que también se requieren para el diseño de la solución, como Node.js.

La base de datos no sigue el modelo entidad relación, pero se ha definido una estructura para facilitar el uso de la información:

Sensores:

- **Id:** Identificador único de un sensor.
- **Lat:** Latitud del sensor.
- **Lon:** Longitud del sensor.
- **LastChange:** Marca de tiempo (timestamp) del momento en el que el sensor cambió de estado por última vez.
- **Data:** De tipo si/no. Indica si se detecta un vehículo o no.

Estacionamientos:

- **SensorId:** Identificador del sensor que ha detectado el estacionamiento.
- **Start:** Marca de tiempo del momento en el que se inició el estacionamiento.
- **End:** Marca de tiempo del momento en el que el vehículo dejó de estar estacionado.

3.3.4 NODE.JS

Para desarrollar la aplicación se ha optado por utilizar node.js como servidor web, ya que permite adaptarla a un gran número de dispositivos (responsive). Node se encarga de proporcionar los servicios a quien realice una petición.

Dicha página utiliza html y javascript para la generación del contenido, en este caso un mapa en el que se muestran:

- Posición del usuario (azul).
- Aparcamientos ocupados (rojo).
- Aparcamientos libres (verde).
- Destino (cruce).
- Ruta que une posición del usuario → aparcamiento libre más cercano al destino → destino.

Para utilizar mapas y la generación de rutas se utiliza ArcGIS, de ESRI. Se ha de considerar el precio por utilizar la capa de generación de rutas de ArcGIS.

Mediante la utilización de capas de gráficos a la hora de añadir elementos sobre los mapas, se muestran aparcamientos libres sobre ocupados, y todos ellos sobre la ruta y el punto de destino.

3.3.5 APACHE CORDOVA

Apache Cordova es el entorno de desarrollo que se emplea para construir la aplicación.

Este entorno permite construir aplicaciones utilizando CSS3, HTML5 y JavaScript en lugar de utilizar APIs específicas para cada plataforma (Android, iOS o Windows Phone). Es decir, permite extender las características del HTML y JavaScript para trabajar con diferentes plataformas.

Como resultado se obtienen aplicaciones híbridas, esto significa que la representación gráfica se realiza vía vistas de web, sin embargo, no son puramente web ya que se empaquetan como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo.

Hay una mezcla de código nativo e híbrido desde la versión 1.9, y esto permite un desarrollo más potente.

¿Por qué usar Apache Cordova?:

1. Es gratuito.
2. Código abierto.
3. Desarrollo multiplataforma (Android, iOS, Windows Phone, etc.).
4. Mezcla de código nativo e híbrido, lo cual proporciona una mayor potencia en el desarrollo de aplicaciones.

5. Como framework de desarrollo utilizamos Ionic Framework 3, el cual integra Apache Cordova como base. Se trata de una herramienta gratuita y de código abierto, que permite desarrollar aplicaciones híbridas en HTML5, CSS y JavaScript.



Figura 7: Imagen Apache Cordova

¿Por qué usar Ionic Framework?

1. Nos permite disponer de todas las ventajas proporcionadas por Apache Cordova, ya que se apoya sobre él.
2. Dispone de una arquitectura robusta y seria para el desarrollo de aplicaciones.
3. Está inspirado en las SDK de desarrollo de aplicaciones nativas, por lo que es fácil de aprender para la gente que ha construido alguna aplicación nativa.
4. Dispone de un diseño limpio, sencillo y funcional que permite la compatibilidad con cualquier dispositivo móvil actual.
5. Dispone de una potente interfaz de línea de comandos que te permite crear, construir, probar y compilar tus aplicaciones en cualquier plataforma (Dispositivo físico, emulador, navegador, etc.).



Figura 8: Imagen Ionic Framework

Es importante mencionar que se emplea esta tecnología para obtener la ubicación del dispositivo físico que está utilizando la aplicación, para así facilitarla a la página web que dispone de los mapas. También se debe mencionar al lector que se emplea una librería de código abierto llamada Leaflet, con la cual se pueden crear mapas base de Google, ESRI, OpenStreetMap, etc.



Figura 9: Imagen Leaflet

Sin embargo, solo se emplea la librería específica *leaflet-geosearch*, la cual se emplea para obtener un listado de lugares públicos a través de una barra de búsqueda. Estos lugares públicos los proporciona el proveedor de OpenStreetMap a través de la librería anteriormente citada, aunque se podría elegir otro proveedor como Bing, ESRI, Google, etc.



Figura 10: Imagen OpenStreetMap

3.4 MEDIOS HARDWARE

3.4.1 SENSORES

Cuando se trata de seleccionar los sensores se tiene que tener en cuenta los siguientes aspectos:

- El coste de los sensores.
- Las características de los mismos.
- La conectividad que estos emplean.

Se barajan por tanto, tres estrategias con el objetivo de seleccionar el mejor sensor, este será el que cumpla las siguientes premisas:

- Minimizar el coste lo máximo posible, en relación calidad precio.
- Que sea resistente ante forzamientos y condiciones climatológicas.
- Que posea los elementos de conectividad necesarios.

Y las tres estrategias que se consideran son:

- Utilizar sensores de bajo coste, incluyendo los elementos de conectividad.
- Utilizar sensores comerciales que han sido diseñados previamente por compañías del sector.

- Solicitar a una compañía un diseño exclusivo de sensor.

Se ha realizado un estudio completo para cada una de estas tres estrategias.

SENSORES DE BAJO COSTE

Los sensores de bajo coste son aquellos que se adquieren de manera fácil. Obviamente su coste es mucho menor, pero presentan una serie de problemas que se deben resolver (por ejemplo: este tipo de elementos no incluyen su correspondiente instalación ni soporte). Además se requiere adquirir por otra parte los elementos conectivos necesarios, con lo cuál se presentan las limitaciones LIM-2 y LIM-4. Se debe tener en cuenta que en principio son la alternativa de menor coste de compra, sin embargo, el coste sería mucho mayor debido a los costes derivados de la instalación (soportes que eviten el forzamiento y eviten vulnerabilidades climatológicas).

SENSORES COMERCIALES

Tras una investigación exhaustiva de los diferentes sensores diseñados del mercado, el sensor más adecuado ha sido el U-Spot.

U-Spot es un sensor magnético autónomo e inalámbrico. Desarrollado por Urbiótica. La fiabilidad de detección de un vehículo está en torno al 98 %, aportando, a su vez el tiempo de estacionamiento.

Como se ha comentado, se trata de un sensor magnético que mide la variación del campo magnético cuando se sitúa un vehículo cerca del mismo. Tras la detección de una variación notable, envía la información a la nube. Además, estos sensores se adaptan a posibles cambios del campo magnético por la influencia de vehículos vecinos a la plaza de aparcamiento y elementos metálicos cercanos.

Estos sensores pueden ser instalados en 10 minutos sin cables y con mínima obra civil. Su calibración es realizada de manera automática, evitando así un exceso de mantenimiento. Puede ser instalado en el bordillo, en el suelo o en la acera.

La vida útil es de 10 años, tras este periodo será necesario el remplazo de las baterías. Poseen un nivel de protección IP67, resistencia mecánica IK10 (robustez a inclemencias meteorológicas) y rango de temperatura de funcionamiento entre los -33°C a 65°C.

El protocolo de comunicación utilizado es el protocolo estándar IEEE 802.15.4 a 2,4GHz (REST, AMQP), permitiendo una comunicación inalámbrica de hasta 200 metros con la red de comunicación.

Otra posible idea es la utilización de cámaras que vigilen las plazas de aparcamiento. Haciendo uso de inteligencia artificial sería posible saber estimar las plazas disponibles. Esta idea sería interesante, ya que no sería necesario utilizar tantos sensores. Pero se presentan varios inconvenientes, entre los que cabe destacar la falta de precisión de estos sistemas, así como la desconfianza que presentaría para los ciudadanos la presencia de estas cámaras en todas las calles.

Se puede observar como la empresa proporciona todos los elementos de instalación necesarios, pero esto incrementa notablemente el coste inicial por plaza de estacionamiento.

SENSORES DE DISEÑO EXCLUSIVO

Esta alternativa se considera como la más viable, se podría realizar un encargo de sensores magnéticos de bajo coste, solicitando al diseñador la inclusión de los elementos de soporte y de mantenimiento. Esto facilita-

r  que los sensores sean de facil reemplazo asegurando que el coste ser  m nimo con respecto a la limitaci n LIM-1. Esta soluci n tendr a un coste inicial alto debido al dise o, pero los costes futuros se ver an reducidos, al disponer de los medios necesarios para el reemplazo y sustituci n de sensores de bajo coste.

Finalmente se muestra al lector las conclusiones extraidas del an lisis de mercado, seleccionando como mejor opci n el encargo del dise o de los sensores a una empresa del sector, incrementando notablemente el coste inicial del proyecto, pero reduciendo al m ximo el coste futuro de implantar la soluci n. Ya que se otorga a la empresa la responsabilidad de garantizar las condiciones antes mencionadas, eliminando as  las limitaciones citadas en la tabla.

TABLA DE COSTES ESTIMADOS Y LIMITACIONES PRESENTES		
TIPO DE SENSOR	PRECIO ESTIMADO	LIMITACIONES PRESENTES
SENSORES DE BAJO COSTE	65� por plaza	LIM-2 LIM-4
SENSORES COMERCIALES	250-300� por plaza	LIM-1 LIM-2
SENSORES DE DISE�O EXCLUSIVO	Por Determinar	Se busca solventar las limitaciones en el encargo y reducir el coste futuro de mantenimiento

3.4.2 ELEMENTOS DE CONECTIVIDAD

Antes de seleccionar un elemento de conectividad u otro se debe tener en cuenta el paso de informaci n por el sistema generado por los sensores.

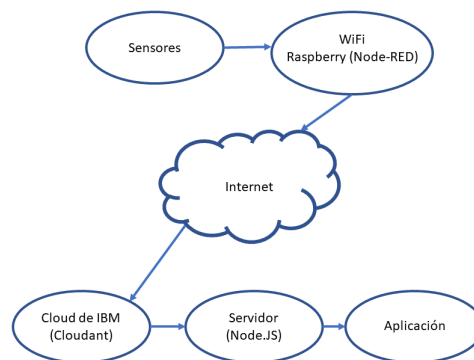


Figura 11: Flujo de Informaci n del Sistema ante la captaci n de un sensor.

Para los elementos de conectividad se proponen dos posibles alternativas:

- Repetidores y Sensores Wemos de Bajo Coste.
- Elementos de SmartCity.

REPETIDORES Y SENSORES WEMOS DE BAJO COSTE

A continuaci n, se procede a realizar un breve an lisis de los chips Wemos D1, la cual es una tarjeta de desarrollo similar a Arduino orientada al Internet de las cosas.

Dispone de un SOC (procesador central) de 32 bits a 160 MHz, además de disponer de conectividad WiFi y Bluetooth, todo ello por alrededor de 3€.

ESPECIFICACIONES TÉCNICAS:

- Voltaje de Alimentación: 5V DC.
- Voltaje Entradas/Salidas: 3.3V DC.
- SoC: ESP8266 (Módulo ESP-12E).
- CPU: Tensilica Xtensa LX3 (32 bit).
- Frecuencia de Reloj: 80MHz/160MHz.
- Instruction RAM: 32KB.
- Data RAM: 96KB.
- Memoria Flash Externa: 4MB.
- Pines Digitales GPIO: 11 (pueden configurarse como PWM a 3.3V).
- Pin Analógico ADC: 1 (0-1V).
- UART: 1.
- Chip USB: CH340G.
- Certificación FCC.
- Antena en PCB.
- Corriente Standby: 40uA.
- Corriente Pico: 400mA.
- Consumo corriente promedio: 70mA.
- Consumo de potencia Standby <1.0mW (DTIM3).
- Dimensiones: 34.2mm x 25.6mm.
- Peso: 10g.

NETWORKING:

- 802.11 b/g/n.
- Wi-Fi Direct (P2P), soft-AP.
- Stack de Protocolo TCP/IP integrado.
- Procesador MAC/Baseband integrado.
- Módulos WEP, TKIP, AES y WAPI integrados.
- PLLs, reguladores, DCXO y manejo de poder integrados.
- Potencia de salida de +19.5dBm en modo 802.11b

- SDIO 2.0, SPI, UART.

RASPBERRY Tal y como se citó en el apartado de elementos software, Node-Red no puede ser implementado directamente en los sensores, ya que los sensores únicamente toman mediciones. Por ello, se tiene en cuenta la tecnología Raspberry donde se implementará el servicio de Node-Red.

Por tanto, la Raspberry será el intermediario entre los sensores y Cloudant.

¿Qué es una Raspberry?

- Raspberry Pi es un computador de placa reducida de bajo costo.
- Su software es código abierto.
- Dispone de conectividad WiFi y Bluetooth, en su última versión, que cuesta alrededor de 35€.

ELEMENTOS DE SMARTCITY

Uno de los puntos más importantes para que una ciudad sea considerada una Smart City, es ofrecer conectividad tanto para los ciudadanos como para la infraestructura de la ciudad en sí misma.

Es posible que parezca una solución de alto coste, sin embargo, es posible utilizar las infraestructuras existentes para ampliar los servicios que ofrecen.

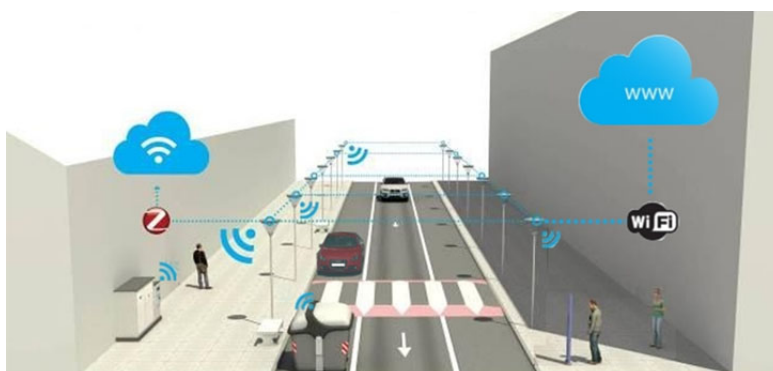


Figura 12: Planteamiento de una smartcity

Una de las alternativas que se ofrece, es el uso de farolas inteligentes capaces de albergar multitud de sensores tales como cámaras de seguridad 4K, módulos 5G, WiFi Gigabit, etc. Esto permitirá mantener una conectividad en tiempo real e indefinida con los sensores.

Puesto que estos elementos son de alto coste y destinar su despliegue para un único uso puede considerarse un gasto innecesario. Se propone la alternativa de proporcionar con los mismos conectividad a los ciudadanos de una serie de servicios gratuitos, relacionados con noticias relevantes de la ciudad, intereses culturales, mapas de visita o cualquier tipo de información que resulte de interés tanto para ciudadanos o visitantes.

4. EXPERIENCIA DE USUARIO

4.1 PRINCIPIOS DE DISEÑO

El diseño de la interfaz es un papel fundamental en cualquier aplicación, está demostrado que más del 70 % del esfuerzo en el desarrollo de las aplicaciones interactivas está enfocado a la interfaz.

Lo primero que hay que tener en cuenta es que la aplicación sea intuitiva, es decir, que se sepa usar a “simple vista”. También, hay que tener en cuenta el uso de metáforas, el uso de ciertos iconos o colores no tienen por qué ser entendidos de la misma manera por toda la sociedad, ya que las diferencias culturales varían dependiendo del lugar de nacimiento, étnia, religión, etc.

Otro aspecto para tener en cuenta son las posibles limitaciones o capacidades que tienen los usuarios de una aplicación, por ejemplo algún tipo de discapacidad [permanente o temporal]. Las posibles limitaciones que se pueden encontrar son:

- **Problemas de percepción del color:** El usuario puede ser daltónico, esto provocaría problemas a la hora de comprender algunas partes del sistema. Problemas auditivos. El usuario podría padecer algún tipo de defecto auditivo, dicho defecto podría ser permanente o temporal (por ejemplo, la presencia de ruidos en el entorno).
- **Problemas de lectura:** Todos los usuarios no pueden leer con la misma velocidad que otros, es por ello por lo que habría que evitar los mensajes largos y que estos desaparezcan muy rápido.
- **Problemas de movilidad:** El usuario podría presentar problemas en el movimiento de ciertas articulaciones, es por ello por lo que se debe de presentar una interacción válida para estos usuarios. Una posible solución sería la implementación de botones grandes o los reconocimientos de voz. Igualmente se recuerda al lector que el contexto de uso de la aplicación requiere de una interacción mínima para comenzar a recibir información.
- **Problemas de idioma:** Los usuarios de la aplicación pueden conocer diferentes idiomas y no en los que se ha desarrollado la aplicación. Es por ello por lo que se debe ofrecer alternativas de idiomas para tener mayor abanico de usuarios potenciales.

4.1.1 INTERFAZ DE LA APLICACIÓN

Lo primero que se mostrará al usuario al iniciar la aplicación es una ventana con el logo de la aplicación y un icono de pre-carga, así el usuario sabe en todo momento que la aplicación está realizando sus correspondientes operaciones. Esta información es importante, ya que hay que procurar en todo momento que el usuario sepa que su acción (abrir la aplicación) está teniendo una consecuencia (una respuesta por parte de la misma).



Figura 13: Imagen Inicio de Aplicación.

Una vez iniciada la aplicación, se proporcionan dos opciones de funcionalidad (Ir a sitio de interés y buscar aparcamiento). Se podría haber considerado añadir más funcionalidades, pero hay que recordar al lector que en la simplicidad está la clave de uso de esta solución, hay que hacer una aplicación sencilla que cumpla con su cometido inicial y que la interacción cumpla los requisitos anteriormente mencionados.



Figura 14: Imagen Menú Inicio.

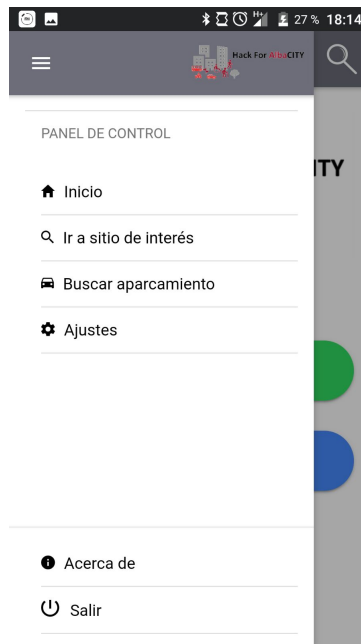


Figura 15: Imagen Menú Lateral de Opciones.

Se incluye en la misma un panel lateral proporcionando una serie de funciones, además de otras funciones que no serán utilizadas de manera habitual por el usuario (es por ello que se ha decidido ponerlo aquí y no en otro sitio). Dos de esas funcionalidades son: Ajustes (solo necesario en la primera configuración del programa) y Acerca de (donde la mayoría de las veces no es visto por el usuario, pero es necesario para reconocer el esfuerzo realizado por los desarrolladores de una aplicación o su correspondiente responsabilidad del uso de la misma).

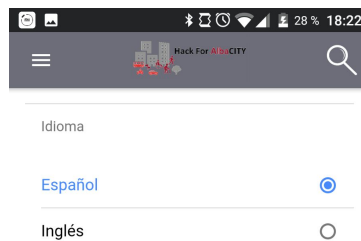


Figura 16: Imagen Ajustes de Idioma.



Figura 17: Imagen Apartado Acerca de.

Como se muestra en el menú principal, se permite acceder a dos funcionalidades principales:

- **Ir a sitio de interés:** Tiene el objetivo principal de que el usuario indique a que lugar de la ciudad quiere visitar y que la aplicación proporcione los aparcamientos disponibles por la zona deseada. Además, la aplicación ofrece una ruta desde su posición actual hasta el aparcamiento más cercano disponible al lugar deseado por el usuario, además de indicar la ruta desde dicho aparcamiento hasta su lugar (esta opción es muy interesante para turistas que no conocen la ciudad, proporcionando la ruta necesaria).

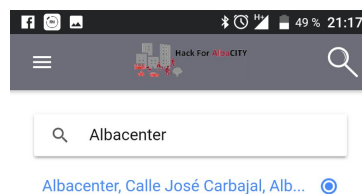


Figura 18: Imagen Apartado Búsqueda de Punto de Interés.



Figura 19: Imagen Ruta Aparcamiento a lugar Deseado.

- **Buscar aparcamiento:** Tiene como objetivo proporcionar al usuario la ruta más cercana desde su posición hasta el aparcamiento más cercano disponible. Es una función útil si el usuario se encuentra conduciendo y decir realizar el estacionamiento de su vehículo. Además, la aplicación ofrece la ruta desde el aparcamiento hasta la ubicación inicial donde el usuario realizó la solicitud. De esta manera, si el usuario desconoce la ciudad, podrá volver al lugar donde requirió realizar la maniobra de estacionamiento.

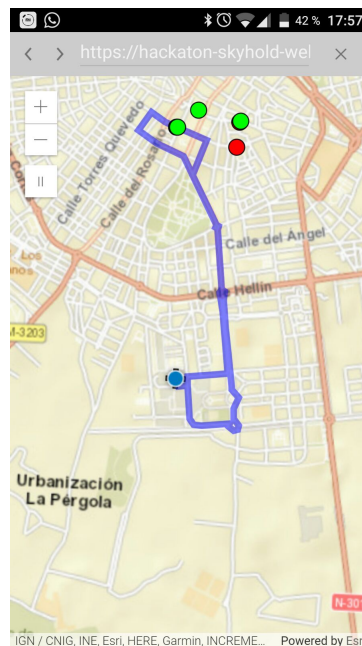


Figura 20: Imagen Ruta más Cercana.

Es importante resaltar que en ambas opciones se usan la misma metáfora de colores (de tipo semáforo) para indicar las plazas disponibles (verde), plazas ocupadas (rojo), posición actual (azul), ruta (línea en azul), destino

seleccionado(aspas), recoger ubicación (brújula), aumentar/reducir zoom del mapa (+/-). De esta manera, el usuario no necesita memorizar el sentido de dichos iconos, ya que se usan en las dos funcionalidades y además son símbolos comúnmente conocidos por el uso de otras aplicaciones con mayor repercusión (por ejemplo, Google Maps).

4.2 SOLUCIONES DE ACCESIBILIDAD

En nuestra solución se prevee lo descrito anteriormente en el apartado 4.1. Se debe tener en cuenta que es una aplicación que se usará principalmente en un vehículo. Aunque no se debe emplear la aplicación mientras que se conduce, es inevitable que el usuario lo haga, aunque sea por un periodo de tiempo reducido, es por ello por lo que nuestra interfaz debe de estar diseñada de tal forma que se pueda usar mientras que se conduzca (aunque no se deba) reduciendo al máximo el período de interacción.

La aplicación se ha diseñado de la forma más intuitiva posible, esto ha sido posible haciendo uso de iconos reconocidos por el usuario medio “Average Joe” (usuario con conocimientos básicos/medios en interfaces de GPS) y diseñando la interfaz con un planteamiento similar a otras aplicaciones con mayor reconocimiento en el sector (Google Maps, etc.) y de una manera muy simple (ajustado exclusivamente a la funcionalidad de nuestra aplicación).

Como se cito anteriormente, el uso de metáforas es un aspecto que se debe tener muy en cuenta. En nuestro diseño de la aplicación, se emplea el uso de los colores rojo y verde para indicar que una plaza está ocupada o libre (haciendo referencia a las prohibiciones de un semáforo); además, se emplean los iconos de un punto para indicar las plazas sobre el mapa, la posición del usuario y el destino deseado.

En cuanto a las limitaciones del usuario citadas anteriormente, no se realizó la implementación de la mayoría, puesto que lo desarrollado sólo corresponde a un prototipo, estas limitaciones son las siguientes:

- **Problemas de percepción del color:** No se incluye la implementación de la configuración de los colores de la aplicación, aunque su implementación sería muy simple. Se añadiría una sección de ajustes y se podría realizar una pequeña prueba de daltonismo para determinar si el usuario capta los colores perfectamente.
- **Problemas auditivos:** La aplicación diseñada no dependerá únicamente de avisos acústicos, haciendo uso de indicaciones y avisos el usuario podrá conocer el estado de la aplicación. Hay que anticiparse a que el usuario se puede encontrar en un entorno muy ruidoso y no podrá escuchar la aplicación y sus indicaciones dependiendo del contexto de uso.
- **Problemas de lectura:** Los mensajes utilizados para comunicarse con el usuario son cortos y fácilmente entendibles, ya que el lugar de implementación de nuestra aplicación (en conducción) obliga a una rápida actuación por parte del usuario.
- **Problemas de movilidad:** Se debe de tener en cuenta que los usuarios que controlan nuestra aplicación poseen una movilidad suficiente para utilizar un automóvil. Aunque no sería un problema de movilidad, sino que el hecho de controlar la aplicación en conducción implicaría una mala precisión por parte del usuario para interactuar con la aplicación. La solución que se ha implementado es la inclusión de botones grandes y distinguibles; además de requerir la atención del usuario lo mínimo posible, evitando una interacción continua del usuario.
- **Problemas de idioma:** Esta opción se ha implementado parcialmente, básicamente en el menú de Ajustes el usuario podrá indicar el idioma deseado; además también se podría haber considerado la autodetección del idioma del dispositivo del usuario de manera automática, teniendo en cuenta el idioma de su dispositivo.

5. CONCLUSIONES

Se concluye, qué la solución planteada proporciona los beneficios necesarios para una ciudad en constante crecimiento demográfico y tecnológico.

Cabe mencionar, que la elección de un correcto presupuesto, es vital para la viabilidad de la solución, por ello se proporcionan diferentes alternativas. Estas deben ser evaluadas con mayor precisión, independientemente del análisis aportado por este documento, con el fin de escoger la mejor alternativa posible y garantizar la viabilidad del proyecto.

Otro punto que el lector debe tener en cuenta, es el diseño de la arquitectura planteada. Se plantea como la arquitectura más adecuada para la pequeña y mediana empresa, dado que permite una alta escalabilidad, beneficiando en tiempo de desarrollo y en coste asociado. Se recuerda al lector que los elementos que forman de la misma pueden ser modificados sin provocar rediseños en los demás componentes, esta es la ventaja de emplear una arquitectura por capas.

Este proyecto no se limita a un único escenario o ciudad, es por ello por lo que se amplía la solución en el apartado de trabajos futuros, es un proyecto muy interesante e innovador para ser desarrollado con más tiempo y dedicación de la que se dispone.

Finalmente recordamos que otra bondad de los elementos conectivos es la capacidad de proveer otros servicios a los habitantes y visitantes de una ciudad. Brindando a los ayuntamientos la posibilidad de ofrecer bienes de interés cultural, turismo, transporte y noticias relevantes. Enriqueciendo y haciendo accesible al público contenido de utilidad e interés introduciendo dicha ciudad como referente a nivel tecnológico y cultural.

6. REFERENCIAS

- [1] Apache Cordova, [cordova.apache.org](https://cordova.apache.org/docs/en/latest/). Consultado el 10 de Abril del 2018. <https://cordova.apache.org/docs/en/latest/>
- [2] ArcGIS for Developers, developers.arcgis.com. Consultado el 14 de Abril del 2018. <https://developers.arcgis.com/>
- [3] Bouskela, Mauricio., et al (2016), *La ruta hacia las Smart Cities: Migrando de una gestión tradicional a la ciudad inteligente*. Banco Interamericano de Desarrollo.
- [4] Dirección General de Tráfico, www.dgt.es. Consultado el 13 de Abril de 2018. <http://revista.dgt.es/es/categorias/distracciones-moviles.shtml>
- [5] IBM Cloudant, [www.ibm.com](https://www.ibm.com/cloud/cloudant). Consultado el 14 de Abril del 2018. <https://www.ibm.com/cloud/cloudant>
- [6] IBM Bluemix, [www.ibm.com](https://www.ibm.com/cloud-computing/bluemix/es). Consultado el 14 de Abril del 2018. <https://www.ibm.com/cloud-computing/bluemix/es>
- [7] Ionic, [www.ionicframework.com](https://ionicframework.com/docs/). Consultado el 11 de Abril del 2018. <https://ionicframework.com/docs/>
- [8] Marco de Desarrollo de la Junta de Andalucía, *Conceptos sobre Escalabilidad*, Consultado el 10 de Abril de 2018. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220>
- [9] Node-RED, [www.nodered.org](https://nodered.org/docs/platforms/bluemix). Consultado el 16 de Abril del 2018. <https://nodered.org/docs/platforms/bluemix>
- [10] Pandorafms, *Bases de Datos NoSQL*, [pandorafms.org](https://blog.pandorafms.org/es/bases-de-datos-nosql/). Consultado el 12 de Abril de 2018. <https://blog.pandorafms.org/es/bases-de-datos-nosql/>
- [11] Universidad de Castilla-la Mancha. www.uclm.es. Consultado el 9 de Abril de 2018. <https://eventos.uclm.es/19833/detail/hack-for-albacity.html>
- [12] URBBIOTICA (2008), [www.urbiotica.com](http://www.urbiotica.com/producto/u-spot/). Consultado el 18 de Abril del 2018. <http://www.urbiotica.com/producto/u-spot/>
- [13] W3Schools (1998), [www.w3schools.com](https://www.w3schools.com/nodejs/default.asp). Consultado el 15 de Abril del 2018. <https://www.w3schools.com/nodejs/default.asp>

ANEXOS

ANEXO I ENLACES DE CONTENIDO

- **Enlace a la WebApp:**
<https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-Web>
- **(Aplicación de mapas en ejecución):**
<https://hackaton-skyhold-web.eu-gb.mybluemix.net/>
- **Documentación (presentación y memoria):**
<https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-Web/tree/master/DOCS>
- **Node-RED:**
<https://git.eu-gb.bluemix.net/HernanIndibil.LaCruz/Hackaton-Skyhold-RED>
- **(Editor de Node-RED):**
<https://hackaton-skyhold-red.eu-gb.mybluemix.net/red/>
- **Repositorio de la Aplicación:**
<https://github.com/Mowstyl/Hackaton-Skyhold-App>

ACLARACIÓN: *Los diferentes servicios serán públicos o se proveerán los accesos necesarios a los mismos.*