

IMPLEMENTACIÓN DEL SOFTWARE DE GESTIÓN PARA EL PICO Y PLACA QUE  
PERMITE LA MOVILIDAD DEL PARQUE AUTOMOTOR EN LA UNIVERSIDAD  
CATÓLICA DE ORIENTE (UCO).



DANIEL ALEJANDRO ESQUIVEL CORREA

UNIVERSIDAD CATÓLICA DE ORIENTE

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

RIONEGRO

2021

IMPLEMENTACIÓN DEL SOFTWARE DE GESTIÓN PARA EL PICO Y PLACA QUE  
PERMITE LA MOVILIDAD DEL PARQUE AUTOMOTOR EN LA UNIVERSIDAD  
CATÓLICA DE ORIENTE (UCO).

Daniel Alejandro Esquivel Correa

Trabajo para obtener el título de ingeniero de Sistemas

**Asesor**

Alirio Antonio Gutiérrez Quintero

Universidad Católica de Oriente

Facultad de Ingenierías

Programa de Ingeniería de Sistemas

Rionegro

2021

**Nota de aceptación:**

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

**Dedicatoria:**

A mi madre

Por ser el pilar fundamental en mi proceso de formación, por su apoyo incondicional en toda mi carrera, porque siempre busco la forma de que yo continuara mis estudios mientras conseguía empleo, por sus consejos y por siempre creer que iba a lograr grandes cosas.

A mi padre

Porque cuando tuve momentos de crisis siempre me dio una voz de aliento para que pudiera continuar, al igual que me apoyo durante mis primeros semestres en Medellín.

A mi abuela

Por siempre colaborarme en todos los aspectos, por confiar en mis capacidades, por motivarme siempre a ser una mejor persona, un mejor profesional y por nunca abandonarme en este sueño.

A mi abuelo

Mi ángel en el cielo, el cual desde allí ve mi crecimiento profesional y humano. Fuiste mi ejemplo a seguir, un hombre enfocado y echado para adelante, que desde pequeño me enseñó valores, me dio todo su amor y me apoyo en todos mis objetivos.

A Yudy

Mi apoyo incondicional, la persona que siempre me dio una voz de aliento, que en los momentos difíciles donde pensé no continuar, siempre estuvo ahí para darme una mano, apoyarme y hacerme ver que podría lograr todo lo que me propusiera.

A mis amigos

Por siempre confiar en mis capacidades, apoyarme en este sueño y por darme la oportunidad de crecer poco a poco siendo parte de sus proyectos personales.

## **Agradecimientos**

Quiero agradecer a mis padres, amigos y familiares quienes me han brindado siempre su apoyo a lo largo de mi vida.

Agradezco a mi amiga Elizabeth Martínez, quien me apoyo durante la construcción del prototipo final donde se simula el ingreso a los estacionamientos del campus universitario. Muy agradecido por su paciencia, por invertir su tiempo en los días de traspasado y por brindarme su sincera amistad.

Agradezco a mi amigo Cristian Aguirre Baena, por su apoyo en la construcción del resumen y traducción del mismo. Siempre voy a estar agradecido porque en todos los años de conocerlo me ha enseñado que todo lo que uno se propone lo puede lograr si tiene disciplina, humildad, constancia y sobre todo si se hacen las cosas con amor.

Al docente Alirio Antonio Gutiérrez Quintero, asesor del presente trabajo de grado, por toda su colaboración, comprensión y apoyo durante el proceso de titulación.

A todos mis profesores a lo largo de mis estudios universitarios, quienes supieron compartir siempre su conocimiento de la mejor manera y que siempre me animaron a hacer las cosas de forma excelente.

Finalmente agradezco a todas las personas que de una u otra forma me brindaron siempre su ayuda, me dieron su voz de aliento y creyeron que podría ser un gran profesional.

## RESUMEN

Varias organizaciones están tratando de contrarrestar los efectos adversos de la movilidad vehicular en la actualidad y en el presente trabajo, se aborda el control del pico y placa para el ingreso a la Universidad Católica de Oriente, ubicada en el municipio de Rionegro, Colombia. Se tuvieron en cuenta las dificultades para el estacionamiento de vehículos, en su mayoría generadas por la ocupación de un 100% de los parqueaderos. En una serie de observaciones, se evidenciaron riesgos de distinta índole, debido a los tiempos que requiere el usuario para estacionarse, entre los que se incluye el ingreso tardío a las actividades académicas y afectaciones por golpes entre vehículos.

El objetivo de estudio, es implementar un sistema de control para la gestión del ingreso al campus universitario mediante un lector RFID. Este dispositivo permite verificar la asignación diaria del pico y placa, de manera que se autorice o se niegue el ingreso del vehículo; la validación se hace mediante consulta en una base de datos MySQL que alberga las placas de los vehículos de los miembros de la institución antes mencionada. Adicionalmente, se creó un usuario administrador manejado por el personal encargado de las zonas de ingreso, quienes autorizan el acceso de terceros que vayan a permanecer en las instalaciones durante cortos periodos de tiempo.

Para la codificación del software, se utilizó Arduino, un IDE de código abierto que permite desarrollar de forma libre los componentes que establecen la conexión a la base de datos MySQL. Así mismo, este IDE permite interactuar con el hardware y compilar el software que contiene toda la lógica de control del sistema.

Se concluyó que la implementación de este sistema ayudará a reducir el aforo total de las celdas del parqueadero del campus universitario en un 80%, lo que minimizará los tiempos de estacionamiento, además de prevenir daños a otros vehículos mientras se estacionan en espacios reducidos o en zonas no autorizadas.

## **ABSTRACT**

Several organizations are trying to counteract the adverse effects of vehicular mobility today, we address the control of the *Pico y Placa* (rush hour traffic restriction) to enter the headquarters of the Universidad Católica de Oriente, located in the municipality of Rionegro, Colombia. The difficulties for parking vehicles, mostly generated by the occupation of 100% of the parking spaces, were taken into consideration. In a series of observations, risks of different types were evidenced, due to the time needed for parking, including late entry to academic activities and damage caused by collisions between vehicles.

The objective of the study is to implement, a control system was designed in order to manage the entrance to the campus by using an RFID reader. This device allows to check the daily assignment of the *Pico y Placa*, so that the entry of a vehicle can be authorized or denied. This validation is done by consulting a MySQL database that stores the license plate numbers of vehicles that belong to the members of the above-mentioned university. Additionally, an administrator user was created, which is managed by the personnel in charge of the entrance areas; they authorize the access of third parties who will remain in the facilities for short periods of time.

For the codification of the software, Arduino was used, an open-source IDE that allows the free development of the components that set up the connection to the MySQL database. Moreover, this IDE allows the interaction with the hardware and it compiles the software that contains all the control logic of the system.

It was concluded that the implementation of this system will help reduce the total use of the university campus parking spaces by 80%. It will minimize the time to park, in addition to preventing damage to other vehicles while parking in narrow spaces or in unauthorized areas.

### **Palabras claves**

*Lector RFID, MySQL, pico y placa, sistema de control, Arduino, ESP32, servomotor, Eeprom, LCD, Wifi, software, IDE, vehículos, zonas de ingreso, hardware, compilación.*

## Contenido

RESUMEN .....	6
ABSTRACT .....	7
CAPÍTULO 1: ANTECEDENTES.....	10
1.1 Antecedentes nacionales.....	11
1.2 Antecedentes internacionales.....	13
CAPÍTULO 2: PLANTEAMIENTO DEL PROBLEMA.....	15
CAPÍTULO 3: JUSTIFICACIÓN .....	19
CAPÍTULO 4: OBJETIVOS.....	21
<b>4.1 General:</b> .....	21
<b>4.2 Específicos:</b> .....	21
CAPÍTULO 5: MARCO TEÓRICO.....	21
5.1 Fundamento teórico.....	21
5.2 Metodología .....	28
5.2.1 Método experimental .....	28
5.3 POBLACIÓN Y MUESTRA .....	29
5.3.1 Población.....	29
5.3.2 Muestra .....	29
CAPÍTULO 6: DISEÑO METODOLÓGICO .....	30
CAPÍTULO 7: RESULTADOS .....	31
7.1 Implementación y Desarrollo .....	31
7.1.1. Modelo del funcionamiento del Circuito .....	31
7.1.2. Configuración de Microcontrolador Pinout ESP32 .....	32
7.1.3. Configuración de LCD ESP32.....	34
7.1.4. Configuración de identificador RFID.....	38
7.1.5. Configuración de WIFI ESP32 .....	42



7.1.6. Configuración de EEPROM.....	45
7.1.7. Configuración de MySQL .....	47
7.1.8. Configuración de Servomotor .....	49
7.1.9. Sistema de acceso .....	51
<b>7.2. Funcionamiento .....</b>	<b>60</b>
7.2.1 Introducción .....	60
7.2.2 Entrada a la institución .....	60
7.2.3 Verificación de ingreso.....	60
7.2.4 Autorización de ingreso .....	61
7.2.5 Ingreso no permitido .....	63
7.2.6 Autorización de ingreso por el administrador .....	66
CONCLUSIONES .....	67
RECOMENDACIONES.....	69
BIBLIOGRAFÍA .....	69

## **CAPÍTULO 1: ANTECEDENTES**

- **TÍTULO:** EL PROBLEMA DE MOVILIDAD EN CAMPUS UNIVERSITARIOS. CASO APLICADO: UNIVERSIDAD DE ANTIOQUIA<sup>1</sup>
- **AUTOR:** DIANA PATRICIA MORENO PALACIO
- **AÑO DE PUBLICACIÓN:** 2008
  
- **TÍTULO:** EL EXCESO DE DEMANDA DE PARQUEADEROS EN LA UNIVERSIDAD EAFIT: ANÁLISIS DE ESTRATEGIAS SOBRE LA GESTIÓN DE LA DEMANDA<sup>2</sup>
- **AUTORES:** ALEJANDRO MORA MORA Y JUAN DAVID RAMÍREZ GALVIS
- **AÑO DE PUBLICACIÓN:** 2008
  
- **TÍTULO:** ANÁLISIS DEL “PICO Y PLACA” COMO RESTRICCIÓN A LA CIRCULACIÓN VEHICULAR EN MEDELLÍN BASADO EN VOLÚMENES VEHICULARES<sup>3</sup>
- **AUTORES:** JOHN JAIRO POSADA HENAO, VIVIANA FARBIARZ CASTRO Y CARLOS ALBERTO GONZALES CALDERÓN
- **AÑO DE PUBLICACIÓN:** 2010
  
- **TÍTULO:** ANÁLISIS DE LA APLICACIÓN DEL PICO Y PLACA EN LA CIUDAD DE QUITO<sup>4</sup>
- **AUTORES:** ABEL POLIVIO REMACHE COYAGO Y SANTIAGO FERNANDO CELI ORTEGA
- **AÑO DE PUBLICACIÓN:** 2017

---

<sup>1</sup> (Palacio, repositorio.unal.edu.co, 2008)

<sup>2</sup> (Galviz, 2020)

<sup>3</sup> (JOHN JAIRO POSADA HENAO, 2020)

<sup>4</sup> (Abel Polivio Remache Coyago, 2020)

- **TÍTULO:** ANÁLISIS Y SOLUCIÓN AL CONGESTIONAMIENTO VEHICULAR EN HORAS PICO UTILIZANDO UNA APLICACIÓN MÓVIL CON GPS<sup>5</sup>  
**AUTORES:** PATRICIO DAVID MENDOZA SOLÓRZANO Y CRISTHIAN VILLACIS VARGAS  
**AÑO DE PUBLICACIÓN:** 2014
- **TÍTULO:** CARACTERIZACIÓN DE LA MOVILIDAD EN BICICLETA EN EL CAMPUS UNIVERSITARIO RODRIGO FACIO, UNIVERSIDAD DE COSTA RICA<sup>6</sup>  
**AUTOR:** JOSÉ FERMÍN ROZAS RODRÍGUEZ  
**AÑO DE PUBLICACIÓN:** 2017

### 1.1 Antecedentes nacionales

**Autor:** Diana Patricia Moreno Palacio

**Año de publicación:** 2008

En esta investigación se realizó un análisis del problema que tiene la movilidad en los campus universitarios enfocado en la Universidad de Antioquia, como metodología de investigación se aplicaron unas encuestas con el método de preferencias declaradas (PD) a los estudiantes de esta universidad. Con esta metodología se pudo evidenciar que al campus ingresan 4200 vehículos en un día normal de la semana, de los cuales 2554 son automóviles y 1593 son motocicletas. Con una encuesta de preferencias reveladas que se hizo a los conductores se pudo determinar que la hora de mayor congestión era a las 10:30 a.m. en donde se ocupaban 837 parqueaderos, es decir el 88.95% de la totalidad, que en este caso el 100% son 941 parqueaderos. En la tarde, en el horario de 04:00 p.m. se obtuvo que ingresaban 689 automóviles, lo cual es 73.22% de ocupación de los parqueaderos siendo la franja horaria de menor congestión, adicionalmente se encontró que el tiempo promedio de parqueo por persona es de 3.2 horas, ocasionando esto una alta demanda de ocupación, pudiendo llegar al 100% de su capacidad.

---

<sup>5</sup> (Vargas, 2020)

<sup>6</sup> (Rodríguez J. F., 2020)

Otro resultado que se pudo obtener es que a medida que pasa el tiempo, la cantidad de motocicletas sigue aumentando, ya que estos usuarios son los que permanecen en la universidad todo el día, lo cual demuestra un uso inadecuado de los parqueaderos, y por lo que otros usuarios parquean en zonas no permitidas ya que este llega a su capacidad del 100%. Sin embargo, gracias a la implementación del pico y placa que se empezó a aplicar en el campus universitario desde el año 2007, solo se permite el ingreso del 80% de la capacidad de los parqueaderos, permitiendo un decrecimiento en la congestión vehicular.

Gracias a este estudio se pudo concluir que las motocicletas son los vehículos que más ingresan a la universidad, debido a su facilidad económica para obtener una. Basado en estos resultados se puede ejercer un mayor control sobre estas y así mantener la congestión vehicular en un bajo rango, más que todo en las horas pico.

**Autores:** Alejandro Mora Mora y Juan David Ramírez Galvis

**Año de publicación:** 2015

En este trabajo se estudió el exceso de demanda de parqueaderos en la universidad de EAFIT, como metodología de investigación se utilizaron modelos econométricos de decisión discreta, los cuales permitían analizar la elección que, hacía cada miembro de la institución en usar su vehículo para llegar al campus, estuviera o no activa la medida de restricción del pico y placa. Se obtuvo que en la universidad de EAFIT ingresan diariamente cerca de 18000 personas a la semana, de los cuales 3600 son ingresos de vehículos. A parte de esto la universidad también estima que el 37% de estos vehículos ingresan más de una vez al campus, lo que eleva el promedio de ingresos en vehículo a 13000, generando un mayor número de ocupación en las zonas de parqueo y una congestión vehicular al ingreso y al momento de estacionar, ya que, según resultados obtenidos, las personas que ingresan al campus, se pueden demorar aproximadamente 24 minutos en poder estacionar su vehículo debido a la congestión vehicular.

El campus de esta universidad cuenta con 1038 celdas para estacionar, y esto genera un desequilibrio, ya que no hay suficientes lugares para poder estacionar.

Como conclusión en esta investigación, se evidencia que es posible mejorar la distribución de los parqueaderos y la medida de pico y placa, para ayudar al fácil ingreso en el campus y reducir los tiempos de espera al momento de estacionar los vehículos.

**Autores:** John Jairo Posada Henao, Viviana Farbiarz Castro y Carlos Alberto Gonzáles Calderón

**Año de publicación:** 2010

Este trabajo trata sobre la implementación del pico y placa en Medellín para disminuir la congestión vehicular en las horas pico. Como metodología de investigación se tomó una estación maestra de cada corredor vial y se analizaron los ADR de años anteriores, con esto se observa el comportamiento del tráfico vehicular y se hace una comparación entre las horas pico y las horas normales, para evidenciar si sí hay mejoras con la implementación de la restricción de pico y placa. A partir de estos resultados se permite observar que gracias a la restricción de pico y placa hay una reducción del 18% de volumen en circulación de vehículos en las horas pico, lo cual permite una reducción del 20% en el flujo vehicular, ayudando a reducir los tiempos de espera que serían mucho más elevados si esta medida de pico y placa no se implementara.

## **1.2 Antecedentes internacionales**

**Autor:** Abel Polivio Remache Coyago y Santiago Fernando Celi Ortega

**Año de publicación:** 2017

En esta tesis los autores analizan la aplicación del pico y placa en la ciudad de Quito, lo que pretende fomentar el uso del transporte público para disminuir el tiempo de viaje gracias a que hay una menor cantidad de vehículos circulando.

Como metodología de estudio se contrató a unos especialistas de la facultad de arquitectura de la Universidad Central del Ecuador para estudiar todo lo que tenía que ver referente a la congestión vehicular y que así ellos propusieran una medida de mejora para disminuir esta congestión, luego de este estudio, se inició con la implementación de restricción de pico y placa de los cuales se evaluaron los resultados en dos fases, una luego de 3 meses de iniciar la medida y otra fase a los 6 meses. Como resultados se obtuvo que las principales horas de congestión vehicular eran a las 06:00 y a las 18:00 y que gracias a la restricción del pico y placa hubo una reducción del 12% en vehículos particulares, un 17% en motocicletas y un 25% para transporte de carga pesada. Gracias

a estos resultados la restricción del pico y placa mostró una reducción del 8% de tráfico en la mañana y un 12% en la tarde. Además, gracias a esta medida se puede evidenciar una disminución en la contaminación del medio ambiente, debido a la disminución en la velocidad de los vehículos.

**Autores:** Patricio David Mendoza Solórzano y Cristhian Ángel Villacis Vargas

**Año de publicación:** 2014

En esta investigación se quiere dar solución a la congestión vehicular en horas pico por medio de una aplicación móvil que utilice el GPS. Esta aplicación pretende mejorar el sistema de transporte enviando una ruta más rápida por medio de una red social, lo que permite llegar en menor tiempo al lugar que se necesita.

Como metodología de investigación se analizó el problema que estaba generando la congestión vehicular por la gran cantidad de vehículos transitando, se utilizó el método deductivo para entender el gran problema crítico que tenía la circulación vehicular y así resolver la problemática y ayudar a mejorar la medida por medio de la aplicación. Esta aplicación ayudó a resolver en un porcentaje alto la congestión vehicular, al permitir a las personas que necesitaban llegar rápido, obtener unas buenas rutas para disminuir su tiempo de llegada.

**Autor:** José Fermín Rozas Rodríguez

**Año de publicación:** 2017

En este trabajo de grado, el autor pretende incentivar el uso de la bicicleta para el ingreso al Campus universitario, ya que la alta demanda de vehículos genera gran congestión al momento de ingresar a la universidad. Como metodología de investigación se realizó un cuestionario electrónico que fue publicado en un portal de internet, esta encuesta fue contestada por 5323 estudiantes, con esta encuesta se pretende conocer cuál es la cantidad de personas que estaría dispuesta a desplazarse en bicicleta con el fin de disminuir la congestión vehicular y cuál es la cantidad de personas que ya se desplazan en bicicleta hacia el campus. Como resultado se pudo obtener que el 1% ya se desplaza en bicicleta y que se puede mejorar la movilidad en bicicleta como medio de transporte en el campus universitario.

Como conclusión se puede apreciar que si todos los estudiantes o al menos la gran mayoría decidieran desplazarse en bicicleta esto ayudaría a disminuir la cantidad de vehículos en la universidad, la congestión vehicular en las horas pico y aportaría un gran cambio para el medio ambiente, ya que la contaminación disminuiría y se tendría un campus universitario mucho más saludable y libre de contaminación.

## **CAPÍTULO 2: PLANTEAMIENTO DEL PROBLEMA**

Teniendo en cuenta que la universidad no tiene un control estricto con las normas de pico y placa establecidas, se pretende con el desarrollo del proyecto propuesto, el desarrollo y la implementación de una aplicación que lleve a cabo un proceso de control automático, para garantizar el cumplimiento de la norma de pico y placa que se impone en la UCO en sus respectivos días, para el ingreso a institución y la utilización del servicio de parqueo.

La aplicación de control automático para el acceso vehicular, que asegure el cumplimiento de la norma de pico y placa en las instalaciones de la UCO, se propone teniendo en cuenta el modelo TAG que utilizan los vehículos registrados en Flypass para el pago automático de peajes, este sistema llamado “TAG” tiene asociado un código a la placa del vehículo que está registrado en la base de datos, es decir, una placa electrónica. Para el caso de la aplicación de la UCO, la placa electrónica estaría asociada al carné institucional propio de cada usuario, el cual permite validar el ingreso del solicitante a la institución.

Alrededor del mundo existen universidades donde también se presentan dificultades de movilidad al momento de encontrar un sitio donde parquear, una de ellas es la Universidad de Salamanca, de la ciudad de Salamanca - España. En un estudio que se realizó en esta universidad, se pudo evidenciar el problema que genera la movilidad en transporte privado al momento de ingresar al campus universitario, esta universidad posee 1.100 celdas de aparcamiento + 18 para minusválidos que pertenecen a la universidad, pero en este trabajo se explica que en un día normal, en la mañana se hacen aproximadamente 2.500 viajes, y en la tarde aproximadamente 1.800 viajes<sup>7</sup>, lo cual

---

<sup>7</sup> (Rodríguez J. F., 2020)

evidencia que hay un desborde por encima de la capacidad máxima que tienen los parqueaderos de esta universidad.

Esta universidad busca mejorar su movilidad, ya que consideran que tener tantos vehículos circulando en el campus presenta un riesgo para las personas que llegan al campus caminando, en bicicleta o en otros medios de transporte diferentes, ya que, al haber tanta congestión vehicular al momento de ingresar al campus, se presentan aglomeraciones de vehículos en lugares que son únicamente para el paso de personas y llegada de transporte público.

En Latinoamérica también se presentan estos inconvenientes, en el campus universitario Rodrigo Facio, ubicado en la ciudad de San José de Costa Rica, se hizo un estudio donde se muestra que el 60% del campus universitario es ocupado por vehículos<sup>8</sup>, invadiendo lugares que no están destinados para parquear, y que perjudican a los miembros de la comunidad universitaria, ya que en esta institución no se ejerce un control de pico y placa o de restricción de movilidad para regular el tráfico dentro del campus.

A nivel nacional, hay otras universidades que también implementan la restricción de pico y placa en sus campus universitarios. El primer caso que podemos observar es el de la Universidad de Antioquia, la cual implementa el pico y placa programado durante todo el día, de esta manera solo se permite el ingreso del 80% de vehículos a su campus, este 80% se representa en 4200 vehículos, entre automóviles (2554) y motocicletas (1593).<sup>9</sup>

Otro caso que podemos encontrar es el de la Universidad EAFIT, ya que ellos también reportan una alta congestión de vehículos al momento de ingresar a los parqueaderos de su campus universitario, por lo cual la universidad aplica el sistema de pico y placa al igual que el cobro de parqueadero, sin embargo ellos reportan que su capacidad al día se excede en un 68%, dado que solo poseen 1038 celdas disponibles para distribuir entre sus usuarios y según el estudio que realizó el departamento de planta física de EAFIT en el año 2015, asumen que el uso de parqueaderos en

---

<sup>8</sup> (Henry Hernández Vega, 2020)

<sup>9</sup> (Palacio, Bdigital, 2020)



su campus universitario sería de aproximadamente 1750 usuarios por día<sup>10</sup>, lo cual evidencia un desequilibrio a la hora de llegar en vehículo propio.

Como estudiante y visitante de universidades como la Universidad Católica Luis Amigó (Funlam) y Politécnico Colombiano Jaime Isaza Cadavid, pude observar que para controlar la medida de pico y placa que establece cada universidad hay una alternativa, la cuál es que uno de los vigilantes observa en la portería placa por placa de los vehículos que ingresan a la universidad y da el aval para que ingresen o no al establecimiento.

En otras universidades, como lo es la UCO, ni siquiera hay un control así, hay una restricción de pico y placa anunciada, pero nadie se encarga de velar porque esto se cumpla, se puede observar que ingresan todos los vehículos y al no haber un control establecido, a las miembros de estas instituciones no les interesa cumplir esta medida.

Algunos de los principales problemas que se observan en la UCO, sobre la medida de pico y placa, para la utilización de las zonas de parqueo al interior de la institución, son:

El poco control que tiene la implementación del pico y placa para los vehículos dentro del campus, ya que la mayoría de los usuarios que utilizan el parqueadero, no respetan esta medida, dado que no hay conciencia y cultura sobre esto.

No se ejerce ningún control para esta medida por parte del personal en la zona de ingreso de la universidad. Pues en general en las horas de alto flujo vehicular de ingreso a las instalaciones de parqueo en la UCO, para el personal no es viable ejecutar para todos el control manual o por observación de aquellos usuarios que no tienen permitido el ingreso debido al control de pico y placa.

Al no poder aplicar efectivamente el control de pico y placa para el ingreso de vehículos a las zonas de parqueo de la institución, se generan diferentes inconvenientes, que afectan en diversa forma a la comunidad universitaria en general. Tales como:

---

<sup>10</sup> (Galviz, 2020)

- Altos niveles de congestión vehicular, según observaciones hechas en diferentes momentos y horas de la jornada de prestación del servicio de parqueo, se puede evidenciar que la mayoría de los usuarios de este servicio (estudiantes, docentes, empelados o visitantes de la UCO) no cumplen la norma de pico y placa para el control de los vehículos dentro del campus.
- Ocupación total permanente de las zonas de parqueo, durante la jornada de prestación del servicio, llegando al punto de que en el parqueadero de las motocicletas no encuentran más lugar, generando que los usuarios decidan incluso dejar estacionados sus vehículos de forma inadecuada, obstaculizando el paso de otros usuarios, que deben entrar o salir de los parqueaderos y en algunos otros casos, se ocasiona que algunos usuarios incluso utilicen espacios que no están destinados para este propósito, tanto al interior como en zonas externas aledañas a la institución, generando incomodidad para la comunidad, no solo al interior de la institución, sino en el exterior de la misma, ocasionando de esta forma eventuales disgustos de la comunidad vecina a la institución, expresada mediante quejas verbales o escritas ante las directivas de la institución o ante las autoridades municipales de tránsito y transporte, lo cual conlleva a que la institución como tal tenga eventualmente una imagen poco positiva ante la comunidad de vecinos.
- Falta de cultura y de respeto por las normas y políticas de control de acceso vehicular a nivel institucional y en las zonas aledañas a la institución y en zonas de acceso a la misma, frente al no cumplimiento del pico y placa, debido a que no es posible llevar a cabo el control correspondiente de cumplimiento de esta, por parte del personal de seguridad de la institución, ya que lo deben hacer manualmente y/o por observación directa de cada vehículo que debe ingresar.

Continuar con este problema de pico y placa en la UCO genera algunos problemas, ya que la alta congestión de flujo vehicular genera contaminación del medio ambiente, malestar e incomodidad para la comunidad en general interna y externa a la institución, se obstaculiza total o parcialmente la movilidad no solo vehicular, sino de la comunidad peatonal que debe circular al interior de la

institución en algunas zonas cercanas a la zona de parqueadero, lo cual genera altos niveles de estrés a los miembros de la comunidad universitaria, especialmente a los estudiantes, pues el ingreso en las horas picos, es demorado, ya que en dichas horas, usualmente se presenta el colapso de las zonas de parqueadero, lo cual genera tiempos de demora para ubicar un lugar disponible y retrasos importantes en su llegada a las aulas de clase o en su salida hacia sus lugares de destino; en algunas otras oportunidades el usuario de parqueo de motocicletas que llega, debe mover algunas de las que ya están estacionadas para abrir espacio y ubicar la suya propia, lo cual eventualmente puede conllevar a ocasionar daños en su propia motocicleta o en las que ya están estacionadas y deben reubicar.

Con esta investigación, se pretende aportar a la universidad una solución automatizada por medio de un software que controle este ingreso, al igual que concientizar a toda la comunidad universitaria, de que hay otras alternativas disponibles para asistir a la universidad, de esta manera se va a lograr una disminución en la contaminación del medio ambiente, como también se va a alivianar el ingreso de vehículos durante el día al campus universitario, permitiendo así una mayor tranquilidad dentro de la universidad y evitando congestión vehicular dentro de la misma.

El desarrollo de este proyecto deja a la universidad muy bien vista, ya que muestra el interés por ayudar a bajar la contaminación del medio ambiente, aportando para la creación de una nueva cultura personal e institucional, como también ayuda a mejorar la comodidad para los miembros de la institución. Con esta implementación la universidad puede ser vista como una universidad innovadora, apostándole al desarrollo del Innovamater, el cual siempre se ha mostrado como un proyecto amigable con el medio ambiente.

### **CAPÍTULO 3: JUSTIFICACIÓN**

La Universidad Católica de Oriente mediante el siguiente trabajo de investigación muestra sus ganas de dar un paso adelante en innovación, y demuestra que tiene estudiantes con ideas diferentes y que pueden solucionar muchos problemas. Con la implementación de esta aplicación se muestra que el flujo vehicular disminuye en una gran cantidad, lo cual impacta de manera significativa el tránsito dentro de la UCO al momento de estacionar los respectivos vehículos; ya que la implementación de esta aplicación de control

de acceso para el ingreso de vehículos a la institución garantiza una mejor movilidad dentro del campus universitario y sus zonas aledañas, lo cual permite que los miembros de la institución, sean estudiantes, empleados o visitantes tengan una mejor tranquilidad dentro de la institución, que no vivan con el estrés de que no van a llegar a tiempo a clase porque no van a encontrar lugar para estacionar. Sin duda alguna la implementación de esta aplicación mejorará en grandes cantidades la tranquilidad de los miembros de la UCO.

Dicho lo anterior, esta aplicación ayuda a mejorar el control, puesto que los parqueaderos van a disminuir su cantidad de vehículos, y demostrará a todos los miembros de la institución el cambio cultural que va a tener dentro de la familia uconiana, el cual crea conciencia en cada uno de los miembros de la institución para ser más responsables con el medio ambiente al buscar otras alternativas de transporte a la hora de llegar a la institución, lo cual permite ayudar a disminuir la congestión vehicular y a reducir la contaminación ambiental.

Con la implementación de este sistema se benefician: primero el medio ambiente, ya que es una iniciativa para crear cultura y que los estudiantes y empleados de la UCO tomen conciencia para usar otro medio de transporte el día que no pueda ingresar al campus en su vehículo, y segundo se beneficia la institución, porque esto permite que no se vean vehículos ni motocicletas mal estacionadas y ayudaría en el flujo dentro del campus universitario, mejorando la calidad de vida dentro de la institución, lo cual genera una mayor tranquilidad para los miembros de la institución al observar que no hay necesidad de preocuparse por no encontrar parqueadero, como también mejorar la movilidad dentro de la institución y a los alrededores de ella, evitando congestión vehicular en las zonas de ingreso y salida del campus universitario..

Esta aplicación mejora la comodidad de los miembros de la institución, ya que les facilita la vida al momento de estacionar su vehículo, ya no es necesario que la persona este durante varios minutos buscando un lugar donde estacionar por la gran cantidad de vehículos estacionados, ya se puede encontrar de manera rápida un lugar donde estacionar, y gracias a esto ya no se corre el riesgo de llegar tarde a clase.

Controlar el ingreso de vehículos con la medida de pico y placa facilita la vida para todas las personas que hacen parte de la institución.

## **CAPÍTULO 4: OBJETIVOS**

### **4.1 General:**

Desarrollar un software que permita gestionar la restricción del pico y placa al momento de ingresar vehículos a la UCO

### **4.2 Específicos:**

- Analizar los requisitos y necesidades que hay para realizar la aplicación.
- Diseñar la arquitectura de la aplicación, la cuál va a estar vinculada con el sistema de torniquetes de ingreso, mediante el carnet institucional.
- Programar el código que le va a dar el funcionamiento lógico a la aplicación.

## **CAPÍTULO 5: MARCO TEÓRICO**

### **5.1 Fundamento teórico**

En esta sección se muestra la implementación de un sistema de ingreso inteligente para vehículos en la universidad. Se iniciará con la definición de un sistema de control de acceso, los beneficios que esté traerán para la universidad y para sus visitantes como también el cambio que se podrá ver con el pasar de los meses. Luego se va a mostrar la tecnología que se va a implementar en este para simular el funcionamiento del mismo, sus principales componentes y sus beneficios al estar implementado y funcionando.

- Diseño e implementación de un parqueo inteligente utilizando Arduino Yun basado en internet de las cosas (IoT) (Lindao, 2020)
- Propuesta de un sistema de estacionamiento inteligente con sensores IoT (inicio Ramos Valencia, 2020)
- Implementación de sensor de proximidad audible a través de placa Arduino para el parqueo de vehículos (Cueva, 2020)
- Diseño de un prototipo de sistema de parqueo inteligente para el edificio de la FIE utilizando tecnología basado en el internet de las cosas (Taípe, 2020)
- Diseño de un sistema de parqueo inteligente con aplicación móvil para mostrar espacios disponibles tiempo y valor a cancelar (Castro, 2020)

- Estudio de factibilidad de la implementación de un sistema domótico empleando Raspberry Pi y Arduino para el monitoreo automatizado de los parqueaderos del edificio de la corte provisional del Guayas (Isaac Andrés Sánchez Salazar, 2020)
- Simulación de un parqueadero inteligente para la universidad de las Américas en la sede Queri (Sangoluisa, 2020)
- Arquitectura IoT para parqueaderos inteligentes en la ciudad de Bogotá (Nicolás Cubillos Gracia, 2020)
- Desarrollo de un sistema de asistencia para búsqueda de disponibilidad de estacionamientos en la Universidad Católica de Santiago de Guayaquil UCSG (Jalón, 2020)
- Desarrollo de un prototipo de control de espacios de estacionamientos libres mediante el uso de visión artificial en el edificio de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad de Guayaquil (Marco Antonio Avendaño Quinde, 2020)
- Diseño de un modelo de ciclo de parqueadero inteligente soportado en tecnología RFID/NFC y aplicación móvil sobre Android para la sede de ingeniería de la Universidad Distrital Francisco José de Caldas (Katherine Andrea Cuartas Castro, 2020)

### **Acceso inteligente**

Hace referencia a monitorear y controlar el ingreso de personas, vehículos, animales, etc. A un determinado lugar que cuenta con un sensor inteligente que se encarga de analizar y guardar la información que se requiere, todo esto se controla por medio de un software que permita identificar y guardar la información, para más adelante tener un reporte bien elaborado de las actividades que se realizan y que ha captado el sensor de ingreso inteligente instalado en determinado lugar. (Locken, 2020)

### **Modulo RFID (Identificador por radio frecuencia)**

Es una tecnología diseñada para leer a una distancia determinada, de forma inalámbrica tarjetas o etiquetas. Para iniciar la comunicación, el lector emite una señal dentro del alcance que tiene, luego, este responde procesando la información almacenada con la que se programó previamente el lector. (Llamas, 2020)

### **Tarjeta Relé Opto acoplada**

Circuito eléctrico que controla una conexión aislada entre circuitos que manejen un voltaje diferente. También permite controlar dispositivos en el hogar o en la industria. (Electronilab, 2020)

### **ESP-32**

Es un módulo genérico utilizado en diferentes aplicaciones en el que podemos trabajar por medio de Bluetooth o WIFI, se puede utilizar en tareas muy exigentes o en sensores con una potencia muy baja. Con este módulo se puede transmitir música, como también realizar codificaciones y decodificaciones de archivos. (SoloArduino, 2020)

### **Display LCD con Módulo I2C**

Es una pantalla donde se pueden programar mensajes que se le van a mostrar al usuario sobre el éxito o la falla del proceso. El módulo I2C permite manipular de una forma más sencilla los recursos del controlador que tiene permitidos Arduino. (Arduino, 2020)

### **Servo motor**

Es un motor especial que permite controlar la posición de un eje que se quiera mostrar en un determinado periodo de tiempo, con este motor lo que se va a controlar es el accionamiento de la barra que bloquea el ingreso al parqueadero cuando el usuario está autorizado para ingresar. (González, 2020)

### **Tarjeta RFID**

Es una tarjeta de proximidad que permite almacenar o recuperar datos por medio de un número de serie o un objeto previamente establecido, todo por medio de una señal, donde puede tener restricciones establecidas en una base de datos. Con esta tarjeta se va a simular la placa vehicular que tendría un vehículo y controlar cuando podría o no entrar al momento del reconocimiento de su número de serie. (HID, 2020)

## **Beneficios de esta tecnología**

La implementación de esta tecnología mejora el control de los vehículos dentro del campus universitario, como también permite administrar el ingreso de vehículos a la institución, reduciendo la cantidad de vehículos en un porcentaje considerable, que puede ayudar a mejorar la movilidad dentro de la institución y sus alrededores.






Esta tecnología ayudará a reducir los tiempos que los usuarios de la universidad invierten normalmente buscando un sitio para estacionar, como también permitirá disminuir la contaminación que generan los vehículos, reduciendo la cantidad de vehículos que ingresan y el tiempo que permanecen encendidos.



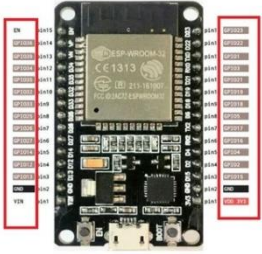



- **Modulo RFID:** Es un lector que permite identificar la información que almacena una tarjeta RFID.
- **Tarjeta Relé Opto Acoplada:** Permite controlar dispositivos industriales o del hogar por medio del voltaje.
- **ESP-32:** Es un módulo que permite codificar o decodificar de manera sencilla por medio de una conexión WIFI o Bluetooth.
- **Display LCD con Módulo I2C:** Es una pantalla LCD en la que se pueden mostrar mensajes programados en el código.
- **Servo motor:** Es un motor que permite controlar los ejes que se quieren mostrar.
- **Tarjeta RFID:** Es una tarjeta que almacena información en un ID que puede ser leída para autorizar ingresos u otras acciones.



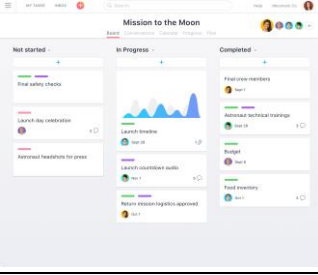
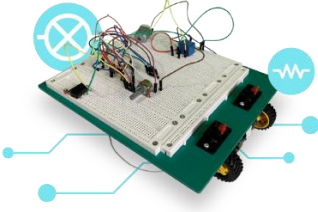




## Glosario de términos

Tabla 1. Glosario de términos

Descripción	Imagen	fuente
Figura 1 – Arduino	 <p>Diagrama de un Arduino Uno con los pines numerados del 1 al 24. Los pines están agrupados en cabezales de 5 pines cada uno. El pin 1 está en la esquina superior izquierda del cabezal inferior izquierdo, y el pin 20 está en la esquina inferior derecha del mismo cabezal. Los pines superiores están numerados de 10 a 5, y los inferiores de 2 a 15.</p>	Fuente: (Zaragoza, 2020)
Figura 2 – Bases de datos	 <p>Diagrama que muestra un sistema gestor de base de datos. En el centro hay un recuadro con el texto "Sistema gestor de base de datos". A la izquierda hay un icono de una base de datos con el texto "Base de datos". A la derecha hay un icono de un usuario con el texto "Usuarios". En el centro superior hay logos de MySQL y ORACLE. En el centro inferior hay un logo de SQL Server. Flechas rojas indican la conexión entre los elementos.</p>	Fuente: (Alberto, 2020)
Figura 3 – Lenguajes de programación	 <p>Diagrama que muestra un escritorio con un laptop y varios iconos de lenguajes de programación: PHP, JavaScript, Python, Java, Perl, C++, .NET, y Ruby. El fondo es azul con código de programación visible.</p>	Fuente: (Aprendum, 2020)
Figura 4 – Sensores	 <p>Colección de varios tipos de sensores electrónicos, como sensores de temperatura, humedad, luz, y otros.</p>	Fuente: (Google, 2020)
Figura 5 – Pantalla LCD	 <p>Una pantalla LCD de color azul montada en una placa verde con pines de conexión.</p>	Fuente: (Google, 2020)

<p>Figura 6 – Tarjeta RFID</p>		<p>Fuente: (Merca Electrónico, 2020)</p>
<p>Figura 7 – Servo motor</p>		<p>Fuente: (Robotisan, 2020)</p>
<p>Figura 8 – ESP 32</p>		<p>Fuente: (Arca Electrónica, 2020)</p>
<p>Figura 9 – Opto acoplada</p>		<p>Fuente: (Bigtronica, 2020)</p>
<p>Figura 10 – MySQL</p>		<p>Fuente: (AWS, 2020)</p>
<p>Figura 11 – IDE</p>		<p>Fuente: (Snurmacher, 2020)</p>

<p>Figura 12 – Desarrollo de software</p>		<p>Fuente: (Sánchez, 2020)</p>
<p>Figura 13 – Levantamiento de requisitos</p>		<p>Fuente: (PMO Informática, 2020)</p>
<p>Figura 14 – Kanban</p>		<p>Fuente: (Asana, 2020)</p>
<p>Figura 15 – Fundamentos de electrónica</p>		<p>Fuente: (Google, 2020)</p>
<p>Figura 16 – Scrum</p>	<p style="text-align: center;"><b>Metodología SCRUM</b></p> 	<p>Fuente: (Google, 2020)</p>
<p>Figura 17 – Codificación</p>		<p>Fuente: (Efecto digital, 2020)</p>

<p>Figura 18 – Tag</p>		<p>Fuente: (Indiamart, 2020)</p>
<p>Figura 19 – Control de acceso</p>		<p>Fuente: (Itsitio, 2020)</p>

## 5.2 Metodología

Cuando se realiza una investigación o un estudio sobre una problemática, donde se comparan la teoría o hipótesis planteada y la realidad que se vive en el entorno donde se va a llevar a cabo el estudio, se puede afirmar que este estudio se va a enfocar en un entorno experimental, en el cual se tiene control sobre las variables de estudio y nos podremos aproximar al resultado que se va a obtener cuando finalice el desarrollo del estudio.

### 5.2.1 Método experimental

Para el desarrollo de este proyecto nos basamos en la modificación de hipótesis, la toma de mediciones, la experimentación y la observación sistemática, por ende, podemos manipular y controlar las variables que se presentan en la investigación y así poder estudiar la relación que se encuentran entre ellas. Con este método vamos a estudiar el fenómeno o problema que se planteó, que es la falta de control que se tiene con el pico y placa de los vehículos al momento de ingresar a la Universidad Católica de Oriente. Mediante este método podemos observar el comportamiento de las variables que se presentan en el problema planteado y así podemos anticipar los resultados que se van a obtener al final para buscar una solución que sería el software que permitirá controlar de manera automatizada esta medida de control de pico y placa.

Al final, podremos obtener como las variables influyen entre sí y de igual forma, se conseguirá la información exacta de las afectaciones.

En esta metodología se siguen una serie de pasos para lograr un resultado satisfactorio, los cuales son:

- Plantear el problema y las observaciones.
- Plantear la hipótesis.
- Definir las variables.
- Definir un diseño experimental.
- Llevar a cabo el procedimiento y recoger los datos correspondientes.
- Analizar los datos.
- Generalizar.
- Predecir.
- Presentar las conclusiones finales.<sup>11</sup>

## **5.3 POBLACIÓN Y MUESTRA**

### **5.3.1 Población**

En el caso del sistema de control de pico y placa, la población está definida por los estudiantes, docentes, administrativos y demás empleados de la Universidad Católica de Oriente, que utilizan un vehículo propio, sea moto o carro, para transportarse y acceder a las instalaciones y estacionamientos de la institución.

### **5.3.2 Muestra**

La muestra será tomada sobre los tres parqueaderos que la universidad tiene designados para los miembros de la institución. Cada parqueadero cuenta con un sistema de torniquetes que lee el carnet de la institución y autoriza el ingreso del vehículo que lo solicita, a este sistema, se le va a integrar el nuevo software que permita verificar si el solicitante tiene acceso a la universidad, dependiendo la restricción que haya ese día.

---

<sup>11</sup> (Rodríguez D. , 2020)

## CAPÍTULO 6: DISEÑO METODOLÓGICO

**Fase-1:** Analizar los requisitos y necesidades que hay para realizar la aplicación.

- Etapa-1: Conocer las necesidades de la UCO
  1. Estudiar la problemática que se vive en la institución.
  2. Conocer la forma en que se maneja el control actual del pico y placa.
  3. Proponer una alternativa para solucionar la problemática tratada.
- Etapa-2: Realizar el levantamiento de requerimientos
  1. Identificación y documentación de los requerimientos del sistema.
  2. Entrevistas con los directivos.
  3. Encuesta a los miembros de la institución.
- Etapa-3: Levantamiento de requisitos
  1. Entrevistas con los usuarios
  2. Requisitos funcionales y no funcionales
  3. Casos de uso
  4. Historias de usuario

**Fase-2:** Diseñar la arquitectura de la aplicación, la cuál va a estar vinculada con el sistema de torniquetes de ingreso, mediante el carnet institucional.

- Etapa-1: Analizar la estructura funcional del sistema de torniquetes de la universidad
  1. Realizar pruebas de funcionamiento al sistema de torniquetes para entender su funcionamiento.
- Etapa-2: Diseñar la arquitectura de la aplicación.
  1. Diseñar la arquitectura para ver cómo va a funcionar la aplicación.

**Fase-3:** Programar el código que le va a dar el funcionamiento lógico a la aplicación.

- Etapa-1: Crear la base de datos que va a tener la placa electrónica.
  1. Crear una base de datos que esté vinculada con el carnet institucional y la placa del vehículo del estudiante.
- Etapa-2: Desarrollar el código fuente que va a consumir la base de datos.
  1. Programar el código Java que va a realizar la conexión con la base de datos.
- Etapa-3: Implementar el código y la base de datos con el sistema de torniquetes.
  1. Realizar la conexión para el funcionamiento de la aplicación junto al sistema de torniquetes de la universidad.

**Fase-4:** Realizar las pruebas necesarias para que la aplicación funcione correctamente.

- Etapa-1: Realizar pruebas a la aplicación.
  1. Realizar pruebas unitarias para verificar el correcto funcionamiento de la aplicación.
  2. Revisar el correcto funcionamiento de la base de datos.
- Etapa-2: Realizar pruebas al sistema de ingreso integrado con la aplicación.

1. Verificar que el sistema si se conecte correctamente con la aplicación y con la base de datos.

Probar que el sistema de torniquetes si reconozca el carnet por medio de la placa electrónica y no presente excepciones.

## CAPÍTULO 7: RESULTADOS

### 7.1 Implementación y Desarrollo

En este capítulo, se detallan el diseño y los criterios del sistema que se desarrolló; se presentan los procesos de diseño y construcción de una tarjeta electrónica para recepción de señales transmitidas por cada uno de los sensores ultrasónicos y como se debe configurar nuestra tarjeta principal Arduino para que administre los datos y sean visualizados en el motor de base de datos MySQL, así como la instalación de la infraestructura donde se instalará el sistema.

#### 7.1.1. Modelo del funcionamiento del Circuito

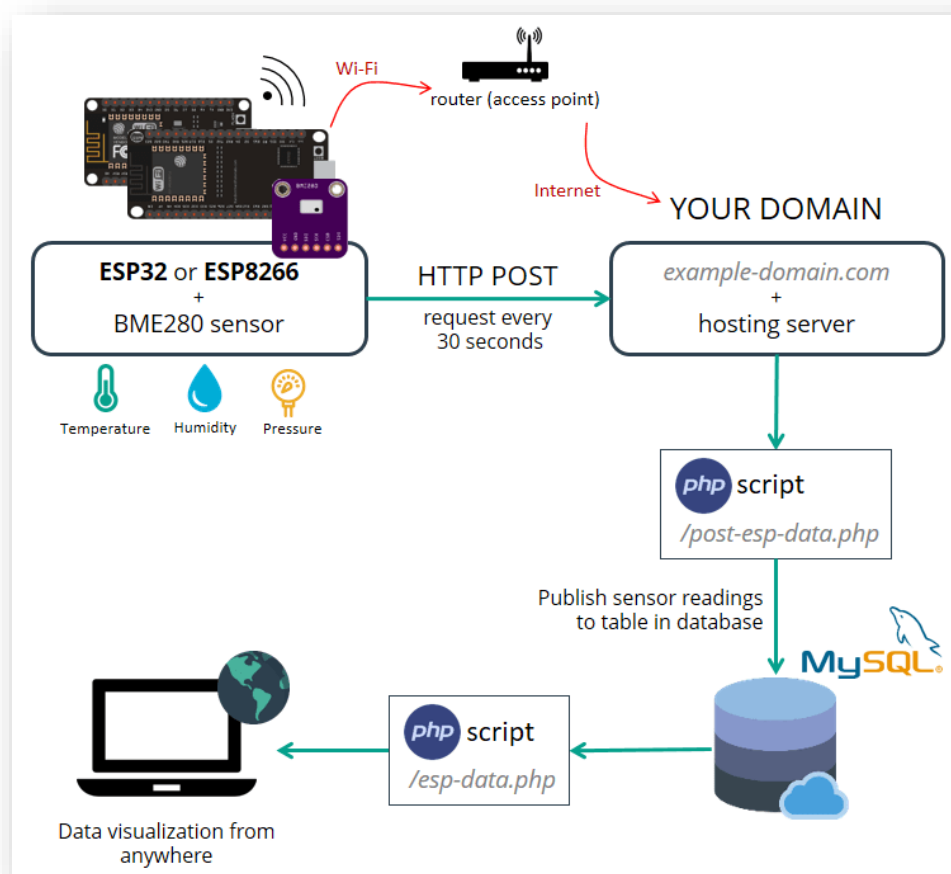


Figura 20 – Modelo de funcionamiento Autor: (Random Nerd Tutorials, 2020)



### 7.1.2. Configuración de Microcontrolador Pinout ESP32

Como primer paso se va a utilizar una tarjeta Pinout ESP32 sobre el cual se va a programar el código para que cumpla con el funcionamiento del proyecto, como se puede observar en la figura 21.

Este código permitirá que el sensor valide si el vehículo está autorizado o no para acceder a las instalaciones de la UCO, y dependiendo la señal que esté envíe, habilitará el ingreso del vehículo por medio de un servo motor.

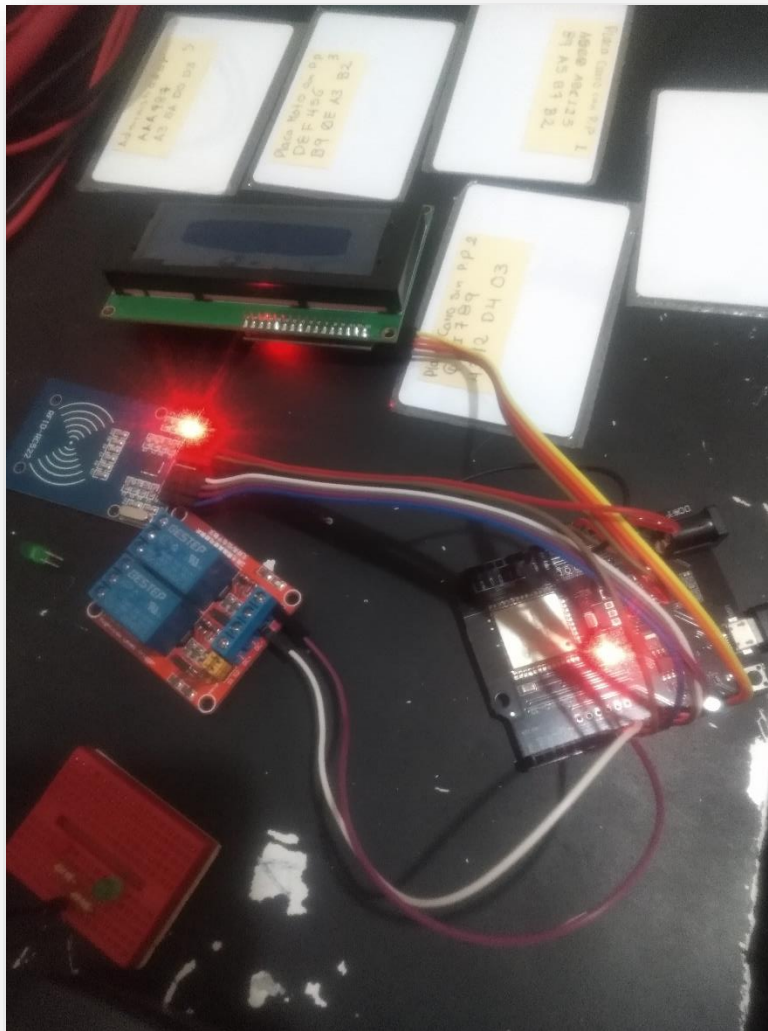


Figura 21 – Hardware de programación Pinout ESP32, Autor: Daniel Esquivel



## Código para el microcontrolador ESP32:

```
//hace una lectura del bloque y del Buffer
status = mfrc522.MIFARE_Read(block, buffer2, &tamano_codigo);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

//String data=""; //variable para guardar el vector de caracteres en una cadena
for (uint8_t i = 0; i < 16; i++) {
    data+=(char)buffer2[i];
}
placa=data;
Serial.println(data);
Serial.println(F(" "));
Serial.println(F("\n**End Reading**\n"));
delay(1000); //change value if you want to read cards faster
mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
}
//-----Funcion lectura de placas-----
void lecturaPlacas(){
    //-----placa1-----
    if(placa.indexOf("ABC123")>=0){
        digitalWrite(salida1 , HIGH);
        digitalWrite(salida2 , LOW);
        Serial.println("No autorizado");
        mensaje="No autorizado";
        //lcdESP32(mensaje);
        lcd.setCursor(1, 0);
        lcd.print(mensaje);
        delay(1000);
        lcd.clear();
    }
    //-----placa2-----
    if(placa.indexOf("GHI789")>=0){
        Serial.println("Autorizado");
        digitalWrite(salida1 , LOW);
        digitalWrite(salida2 , HIGH);
        mensaje="Autorizado";
        //lcdESP32(mensaje);
        lcd.setCursor(1, 0);
        lcd.print(mensaje);
        delay(1000);
        lcd.clear();
    }
}
```

```

}
//-----placa3-----
if(placa.indexOf("DEF456")>=0){
  Serial.println("No autorizado");
  digitalWrite(salida1 , HIGH);
  digitalWrite(salida2 , LOW);
  mensaje="No autorizado";
  //lcdESP32(mensaje);
  lcd.setCursor(1, 0);
  lcd.print(mensaje);
  delay(1000);
  lcd.clear();
}
//-----placa4-----
if(placa.indexOf("JKL123")>=0){
  Serial.println("Autorizado");
  digitalWrite(salida1 , LOW);
  digitalWrite(salida2 , HIGH);
  mensaje="Autorizado";
  //lcdESP32(mensaje);
  lcd.setCursor(1, 0);
  lcd.print(mensaje);
  delay(1000);
  lcd.clear();
}
//-----administrador-----
if(placa.indexOf("AAA987")>=0){
  Serial.println("admin");
  digitalWrite(salida1 , LOW);
  digitalWrite(salida2 , HIGH);
  mensaje="admin";
  //lcdESP32(mensaje);
  lcd.setCursor(1, 0);
  lcd.print(mensaje);
  delay(1000);
  lcd.clear();
}
//-----
}

```

### 7.1.3. Configuración de LCD ESP32

En el desarrollo del circuito, esta pantalla mostrará un mensaje informando si el acceso del vehículo fue autorizado o rechazado.



Figura 22 – LCD, Autor: Daniel Esquivel

### Código para configurar la pantalla LCD:

```
//----- Librerias ESP-32_UNO_R3
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>

//-----MODULO LCD-----
//---Configuracion de la lcd
int lcdColumns = 20;
int lcdRows = 4;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
//-----
//-----RFID-----
constexpr uint8_t RST_PIN = 13; // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = 5; // Configurable, see typical pin layout above
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
char c;

//-----NÚMERO DE LAS PLACAS-----
String placa_1="ABC123";
String placa_2="GHI789";
String placa_3="DEF456";
String placa_4="JKL123";
String Adminstrador="AAA987";
//-----VARIABLES-----
String mensaje;
```

```

String data="";
String placa="";
int salida1=12;
int salida2=14;
//-----
void setup() {
  Serial.begin(9600);
  //-----LCD
  lcd.init();
  lcd.backlight();
  //-----RFID
  SPI.begin();          // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522 card
  //-----config Pinput
  pinMode(salida1, OUTPUT); // A como entrada
  pinMode(salida2, OUTPUT); // B como entrada
  digitalWrite(salida1 , LOW);
  digitalWrite(salida2 , LOW);

}
//-----CICLO PRINCIPAL-----
void loop() {
  Serial.println("Digiti 1 para LEER:");
  while(!Serial.available()){
  c=Serial.read();
  switch(c)
  {
    case '1':
      leerRFID();
      break;
  }
  lecturaPlacas();
}
//-----FUNCIÓN LCD -----
String lcdESP32 (String mensaje){
  lcd.setCursor(1, 0);
  lcd.print(mensaje);
  delay(5000);
  lcd.clear();
}
void leerRFID()
{
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  //some variables we need

```

```

byte block;
byte tamaño_codigo;
MFRC522::StatusCode status;
Serial.println("Pase su tarjeta");

//-----

// Look for new cards
while ( ! mfrc522.PICC_IsNewCardPresent() ) {
  //return;
  //se queda en el ciclo hasta que detecta una nueva tarjeta
}

// Select one of the cards
if( ! mfrc522.PICC_ReadCardSerial() ) {
  //return;
  Serial.println("Tarjeta No Compatible");
}
Serial.println("Tarjeta OK");
//Serial.println(F("**Tarjeta OK:**"));

tamaño_codigo = 18;

//-----

//mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid)); //dump some details about the card

//mfrc522.PICC_DumpToSerial(&(mfrc522.uid)); //uncomment this to see all blocks in hex
//aumentamos el tamaño para que haya un rango más amplio de lectura
tamaño_codigo = 18;
//----- GET LAST NAME

//declaramos un buffer de la misma longitud de tamaño_codigo
byte buffer2[18];
block = 1;

status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfrc522.uid)); //line 834
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Authentication failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}

```

#### 7.1.4. Configuración de identificador RFID

Es un controlador que permite escribir y leer información de otros dispositivos, tarjetas o cualquier otro objeto que tenga una etiqueta RFID. Este objeto permite transmitir la información que tiene un objeto y así, poder controlarla o restringirla.



Figura 23 – Identificador RFID, Autor: Daniel Esquivel

#### Código para el identificador RFID:

```
#include <SPI.h>
#include <MFRC522.h>

constexpr uint8_t RST_PIN = 13; // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = 5; // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
char c;

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card
}

void loop()
{
```

```

Serial.println("1. Escribir");
Serial.println("2. Leer");
while(!Serial.available()){
c=Serial.read();
switch(c)
{
case '1':
Escribir();
break;
case '2':
Leer();
break;
}
}
void Escribir()
{

// Prepare key - all keys are set to FFFFFFFFh at chip delivery from the factory.
MFRC522::MIFARE_Key key;
byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte tamaño_codigo;
Serial.println("Escribe nuevo código y finaliza con #");
Serial.setTimeout(20000L);
tamaño_codigo = Serial.readBytesUntil('#', (char *) buffer, 30);

//llena con espacio vacío lo que no usamos de espacio para el código
for (byte i = tamaño_codigo; i < 10; i++) buffer[i] = ' ';

//inicializamos la key que nos permite escribir en la tarjeta
for(byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

Serial.println("Ingresa tu tarjeta");
//Look for new cards
while(! mfr522.PICC_IsNewCardPresent()){
//return;
//ciclo infinito. Sale hasta que se ingresa la tarjeta
}

//select one of the cards
if( ! mfr522.PICC_ReadCardSerial()){
//return;
Serial.println("Tarjeta no compatible");
}
}

```

```

for(byte i = 0; i < mfr522.uid.size; i++){
}

//especificamos el tipo de tarjeta
MFRC522::PICC_Type piccType = mfr522.PICC_GetType(mfr522.uid.sak);

//asignamos un bloque de escritura. 7 bloques de 30 caracteres
block = 1;

//autenticacion de la tarjeta
status = mfr522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfr522.uid));
//En caso de haber algun error con la key o con el bloque, etc. arroja un error
if(status != MFRC522::STATUS_OK){
    return;
}

//realizamos la escritura en el bloque 1 lo que hay en el buffer limitado a 16 bits
status = mfr522.MIFARE_Write(block, buffer, 16);
//si hay algun problema en la escritura lanza un error
if(status != MFRC522::STATUS_OK){
    return;
}

Serial.println("Escrito correctamente");
mfr522.PICC_HaltA(); //Halt PICC
mfr522.PCD_StopCrypto1(); //stop encryption
}

void Leer()
{
    MFRC522::MIFARE_Key key;
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

    //some variables we need
    byte block;
    byte tamano_codigo;
    MFRC522::StatusCode status;
    Serial.println("Pase su tarjeta");

    //-----

    // Look for new cards
    while ( ! mfr522.PICC_IsNewCardPresent() ) {

```



```

//return;
//se queda en el ciclo hasta que detecta una nueva tarjeta
}

// Select one of the cards
if( ! mfr522.PICC_ReadCardSerial() ) {
  //return;
  Serial.println("Tarjeta No Compatible");
}
Serial.println("Tarjeta OK");
//Serial.println(F("**Tarjeta OK:**"));

tamano_codigo = 18;

//-----

//mfr522.PICC_DumpDetailsToSerial(&(mfr522.uid)); //dump some details about the card

//mfr522.PICC_DumpToSerial(&(mfr522.uid)); //uncomment this to see all blocks in hex
//aumentamos el tamaño para que haya un rango más amplio de lectura
tamano_codigo = 18;
//----- GET LAST NAME

//declaramos un buffer de la misma longitud de tamano_codigo
byte buffer2[18];
block = 1;

status = mfr522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfr522.uid)); //line 834
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Authentication failed: "));
  Serial.println(mfr522.GetStatusCodeName(status));
  return;
}

//hace una lectura del bloque y del Buffer
status = mfr522.MIFARE_Read(block, buffer2, &tamano_codigo);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Reading failed: "));
  Serial.println(mfr522.GetStatusCodeName(status));
  return;
}

String data=""; //variable para guardar el vector de caracteres en una cadena
for (uint8_t i = 0; i < 16; i++) {
  data+=(char)buffer2[i];
}

```

```

Serial.println(data);
Serial.println(F(" "));

//-----
String placa="";
placa=data;
Serial.println(data.length());
Serial.println(placa.length());
if(placa.indexOf("ABC123")>=0){
  Serial.println("abrir");
}
//-----

Serial.println(F("\n**End Reading**\n"));

delay(1000); //change value if you want to read cards faster

mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
}

```

### **7.1.5. Configuración de WIFI ESP32**

Con el módulo de WIFI podemos realizar la interconexión inalámbrica con cualquier dispositivo electrónico que se vaya a integrar a la estructura del proyecto.

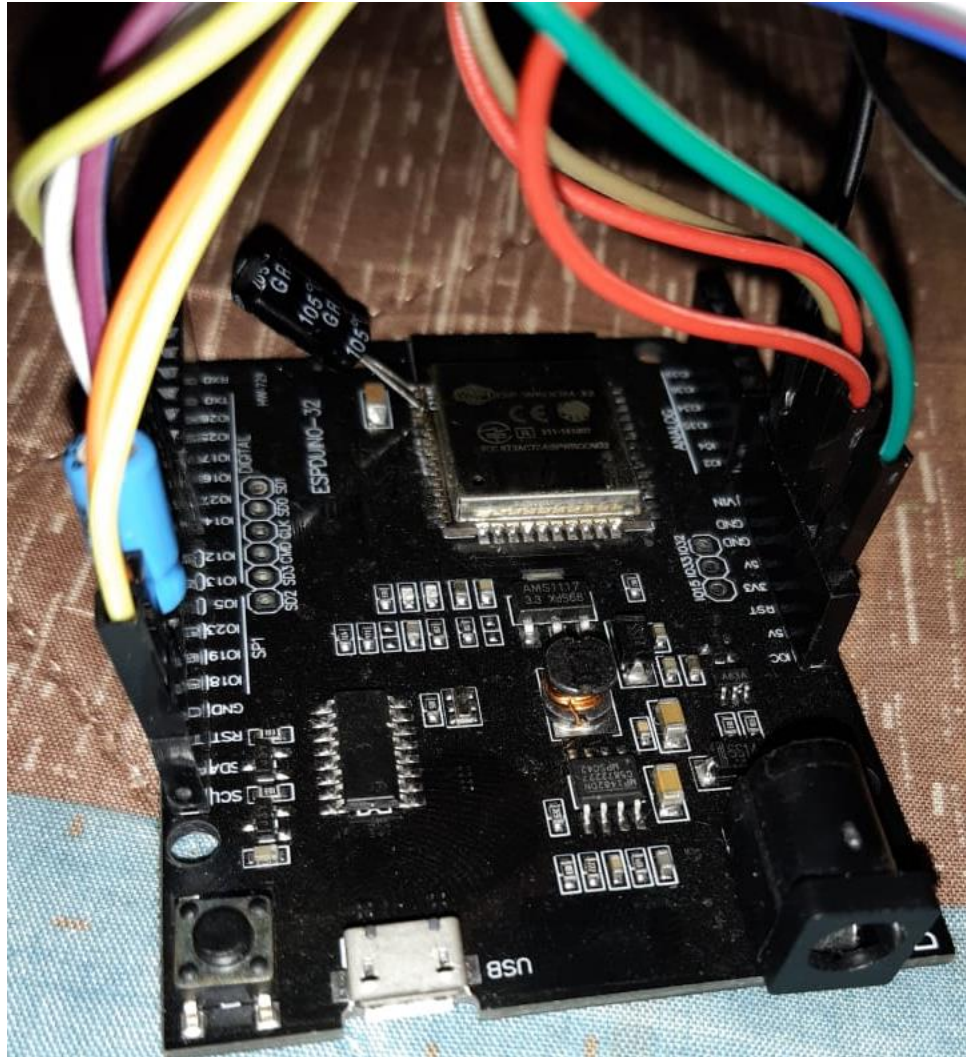


Figura 24 – WIFI ESP32, Autor: Daniel Esquivel

### Código para la configuración del módulo WIFI:

```
#include <WiFi.h>
```

```
const char* ssid = "KATALINA_P1";  
const char* password = "68A34VJ63GH";
```

```
WiFiServer server(80);
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);  
  pinMode(5, OUTPUT);    // set the LED pin mode
```

```
  delay(10);
```

```

// We start by connecting to a WiFi network

Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

server.begin();

}

int value = 0;

void loop(){
WiFiClient client = server.available(); // listen for incoming clients

if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
        if (client.available()) { // if there's bytes to read from the client,
            char c = client.read(); // read a byte, then
            Serial.write(c); // print it out the serial monitor
            if (c == '\n') { // if the byte is a newline character

                // if the current line is blank, you got two newline characters in a row.
                // that's the end of the client HTTP request, so send a response:
                if (currentLine.length() == 0) {
                    // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
                    // and a content-type so the client knows what's coming, then a blank line:
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-type:text/html");
                    client.println();
                }
            }
        }
    }
}

```

```

// the content of the HTTP response follows the header:
client.print("Click <a href=\"/H\">here</a> to turn the LED on pin 5 on.<br>");
client.print("Click <a href=\"/L\">here</a> to turn the LED on pin 5 off.<br>");

// The HTTP response ends with another blank line:
client.println();
// break out of the while loop:
break;
} else { // if you got a newline, then clear currentLine:
currentLine = "";
}
} else if (c != '\r') { // if you got anything else but a carriage return character,
currentLine += c; // add it to the end of the currentLine
}

// Check to see if the client request was "GET /H" or "GET /L":
if (currentLine.endsWith("GET /H")) {
digitalWrite(5, HIGH); // GET /H turns the LED on
}
if (currentLine.endsWith("GET /L")) {
digitalWrite(5, LOW); // GET /L turns the LED off
}
}
}
// close the connection:
client.stop();
Serial.println("Client Disconnected.");
}
}

```

### 7.1.6. Configuración de EEPROM

EEPROM es la memoria que se puede programar, borrar y reprogramar de forma eléctrica. En este chip se guarda información sin necesidad de energía, pero para borrar la información si necesita más voltaje que lo normal, y para esto utiliza circuitos lógicos, su funcionamiento es parecido al de una memoria RAM, pero su funcionamiento es mucho más lento.

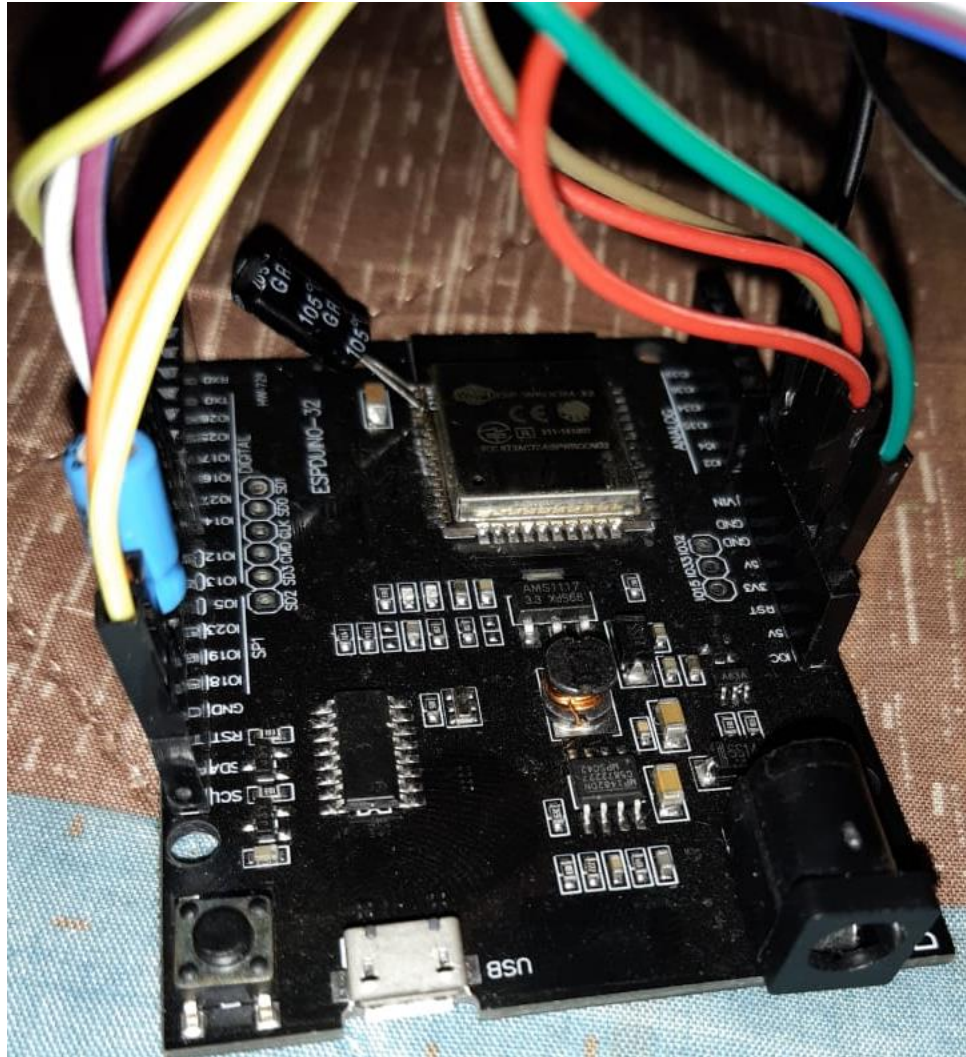


Figura 25 – EEPROM, Autor: Daniel Esquivel

### Código para la configuración:

```
// include library to read and write from flash memory
#include <EEPROM.h>
// define the number of bytes you want to access
#define EEPROM_SIZE 1

void setup() {
  Serial.begin(9600);
  // initialize EEPROM with predefined size
  EEPROM.begin(EEPROM_SIZE);
}

void loop() {
  EEPROM.write(0, 12);
  EEPROM.commit();
  Serial.println(EEPROM.read(0));
  delay(1000);
}
```

}

### 7.1.7. Configuración de MySQL

MySQL es un motor de base de datos relacional, el cual se utilizará para almacenar la información de los vehículos. Al tener la información almacenada en la base de datos, el identificador RFID, consultará la información directamente en la base de datos y dependiendo la señal que devuelva se autorizará o no, el ingreso de los vehículos a las instalaciones de la institución.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	<b>id</b>		int(11)		No	Ninguna	AUTO_INCREMENT	Cambiar  Eliminar  Más
<input type="checkbox"/>	2	<b>nombre</b>		varchar(15)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	3	<b>apellido</b>		varchar(15)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	4	<b>cedula</b>		int(20)		No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	5	<b>placa</b>		varchar(10)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	6	<b>vehiculo</b>		varchar(10)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	7	<b>idhexrfid</b>		varchar(20)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más
<input type="checkbox"/>	8	<b>iddecrfid</b>		varchar(20)	utf8mb4_general_ci	No	Ninguna		Cambiar  Eliminar  Más

Figura 26 – MySQL, Autor: Daniel Esquivel

#### Código para configurar la base de datos:

```
#include <WiFi.h>
```

```
const char* ssid = "KATALINA_P1";  
const char* password = "68A34VJ63GH";
```

```
//Inicializar instancia de la libreria ethernet  
WiFiClient client;
```

```
//Dirección IP del servidor con la página PHP  
char server[] = "192.168.0.102";
```

```
//Variables que necesita para funcionar  
String codigo; //Aquí se almacena la respuesta del servidor  
String nombre; //Aquí se almacena el nombre que recuperamos de MySQL  
boolean fin = false;  
boolean pregunta = true;
```

```
void setup() {  
  // Iniciar puerto serie  
  Serial.begin(9600);  
  // We start by connecting to a WiFi network  
  Serial.println();  
  Serial.println();  
  Serial.print("Connecting to ");
```

```

Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {
    //Comprobamos si tenemos datos en el puerto serie
    if (pregunta == true) Serial.print("Escribe el identificador de la persona: ");
    pregunta = false;
    if (Serial.available()>0){
        //leemos el identificador
        int identificador=Serial.read()-48;
        //llamamos a la función que nos permitirá comunicarnos con el servidor
        httpRequest(identificador);
        pregunta = true;
    }
}

// Con esta función hacemos la conexión con el servidor
int httpRequest(int identificador) {
    // Comprobar si hay conexión
    if (client.connect(server, 80)) {
        Serial.println("Conectado");
        // Enviar la petición HTTP
        //Dirección del archivo php dentro del servidor
        client.print("GET http://localhost/acceso/datosmysql.php?id=");
        //Mandamos la variable junto a la línea de GET
        client.print(identificador);
        client.println(" HTTP/1.1");
        //IP del servidor
        client.println("Host: 192.168.0.102");
        client.println("Connection: close");
        client.println();
    }
    else {
        // Si no conseguimos conectarnos
        Serial.println("Conexión fallida");
        Serial.println("Desconectando");
        client.stop();
    }
}

```



```

delay(500);
//Comprobamos si tenemos respuesta del servidor y la
//almacenamos en el string ----> codigo.
while (client.available()) {
    char c = client.read();
    codigo += c;
    //Habilitamos la comprobación del código recibido
    fin = true;
}
//Si está habilitada la comprobación del código entramos en el IF
if (fin) {
    // Serial.println(codigo);
    //Analizamos la longitud del código recibido
    int longitud = codigo.length();
    //Buscamos en qué posición del string se encuentra nuestra variable
    int posicion = codigo.indexOf("valor=");
    //Borramos lo que haya almacenado en el string nombre
    nombre = "";
    //Analizamos el código obtenido y almacenamos el nombre en el string nombre
    for (int i = posicion + 6; i < longitud; i++){
        if (codigo[i] == ';') i = longitud;
        else nombre += codigo[i];
    }
    //Deshabilitamos el análisis del código
    fin = false;
    //Imprimir el nombre obtenido
    Serial.println("Valor de la variable nombre: " + nombre);
    //Cerrar conexión
    Serial.println("Desconectarn");
    client.stop();
}
//Borrar código y salir de la función//Dirección IP del servidor
codigo="";
return 1;
}

```

### 7.1.8. Configuración de Servomotor

Es un tipo de motor que permite girar hasta 180 grados, el cual se controla por el envío de señales eléctricas, y por medio de estas se le informa la posición en la cual se debe mover.

Para mover a la posición que se le indicó, se utiliza un sensor acompañado de un motor normal.



*Figura 27 – Servomotor, Autor: Daniel Esquivel*

### **Código de configuración del servo:**

```
#include <Servo.h>

// Declaramos la variable para controlar el servo
Servo servoMotor;

void setup() {
  // Iniciamos el monitor serie para mostrar el resultado
  Serial.begin(9600);

  // Iniciamos el servo para que empiece a trabajar con el pin 9
  servoMotor.attach(12);
}

void loop() {

  // Desplazamos a la posición 0°
  servoMotor.write(0);
  // Esperamos 1 segundo
  delay(1000);

  // Desplazamos a la posición 90°
  servoMotor.write(90);
  // Esperamos 1 segundo
```

```
delay(1000);  
  
// Desplazamos a la posición 180°  
servoMotor.write(180);  
// Esperamos 1 segundo  
delay(1000);  
}
```

### 7.1.9. Sistema de acceso

Es la maqueta que representa el parqueadero de la universidad, en el cual se va a implementar el control de acceso para autorizar el ingreso de los vehículos si estos no tienen la restricción del pico y placa. Es un modelo representativo de cómo se vería y funcionaría el parqueadero del campus universitario.



Figura 28 – Maqueta vista desde arriba, Autor: Daniel Esquivel



Figura 29 – Maqueta lateral 1, Autor: Daniel Esquivel



Figura 30 – Maqueta lado 2, Autor: Daniel Esquivel

**Código de control de acceso:**

```
#include <WiFi.h>
#include <MFRC522.h>
#include <SPI.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32_Servo.h>
//-----SERVO-----
Servo myservo;
int servoPin = 12;
//-----MODULO LCD-----
```

```

int lcdColumns = 20;
int lcdRows = 4;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
byte N[8] = {
B11111,
B11111,
B11111,
B11111,
B11111,
B11111,
B11111,
B11111,
};

//-----MODULO WIFI-----
const char* ssid = "KATALINA_P1"; //Nombre de la red wifi
const char* password = "68A34VJ63GH"; //Password WiFi
WiFiClient client;

//-----MODULO PHP-----
char server[] = "192.168.0.103"; //Dirección IP del servidor con la página PHP

//-----MODULO RFID-----
constexpr uint8_t RST_PIN = 13;
constexpr uint8_t SS_PIN = 5;
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
char c;
byte ActualUID[4]; //almacenará el código del Tag leído

//-----OTRAS VARIABLES-----
int estado=1;
int dato_1=0;
bool cicloWhile1=true;
String codigo; //Aquí se almacena la respuesta del servidor
boolean fin = false;
String nombre; //Aquí se almacena el nombre que recuperamos de MySQL
String identificador;
String identificador1;
byte UID[4]; //almacenará el código del Tag leído
String placa="";
String peticion;
String IDusuario="";
bool userAcceso;
bool acceso;

```

```

//-----
char picoPlaca[] ="123";
char picoPlacaM[] ="MXXX5XX";//Pico y placa moto
char picoPlacaC[] ="CXXXXX3";//Pico y placa carro
char picoPlacaA[] ="AXXXXXX";//administrador

//.....
void setup() {
  Serial.begin(9600); //Iniciar puerto serie
  //-----INICIO SERVO- -----
  myservo.attach(servoPin,100, 2800);
  //-----INICIO LCD- -----
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 1);
  lcd.print("Iniciando Wifi...");
  //-----INICIO RFID -----
  SPI.begin();
  mfrc522.PCD_Init();
  //-----INICIO CONEXIÓN WIFI-----
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  //-----
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("Conectado a red WIFI");
  delay(3000);
  lcd.clear();
  moduloLCDsms1(); //mensaje bienvenido

  //-----INICIO VARIABLES-----
}
//.....
void loop() {
  switch (estado) {
    //-----

```

```

case 1://INICIO
Serial.println("1. INICIO");
cicloWhile1=true;
moduloLCDsms2(); //mensaje de acceso
moduloRFID();
break;
//-----
case 2://LECTURA
Serial.println("2. LECTURA");
//identificador="137165183178";
placa=httpRequest(identificador);
peticion=""; //reiniciar variable
moduloLCDsms3(); //Mensaje de petición
break;
//-----
case 3://PETICION
Serial.println("3. PETICION");
userAcceso=UserSolicitud(placa);
placa=""; //reiniciar variable
if(acceso==true){
    estado=4;
}
else{
    estado=5;
}
acceso=false;
break;
//-----
case 4: // ACCESO AUTORIZADO
Serial.println("4. ACCESO AUTORIZADO");
lcd.clear();
lcd.setCursor(4, 0);
lcd.print("AUTORIZADO");
lcd.setCursor(0, 1);
lcd.print("Cerrando en:");
Serial.println("Autorizado");
//.....
myservo.write(10);
delay(100);
myservo.write(100);
delay(1000);
//.....
lcd.createChar (7,N);
for(int i=0; i<=19; i++){
    int count=19-i;
    lcd.setCursor(i, 2);
    lcd.write (byte (7));
}

```

```

        lcd.setCursor(13, 1);
        lcd.print(count);
        delay(1000);
    }
    //.....
    myservo.write(10);
    delay(1000);
    //.....
    estado=1;
    break;
    //-----
    default: // ACCESO NO AUTORIZADO
    Serial.println("4. ACCESO NO AUTORIZADO");
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("NO AUTORIZADO");
    Serial.println("No Autorizado");
    delay(1000);
    estado=1;
    break;
    //-----
}
}
//.....
void moduloRFID() {
    // Revisamos si hay nuevas tarjetas presentes
    while(cicloWhile1==true)
    if(mfrc522.PICC_IsNewCardPresent())
    {
        Serial.println("dentro del while");
        //Seleccionamos una tarjeta
        if ( mfrc522.PICC_ReadCardSerial())
        {
            // Enviamos serialmente su UID
            Serial.println("Card UID:");
            for (byte i = 0; i < mfrc522.uid.size; i++) {
                UID[i]=mfrc522.uid.uidByte[i];
            }
            String myString1 = String(UID[0]);
            String myString2 = String(UID[1]);
            String myString3 = String(UID[2]);
            String myString4 = String(UID[3]);
            identificador=myString1+myString2+myString3+myString4;
            Serial.println(identificador);
            mfrc522.PICC_HaltA();
            //-----
            estado=2;

```



```

        cicloWhile1=false;
        //-----
    }
}
else{
    cicloWhile1=true;
}

}
//-----
void moduloLCDsms1(){
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("Bienvenidos");
    lcd.setCursor(0, 1);
    lcd.print("Cargando....");
    lcd.createChar (7,N);
    for(int i=0; i<=19; i++){
        lcd.setCursor(i, 2);
        lcd.write (byte (7));
        delay(300);
    }
}
//-----
void moduloLCDsms2(){
    lcd.clear();
    lcd.setCursor(6, 0);
    lcd.print("USUARIO");
    lcd.setCursor(0, 2);
    lcd.print("Colocar la tarjeta");
    lcd.setCursor(0, 3);
    lcd.print("Sobre el lector");
}
//-----
void moduloLCDsms3(){
    lcd.clear();
    lcd.setCursor(5, 0);
    lcd.print("Accediendo");
    estado=3;
}
//-----
// Con esta función hacemos la conexión con el servidor
String httpRequest(String identificador) {
    // Comprobar si hay conexión
    if (client.connect(server, 80)) {
        Serial.println("Conectado");
        // Enviar la petición HTTP

```

```

//Dirección del archivo php dentro del servidor
client.print("GET http://localhost/acceso/datosmysql.php?iddecrfid=");
//Mandamos la variable junto a la línea de GET
client.print(identificador);
client.println(" HTTP/1.1");
//IP del servidor
client.println("Host: 192.168.0.103");
client.println("Connection: close");
client.println();
}
else {
// Si no conseguimos conectarnos
Serial.println("Conexión fallida");
Serial.println("Desconectando");
client.stop();
}
delay(500);
//Comprobamos si tenemos respuesta del servidor y la
//almacenamos en el string ----> codigo.
while (client.available()) {
char c = client.read();
codigo += c;
//Habilitamos la comprobación del código recibido
fin = true;
}
//Si está habilitada la comprobación del código entramos en el IF
if (fin) {
// Serial.println(codigo);
//Analizamos la longitud del código recibido
int longitud = codigo.length();
//Buscamos en que posición del string se encuentra nuestra variable
int posicion = codigo.indexOf("placa=");
//Borramos lo que haya almacenado en el string nombre
nombre = "";
//Analizamos el código obtenido y almacenamos el nombre en el string nombre
for (int i = posicion + 6; i < longitud; i ++){
if (codigo[i] == ';') i = longitud;
else nombre += codigo[i];
}
//Deshabilitamos el análisis del código
fin = false;
//Imprimir el nombre obtenido
//-----
peticion=nombre;
Serial.print("Mysql: ");
Serial.println(peticion);
//-----

```

```

        Serial.println("Valor de la variable nombre: " + nombre);
        //Cerrar conexión
        Serial.println("Desconectarn");
        client.stop();
    }
    //Borrar código y salir de la función//Dirección IP del servidor
    codigo="";
    return peticion;
}
//-----
bool UserSolicitud(String placa){
    int str_len = placa.length() + 1;
    char placaArray[str_len];
    placa.toCharArray(placaArray, str_len);
    Serial.println("Placa base de datos:");
    Serial.print(placaArray[1]);
    Serial.print(placaArray[2]);
    Serial.print(placaArray[3]);
    Serial.print(placaArray[4]);
    Serial.print(placaArray[5]);
    Serial.print(placaArray[6]);
    Serial.println(placaArray[7]);
    //-----
    if(placaArray[1]==picoPlacaM[0]){
        Serial.println("Moto");
        if(placaArray[5]==picoPlacaM[4]){
            acceso=false;
        }
        else{acceso=true; }
    }
    //-----
    if(placaArray[1]==picoPlacaC[0]){
        Serial.println("Carro");
        if(placaArray[7]==picoPlacaC[6]){
            acceso=false;
        }
        else{acceso=true; }
    }
    //-----
    if(placaArray[1]==picoPlacaA[0]){
        Serial.println("Administrador");
        acceso=true;
    }
    //-----
    return acceso;
}

```

## **7.2. Funcionamiento**

### **7.2.1 Introducción**

El sistema de acceso para controlar la restricción del pico y placa, está diseñada para disminuir la cantidad de vehículos en el campus universitario, y disminuir la dificultad para encontrar un lugar donde estacionar. Esta implementación les permitirá a los usuarios de la institución reducir los tiempos buscando un lugar para estacionar, como también, las posibilidades de encontrar el vehículo dañado al haber más espacio en los parqueaderos. Para acceder al sistema de la institución existen dos roles: Usuario administrador y Usuario estándar de la institución.

### **7.2.2 Entrada a la institución**

Para ingresar a la institución, el usuario debe de hacer uso de su carnet institucional para poder acceder al sistema de ingreso, este carnet será el que el sistema validará para ver si tiene permiso de ingreso por el pico y placa de ese día.

- **Importante**

- En caso de que el usuario tenga ese día pico y placa, el sistema no le permitirá el ingreso a la institución.
- En caso de que el usuario sea un externo a la universidad, y solo este de paso para realizar algún trámite, la persona encargada de la portería tendrá un carnet que le podrá permitir el acceso.

### **7.2.3 Verificación de ingreso**

Cuando el vehículo llega a la portería de la universidad, el sistema solicita que se coloque el carnet de acceso sobre el lector, y esta informará si el usuario tiene autorización o no para ingresar a la institución.



*Figura 31 – Verificación de ingreso, Autor: Daniel Esquivel*

#### **7.2.4 Autorización de ingreso**

Cuando el usuario coloca el carnet sobre el lector, este informará si el usuario no tiene restricción de pico y placa y autoriza el ingreso a la institución.



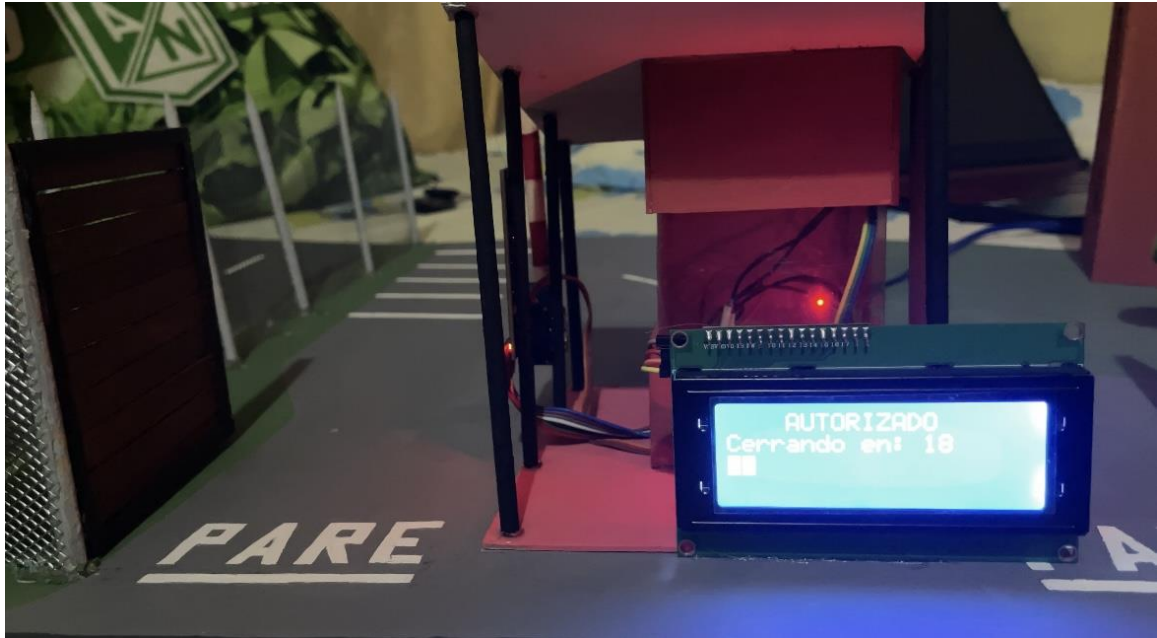
Figura 32 – Ingreso autorizado, Autor: Daniel Esquivel

El sistema consulta en la base de datos y este verifica si el usuario tiene o no acceso a la institución, devolviendo la respuesta al sistema de acceso.

```
COM5
dentro del while
Card UID:
67182123
2. LECTURA
Conectado
Mysql: MCOZ92E
Valor de la variable nombre: MCOZ92E
Desconectarn
3. PETICION
Placa base de datos:
MCOZ92E
Moto
4. ACCESO AUTORIZADO
Autorizado
```

Figura 33 – Consola de IDE 1, Autor: Daniel Esquivel

En el momento en que se autoriza el ingreso del usuario a la institución, la valla vehicular se moverá en un ángulo de 90° para permitir que el vehículo ingrese a la universidad, y este estará así 20 segundos para luego bajar y volver a su estado inicial.



*Figura 34 – Autorización de la maqueta, Autor: Daniel Esquivel*

### **7.2.5 Ingreso no permitido**

Cuando el usuario intenta ingresar a la universidad en un día que tiene restricción de pico y placa, este no le permitirá su ingreso a la institución y la pantalla mostrará un mensaje donde dice que no fue autorizado su ingreso a la universidad.



*Figura 35 – Ingreso no autorizado, Autor: Daniel Esquivel*





Figura 36 – Ingreso no autorizado - maqueta, Autor: Daniel Esquivel

El sistema consulta en la base de datos y este verifica si el usuario tiene o no acceso a la institución, devolviendo la respuesta al sistema de acceso.

```
COM5
dentro del while
Card UID:
12112671179
2. LECTURA
Conectado
Mysql: MOON53E
Valor de la variable nombre: MOON53E
Desconectarn
3. PETICION
Placa base de datos:
MOON53E
Moto
4. ACCESO NO AUTORIZADO
No Autorizado
1. INICIO
 Autoscroll  Mostrar marca temporal
Nueva línea 9600 baudio Limpiar salida
```

Figura 37 –

Consola de IDE 2, Autor: Daniel Esquivel

### 7.2.6 Autorización de ingreso por el administrador

Cuando un usuario externo a la institución necesita ingresar a realizar una diligencia, o una visita a la universidad, y este por no ser miembro activo de la comunidad universitaria no posee un carnet de ingreso, la persona encargada de la portería de la universidad, tendrá un carnet de administrador el cual le puede autorizar el ingreso a estos casos especiales.



*Figura 38 – Autorización del administrador, Autor: Daniel Esquivel*

En la base de datos se creó un usuario administrador para poder asignar esta tarjeta a las personas encargadas de la portería de la universidad, para así poder autorizar los ingresos de las personas externas a la institución.

```
dentro del while
dentro del while
Card UID:
137165183178
2. LECTURA
Conectado
Mysql: AABC123
Valor de la variable nombre: AABC123
Desconectarn
3. PETICION
Placa base de datos:
AABC123
Administrador
4. ACCESO AUTORIZADO
Autorizado
```

Figura 39 –

*Consola de IDE 3, Autor: Daniel Esquivel*

## CONCLUSIONES

En esta tesis, se creó un sistema de control de pico y placa, el cual se pretende disminuir el aforo de los parqueaderos del campus universitario de la UCO en un 80%. Basados en otros estudios hechos como en la Universidad de Antioquia, se pudo volver a demostrar que estas medidas benefician a los miembros de las instituciones, al permitir reducir los tiempos de ingreso, estacionamiento y llegada a sus actividades.

En esta tesis, se logró desarrollar un aplicativo con base de datos, que permitió simular el control para la restricción de pico y placa en la UCO, ya que, al momento de ingresar y encontrar un sitio para estacionar, se dificultaba para los miembros del plantel y generaban retrasos para llegar a sus actividades.

En esta tesis, se realizó un análisis de cuáles eran las herramientas que se iban a necesitar, para lograr implementar un sistema, que pudiera controlar el ingreso a la universidad cuando los vehículos tuvieran pico y placa.

En esta tesis, se diseñó una arquitectura de comunicación cliente/servidor, lo cual permitió la interacción del usuario final, es decir el estudiante, con la aplicación de control del pico y placa, para simular como iba a ser el ingreso a la institución.

En esta tesis, se desarrolló el código fuente del software en lenguaje Arduino, este software, albergó toda la lógica de programación que integraba el hardware ESP32 con la base de datos y la aplicación, y así se logró simular como sería el ingreso a la Universidad para los miembros de la misma.

Mediante la instalación e implementación del sistema de control del pico y placa, se simuló como sería el comportamiento en la Universidad Católica de Oriente, con lo que se observó cómo sería

el ingreso de los vehículos que tienen restricción, la cual no se controlaba hasta antes de la implementación de dicho sistema.

La disminución de vehículos dentro de la universidad, permite reducir el riesgo de realizar daños a otros vehículos, ya que antes, los motociclistas debían mover otras motos para abrir algún espacio y poder estacionar, corriendo el riesgo de realizar algún daño a un tercero.

MySQL, el motor de base de datos seleccionado para administrar la información de los usuarios fue la mejor opción, este permitió manipular la información de una forma sencilla y confiable gracias a sus características de ser un motor de base de datos relacional, entre las cuales se podían realizar sentencias SQL de forma concisa para obtener la información necesaria sobre los usuarios.

Con la implementación de MySQL como motor de base de datos, se facilita la integración de diferentes funcionalidades a futuro, como también la unión con otras tablas de información si estas se llegaran a implementar a futuro.

La construcción del sistema, se realizó con una arquitectura cliente – servidor, donde las tareas van repartidas entre el servidor y el cliente. Con esta arquitectura, se puede diferenciar el trabajo que realiza cada aplicación, como también centralizar la información que se va a gestionar y así se pueden separar las responsabilidades del sistema, lo cual facilita el diseño y el entendimiento del sistema.

Php MyAdmin, entorno de desarrollo que se utilizó para organizar la información, crear la base de datos y sus respectivas tablas, permitió agilizar el proceso de desarrollo, ya que es un software con bajo consumo en la máquina y que permite realizar las operaciones con una velocidad mayor que otros gestores de bases de datos.

El módulo ESP32 que se utilizó para la integración de los componentes electrónicos fue una correcta elección, este permitió integrar de forma sencilla las librerías de Arduino para el desarrollo de la lógica del sistema, como también permitió de forma eficaz, conectar con la base de datos y extraer la información necesaria para autorizar el ingreso a la institución.

Las características que tiene el ESP32, entre ellas la conexión Wi-Fi, facilitaron la instalación del sistema, ya que solo con estar conectado a una red, puede funcionar perfectamente y acceder a la información del servidor donde esté almacenado.

Arduino fue el IDE de desarrollo escogido para realizar el desarrollo e integración de los microcontroladores, este entorno de desarrollo nos permitió avanzar de forma rápida, ya que es un entorno de programación simple y directo, que funciona en multi plataforma y que nos simplificó el proceso de trabajar con los microcontroladores.

En un estudio que se hizo en la Universidad de Antioquia, se pudo evidenciar que la implementación de la restricción de pico y placa en su campus universitario, disminuyó del 100% de su capacidad en parqueaderos a un 80%, esta medida redujo los tiempos de estacionamiento de los usuarios.

## RECOMENDACIONES

Se sugiere a los administradores del sistema de ingreso a la institución educativa, realizar una integración de la información del carnet institucional con la información de los vehículos, para así tener toda la información centralizada en un mismo lugar y que sea más fácil para los administradores y los usuarios acceder a este sistema de control.

Se recomienda realizar una revisión periódicamente de la información almacenada, para controlar que la información de las placas de los vehículos si corresponda a su respectivo dueño, y evitar información errónea dentro de la base de datos.

Se aconseja a la Universidad Católica de Oriente implementar características adicionales que el sistema puede soportar y que no es compleja su integración, como un lector inteligente que valide que la placa en el carnet institucional coincide con la que tiene el vehículo que está ingresando a la institución.

## BIBLIOGRAFÍA

- Abel Polivio Remache Coyago, S. F. (01 de Julio de 2020). *Dialnet*. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=6076491>
- Alberto, J. (04 de octubre de 2020). *Printerest*. Obtenido de [https://co.pinterest.com/pin/464152305323583855/?autologin=true&nic\\_v2=1a6omkKQF](https://co.pinterest.com/pin/464152305323583855/?autologin=true&nic_v2=1a6omkKQF)
- Android*. (s.f.). Obtenido de <https://developer.android.com/sdk/index.html>
- Aprendum, E. (04 de octubre de 2020). *Aprendum* . Obtenido de <https://www.aprendum.com/blog/lenguajes-de-programacion-mas-demandados-2017/>
- Arca Electrónica*. (04 de octubre de 2020). Obtenido de <https://www.arcaelectronica.com/products/tarjeta-de-desarrollo-esp32-modulo-wifi-bluetooth-esp-32>
- Arduino, A. (03 de octubre de 2020). *Aprendiendo Arduino*. Obtenido de <https://aprendiendoarduino.wordpress.com/2018/10/17/pantalla-lcd-i2c-en-arduino/>
- Asana*. (04 de octubre de 2020). Obtenido de <https://asana.com/es/uses/kanban-boards>
- AWS*. (04 de octubre de 2020). Obtenido de <https://aws.amazon.com/es/rds/mysql/>
- Bigtrónica*. (04 de octubre de 2020). Obtenido de <https://www.bigtronica.com/centro/componentes-electromecanicos/rele-s/1352-tarjeta-rele-optoacoplada-5v-hi-1-canal-5053212013523.html>
- Brown, S. (2013). *Software Architecture for Developers*. LeanPub.
- Castro, F. L. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/27411/1/TESIS%20SANCHEZ%20CASTRO%20FRANCISCA%20LORENA.pdf>



alidad\_en\_bicicleta\_en\_el\_Campus\_Universitario\_Rodrigo\_Facio\_Universidad\_de\_Costa\_Rica

HID, T. (03 de octubre de 2020). *Tarjetas HID*. Obtenido de [https://www.tarjetashid-mifare-rfid.com/que-es-tarjeta-proximidad-rfid.html#:~:text=RFID%20\(siglas%20de%20Radio%20Frequency,tarjetas%2C%20transpondedores%20o%20tags%20RFID](https://www.tarjetashid-mifare-rfid.com/que-es-tarjeta-proximidad-rfid.html#:~:text=RFID%20(siglas%20de%20Radio%20Frequency,tarjetas%2C%20transpondedores%20o%20tags%20RFID).

*IEEE*. (s.f.). Obtenido de <http://standards.ieee.org/index.html>

IEEE. (1990). *Glosario Estándar de Ingeniería de Software Terminología*. New York.

*IEEE830*. (s.f.). Obtenido de <http://standards.ieee.org/findstds/standard/830-1998.html>

*Indiamart*. (04 de octubre de 2020). Obtenido de <https://www.indiamart.com/proddetail/rfid-tags-7563075688.html>

Isaac Andrés Sánchez Salazar, K. C. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/28134/1/B-CINT-PTG-N.286%20S%20a1nchez%20Salazar%20Isaac%20Andres.pdf>

*Itsitio*. (04 de octubre de 2020). Obtenido de <https://www.itsitio.com/cl/las-soluciones-control-acceso-zkteco-llegan-intcomex/>

Jalón, L. A. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://192.188.52.94/bitstream/3317/7653/1/T-UCSG-PRE-ING-CIS-142.pdf>

*JAVA*. (s.f.). Obtenido de <http://www.oracle.com/technetwork/java/index.html>

JOHN JAIRO POSADA HENAO, V. F. (06 de Junio de 2020). *Scielo*. Obtenido de <http://www.scielo.org.co/pdf/dyna/v78n165/a11v78n165.pdf>

*kanban*. (s.f.). Obtenido de <http://www.desarrolloweb.com/articulos/desarrollo-agil-kanban.html>

Katherine Andrea Cuartas Castro, J. C. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://repository.udistrital.edu.co/bitstream/11349/5903/1/CruzadoJimenezJoseCarlos2017.pdf>

Ken Schwaber, J. S. (2011). *La Guía de Scrum*.

Lindao, L. N. (03 de octubre de 2020). *Google Scholar*. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/13461/1/UPS-GT001798.pdf>

Llamas, L. (03 de octubre de 2020). *Luisllamas*. Obtenido de [https://www.luisllamas.es/arduino-rfid-mifare-rc522/#:~:text=El%20RFID%20\(Identificador%20por%20radiofrecuencia,aut%C3%B3mata%20o%20procesador%20como%20Arduino](https://www.luisllamas.es/arduino-rfid-mifare-rc522/#:~:text=El%20RFID%20(Identificador%20por%20radiofrecuencia,aut%C3%B3mata%20o%20procesador%20como%20Arduino).

Locken. (03 de octubre de 2020). *IAGua*. Obtenido de <https://www.iagua.es/market/locken/acceso-inteligente>

Marco Antonio Avendaño Quinde, C. A. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/48872>

*Merca Electrónico*. (04 de octubre de 2020). Obtenido de <https://tienda.mercaelectronico.com/accesorios-para-control-de-acceso/tarjeta-rfid.html>

*MYSQL*. (s.f.). Obtenido de <http://www.mysql.com/>

Nicolás Cubillos Gracia, J. S. (03 de octubre de 2020). *Google Scholar*. Obtenido de [https://repository.ucatolica.edu.co/bitstream/10983/22524/1/ArquitecturaIoTParaParqueaderosInteligentesEnLaCiudadDeBogota\\_625372\\_625386\\_2018\\_11\\_19.pdf](https://repository.ucatolica.edu.co/bitstream/10983/22524/1/ArquitecturaIoTParaParqueaderosInteligentesEnLaCiudadDeBogota_625372_625386_2018_11_19.pdf)

Palacio, D. P. (Agosto de 2008). *Repositorio UNAL*. Obtenido de <https://repositorio.unal.edu.co/handle/unal/70221>

Palacio, D. P. (Agosto de 2008). *repositorio.unal.edu.co*. Obtenido de <https://repositorio.unal.edu.co/handle/unal/70221>

Palacio, D. P. (06 de Junio de 2020). *Bdigital*. Obtenido de [http://bdigital.unal.edu.co/2406/2/43203448.2008\\_2.pdf](http://bdigital.unal.edu.co/2406/2/43203448.2008_2.pdf)

*PMO Informática*. (04 de octubre de 2020). Obtenido de <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>

*Primefaces*. (s.f.). Obtenido de <http://www.primefaces.org/showcase/mobile/index.xhtml>

Primefaces. (s.f.). *Primefaces*. Obtenido de <http://www.primefaces.org/whyprimefaces>

*programación XP*. (s.f.). Obtenido de <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>

*Random Nerd Tutorials*. (4 de Noviembre de 2020). Obtenido de <https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/>

*Robotisan*. (04 de octubre de 2020). Obtenido de <https://www.robotistan.com/tower-pro-sg90-rc-mini-servo-motor-eng>

Rodríguez, D. (11 de octubre de 2020). *Lifeder*. Obtenido de <https://www.lifeder.com/metodo-cientifico-experimental/>

Rodríguez, J. F. (20 de Junio de 2020). *Conama9*. Obtenido de [http://www.conama9.conama.org/conama9/download/files/CTs/2717\\_JRozas.pdf](http://www.conama9.conama.org/conama9/download/files/CTs/2717_JRozas.pdf)

Sánchez, N. (04 de octubre de 2020). Obtenido de <https://ineverycrea.net/comunidad/ineverycrea/recurso/diferencias-entre-programacion-y-desarrollo-de/eb6a4971-24b9-de53-b38a-8a6ac2ce2a13>

Sangoluisa, G. A. (03 de octubre de 2020). *Google Sccholar*. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/20699/1/CD%2010204.pdf>

*Secretaría de movilidad de Medellín*. (14 de Agosto de 2019). Obtenido de <https://www.medellin.gov.co/SIMM/fotodeteccion>

Snurmacher, A. C. (04 de octubre de 2020). Obtenido de <https://dronprofesional.com/blog/arduino-pro-ide-la-nueva-version-del-ide-de-arduino/>

SoloArduino. (03 de octubre de 2020). *SoloArduino*. Obtenido de <https://soloarduino.blogspot.com/2017/03/que-es-un-esp32.html>

Taipe, E. J. (03 de octubre de 2020). *Google Scholar*. Obtenido de <http://dspace.esPOCH.edu.ec/bitstream/123456789/10930/1/98T00224.pdf>

*Universidad de Buenos Aires*. (s.f.). Obtenido de Facultad de Ciencias Exactas y Naturales: <http://triton.exp.dc.uba.ar/datamining/index.php/que-es-data-mining>

Vargas, P. D. (06 de Junio de 2020). *Dspace*. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/6505/1/UPS-GT000596.pdf>



inicio Ramos Valencia, R. B. (03 de octubre de 2020). *Google Scholar*. Obtenido de <https://search.proquest.com/openview/5b2c56a6dda949cb7b5f8875c88b1437/1?pq-origsite=gscholar&cbl=1006393>

Zaragoza, M. (04 de octubre de 2020). *Educar para el cambio*. Obtenido de <https://educarparaelcambio.com/arduino/que-es-arduino-y-para-que-sirve/>