|  |  |
| --- | --- |
|  | |
|  | |
| **Elaborado para:** | Código IoT |
|  |  |
| **Fecha de elaboración:** | 6 Mayo 2023 |
| **Vigencia:** | 30 días naturales |
|  |  |
| **Elaborado por:**  **Revisado por:** | David Sánchez y David Torres |
|  |  |
| **Documento:** | Plan de acción del Proyecto Capstone |
|  | |

Formato Kardex

Alumnos

|  |  |
| --- | --- |
| Nombre del proyecto: | Control Interactivo Vehicular |
| Fecha de inicio del proyecto: | 8-marzo-2023 |
| Fecha de conclusión del proyecto: | 26 de mayo del 2023 |
| Descripción: | Este proyecto consiste en la automatización de un control de acceso vehicular, basado en un sistema de detección de placas vehiculares mediante el uso de microcontroladores y la aplicación de tecnologías del internet de las cosas (IOT). |
| Aplicaciones: | Sector público y privado. Control de entradas y salidas de vehículos de uso oficial de la universidad, así como de particulares. Estacionamientos universitarios de diferentes unidades académicas. Empresas, organismos y asociaciones. |
| Objetivo general: | Incrementar el control vehicular, en el acceso y salida de estacionamientos a través de la autorización y no autorización por parte del propietario, y con esto reducir la inseguridad de robo dentro de la Universidad Autónoma del Estado de Morelos. |
| Objetivos específicos: | Brindar mayor seguridad dentro de la universidad mediante el control de accesos vehicular. |
|  | Automatización de un sistema de accesos. |
| Justificación: | Actualmente en distintas organizaciones, a pesar de un equipo de vigilancia (recurso humano y tecnológico) y brindar espacios de estacionamientos para el uso exclusivo del personal, siguen teniendo mucha inseguridad debido a que se ha sustraído vehículos sin autorización del propietario de los estacionamientos oficiales. El uso de nuevas tecnologías de IoT permitirá tener al propietario notificaciones de entrada y salida de su vehículo, que garantice un control total bajo su autorización de dicho evento, a través del uso de Telegram. |
| Integrantes del equipo: | David Torres Moreno |
|  | David Sánchez Vázquez |
| Validado por: | Hugo Vargas |
|  |  |
| Contenido Temático: | Temario.   1. Introducción 2. Principio de funcionamiento 3. Material necesario 4. Herramientas computacionales 5. Circuitos 6. Envío de información 7. Recepción de información 8. Almacenamiento de información 9. Panel de control 10. Automatización 11. Utilización 12. Visualización de datos 13. Instrucciones de uso |
| Productos: | Sistema de IoT con submódulos de:  - Detección de placa (Script en python)  - Envió de notificación al propietario por Telegram (Bot)  - Flujo del proceso en node red  - Maqueta de funcionamiento  - Base de datos de usuarios registrados  - Código de ESP32- CAM para servidor de cámara y movimiento de pluma. |
| Alcances: | Para poder desarrollar y presentar este proyecto, se realiza una maquetación del prototipo para posteriormente implementarlo en el mundo real, Es así que se modela el proceso con sensores de cámara, servo motor, envió de información e interacción con el usuario. Para poder llevar a cabo la implementación del proyecto en un caso de uso real, se requerirá de contar con la pluma de estacionamiento, cámara y un servidor para procesar la captura de la placa del vehículo a través de una cámara. |
| Requisitos: | Contar con los conocimientos en programación en Python, Arduino, Node-red, bases de datos y electrónicos para configurar el circuito y el prototipo. |
| Software: | Ubuntu 20.4  Node-Red  Telegram  MySQL  MQTT  Python |
| Hardware: | ESP32-CAM: Tarjeta de desarrollo IoT ideal para transmitir vídeo e imágenes por internet en diseño compacto.  FTDI: Convertidor USB Serial FTD  Raspberry Pi: Serie de ordenadores monoplaca u ordenadores de placa de bajo costo.  Sensores: Camára del ESP32-CAM  Actuadores: Servo motor  Circuito: Protoboard |

**Introducción**

La inseguridad como problemática social, ha sido estudiada por organismos públicos como el Instituto Nacional de Estadística y Geografía quienes a través de la Encuesta Nacional de Seguridad Pública Urbana (ENSU), en su informe del cuarto cuatrimestre del 2022, arrojaron datos indicando la prevalencia de inseguridad en zonas urbanas, a decir de la ENSU, el 26.3% de la población encuestada, manifestó que al menos un integrante de la familia fue víctima de robo total o parcial de vehículo, robo o asalto en la calle o transporte público, entre otros delitos.

Cómo se observa la inseguridad es un fenómeno que aqueja a la población a nivel nacional, en el caso de Morelos, en 2022, el estado ocupaba el segundo lugar en robo de vehículos a nivel país, se contabilizaron en el año aproximadamente cuatro mil 359 robos, de esta cantidad, mil 417 fueron sustraído en la capital del estado. El contexto en el que se han suscitado los robos indica que un 50% han sido robos con violencia y el otro 50% han sido en automóviles estacionados (Diario de Morelos, 2023).

El impacto de las aplicaciones del Internet de las Cosas (IoT) han generado una marca positiva en la calidad de vida de las personas, de acuerdo con Damayanthy, Galván, Guzmán y Orrante (2022) las áreas más beneficiadas son la seguridad, asistencia sanitaria y educación. Este proyecto, se enfoca en el área de seguridad y en el control vehicular. Inicialmente las aplicaciones del Internet de las Cosas (IoT) al área de transporte, proveían beneficios tales como:

* Gestión y logística de flotas
* Estacionamiento y tráfico inteligente
* Navegación avanzada
* Seguridad del conductor y del pasajero
* Reducción del impacto medioambiental

Actualmente, se realizan esfuerzos por mejorar las experiencias en el sector automotriz, sin embargo, para el tema que ocupa este proyecto, es importante diseñar y fortalecer un sistema de control de acceso vehicular ya que está relacionado con situaciones de inseguridad que aquejan el espacio en donde se aplicará dicho proyecto.

En este sentido, es importante mencionar que el área de aplicación es la Universidad Autónoma del Estado de Morelos, especialmente el estacionamiento de la Facultad de Contaduría, Administración e Informática. Como se ha mencionado, se busca crear un control que brinde seguridad a quienes hacen uso del estacionamiento.

La importancia radica en la necesidad de mantener un mayor control y supervisión de quienes entran y salen del estacionamiento, generando mayor seguridad para los usuarios, debido a que, en años pasados, se han reportado robos al interior de la Universidad Autónoma del Estado de Morelos, al respecto el periódico El Sol de Cuernavaca, señala que el 3 de agosto del 2021, un grupo de sujetos amagó al guardia de seguridad y robaron una pipa y un vehículo marca Ford propiedad de la UAEM. De acuerdo con los informes, se trató de un robo de ladrones comunes que a través de la fuerza sometieron al guardia y extrajeron los vehículos mencionados de la UAEM. Frente a esta problemática la Universidad Autónoma del Estado de Morelos emitió un informe el día 03 de agosto del 2021 en el que señalan las medidas de seguridad tomadas a raíz del evento, tales como reacomodo de cámaras de vigilancia, así como luminarias en aquellas zonas que carecen de iluminación (UAEM, 2021).

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza bajaPor otro lado, frente a las quejas de robo de vehículos por parte de particulares en la UAEM, en el 2019, se solicitó la presencia policiaca del estado al interior de la institución. Ubaldo González Carretes, quien fuera en su momento el Coordinador de Protección de la Dirección de Protección y Asistencia de la institución, argumentó que las quejas por robo de vehículos son pocas al grado de que no existen cifras. Sin embargo, fue necesario solicitar la presencia de autoridades para generar un sistema de supervisión y control.

Imagen que contiene pequeño, tabla, agua, barco

Descripción generada automáticamenteEn la actualidad, la interacción entre la inteligencia artificial, el Internet de las Cosas (IoT) y los microcontroladores ha abierto un nuevo horizonte en el desarrollo de sistemas de control vehicular. Estos avances tecnológicos han permitido la creación de un control vehicular interactivo de vanguardia, capaz de ofrecer una experiencia de conducción completamente integrada, personalizada, pero sobre todo segura. Es así, que los modelos tradicionales de como operan actualmente los estacionamientos, se transforman con el uso de IoT.

El control vehicular interactivo con inteligencia artificial, IoT y microcontroladores combina múltiples elementos para optimizar la eficiencia y seguridad dentro de los planteles universitarios. La inteligencia artificial proporciona al sistema la capacidad de aprender y adaptarse a las preferencias y necesidades del conductor, permitiendo una interacción más intuitiva y eficaz.

Por otro lado, el Internet de las Cosas permite la conexión y comunicación en tiempo real entre el vehículo y su entorno, mientras que los microcontroladores desempeñan un papel fundamental en este sistema, ya que son responsables de coordinar y ejecutar las diversas tareas y funciones del control vehicular interactivo. Estos dispositivos compactos y potentes permiten la integración de componentes electrónicos, como cámaras y mucho más, brindando un mayor control de acceso.

La sinergia entre la inteligencia artificial, el IoT y los microcontroladores en el control vehicular interactivo abre las puertas a una serie de beneficios revolucionarios pues este control vehicular interactivo representa un salto significativo en la evolución de los sistemas de control automotriz. Con capacidades de aprendizaje automático, conectividad total y procesamiento en tiempo real.

**Principio de funcionamiento**

El principio de funcionamiento del proyecto Control Interactivo Vehicular (CIV) se basa en un proceso de captura, procesamiento y análisis de imágenes. A través de una cámara integrada en el acceso al estacionamiento se capturan imágenes de las placas de los automóviles, estas se envían al microcontrolador para su procesamiento utilizando algoritmos de visión por computadora y técnicas de procesamiento de imágenes, donde se extraen las placas (caracteres que la conforman) vehiculares de las imágenes capturadas. La IA utiliza técnicas de aprendizaje automático para procesar y comprender la información de las placas.

Con la ayuda de bases de datos de registro de vehículos, la inteligencia artificial puede comparar la información de las placas con la información almacenada y realizar diversas acciones. Por ejemplo, puede verificar si el vehículo está relacionado a algún trabajador de la universidad, a un alumno, etc.

La información se enviará a través de salas de MQTT para su procesamiento. Una vez que se ha identificado la placa que se encuentra relacionada a un usuario, se le notificará al usuario por medio de un Bot generado en Telegram. El usuario podrá tener acceso y podrá realizar diversas funciones dentro del sistema por medio de su dispositivo móvil.

Además, la información de las placas también puede ser utilizada para realizar acciones relacionadas con la seguridad, como para buscar información del usuario que se encuentre relacionado a esta placa o informar al usuario del vehículo que tiene esas placas sobre la entrada o salida inesperada y sucesivamente avisar a las autoridades pertinentes sobre esta acción.

**Material necesario:**

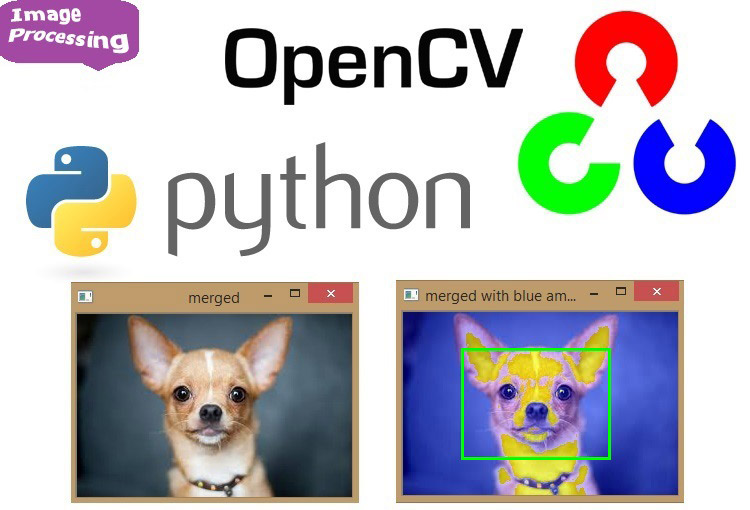
Se requiere de diferentes tipos de hardware y software para poder realizar el prototipo funcional para el proyecto de Control Interactivo Vehicular para poder ponerlo en práctica en la Facultad de Contaduría, Administración e Informática (FCAeI).

El software necesario es el siguiente: Ubuntu 20.4, Node-Red, Telegram, MySQL, MQTT, Python, Fritzzing.

Para el caso del hardware: ESP32-CAM, FTDI, Raspberry Pi, Sensores: Camára de ESP32-CAM, Actuadores: servo motor y Circuito: protoboard.

**Herramientas computacionales:**

Las herramientas computacionales que se utilizaron para este proyecto son: Una computadora con sistema operativo Ubuntu 20.04 con Python, Node-red. Software como se ha mencionado, el IDE de Arduino para poder cargar el programa que ejecutará, por un lado la ESP32-CAM y por otro, lo que requiere de configuración la Raspeberry.: Mysql, node-red, MQTT, el acceso a través de SSH para configurar la Rasberry. Aunado a esto se requieren las siguientes librerías:

También para el proceso de detección de placa si utilizó la librería CV2. OpenCV es una enorme biblioteca de código abierto para visión por computadora, aprendizaje automático, y procesamiento de imágenes y ahora juega un papel importante en la operación en tiempo real, lo cual es muy importante en los sistemas actuales y proyectos de IoT. Al usarlo, uno puede procesar imágenes y videos para identificar objetos, caras o incluso la letra de un humano. En este sentido, se aplicó para realizar un procesamiento de las imágenes de las placas para detectar los caracteres que la identifican. Esta biblioteca utiliza otras librerías como Numpy para trabajar la imagen como una la estructura de matriz OpenCV para su análisis. Para identificar patrones de imagen y sus diversas características, se utiliza el espacio vectorial y operaciones matemáticas en estas características. Para instalar OpenCV, uno debe tener Python y PIP preinstalados en su sistema. [[1]](#footnote-1)

La version de **opencv-python** utilizada es**: 4.7.0.72**. Con el comando siguiente se instala desde pip de Python: **pip install opencv-python**

Diagrama

Descripción generada automáticamentePor otro lado, necesitamos detectar caracteres por lo que es indispensable usar la librería**: Tesseract**. Tesseract es un motor de reconocimiento de texto de código abierto (OCR), disponible bajo la licencia Apache 2.0. Se puede usar directamente, o (para programadores) usando una API para extraer texto impreso de imágenes. Admite una amplia variedad de idiomas. Tesseract es compatible con muchos lenguajes de programación, nosotros lo vinculamos con Python. Se puede usar con el análisis de diseño existente para reconocer texto dentro de un documento grande, o se puede usar junto con un detector de texto externo para reconocer el texto de una imagen de una sola línea de texto.En la anterior figura, se muestra el proceso para identificación de texto a partir de una imagen.

**Circuitos:**

**Diagrama

Descripción generada automáticamente**Para poder armar los circuitos necesarios para este proyecto, requerimos de un software llamado Fritzzing. Este software permite modelar el circuito sobre una protoboard virtual. A continuación se muestra como es que se conectan los diferentes dispositivos que utilizamos:

**Diagrama

Descripción generada automáticamente**Por otro lado, debemos tener cuidado al realizar las conexiones para poder identificar problemas con la conexión de los diferentes dispositivos. A continuación, se muestra la conexión electrónica:

**Envío de información**

Como se ha mencionado anteriormente, el envío de la información es principalmente por MQTT. MQTT son las siglas de Message Queuing Telemetry Transport. Se trata de un protocolo de mensajería ligero para usar en casos de clientes que necesitan enviar datos, que están conectados a redes no fiables o con recursos limitados en cuanto al ancho de banda. Se utiliza principalmente para comunicaciones de máquina a máquina (M2M) o conexiones del tipo de Internet de las cosas.

Para poder enviar información, de detección de placa o apertura de pluma utilizamos dos diferentes salas para cada evento:

1. codigoIoT/mqtt/civ/placas
2. codigoIoT/mqtt/civ/pluma

Ejemplo de como se ven los nodos en Node-Red.

Forma, Rectángulo

Descripción generada automáticamente

Imagen que contiene Icono

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

La configuración para que funcione MQTT es estándar y se realiza sobre la Raspberry Pi o un servidor para que a través de los nodos de Node-red se recuperé esa información y se ejecuten diversos eventos de acuerdo con los datos procesados.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

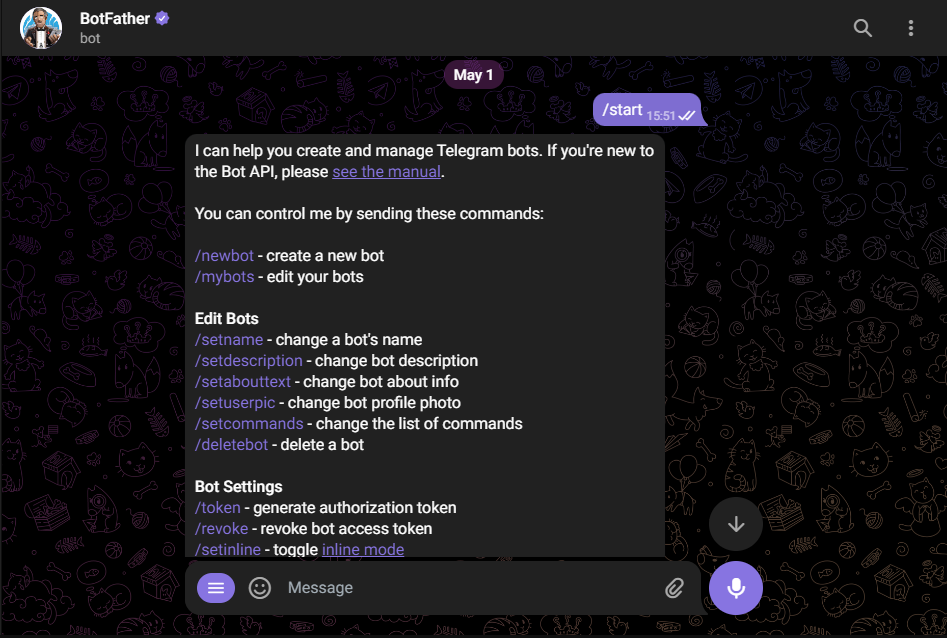
Descripción generada automáticamente

Por otro lado, tenemos la comunicación con el propietario del vehículo para ejecutar ciertas acciones. Para esto se desarrollo un Bot de Telegram, para poder recibir comandos que el sistema pueda ejecutar. A continuación se muestra el procedimiento para generar esta aplicación.

**¿Cómo generar un bot de Telegram?**

Para crear un bot de telegram necesitamos tener una cuenta activa desde la que buscaremos al BotFather el cual nos guiará sobre como crear nuestro propio bot que usaremos para esta práctica.

1. Lo primero que haremos será buscar desde nuestra cuenta el usuario @BotFather e iniciaremos la conversación
2. Este nos contestara con los comandos que utilizaremos para crear nuestro propio bot.



1. Empezaremos insertando el siguiente comando para crear nuestro propio bot

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Ya que se haya creado nos pide que le asignemos como se llamara nuestro bot, insertamos el nombre y enseguida nos pedirá un nombre de usuario por el cual se identificara.
2. Una vez asignado el nombre y el nombre de usuario correctamente nos generará un mensaje de felicitaciones en el cual encontraremos un token único y privado el cual nos permitirá tener acceso a este bot desde nuestro Node Red y con el cual estaremos trabajando durante todo el proyecto.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**¿Cómo descargar el nodo de Telegram?**

Dentro del entorno de Node Red debemos descargar desde el manage palette el nodo “node-red-contrib-telegrambot” el cual contiene un nodo receptor y un nodo emisor que actúan como un bot de Telegram mediante un token generado por @botfather.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ya que se haya instalado correctamente el paquete de nodo de Telegram nos aparecerá en el costado izquierdo donde se encuentran todos los nodos disponibles que tenemos.

Interfaz de usuario gráfica

Descripción generada automáticamente

**¿Como configurar un bot de telegram desde Node Red?**

Ya instalados los nodos de telegram lo próximo por hacer será configurar los nodos receptores y emisor para que puedan tener conexión con nuestro bot anteriormente creado, para esto lo que haremos será añadir en bot a nuestro Node red de la siguiente manera;

* Seleccionaremos un nodo receiver y daremos doble clic sobre él, después nos abrirá un panel en el que vamos a agregar nuestro bot creado.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Damos clic en el incono de editar, nos desplegará un formulario en el cual agregaremos el nombre de nuestro bot, así como el token generado por el @botfather que será nuestra contraseña de acceso al bot.
* Con motivos prácticos y de proyecto solo agregaremos esos dos datos, pero también tiene la opción de agregar a los usuarios que pueden tener acceso a este bot, de la misma manera nos deja agregar todos los chatIds de estos usuarios para que automáticamente se envíe la información necesaria única y exclusivamente a estos usuarios, una vez especificados los datos necesarios daremos clic en el bot ADD para agregar nuestro bot a Node Red.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Una vez que se haya agregado nuestro bot lo único que falta es añadirlo a nuestro nodo, y de esta manera en todos los nodos que utilicemos ya solo bastara con seleccionar el bot, ya que ya se encuentra guardado en el Flow.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

***¿Cómo funciona la respuesta del bot?***

El sistema funciona por medio de un bot de telegram por el cual el usuario enviara mensaje por medio de su dispositivo móvil para realizar diversas acciones. Comenzará por acceder al chat del bot el cual cuando es la primera vez iniciara mando el comando “/start” con el que se ejecutara por primera vez este boto y mandara una respuesta al usuario, dentro del sistema Node-Red esta configuración está representada de la siguiente manera:

Diagrama

Descripción generada automáticamente

Cuando el usuario inicia el chat con el comando “start” o “Hola” (Todos los comandos inician por “/” antes de la palabra para que Node red pueda leer la acción) el sistema ejecuta una función la cual le retornara un mensaje al chat del usuario por medio de su chatId el cual es el identificador único de cada usuario que Telegram genera a cada chat.

Texto

Descripción generada automáticamente

***Consultar mi Información***

Si un usuario quiere consultar su información que está registrada dentro del sistema usara el comando “consultar” el cual ejecutara una función en la que se guardara el chatId correspondiente a este usuario para después realizar la consulta SQL donde el chatId del usuario sea el mismo que el que este dentro de la base de datos.

|  |
| --- |
| flow.set("chatId", msg.payload.chatId);  var chatId = global.get("chatId");  msg.topic = "SELECT \* FROM Personas WHERE CHATID = '" + chatId + "';";  msg.leer = true;  return msg; |

Esta función estará conectada a una base de datos la cual contendrá toda la información del sistema, una vez que esta haya recuperado toda la información del usuario que consulto, el flujo manda a otra función que será la encargada de darle el formato a la información y con el nodo sender se enviara la información final al usuario.

|  |
| --- |
| var consulta = msg.payload;  var message;  for(var i = 0; i < consulta.length; i++){  var row = consulta[i];  message += "\n";  message = "Nombre: " + row.NOMBRE + "\n";  message += "Vehiculo: " + row.VEHICULO + "\n";  message += "Correo: " + row.CORREO + "\n";  message += "Telefono: " + row.TELEFONO + "\n";  message += "Placa: " + row.PLACA + "\n";  message += "Folio: " + row.FOLIO + "\n";    }  var chatId = flow.get("chatId");  msg.payload ={  "chatId": chatId,  "type": "message",  "content": message  }  return msg; |

***Flujo en Node Red***

Gráfico

Descripción generada automáticamente

***Respuesta Telegram***

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

***Buscar Persona***

Si un usuario deseara consultar la información de un usuario por medio de la placa del vehículo, el sistema recibirá la placa por medio del bot ejecutando la función que realizara la consulta SQL a la base de datos, ya que la consulta se haya ejecutado correctamente esta será enviada al usuario de regreso con la información pertinente del usuario que tiene asignada es placa.

|  |
| --- |
| flow.set("chatId", msg.payload.chatId);  var placa = msg.payload.content;  var mensaje = placa.split(" ");  msg.topic = "SELECT \* FROM Personas WHERE PLACA = '" + mensaje[1] + "';";  msg.leer = true;  return msg; |

Una vez que ya tengamos nuestra función que recopilara la información del usuario que se quiere buscar por medio de una consulta a la BD, la función recopilara la información del usuario dentro de un payload, el cual utilizaremos después para darle formato y que se le envié por medio de un sender al usuario.

|  |
| --- |
| var chatId = flow.get("chatId");  var consulta = msg.payload;  var message;  for(var i = 0; i < consulta.length; i++){  var row = consulta[i];  message += "\n";  message = "Nombre: " + row.NOMBRE + "\n";  message += "Vehiculo: " + row.VEHICULO + "\n";  message += "Correo: " + row.CORREO + "\n";  message += "Teléfono: " + row.TELEFONO + "\n";  message += "Placa: " + row.PLACA + "\n";  message += "Folio: " + row.FOLIO + "\n";    }  msg.payload ={  "chatId": chatId,  "type": "message",  "content": message  }  return msg; |

***Flujo en Node Red***

Imagen que contiene Gráfico

Descripción generada automáticamente

**Disponibilidad de espacios en estacionamiento:**

Flujo de Node Red donse se visualiza el proceso para obtener los espacios disponibles en el estacionamiento.



**Recepción de información**

La recepción de información sucede en dos canales, primero la información procesada se transmite a través de MQTT. Si el reconocimiento de la placa se realiza de una forma correcta, se recupera el idChat registrado en la base de datos para que posteriormente se le envíe una notificación al usuario, donde puede autorizar la entrada al estacionamiento. Esto genera que se publique la información en la sala (codigo/IoT/mqtt/civ/placas) y con la autorización del usuario en la sala (codigo/IoT/mqt/civ/pluma) de apertura de pluma. Solamente pueden recibir valores True o False (Apertura de pluma y cierre de pluma, respectivamente). Pero en el registro de los usuarios se realiza a través de la comunicación por medio del Bot de Telegram. A continuación, mostraremos como sr recibe la información para el registro de los usuarios.

***Registro de Usuarios***

Dentro del sistema, los usuarios podrán hacer su registro en su dispositivo móvil por medio de Telegram utilizando el comando "/registro" el cual le desplegara un mensaje con lo requerido para poder hacer el registro en el sistema, el usuario deberá contestar el mensaje del bot para que se registre exitosamente en la BD.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

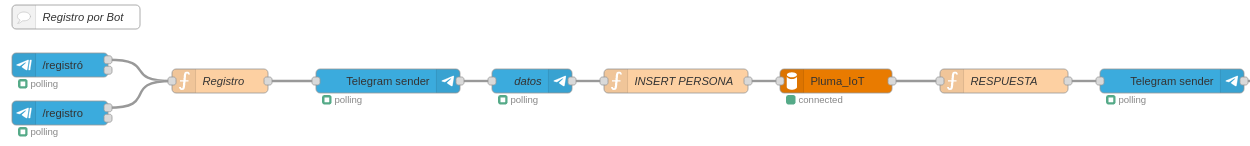
Dentro del flujo de node red se recibirán los datos que el usuario enviara al bot, los cuales se pasaran a la función donde se guardaran para después hacer el registro en la BD. Para hacer el registro conectamos el nodo función al nodo de la BD en la que se guardaran los datos, esta función lo que hará es separar los datos por saltos de línea para convertirlos en arrays, para después guardar cada array en su variable correspondiente y que sea usada para después realizar el insert.

|  |
| --- |
| var chatId = flow.get("chatId");  var mensaje = msg.payload.content;  var mensaje1 = mensaje.split("\n");  var nombre = mensaje1[0];  var vehiculo = mensaje1[1];  var correo = mensaje1[2];  var telefono = mensaje1[3];  var placa = mensaje1[4];  var folio = mensaje1[5];  msg.topic = "INSERT INTO Personas (CHATID, NOMBRE, VEHICULO, CORREO, TELEFONO, PLACA, FOLIO) VALUES ('" + chatId +"','" + nombre + "','" + vehiculo + "','" + correo + "','" + telefono + "','" + placa + "','" + folio + "');";  msg.leer = true;  return msg; |

Ya que se haya creado en registro exitosamente en el sistema, se le enviara un mensaje al usuario, en el que se le confirmara que su registro ha sido exitoso y que puede realizar alguna otra operación.

|  |
| --- |
| var chatId = flow.get("chatId");  msg.payload={  "chatId": chatId,  "type": "message",  "content": "¡Registro exitoso! Vuelva a ejecutar el comando /start o /Hola si desea realizar otra operación"  }  return msg; |

***Flujo en Node Red***



***Respuesta de Telegram***

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

***Flujo de Autorización y apertura de pluma***

A continuación se muestra el flujo de Node red que cambia de estado la pluma para acceder o salir bajo la autorización del propietario del vehículo.

***Diagrama

Descripción generada automáticamente***

**Almacenamiento de información**

***Almacenamiento MySQL***

Para el sistema CIV utilizamos un servidor MySQL en la cual estará montada toda nuestra base de datos. Nuestra BD tendrá la siguiente estructura, la cual constara de 4 tablas principales conectadas entre si con ayuda de las llaves primarias y las llaves foráneas, estas tablas serán:

* ***Tabla Personas***: La cual contendrá la información de los usuarios que estén registrados y que tendrán acceso a los estacionamientos.
* ***Tabla Entrada***: Contendrá todos los registros de las entradas al estacionamiento.
* ***Tabla Salidas***: Contendrá todos los registros de las salidas al estacionamiento.
* ***Tabla Bitácora***: Contendrá un seguimiento de la interacción que tiene los usuarios con el sistema.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Panel de control**

El panel de control, permite obtener datos de quienes y que vehículos ingresan al estacionamiento de la FCAeI. Es así como en este prototipo se ha brindado una interfaz donde se transmite en tiempo real los vehículos que quieren ingresar a dicho estacionamiento. A través de un botón se captura la foto del vehículo, y esto, genera el proceso de identificación de la placa a través de Python y las librerías mencionadas. En este primer prototipo se realizó de esta manera debido a que el dispositivo de la ESP32-Cam no soportó los tres dispositivos: la cámara, el servomotor y el sensor de aproximación. Para un siguiente paso se tiene planeado la incorporación para que se ejecute directamente el proceso de detección de placa si tener que ejecutar el botón de captura de imagen.

Continuando con la explicación del panel de control, permite obtener los datos del vehículo así como la identificación de la placa, esto envía una notificación al propietario y solo si el autoriza se apertura la pluma del estacionamiento. Es así que se la interactividad se brinda al usuario, para permitir ejecutar una acción. En la comunicación por Telegram por parte del usuario ocurren dos cosas. De primera instancia se recibe un comando de autorización (/autoriza) por parte del usuario, ejecutando un proceso de validación y levantamiento de la pluma. Si el comando es de no autorización (/noautoriza) se ejecuta un mensaje para las personas encargadas de tal evento de riesgo en la FCAeI. En este sentido, no se levanta la pluma permitiendo al propietario realizar una inspección de su vehículo y la asistencia por parte de la seguridad.

El sistema también permite tener un conteo de los espacios disponibles en el estacionamiento. El usuario puede acceder a la información para no tener que ir directamente al estacionamiento a validar esto. Esto permitirá que el propietario busque otra opción de estacionamiento.

**Automatización**

En este proyecto la automatización radica principalmente en el registro por parte del usuario a través de una serie de comandos para registrar la información de sus vehículos, sin tener que realizar este proceso manualmente. El registro a través de Telegram solicita: nombre del propietario, información del vehículo e información de comunicación. En este sentido el proceso tiene una mejora con respecto a como se realiza este proceso actualmente. Por ejemplo, en el caso de acceso con una tarjeta, este tiene un costo por creación además de que si se pierde se tiene que crear nuevamente con la información del propietario. En este sentido abogamos al fácil acceso a la plataforma por Telegram, ya que cualquier persona en este momento, cuenta con un dispositivo celular. Otra cosa que se ha automatizado es que el usuario no debe de ir hasta el estacionamiento para saber si existe un espacio si no se puede realizar una solicitud de información a través del bot.

**Utilización**

La utilización del sistema propuesto es para que los propietarios de los vehículos tengan un control de la entrada y salida de su vehículo registrado, a través de un monitoreo por Telegram. A continuación, se describen los pasos para usar:

* Primero un usuario se acerca al estacionamiento para que la cámara pueda detectar la placa correspondiente a su vehículo.



* Segundo, a través del ESP32-Cam se captura una imagen del vehículo para procesarla y obtener la placa detectada. En caso de que se encuentre registrada en la base de datos, se realiza un proceso para recuperar la información de contacto del propietario. Es así que a través del bot de Telegram se le identifica y manda una notificación donde se puede tener respuesta en dos sentidos: autorizar y no autorizar.
* ****Tercero, si el usuario autoriza la entrada al estacionamiento (o salida), es decir responde en Telegram con el comando /autorizar, se publica en la sala correspondiente (código/IoT/mqtt/pluma/) un valor de True que permite ejecutar la acción de cambiar el estado del servo motor a abierto. En el caso que se reciba una respuesta de que no autoriza (/noautorizar) se manda una notificación de que algún evento sospechoso esta ocurriendo. Como por ejemplo, que alguna persona distinta del propietario acceda al vehículo e intente salir del estacionamiento sin el permiso del propietario. En este sentido se alerta al propietario del vehículo y a las autoridades correspondiente encargados de los siniestros. Es aquí la importancia de tener un contacto directo con el propietario debido a que puede estar notificado en tiempo real de diferentes situaciones.

**Visualización de datos**

La visualización de los datos se realiza a través de un dashboard de Node-red. La descripción de los datos que se muestran es como sigue:

* De primera instancia existe un apartado (1) donde se visualiza el vehículo que se encuentra en la entrada y salida del estacionamiento.
* Enseguida de un apartado (2) donde se muestra una tabla con los datos de los vehículos que se encuentran adentro del estacionamiento con los datos de la fecha y hora de entrada. Es decir, solo se muestran los vehículos que han ingresado. Si es el caso que un vehículo sale, entonces ya no aparece más en esta tabla de información.
* El apartado (3) permite ejecutar la función de capturar una imagen del carro en la entrada o salida del estacionamiento, para que se ejecute el proceso de identificación de placa y se muestre cual es la placa detectada.
* En el apartado (4) se muestran los datos del propietario del vehículo identificado a través de la placa.
* Y el apartado (5) muestra los espacios ocupados en el estacionamiento de un total de espacios en el estacionamiento. Es decir, representa el número de vehículos en el estacionamiento.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Instrucciones de uso**

Las instrucciones para poder interactuar y usar este proyecto son las siguientes:

* **Instalar Telegram.** Primero, debemos tener instalado Telegram. Enseguida se deberá iniciar un chat en Telegram con el Bot desarrollado. Esto se puede realizar compartiendo el contacto del Bot. En el caht del bot se requier ejecutar el comando /start o /Hola para generar una comunicación con el bot.
* **Registro de vehículos.** Los propietarios de los vehículos deberán registrar a través del bot de Telegram los datos correspondientes a su vehículo y de contacto. El comando es **/registro** Deberá seguir el formato tal como se solicita para un registro correcto.
* **Consultar información registrada.** Se puede validar el estatus del registro con el comando **/consulta** Este regresará los datos que coincidan en la base de datos.
* **Buscar información por placa.** Se puede realizar una búsqueda con el comando **/buscar PLACA** para obtener la información de alguna placa de un vehículo. Esta operación es para uso exclusivo del personal encargado de la seguridad del estacionamiento.
* **Autorización.** Cuando un vehículo intenta entrar o salir del estacionamiento se envía una solicitud de respuesta a través del bot de Telegram la cual puede ser**: /autorizar y /noautorizar.** Si el propietario del vehículo autoriza entonces se apertura la pluma, en otro caso no se apertura. En este sentido se agregó un comando extra (**/salir**) donde si un vehículo se encuentra adentro del estacionamiento, con este comando puede salir (esto se realizó para excepciones especificas).
* **Disponibilidad de espacios.** Una funcionalidad práctica que existe en este proyecto es que el propietario antes de llegar al estacionamiento y saber si existe algún espacio para estacionarse es el de **/disponibilidad**. El bot regresa la información de cuantos espacios disponibles existen.

**Referencias:**

Cárdenas, K. (2021). Se roban una pipa y una camioneta de la UAEM. El Sol de Cuernavaca, 03 de agosto del 2021. Consultado en: <https://www.elsoldecuernavaca.com.mx/policiaca/se-roban-una-pipa-y-una-camioneta-de-la-uaem-en-morelos-7042516.html>

Damyanthy, Y., Galván, D., Guzmán, I. y Orrante, D. (2022). El impacto del Internet de las Cosas (IoT) en la vida cotidiana. Ciencia Latina Revista Científica Multidisciplinar, 6(2), p. 1-10.

Flores, L. (2023). Incrementa robo de vehículos. Diario de Morelos, 02 de junio del 2023. Consultado en: <https://www.diariodemorelos.com/noticias/incrementa-robo-de-veh-culos>

Instituto Nacional de Estadística y Geografía (2023). Encuesta Nacional de Seguridad Pública Urbana Cuarto Cuatrimestres del 2022. Comunicado de Prensa 21/23. 19 de enero del 2023.

Morelos, R. (2023). En 2022, se robaron cuatro mil359 vehículos en Morelos. La Jornada, 10 de febrero del 2023. Consultado en: <https://www.jornada.com.mx/notas/2023/02/10/estados/en-2022-se-robaron-cuatro-mil-359-vehiculos-en-morelos/>

Universidad Autónoma del Estado de Morelos (2021). Atiende UAEM robo de vehículos al interior de sus instalaciones. Boletín de prensa No. 3883. Ciudad Universitaria, 03 de agosto del 2021.

1. https://opencv.org/ [↑](#footnote-ref-1)