

# PROTOTIPO BASADO EN IoT Y ARDUINO® PARA VALIDAR LA DISPONIBILIDAD DE ESTACIONAMIENTOS EN LA SEDE NORTE DE LA UNIAJC

IoT AND ARDUINO®-BASED PROTOTYPE TO VALIDATE PARKING SPACE AVAILABILITY AT UNIAJC NORTH HEADQUARTERS

## AUTORES

### Fanor Martínez Tenorio:

Ingeniero involucrado en el desarrollo empresarial y emprendimiento. Desde 2004 ha sido profesor en el área de la tecnología, ingeniería y la administración en diversas cátedras. Se ha desempeñado como director académico y gestor de proyectos. Actualmente es miembro del grupo de investigación INTELIGO de la Institución Universitaria Antonio José Camacho, que tiene como líneas de investigación los procesos industriales y la aplicación de sistemas embebidos. Correo: [fmartinezt@admon.uniajc.edu.co](mailto:fmartinezt@admon.uniajc.edu.co)

### Edwar Andrés Benachi Arce:

Egresado de Tecnología en Electrónica de la Institución Universitaria Antonio José Camacho. Correo: [edwar-2008@hotmail.com](mailto:edwar-2008@hotmail.com)

### Cristian Daza Bolaños:

Egresado de Tecnología en Electrónica de la Institución Universitaria Antonio José Camacho. Correo: [falcon\\_m21@hotmail.es](mailto:falcon_m21@hotmail.es)

33

Fanor Martínez Tenorio, Edwar Andrés Benachi Arce, Cristian Daza Bolaños

### Semillero SELECT

Grupo de investigación en Emulación de la Percepción, la Cognición y la Comunicación (INTELIGO)

Institución Universitaria Antonio José Camacho

Recibido: 24/06/2022 - Aceptado: 27/09/2022

Para citar este artículo: Martínez Tenorio, F., Benachi Arce, E. y Daza Bolaños, C. (2022). Prototipo basado en IoT y Arduino® para validar la disponibilidad de estacionamientos en la sede norte de la UNIAJC. Revista Sapiencia, 14(28), 33-47.

## RESUMEN

Este artículo revela los resultados de una investigación articulada al semillero SELET y el grupo de investigación INTELIGO. Donde se desarrolla una solución a problemas de ciudad, específicamente relacionados a la poca oferta de parqueaderos y a la falta de información sobre la disponibilidad de estos espacios. Los retos encontrados por los estudiantes ante este desarrollo son la integración de un módulo de sistema embebido a las soluciones basadas en el internet de las cosas (IoT), que permita determinar la presencia de vehículos en un estacionamiento. Para ello, se emplea una variante compuesta de la metodología de Software RUP y XP, acompañada de un proceso de diseño de arquitectura de hardware, obteniendo unos resultados para establecer una ruta de selección e implementación apropiada de tecnologías para desarrollar el prototipo, además de recomendaciones para posibles evoluciones de este prototipo a un producto mínimo viable.

### PALABRAS CLAVE

Internet de las cosas IoT; Sistema embebido Arduino; Firebase; Servicios OTT (over the top).

## ABSTRACT

This article reveals the results of a research articulated with the SELET research group and the INTELIGO research group. Where a solution to city problems is developed, specifically related to the limited supply of parking spaces and the lack of information on the availability of these spaces. The challenges encountered by the students in this development are the integration of an embedded system module to solutions based on the Internet of Things (IoT), which allows determining the presence of vehicles in a parking lot. For this, a variant composed of the RUP and XP Software methodology is used, accompanied by a hardware architecture design process, obtaining results to establish a route for the selection and appropriate implementation of technologies to develop the prototype, as well as recommendations for possible evolutions of this prototype to a minimum viable product.

### KEYWORDS

Internet of Things IoT; Arduino embedded system; Firebase; OTT (over the top) services.

## INTRODUCCIÓN

**E**n las grandes ciudades del mundo en desarrollo, los tiempos empleados en viajes son generalmente altos y van en aumento. Los destinos accesibles dentro del tiempo dado están disminuyendo. El tiempo promedio de un viaje diario en un solo sentido en Río de Janeiro es de 90 minutos. En Bogotá, de 60 minutos. La velocidad vehicular promedio en Manila es de siete millas por hora. El auto en Bangkok pasa detenido en el tráfico, en promedio, el equivalente a 44 días al año. (Gakenheimer, 1998)

En los últimos años en las ciudades colombianas se ha venido presentando un crecimiento bastante significativo del parque automotor, generando situaciones por la poca oferta de parqueaderos, siendo este un problema de movilidad que presentan las ciudades del siglo XXI.

“La ciudad de Cali ha establecido 12 Zonas de Estacionamiento Regulado - ZER a las cuales se les ha realizado parcialmente estudios de oferta y demanda como insumo para elaborar el Plan Maestro de Estacionamiento” (2019). Dichos estudios han revelado la demanda no atendida en vía en ciertos horarios para estas zonas, la cual se atiende con la capacidad de los estacionamientos reconocidos como fuera de vía.

Para la debida comprensión de los análisis de la oferta y demanda de estacionamientos públicos en vía y fuera de vía, el Plan Integral de Movilidad Urbana (PIMU) de Cali presenta las siguientes definiciones de cada uno de los conceptos expuestos.

**Estacionamiento en vía:** Corresponde a la oferta de cajones o plazas de estacionamiento ubicados sobre infraestructura vial, en la cual se asigna un carril de tráfico mixto al aparcamiento de vehículos.

**Estacionamiento fuera de vía:** Corresponde a la oferta de cajones o plazas de estacionamiento ubicados en parqueaderos en lotes, sótanos o edificios adecuados para tal fin. El tipo de servicio puede ser público o privado. (Departamento Administrativo de Planeación, 2019)

Hoy los más de 737.428 vehículos que circulan en la ciudad, según informe anual de calidad de vida (Informe Anual de Calidad de vida Cali cómo vamos 2021, 2021), se ven obligados a parquear en zonas alternativas, unas reguladas y otras no, lo cual crea diversos tipos de caos vehicular. A toda esta problemática también se ven avocadas las universidades y, en este caso, el parqueadero de la Institución Universitaria Antonio José Camacho de la sede norte, que pertenece a la Zonas de Estacionamiento Regulado 2 (ZER 2) (Departamento Administrativo de Planeación, 2019).

El área del parqueadero de la institución incurre en la misma situación de estacionamientos que presenta la ciudad, donde no puede garantizar la total demanda de estos para su población. Además, sus usuarios al realizar el desplazamiento para verificar si hay espacio o no, pierden la oportunidad de encontrar otro sitio en la zona que le garantice la seguridad de su vehículo.

Por lo tanto, se plantea la hipótesis de que la falta de información sobre la disponibilidad de espacios libres de parqueadero genera diferentes inconvenientes como:

- Congestión en zonas cercanas a la universidad por la cantidad de usuarios del parqueadero universitario.
- Pérdida de tiempo del conductor en la búsqueda de estacionamiento.
- Contaminación ambiental por causa del aumento del trayecto del vehículo.

El contexto anterior hace referencia a unas de las tantas problemáticas de movilidad urbana e invita a comprender el concepto de Ciudades Inteligentes o Smart City donde, según el desarrollo teórico, determina que es una evolución e integración de diferentes tecnologías, por ejemplo: la informática, la electrónica, las TIC, los sistemas de información, los servicios en la nube, etc., que buscan desarrollos sostenibles de las ciudades.

36 Considerando que la Institución Universitaria Antonio José Camacho busca desarrollar diferentes competencias de tipo disciplinar como las mencionadas y además relacionadas a lo que se conoce como competencias ciudadanas. A través de trabajos de grado como este, se pretende ayudar a establecer mayor intervención de los estudiantes en investigación, desarrollo y soluciones a su entorno.

Por lo anterior, el objeto de este artículo es dar a conocer la experiencia del desarrollo de un prototipo como estrategia en la cual, a partir de investigaciones del grupo INTELIGO, estos trabajos hacen parte de sus productos y además ayudan a superar las falencias con las que los estudiantes se enfrentan al no saber qué proyecto de grado realizar, así como la de aprender a generar soluciones integrales.

## MARCO TEÓRICO

La investigación toma como referente inicial lo expresado en el trabajo denominado Transformar lo Público, perspectivas sobre la reforma administrativa de Santiago de Cali de 2016, donde se considera Smart City a las ciudades interconectadas regional, nacional y globalmente, gracias a la intermediación de flujos de capitales económicos o sociales en un contexto en donde las tecnologías de la información toman cada vez más importancia en las relaciones cotidianas, institucionales e interinstitucionales. (Sanabria Pulido & Rodríguez Caporalli, 2019)

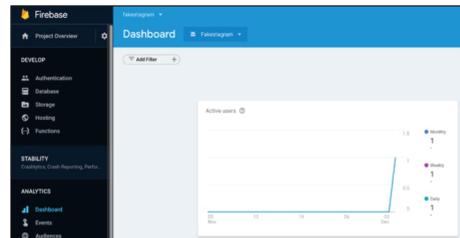
En este contexto, se puede pensar que, para generar inteligencia urbana, la población de estas ciudades demanda una diversidad de servicios no concentrados en un proveedor y, además, que se adapten específicamente a sus requerimientos y actuaciones; lo que invita a una combinación de esfuerzos ingeniosos para mejorar la calidad de vida de los y las habitantes, donde se promueva el crecimiento económico y proteja el medio ambiente de la degradación (ONU, 2016, como se citó en Casas, 2020). Es así como la creación de entidades relacionadas con las TIC, la política de ciudades inteligentes y el mejoramiento del servicio al ciudadano mediante la tecnología son medidas acertadas para el futuro de estos temas en la ciudad.

La tecnología reconocida como el Internet de las Cosas (IoT) brinda utilidades que pueden considerarse y definirse como prácticamente infinitas, ya que cada vez son más los medios que nos rodean y que tenemos a nuestra disposición para implementar, de manera no muy compleja, factible y de fácil uso, con gran ventaja y un posible dispositivo electrónico capaz de dotar a cierto objeto o función de una conexión a la red, ya sea por medio de una tecnología u otra.

Como lo menciona Martínez Fernández (2016), son muchas las aplicaciones móviles y servicios donde se generan cantidad de datos en tiempo real, y esto ha generado los reconocidos sistemas 'Big Data', facilitando así la integración de sensores aplicables prácticamente a cualquier tipo de necesidad.

La tecnología Arduino®, que emplea el procesador ESP8266, es apta para usar en cualquier proyecto IoT. Se programa con LUA o mediante el IDE de Arduino®; el lenguaje que opera dentro de este microcontrolador se llama Wiring, basado en la plataforma Processing y primordialmente en el lenguaje de programación C/C++. Dispone de una extensa comunidad y documentación que permite conectar el proyecto al mundo exterior mediante la conexión Wifi por el conversor USB CH340, que normalmente el sistema operativo lo instala automáticamente, aunque dependiendo de los casos, puede que se requiera instalar el driver específico. (COLDFIRE ELECTRONICA, s.f.)

Firebase es una plataforma de Google para el desarrollo y creación de aplicaciones móviles, con un entorno de trabajo amigable como se observa en la Figura 1. Esta plataforma se encuentra en la nube disponible para iOS, Android y web. Es básicamente una base de datos remota que responde en tiempo real a los cambios que se realizan en los datos suministrados por ciertos dispositivos como sensores o, de igual manera, por un usuario a través de una App.



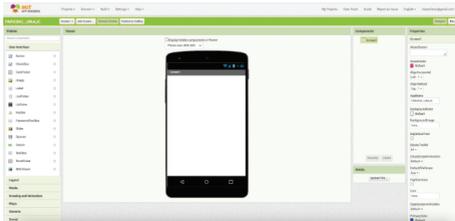
**Figura 1.** Entorno de desarrollo plataforma Firebase.

**Fuente:** <https://firebase.google.com/>.

Los datos recolectados en tiempo real se almacenan en formato JSON (JavaScript Object Notation) y son sincronizados de la misma manera con el cliente o el usuario que esté haciendo uso de estos, esto garantiza una comunicación entre Firebase y los datos empleados por la App, y así la visualización desde cualquier dispositivo móvil. Según la guía de Firebase:

Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones, llamado reglas de seguridad de Firebase Realtime Database, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir. Integrar Firebase Authentication permite que los programadores definan quién tiene acceso a qué datos y cómo acceden a ellos. (Rosero, 2019)

MIT App Inventor, adoptada por Google, es una herramienta con la que se puede crear Apps para dispositivos Android que convierte fácilmente a sus usuarios en creadores de su propia tecnología y estos, a su vez, pueden compartir sus aplicaciones con otras personas haciendo uso de Internet, como se ve en la Figura 2.



**Figura 2.** Entorno de desarrollo en plataforma App Inventor

**Fuente:** <http://ai2.appinventor.mit.edu/#6172155298316288>

Todo este proceso va relacionado con el entorno físico o con acciones que permitan al usuario ver procesos que se ejecutan en la vida real como en lo digital, y el hecho de que los estudiantes puedan llevar sus propias Apps en sus dispositivos es un aliciente para motivarlos a crear e innovar y además a

romper paradigmas en la adquisición de nuevos conocimientos. MIT App Inventor puede ser una excelente herramienta para desarrollar el Pensamiento Computacional en las aulas porque promueve todas las ideas expuestas con anterioridad con el añadido de que incorpora la ubicuidad de la tecnología que usa (Sandoval Quintero, 2018).

## METODOLOGÍA

La investigación se desarrolló a través de una variante compuesta de las metodologías de Software RUP y XP, acompañadas de un proceso de diseño de arquitectura de hardware.

En la metodología RUP se manejaron ciclos iterativos, se define tiempo, se hereda modelos y se define una documentación. La metodología XP establece un enfoque en los usuarios, obteniendo resultados rápidos para interactuar con el proceso y la finalidad de hacer refactorización del código.

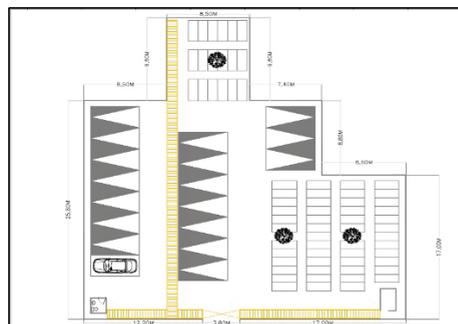
La implementación de la metodología involucró una fase inicial de investigación que permite, en un segundo momento, establecer la documentación de los modelos de software y hardware adoptados para responder a los requerimientos de usuario, posteriormente se realizan las implementaciones y pruebas respectivas del prototipo dando respuesta a la solución del problema. La siguiente tabla permite reconocer las fases expresadas de manera detallada.

<b>Fase de Investigación</b>	<p>Exploración de problemas de ciudad y tecnologías involucradas específicamente en espacios de parqueaderos en diferentes contextos, que permitiera entender los requerimientos de usuarios y escenarios de implementación.</p> <p>Reconocimiento de trabajos y experiencias que involucrarán:</p> <ul style="list-style-type: none"> <li>Tecnologías de captura de variables (sensores) en espacios similares a donde se realizaría la implementación del prototipo.</li> <li>Tecnología embebida y plataformas que permitan articularse con IoT y OTT, de forma fácil y de bajo costo.</li> <li>Protocolos de comunicación y arquitecturas compatibles con las tecnologías encontradas.</li> <li>Desarrollo de aplicaciones APP, base de datos e interfaces de forma ágil y costos bajos.</li> <li>Comprensión de servicios OTT ofertados y apropiados a la solución basada en IoT que se proponía.</li> </ul>
<b>Fase de Documentación</b>	<p>Caracterización de los espacios del parqueadero de la UNIAJC (sede norte) y mapeo del área de uso de estos para determinar el entorno de censado.</p> <p>Especificación de los requerimientos funcionales y no funcionales para el diseño de la arquitectura de hardware y software.</p> <p>Definición de hardware para establecer la captura de variables y la comunicación del prototipo.</p> <p>Definición de los protocolos de comunicación de la plataforma Firebase.</p> <p>Definición de las herramientas de software necesarias para cumplir con los servicios utilizados en el análisis de disponibilidad de parqueaderos.</p> <p>Diseño de interfaz hardware-software que permita la funcionalidad de la App.</p>
<b>Fase de Implementación</b>	<p>Implementación de la arquitectura tecnológica que soporte los requerimientos definidos para los servicios de disponibilidad de parqueaderos.</p> <p>Desarrollo del módulo de hardware y software para el funcionamiento de la solución tecnológica.</p> <p>Desarrollo del módulo App de disponibilidad del parqueadero.</p>
<b>Fase de Pruebas</b>	<p>Pruebas funcionales asociadas al diseño de cada componente del sistema integrado hardware y software.</p> <p>Validación de un servicio OTT a la solución basada en IoT para otorgar la disponibilidad de estacionamientos libres de la UNIAJC sede norte.</p> <p>Pruebas para verificar la funcionalidad de los sistemas (Servicios y Red).</p>

**Tabla 1.** Componentes de las fases desarrolladas  
**Fuente.** Elaboración propia.

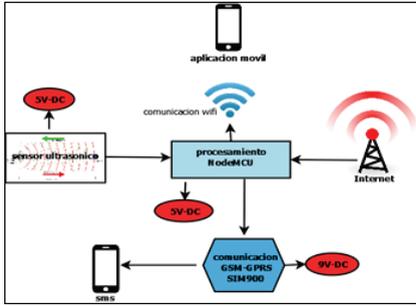
## HALLAZGOS

La caracterización de requerimientos del sitio arrojó un plano ajustado al proyecto de normas para parqueaderos que presentó la Alcaldía de Santiago de Cali en el 2019. Donde estos espacios sean públicos o privados deberán de cumplir esta norma. El plano se desarrolló con medidas reales del sitio a una escala 1:200, y la caracterización determinó 18 estaciones de parqueos de carros y zonas de parqueo para motocicletas con un total de 66 estaciones. Ver Figura 3.



**Figura 3.** Plano ajustado según propuesta de normatividad Santiago de Cali del parqueadero sede norte UNIAJC  
**Fuente:** Elaboración propia.

Dando continuidad al levantamiento de los requerimientos funcionales y no funcionales del proyecto se encuentra que la siguiente arquitectura del sistema, Figura 4, permite cumplir los objetivos técnicos del proyecto y posteriormente seleccionar y justificar los componentes de hardware y software que se acogen a las capacidades de adquisición e implementación por parte del equipo de investigación. Ver la Tabla 2.



**Figura 4.** Diseño de la arquitectura del prototipo para validar la disponibilidad de estacionamientos en la sede norte de la UNIAJC  
Fuente: Elaboración propia.

Referencia dispositivo	Características de selección e implementación encontradas
<p><b>Figura 8.</b> Módulo GSM SIM800L Fuente: <a href="https://nettigo.eu/system/images/1935/original.jpg">https://nettigo.eu/system/images/1935/original.jpg</a></p>	<p>Este módulo proporciona una manera de comunicarse utilizando el teléfono celular o lo que se conoce como GSM a través de la red. Este sistema permite transmisiones digitales de voz y datos, como mensajes de texto (SMS) o mensajes multimedia (MMS).</p> <p>Se requiere un microcontrolador para controlarlo, en el caso del proyecto planteado se usa la NODEMCU, sin embargo, cualquier microcontrolador que tenga comandos UART (transmisor/receptor asincrónico universal), puede enviar y recibir comandos a través de los pines RX/TX.</p>

**Tabla 2.** Componentes seleccionados para el desarrollo del prototipo  
Fuente: Elaboración propia basada en (Benachi Arce y Daza Bolaños, 2019).

Referencia dispositivo	Características de selección e implementación encontradas
<p><b>Figura 5.</b> Sensor ultrasónico JSN-SR04T Fuente: <a href="https://http2.mlstatic.com/modulo-sensor-ultrasonico-impermeable-jsn-sr04t-arduino-D_NQ_NP_927325-MLM25427176918_032017-F.jpg">https://http2.mlstatic.com/modulo-sensor-ultrasonico-impermeable-jsn-sr04t-arduino-D_NQ_NP_927325-MLM25427176918_032017-F.jpg</a></p>	<p>Este sensor tiene un rango de captura desde los 25 cm a 4,5 metros aproximadamente, trabaja a la intemperie y no es tan sensible a los cambios de temperatura ideal para los requerimientos encontrados.</p> <p>Además, se comprueba mayor confiabilidad al realizar comparación con otros sensores como los inductivos o sensores de proximidad que no abarcan una distancia suficiente para emplearse en el proyecto, otros como los magnéticos que solo trabajan para detectar material metálico y los capacitivos que tienen una sensibilidad que no permite estabilidad en la captura de los datos.</p>
<p><b>Figura 6.</b> Tarjeta NODEMCU v3.0 Fuente: <a href="https://www.coldfire-electronica.com/es/item/356/tarjeta-de-desarrollo-nodemcu-v3-esp8266">https://www.coldfire-electronica.com/es/item/356/tarjeta-de-desarrollo-nodemcu-v3-esp8266</a></p>	<p>Esta tarjeta contiene un procesador ESP8266 que permite trabajar por medio de WIFI ideal para los proyectos IoT. Trabaja sobre la plataforma Arduino® IDE siempre y cuando se haga la previa instalación de unas librerías y un controlador para su buen funcionamiento.</p> <p>Para este proyecto, la tarjeta NODEMCU con la que se trabajó es una versión China y obliga a descargar el controlador (DRIVER CH340) e instalarlo en Windows, este driver se puede encontrar en internet de manera gratuita y sirve para muchas otras tarjetas como Arduino® uno, nano, mega, etc. y tarjetas que tengan procesador ESP8266 que sean de esta procedencia. Después, se procedió a hacer la instalación de la librería para el procesador ESP8266, y para ello se ejecuta el software Arduino® IDE.</p>
<p><b>Figura 7.</b> Regulador de voltaje DC-DC Step-Down 3A LM2596 Fuente: <a href="https://naylor-mechatronics.com/841-large_default/convertido-el-voltaje-dc-dc-step-down-3a-lm2596.jpg">https://naylor-mechatronics.com/841-large_default/convertido-el-voltaje-dc-dc-step-down-3a-lm2596.jpg</a></p>	<p>Se emplean dos reguladores de este tipo en el proyecto porque como función entregan un voltaje de salida ajustable entre 1.23V a 37V y estable, frente a variaciones del voltaje de entrada o de carga, dado que es capaz de soportar corrientes de salida de hasta 3A. De esta manera se garantiza entregar 5v DC para el funcionamiento de la tarjeta NODEMCU y 3.5v DC para la tarjeta SIM 800L a partir de un único suministro de energía.</p>

40

Para la integración de las anteriores tecnologías, se realizó el acondicionamiento que permitiera acoplar y obtener comunicación entre el NODEMCU, la plataforma Firebase y el Módulo GSM SIM800L, como se expone a continuación.

Con respecto al código en Arduino®, para la definición de variables que identifiquen los sensores ultrasónicos y la comunicación con la plataforma Firebase, es necesario contar inicialmente con las librerías respectivas. Al mismo tiempo, como el prototipo consta de 2 sensores, se deben nombrar cada uno de forma diferente para no generar conflicto en la ejecución del programa, cada TRIGGER y cada ECHO de los sensores deben diferenciarse como se muestra en la Figura 9.

En el área de conectividad con Firebase se debe colocar el host de la plataforma Firebase y un código que entrega la plataforma para su posterior conexión, al igual que colocar el nombre de la red Wifi y la contraseña.

```

2_sensores_parquadero_tesis_firebase Arduino 1.8.4
Archivo Editar Programa Herramientas Ayuda
2_sensores_parquadero_tesis_firebase
#include <Wire.h>
#include <DS2413.h>
#include <FT232RL.h>
#include <OneWire.h>
#include <SoftwareSerial.h>
SoftwareSerial S28000(13, 12); // Configure el puerto serial para el S28000

//definico como nos conectas pin 01 / RXD0= pin 02
#define DS2413 0 //pin 01
#define RXD01 0 //pin 02
//definico como nos conectas pin 05 / S280= pin 04
#define DS2802 14 //pin 04
#define RXD02 12 //pin 02
#define RXD03 14 //pin 04
#define RXD04 14 //pin 04
#define RXD05 14 //pin 04
#define RXD06 14 //pin 04
#define RXD07 14 //pin 04
#define RXD08 14 //pin 04
#define RXD09 14 //pin 04
#define RXD10 14 //pin 04
#define RXD11 14 //pin 04
#define RXD12 14 //pin 04
#define RXD13 14 //pin 04
#define RXD14 14 //pin 04
#define RXD15 14 //pin 04
#define RXD16 14 //pin 04
#define RXD17 14 //pin 04
#define RXD18 14 //pin 04
#define RXD19 14 //pin 04
#define RXD20 14 //pin 04
#define RXD21 14 //pin 04
#define RXD22 14 //pin 04
#define RXD23 14 //pin 04
#define RXD24 14 //pin 04
#define RXD25 14 //pin 04
#define RXD26 14 //pin 04
#define RXD27 14 //pin 04
#define RXD28 14 //pin 04
#define RXD29 14 //pin 04
#define RXD30 14 //pin 04
#define RXD31 14 //pin 04
#define RXD32 14 //pin 04
#define RXD33 14 //pin 04
#define RXD34 14 //pin 04
#define RXD35 14 //pin 04
#define RXD36 14 //pin 04
#define RXD37 14 //pin 04
#define RXD38 14 //pin 04
#define RXD39 14 //pin 04
#define RXD40 14 //pin 04
#define RXD41 14 //pin 04
#define RXD42 14 //pin 04
#define RXD43 14 //pin 04
#define RXD44 14 //pin 04
#define RXD45 14 //pin 04
#define RXD46 14 //pin 04
#define RXD47 14 //pin 04
#define RXD48 14 //pin 04
#define RXD49 14 //pin 04
#define RXD50 14 //pin 04
#define RXD51 14 //pin 04
#define RXD52 14 //pin 04
#define RXD53 14 //pin 04
#define RXD54 14 //pin 04
#define RXD55 14 //pin 04
#define RXD56 14 //pin 04
#define RXD57 14 //pin 04
#define RXD58 14 //pin 04
#define RXD59 14 //pin 04
#define RXD60 14 //pin 04
#define RXD61 14 //pin 04
#define RXD62 14 //pin 04
#define RXD63 14 //pin 04
#define RXD64 14 //pin 04
#define RXD65 14 //pin 04
#define RXD66 14 //pin 04
#define RXD67 14 //pin 04
#define RXD68 14 //pin 04
#define RXD69 14 //pin 04
#define RXD70 14 //pin 04
#define RXD71 14 //pin 04
#define RXD72 14 //pin 04
#define RXD73 14 //pin 04
#define RXD74 14 //pin 04
#define RXD75 14 //pin 04
#define RXD76 14 //pin 04
#define RXD77 14 //pin 04
#define RXD78 14 //pin 04
#define RXD79 14 //pin 04
#define RXD80 14 //pin 04
#define RXD81 14 //pin 04
#define RXD82 14 //pin 04
#define RXD83 14 //pin 04
#define RXD84 14 //pin 04
#define RXD85 14 //pin 04
#define RXD86 14 //pin 04
#define RXD87 14 //pin 04
#define RXD88 14 //pin 04
#define RXD89 14 //pin 04
#define RXD90 14 //pin 04
#define RXD91 14 //pin 04
#define RXD92 14 //pin 04
#define RXD93 14 //pin 04
#define RXD94 14 //pin 04
#define RXD95 14 //pin 04
#define RXD96 14 //pin 04
#define RXD97 14 //pin 04
#define RXD98 14 //pin 04
#define RXD99 14 //pin 04
#define RXD100 14 //pin 04
// declarando variables
int ocupado = 0;
int libre = 0;
float tiempo_de_espera, distancia; // creamo una variable de flotante: es decir, nos puede dar resultado en decimales.
float tiempo_de_espera, distancia; // creamo una variable de flotante: es decir, nos puede dar resultado en decimales.
bool estado=libre;
bool estado=libre;
bool estado=libre;
bool estado=libre;

```

**Figura 9.** Código Arduino® definición de variables  
**Fuente:** Elaboración propia en Plataforma de Arduino®

Para enviar mensajes de texto a determinado número de celular empleando esta arquitectura y el módulo GSM SIM800L se crea una función "void mensaje", la cual emplea ciertos comandos AT que se aprecian en la Figura 10.

```

2_sensores_parquadero_tesis_firebase Arduino
Archivo Editar Programa Herramientas Ayuda
2_sensores_parquadero_tesis_firebase
void mensaje_sms()
{
  Serial.println("Enviando SMS...");
  S28000.print("AT+CMGF=1"); //Comando AT para mandar un SMS
  delay(100);
  S28000.print("AT+CMGS=\"+573147478554\""); //Número al que vamos a enviar el mensaje
  delay(500);
  S28000.print("Hola te informamos que el parqueadero DHAJAC este totalmente lleno");// Texto del SMS
  delay(100);
  S28000.print((char)26); //Comando de finalizacion -Z
  delay(100);
  S28000.println();
  delay(1000); // Esperamos un tiempo para que envíe el SMS
  Serial.println("SMS enviado");
}

```

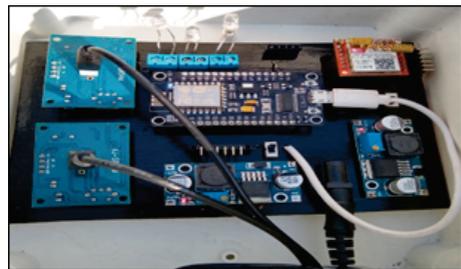
**Figura 10.** Configuración comandos AT para envío de mensaje de texto  
**Fuente:** Elaboración propia en Plataforma de Arduino®

En última instancia, se logra validar la comunicación efectiva de todos los componentes del prototipo a través del desarrollo de la App, que es la interfaz entre el usuario y el sistema. De esta manera, se demuestra como en la actualidad el IoT y los servicios OTT se integran de forma efectiva e intuitiva en el desarrollo de soluciones.

## RESULTADOS

Desde el punto de vista de las competencias académicas, tras el desarrollo de la metodología propuesta, se encuentra en primera instancia falencias de apropiación metodológica en investigación, característica descubierta a través de circunstancias de recelo por parte de los estudiantes quienes hacen parte de este equipo de trabajo al enfrentarse al autoaprendizaje de nuevos conocimientos tanto prácticos como teóricos e interdisciplinarios. Esta situación se pudo superar con procesos de acompañamiento en la construcción y reconstrucción del proyecto que tuvo como consecuencia un mayor tiempo para lograr su culminación.

Con respecto a los resultados en la implementación de la arquitectura propuesta, tal como se aprecia en las figuras 11 y 12, se puede afirmar que fueron exitosos porque, según los requerimientos, se logra con una sola variación conforme a la propuesta inicial, la cual tiene que ver con el módulo GSM/GPRS SIM900 SHIELDSE que presentó problemas de conectividad con la red local por características propias de fabricación, por lo tanto, se llega a la conclusión de remplazarse por el módulo GSM SIM800L.



**Figura 11.** Tarjeta electrónica del prototipo finalizada  
**Fuente:** Tomado de (Benachi Arce & Daza Bolaños, 2019)

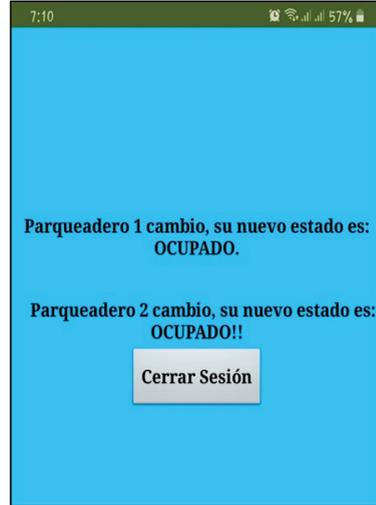


**Figura 12.** Prototipo finalizado  
**Fuente:** Tomado de (Benachi Arce & Daza Bolaños, 2019)

42 Pasando a los resultados de validación de la solución basada en IoT y los servicios OTT para informar la disponibilidad de estacionamientos de la UNIAJC sede norte: Fue necesario el desarrollo de una App que cumple la función de interfaz y presenta características básicas como la creación de usuario y contraseña, con la finalidad del usuario poder iniciar sesión y el sistema dar respuesta mostrando el estado real en el que se encuentra la zona de estacionamiento a través de un mensaje que dice "Parqueadero 1 o 2 cambió. Su nuevo estado es Libre u Ocupado", como se ve en las siguientes figuras.



**Figura 13.** Vista general de la interfaz desarrollada con la que interactuará el usuario  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)



**Figura 14.** Notificación del estado real del parqueadero en la interfaz desarrollada  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)

Los análisis de esta aplicación arrojaron que el tiempo de respuesta en ocasiones se ve afectado por la calidad de señal de la red Wi-Fi, ya que la carga de datos a la plataforma Firebase y la App dependen de la velocidad del internet, que es afectada por variables locales de la red en la que se apoya el dispositivo o la velocidad que está ofreciendo el proveedor del servicio.

Las pruebas para verificar la funcionalidad del sistema completo requieren la ejecución de la aplicación llamada MIT AI2 Companion, que es una herramienta para simular en tiempo real el progreso de la App, con los siguientes resultados:

- En el 100 % de las pruebas realizadas los valores de la distancia que debe detectar el sensor al acercarse un vehículo son correctos, visualizándose en el monitor serial de Arduino® y los estados de las salidas digitales en tiempo real.
- El comportamiento de respuesta de la base de datos desarrollada en la plataforma Firebase a la App, así como el

envío del mensaje de texto a los usuarios de disponibilidad de espacio en el parqueadero fueron satisfactorios.

La App permite de forma confiable al usuario desde su móvil, conocer el estado de disponibilidad de cupos en el parqueadero de la UNIAJC sede norte, como se muestra a continuación con las diferentes situaciones posibles las cuales fueron comprobadas en sitio, tomando como referente solo dos espacios de parqueadero. También es importante aclarar, antes de exponer los resultados de estas pruebas, que el sensor 1 se encarga de detectar lo que sucede en el denominado parqueadero 1 y el sensor 2 en el denominado parqueadero 2 respectivamente.



**Figura 15.** Inicio de primera prueba  
**Fuente.** Pruebas parqueadero UNIAJC sede norte

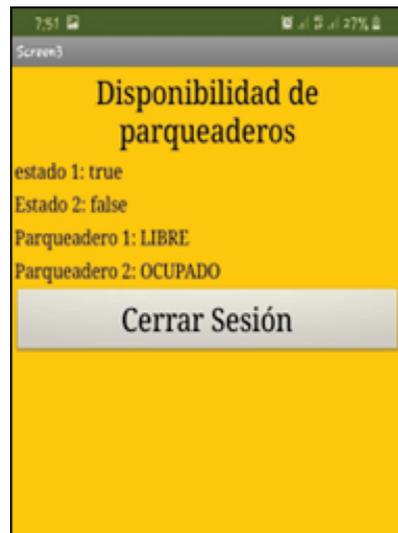


**Figura 16.** Inicio de segunda prueba  
**Fuente.** Pruebas parqueadero UNIAJC sede norte

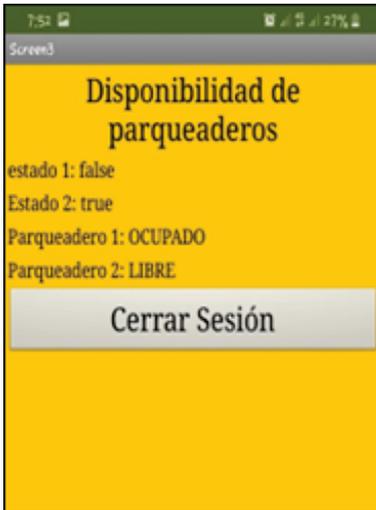
**La primera prueba comienza ocupando uno de los dos cupos del parqueadero y dejando el otro libre.** Para esta situación, el parqueadero 1 lo ocupó el vehículo de color violeta y el parqueadero 2 el vehículo de color plateado, dando como resultado una visualización al usuario de la App como se presenta en la figura 17 que muestra los estados de los sensores y de los parqueaderos respectivamente.

El estado 1 hace referencia al sensor 1, en este caso existe una distancia mayor a 80 cm, que lo confirma con un (true) y así arroja el mensaje "Parqueadero 1 Libre" y el sensor 2 detecta una distancia menor a 80 cm leyendo con un (false), para dar como mensaje "Parqueadero 2 Ocupado".

**En la segunda prueba, al igual que en la anterior, se varía el estado de los sensores, ubicando el vehículo violeta a una distancia menor a 80 cm y el vehículo plateado ubicándolo a una distancia mayor a 80 cm., dando como resultado el "Parqueadero 1 Ocupado" y el "Parqueadero 2 Libre" (ver Figura 18).**

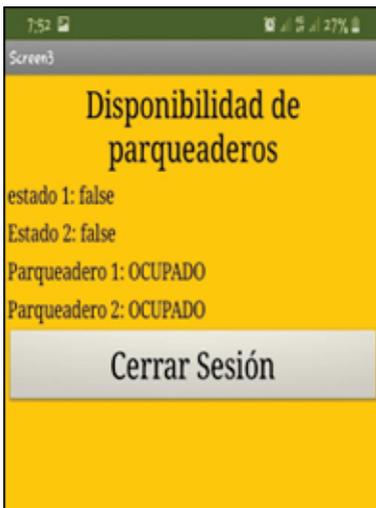


**Figura 17.** Visualización prueba número 1  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)

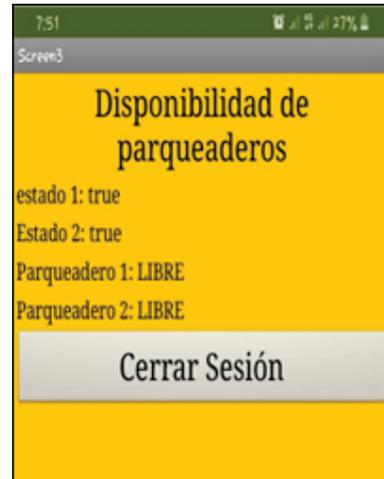


**Figura 18.** Visualización prueba número 2  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)

La tercera prueba implica ocupar los dos parqueaderos con los vehículos, llevando de esta manera ambos sensores a un estado de (false), reflejados en tiempo real en el dispositivo Android como se puede observar en la Figura 19.



**Figura 19.** Visualización prueba número 3  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)



**Figura 20.** Visualización prueba número 4  
**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)

Para la última prueba se realizó el movimiento de ambos vehículos a una distancia mayor a 80 cm. Con el fin de que el sensor enviara un true para ambos estados y enviara un mensaje por medio de la plataforma de App Inventor que dijera "Parqueadero 1 Libre" y "Parqueadero 2 Libre", como se ve reflejado en la Figura 20.

		Plataforma Firebase		Red móvil Envío SMS	Aplicación móvil	
	Acción	Estación 1	Estación 2		Estación 1	Estación 2
Sensor 1	Distancia > 80 cm	True		No	Libre	
	Distancia < 80 cm	False		Si	Ocupado	
Sensor 2	Distancia < 80 cm		False	Si		Ocupado
	Distancia > 80 cm		True	No		Libre
	Indicador led	Ok	Ok	N/A	Ok	Ok
	Tiempo reacción	1sg	1sg	2sg	1sg	1sg
	% de error	0%	0%	0%	0%	0%

**Tabla 6.** Resultados de funcionalidad del sistema

**Fuente.** Tomado de (Benachi Arce & Daza Bolaños, 2019)

Con la investigación realizada se puede concluir que los sistemas basados en IoT cada día tienen más acogida por la población no solo estudiantil sino en la población general, y es de esperarse, ya que estos sistemas convierten muchas ideas en proyectos innovadores que ayudan al crecimiento no solo de los estudiantes, sino que también ayudan a mejorar la calidad de vida de muchas personas, específicamente a solucionar problemáticas que tienden a afectar a toda una comunidad.

El reto de trabajar con las tecnologías propuestas aporta gran cantidad de conocimiento no solo en tema de innovación, también ayudan al estudiante a indagar y definir qué tipos de tarjetas procesadoras o plataformas digitales son las más adecuadas para desarrollar un proyecto IoT, de acuerdo con los alcances que este propone.

46 Se debe trabajar en mejorar competencias en metodologías ágiles y desarrollo de software en los estudiantes para que tengan una mejor adaptación a lo que se reconoce como industria 4.0, lo cual se demostró a través de los desafíos que este proyecto demandó cuando se debe articular la funcionalidad de sus componentes y esta se hace hoy en día por programación embebida o en la nube. Así que se puede proponer a esta situación el desarrollo de una estrategia que involucre los semilleros de investigación del programa académico para motivar a los estudiantes al progreso de estas competencias.

En términos de tecnologías y compatibilidad de estas con los servicios OTT, se debe realizar un mejor estudio de validación y compatibilidad, que ayude a futuros desarrollos a emplear de manera efectiva y en especial que sea apta para poder evolucionar el prototipo a un producto mínimo funcional y convertirse, porque no, en un desarrollo industrial.

En el desarrollo del proyecto se pudo evidenciar que hay muchas opciones para crear un sistema de monitoreo y de censado, pero no todas garantizan el buen funcionamiento del proyecto. En particular para casos que presenten requerimientos similares se concluye que se puede recomendar y es una buena opción, por capacidad, tamaño, economía y los resultados en la validación del sistema IoT, la tarjeta NodeMCU y el módulo de envío de mensajes de texto "SIM800L" para la comunicación por GSM.

---

## REFERENCIAS

**Benachi Arce, E. y Daza Bolaños, C. (2019).**

Desarrollo tecnológico de una solución basada en IoT para validar la disponibilidad de estacionamientos de la UNIAJC en la sede norte. Cali: Institución Antonio José Camacho.

**Casas Toris, U. J., Carrillo Arteaga, A. N., y Rodríguez Aguilar, R. M. (junio de 2020).**

Revisión crítica de sustentabilidad o sostenibilidad, en literatura sobre ciudades inteligentes. CoPaLa, 5(9), 67-90.

**Coldfire Electrónica. (s.f.).** Tarjeta de desarrollo Nodemcu v3-esp8266.

<https://www.coldfire-electronica.com/esp/item/356/tarjeta-de-desarrollo-nodemcu-v3-esp8266>

**Departamento Administrativo de Planeación. (2019).** Plan Integral de Movilidad Urbana de Cali-Visión 2030.

<https://planeacion.cali.gov.co>  
[https://planeacion.cali.gov.co/pimu/nomenclatura/Anexo%201%20Plan%20de%20Accion/Anexo1\\_Plan\\_de\\_Accion\\_PIMU%20%28v.may-15-2019%29.pdf](https://planeacion.cali.gov.co/pimu/nomenclatura/Anexo%201%20Plan%20de%20Accion/Anexo1_Plan_de_Accion_PIMU%20%28v.may-15-2019%29.pdf)

**Elpais.com.co. (jueves de 07 de 2012).** Falta de parqueaderos, problema que infarta las calles y andenes de Cali. El País.

**Friss de Kereki Guerrero, I. (2003).** Modelo para la creación de entornos de aprendizaje basados en técnicas de gestión del conocimiento (Tesis Doctoral). Universidad Politécnica de Madrid.

**Gakenheimer, R. (1998).** Los problemas de la movilidad en el mundo en desarrollo. EURE (Santiago), 24(72), 33-52.  
<https://dx.doi.org/10.4067/S0250-71611998007200002>

**Programa Cali Cómo Vamos. (2021).**

Informe Anual de Calidad de vida Cali cómo vamos 2021.

[https://www.calicomovamos.org.co/\\_files/ugd/ba6905\\_2a6681f601a5475ebffbc95788a108e.pdf](https://www.calicomovamos.org.co/_files/ugd/ba6905_2a6681f601a5475ebffbc95788a108e.pdf)

**Martínez Fernández, N. F. (2016).** Sistema de alarmas, transmisión y monitoreo de datos aplicado a la medición de variables con redes GSM/GPRS. Obtenido de <https://repositorio.ucm.edu.co/handle/10839/1495>

**Rosero A., Y. (2019).** Diseño de sistema de seguridad domótico de cerrojo. Cali.

**Sanabria Pulido, P. P., & Rodríguez Caporalli, E. (2019).** Transformar lo público: perspectivas sobre la reforma administrativa de Santiago de Cali de 2016. Editorial Universidad Icesi - Sello Editorial Javeriano Cali.

**Sandoval Quintero, M. (2018).** Entorno de desarrollo App Inventor.

[https://www.researchgate.net/profile/Mayra\\_Sandoval\\_Quintero/publication/318201021/figure/fig4/AS:512875641147396@1499290489920/Entorno-de-desarrollo-App-Inventor.png](https://www.researchgate.net/profile/Mayra_Sandoval_Quintero/publication/318201021/figure/fig4/AS:512875641147396@1499290489920/Entorno-de-desarrollo-App-Inventor.png)