|  |  |
| --- | --- |
|  | |
|  | |
| **Elaborado para:** | Código IoT |
|  |  |
| **Fecha de elaboración:** | 9 |
| **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: | Control de entradas y salidas de vehículos de uso oficial de la universidad. Estacionamientos universitarios de diferentes unidades académicas. Empresas, organismos y asociaciones. |
| Objetivo general: | Incrementar el control de estacionamiento en el acceso y salida de los vehículos a través de la autorización y no autorización por parte del usuario al reducir la inseguridad de robo dentro de la universidad autónoma del Estado de Morelos. |
| Objetivos específicos: | 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, sigue habiendo mucha inseguridad debido a que se ha sustraído vehículos sin autorización del propietario. El uso de nuevas tecnologías de IoT permitirá tener al usuario un control y notificación de entrada y salida de su vehículo. |
| Integrantes del equipo: | David Torres Moreno |
|  | David Sanchez Vazquez |
| 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  - Envió de notificación al propietario  - Flujo del proceso en node red  - Maqueta de funcionamiento  - base de datos de usuarios registrados |
| Alcances: | Para poder desarrollar y presentar este proyecto, se realiza un maquetación modelando el proceso con sensores de cámara, servo motor, envió de información. Para poder llevar a cabo la implementación del proyecto en un caso de uso real, se requerirá de contar con la pluma 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, Node-red, bases de datos y electrónicos para armar el circuito prototipo. |
| Software: | Ubuntu 20.4  Node-Red  Telegram  MySQL  MQTT  Python |
| Hardware: | ESP32-CAM, raspebrry, servo motor, protoboard |

**Introducción**

En 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 y personalizada.

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 con inteligencia artificial, IoT y microcontroladores 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.

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

Imagen que contiene pequeño, tabla, agua, barco

Descripción generada automáticamente

**Principio de funcionamiento**

El principio de funcionamiento de un CIV se basa en un proceso de captura, procesamiento y análisis de imágenes, que al momento de utilizar cámaras integradas una pluma de acceso para capturar imágenes de las placas de los automóviles, están se envíen al microcontrolador para su procesamiento utilizando algoritmos de visión por computadora y técnicas de procesamiento de imágenes, se extraen las placas vehiculares de las imágenes capturadas.

Una vez que se ha identificado la placa que se encuentra relacionada a un usuario, la información se enviará por medio de un bot generado en Telegram donde el usuario podrá tener acceso y podrá realizar diversas funciones dentro del sistema por medio de su dispositivo móvil. 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.

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**

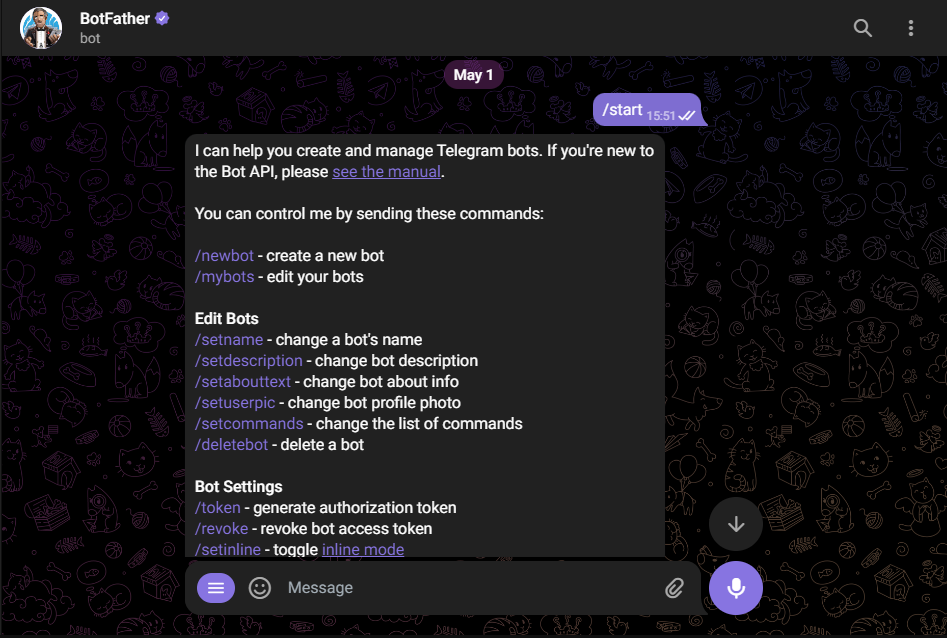
**Herramientas computacionales**

**Circuitos**

**¿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 guiara 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.

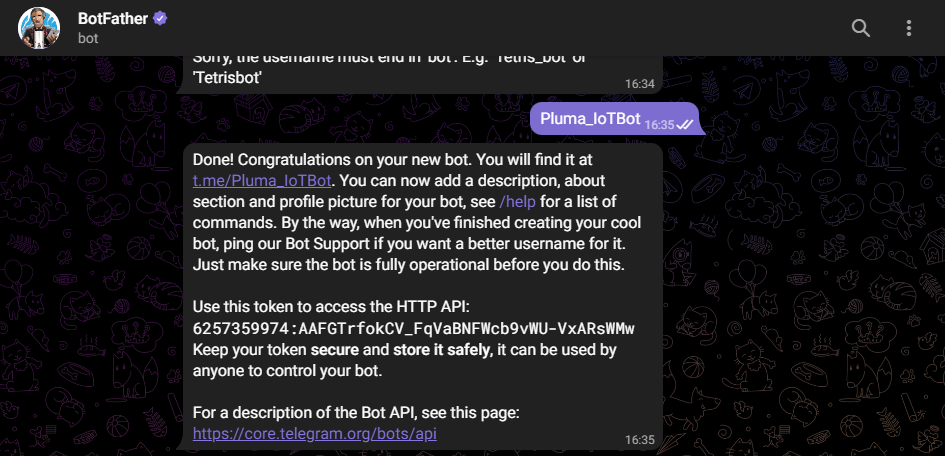


1. Empezaremos insertando el siguiente comando para crear nuestro propio bot

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.



**¿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;

1. 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

1. 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.
2. 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

1. 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

**Envío de información**

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

El sistema funcionara por medio de un bot de telegram por el cual el usuario enviara mensaje por medio de su dispositivo móvil para realizar diversas acciones. Comenzara 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 esta 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.

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

Si un usuario quiere consultar su información que esta 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.

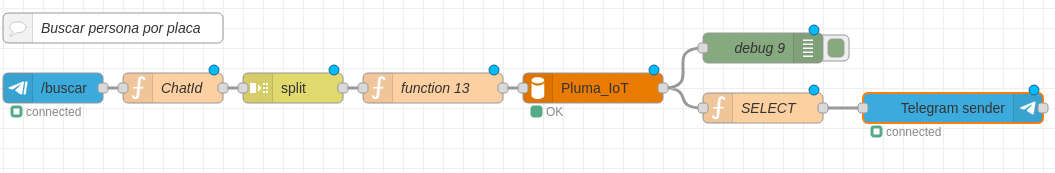
Esta función estará conectada a una base de datos la cual contendrá toda la información del sistema, una vez que esta haya recuperado toda la información del usuario que consulto, el flujo manda a otra función que será la encargada de darle el formato a la información y con el nodo sender se enviara 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.NUM\_TRA + "\n";  }  var chatId = global.get("chatId");  msg.payload ={  "chatId": chatId,  "type": "message",  "content": message  }  return msg; |

Escala de tiempo

Descripción generada automáticamente con confianza media

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.



**Recepción de información**

Diagrama

Descripción generada automáticamente

**Almacenamiento de información**

Almacenamiento MySQL

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Panel de control**

**Automatización**

**Utilización**

**Visualización de datos**

**Instrucciones de uso**