



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

INSTITUTO DE INGENIERÍA INDUSTRIAL Y AUTOMOTRIZ

**“DISEÑO DE UN SISTEMA ELECTRÓNICO DE SEGURIDAD
AUTOMOTRIZ PARA VEHÍCULOS NISSAN VERSA
MODELOS 2017-2021”**

**TESIS PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO AUTOMOTRIZ**

PRESENTA:

HUGO DANIEL MARTÍNEZ OROPEZA

DIRECTOR DE TESIS:

MEDI HERIBERTO ILDEFONSO HERNÁNDEZ MARTÍNEZ

HEROICA CIUDAD DE HUAJUAPAN DE LEÓN, OAXACA; NOVIEMBRE DE 2023

Dedicatoria

Con todo mi cariño, amor y admiración a mis padres Hugo y Magdalena Paz.

A mi hermano Hernando Israel.

A mis abuelitas Matea y Sivilina Fidelfa.

Hugo Daniel.

Agradecimientos

En primer lugar, deseo agradecer infinitamente a mis padres Hugo y Magdalena Paz, por todo su apoyo durante mi vida y formación académica, por motivarme a seguir adelante, hacer que nunca me rindiera. Sinceramente gracias por darme la oportunidad de poder estudiar y culminar la carrera; si estoy aquí, es por ustedes y ahora sí puedo decirles “lo logramos, los quiero”.

A mi hermano Hernando, por ser el mejor compañero de vida, por cuidarme y apoyarme. Gracias por ser un ejemplo y seguir al pendiente en todo momento; sin tu apoyo esto no se hubiera logrado hermano.

A mis abuelitas Matea y Fidelfa, por todo el apoyo y amor que me han brindado, gracias por todo su cariño y consejos, las admiro mucho.

Al profesor Heriberto I. Hernández Martínez, por aceptar mi propuesta de tesis y darle seguimiento, por soportar mis altibajos durante el proceso de este trabajo y por su amistad.

A mis compañeros Santiago, Jorge Francisco, Kevin, Félix, Edrei, Reyna, Lilian y Yael, por brindarme su amistad sincera y ser grandes personas, gracias por brindarme su apoyo para culminar la carrera y la tesis.

A mi novia Valeria, por su apoyo y ánimos, gracias por estar presente en los momentos difíciles y hacer posible este logro.

Al profesor José Eduardo Cruz Mayo, por su apoyo durante la carrera y brindarme a Scott como mi fiel compañero, quien me ha brindado gran felicidad desde que llego a formar parte de la familia.

Al doctor José Antonio Juárez Abad, por el apoyo e interés mostrado para diseñar la PCB del presente trabajo de tesis.

Por último, agradezco a la maestra Tania Martínez Manzo por su valioso apoyo en la implementación de la carcasa del SESA-NiV.

Hugo Daniel.

Índice General

Dedicatoria	iii
Agradecimientos	v
Índice General	vii
Índice de Figuras	ix
Índice de Tablas.....	xiii
Resumen	xv
Capítulo 1. Introducción.....	1
1.1. Antecedentes.....	3
1.2. Planteamiento del Problema	5
1.3. Justificación.....	6
1.4. Limitantes.....	6
1.5. Hipótesis	7
1.6. Objetivos	7
1.6.1. Objetivo General.....	7
1.6.2. Objetivos específicos	7
1.7. Metas	8
1.8. Contenido del Documento de Tesis	8
Capítulo 2. Marco Teórico	11
2.1. Evolución del Sistema de Encendido Automotriz	11
2.2. Sistemas de Encendido.....	12
2.2.1. Sistema de encendido convencional	13
2.2.2. Sistema de encendido transistorizado.....	15
2.2.3. Sistema de encendido electrónico.....	15
2.2.4. Encendido controlado por una ECU	16
2.3. Unidad de Control Electrónico	18
2.4. Sistemas de Seguridad Antirrobo	23
Capítulo 3. Sistema Electrónico de Seguridad Automotriz	27
3.1. Especificación del sistema	27
3.1.1. Sistema de seguridad de los vehículos Nissan Versa	28
3.1.2. Sistema electrónico de seguridad automotriz	31
3.2. División Hardware y Software	37
3.3. Iteración e Implementación.....	38
3.4. Diseño Detallado HW y SW.....	39
3.4.1. Diseño HW	39
3.4.1.1. Módulo ESP32	40

3.4.1.1.1. Integración de los componentes en el módulo ESP32	43
3.4.1.2. Pantalla OLED.....	44
3.4.1.3. Módulo RFID RC522.....	46
3.4.1.4. Módulo SIM800L.....	50
3.4.1.5. Módulo GPS Neo-6m-7m.....	54
3.4.1.6. Dip Switch cuatro posiciones	58
3.4.1.7. Relevadores	60
3.4.2. Diseño SW	61
3.4.2.1. Arduino IDE.....	61
3.4.2.2. MIT App Inventor	62
3.4.2.3. EAGLE	69
3.4.2.4. SolidWorks	70
Capítulo 4. Resultados	73
4.1. Integración de Componentes Hardware y Software	73
4.2. Pruebas del sistema	73
4.2.1. Pruebas de caja gris	73
4.2.2. Pruebas de caja negra.....	77
4.3. Actualización del sistema	85
Capítulo 5. Conclusiones	87
5.1. Líneas Futuras de Investigación	91
Referencias Bibliográficas	93
Anexo A. Instalación del Firmware en el Módulo ESP32	A-1
Anexo B. Diseño de la Placa PCB del SESA-NiV.....	B-1
Anexo C. Diseño de la Carcasa SESA-NiV.....	C-1
Anexo D. Código del SESA-NiV en Arduino	D-1
Anexo E. Código del SESA-NiV en MIT App Inventor	E-1

Índice de Figuras

Figura 1.1. Índice de robos vehiculares en los últimos seis años (AMIS, 2022). .2	
Figura 1.2. Porcentaje de robos por modelos (AMIS, 2022).3	
Figura 2.1. Principio básico del sistema de ignición Kettering (Trevor, 1987)..11	
Figura 2.2. Diagrama del sistema de encendido convencional.....13	
Figura 2.3. Diagrama del sistema de encendido transistorizado.....15	
Figura 2.4. Diagrama del sistema de encendido electrónico.16	
Figura 2.5. Diagrama del sistema de encendido por una ECU (Denton, 2012). 17	
Figura 2.6. Botón de encendido y apagado del motor.....17	
Figura 2.7. Diagrama general de una ECU (Halderman, 2012).19	
Figura 2.8. Bastón bloqueador de volante.25	
Figura 2.9. Sistema de alarma ASIN.....26	
Figura 2.10. Esquema del sistema inmovilizador electrónico.26	
Figura 3.1. Fases del desarrollo de sistemas empotrados.....27	
Figura 3.2. Diagrama eléctrico del sistema de encendido electrónico del vehículo Nissan Versa (Nissan, 2007).29	
Figura 3.3. Diagrama de estados del sistema de seguridad del vehículo Nissan Versa.....30	
Figura 3.4. Diagrama general del SESA-NiV.....33	
Figura 3.5. Diagrama de estados del SESA-NiV.33	
Figura 3.6. Diagrama de flujo del subsistema de encendido electrónico.34	
Figura 3.7. Circuito para el accionamiento de la bomba de combustible por un relevador.....34	
Figura 3.8. Diagrama de flujo del subsistema de seguridad automotriz.35	
Figura 3.9. Diagrama de flujo del subsistema inmovilizador por SMS.35	
Figura 3.10. Diagrama de flujo del subsistema inmovilizador de relevadores. .36	
Figura 3.11. Relevador de activación de inyectores de combustible del SESA-NiV.36	
Figura 3.12. Funcionamiento del sistema de encendido de la bobina DIS y del módulo de activación de la bomba de combustible (Denton, 2012) (Halderman, 2012).39	
Figura 3.13. Módulo ESP32.42	
Figura 3.14. GPIO del módulo ESP32.44	
Figura 3.15. Páginas y columnas de la pantalla 128x64 OLED.....45	
Figura 3.16. Terminales de la pantalla 128x64 OLED.45	
Figura 3.17. Esquema de conexión de los GPIO del ESP32 con la Pantalla 128x64 OLED.46	

Figura 3.18. Transmisión y recepción de datos del módulo RFID RC522.	47
Figura 3.19. Terminales del módulo RFID RC522.	48
Figura 3.20. Sectores de la memoria del módulo RFID RC522.	49
Figura 3.21. Bytes que se almacenan en una línea de un bloque del módulo RFID RC522.	49
Figura 3.22. Esquema de conexión de GPIO del ESP32 con el módulo RFID RC522.	50
Figura 3.23. Comunicación del SIM800L con el módulo ESP32.	51
Figura 3.24. Terminales del módulo SIM800L.	53
Figura 3.25. Esquema de conexión de GPIO del ESP32 con el módulo SIM800L.	53
Figura 3.26. Triangulación GPS para calcular las coordenadas del dispositivo GPS.	54
Figura 3.27. Módulo GPS Neo-6m-7m.	55
Figura 3.28. Terminales del módulo GPS Neo-6m-7m.	56
Figura 3.29. Funcionamiento del módulo GPS Neo-6m-7m con el módulo ESP32 y el módulo SIM800L.	57
Figura 3.30. Esquema de conexión del ESP32 con el módulo GPS Neo-6m-7m.	57
Figura 3.31. Esquema de conexión del ESP32 con el Dip Switch de cuatro posiciones.	59
Figura 3.32. Esquema de conexión del ESP32 con los relevadores.	60
Figura 3.33. Transferencia de datos Bluetooth entre el dispositivo móvil y el módulo ESP32 mediante MIT App Inventor.	62
Figura 3.34. Dispositivo móvil Xiaomi Mi A2 Lite.	63
Figura 3.35. Diagrama de flujo que muestra la secuencia de la aplicación.	64
Figura 3.36. Pantalla de ingreso de contraseña.	65
Figura 3.37. Pantalla para la conexión Bluetooth entre el dispositivo móvil y el módulo ESP32.	65
Figura 3.38. Menú principal de la aplicación del SESA-NiV.	66
Figura 3.39. Pantallas que se abrirán de manera secuencial al autorizar el encendido del vehículo.	67
Figura 3.40. Pantalla que indica al usuario presionar el freno para apagar el vehículo.	67
Figura 3.41. Diagrama de estados de la opción 1: Autorizar encendido.	67
Figura 3.42. Pantalla que se abrirá al pulsar el botón Inmovilizar vehículo.	68
Figura 3.43. Diagrama de estados de la opción 2: Inmovilizar vehículo.	68
Figura 3.44. Pantalla que indica al usuario el proceso de envío de las coordenadas para la localización del vehículo.	69

Figura 3.45. Diagrama de estados de la opción 3: Localizar vehículo por GPS.	69
Figura 4.1. Prueba de caja gris de la pantalla OLED.	74
Figura 4.2. Prueba de caja gris del módulo RFID.	74
Figura 4.3. Validación de la tarjeta leída por el lector RFID.	75
Figura 4.4. Prueba de caja gris del módulo SIM800L.	75
Figura 4.5. Prueba del accionamiento de un actuador al enviar un mensaje SMS de encendido y apagado.	76
Figura 4.6. Mensaje SMS de notificación del encendido y apagado del relevador.	76
Figura 4.7. Prueba de caja gris del módulo GPS.	77
Figura 4.8. Esquema gráfico de las pruebas de caja negra.	77
Figura 4.9. Prueba de caja negra del SESA-NiV.	78
Figura 4.10. Capas Top y Bottom de la PCB del SESA-NiV.	79
Figura 4.11. Placa final del SESA-NiV.	79
Figura 4.12. SESA-NiV conectado a una fuente de 12V.	80
Figura 4.13. Voltajes referenciados a tierra del SESA-NiV.	80
Figura 4.14. Pantallas para ingresar la contraseña y enlazar el SESA-NiV con el dispositivo móvil mediante Bluetooth.	81
Figura 4.15. SESA-NiV emparejado al dispositivo móvil y menú principal de la aplicación.	81
Figura 4.16. Submenús de la aplicación para encender el vehículo.	82
Figura 4.17. Submenús de la aplicación para apagar el vehículo.	82
Figura 4.18. Condiciones que deben cumplirse para la inmovilización del vehículo.	82
Figura 4.19. Envío y recepción de las coordenadas GPS.	83
Figura 4.20. Envío y recepción de mensajes SMS al SESA-NiV.	83
Figura 4.21. Contraseña incorrecta para acceder al SESA- NiV.	84
Figura A.1. IDE Arduino.	A-1
Figura A.2. Pestaña para instalar el URL gestor de placas.	A-2
Figura A.3. Menú del gestor de placas.	A-2
Figura A.4. Menú del tipo de placa al cual se copilará el programa principal.	A-3
Figura B.1. Prueba de funcionalidad del SESA-NiV.	B-1
Figura B.2. Pasos para agregar un nuevo esquemático en EAGLE.	B-2
Figura B.3. Secuencia de pasos para agregar un componente de la librería.	B-2
Figura B.4. Pasos para crear una librería.	B-3
Figura B.5. Crea el recuadro del componente en el esquemático.	B-4
Figura B.6. Menú principal para la creación de un componente.	B-5

Figura B.7. Símbolo en la librería del componente.....	B-5
Figura B.8. Pasos para agregar el Footprint a la librería del componente.	B-6
Figura B.9. Conexión de las terminales del símbolo y del Footprint.	B-6
Figura B.10. Esquemático del SESA-NiV diseñado en EAGLE.	B-7
Figura B.11. Generación de la placa PCB.....	B-8
Figura B.12. Prototipo de la PCB sin las pistas.....	B-10
Figura B.13. Enrutamiento de la capa Top.....	B-11
Figura B.14. Enrutamiento de la capa Bottom.....	B-12
Figura B.15. Iconos Polygon y Rastnest.....	B-12
Figura B.16. Colocación del plano GND en la capa Top.	B-13
Figura B.17. Colocación del plano GND en la capa Bottom.	B-13
Figura B.18. Iconos DRC y ERC.....	B-14
Figura B.19. PCB en la capa Top.	B-15
Figura B.20. PCB en la capa Bottom.	B-15
Figura C.1. Dibujo técnico de la caja.....	C-1
Figura C.2. Dibujo técnico de la tapa.	C-2
Figura C.3. Ensamblaje de la caja.....	C-3
Figura C.4. Impresión 3D de la caja y tapa.	C-3
Figura C.5. Sistema final SESA-NiV.	C-4
Figura E.1. Código en MIT App Inventor para la pantalla Contraseña.	E-1
Figura E.2. Código en MIT App Inventor para la Conexión Bluetooth.	E-1
Figura E.3. Código en MIT App Inventor del Menú principal.....	E-2
Figura E.4. Código en MIT App Inventor de las opciones Accesorios e ignición, Encender vehículo y Apagar vehículo.	E-3
Figura E.5. Código en MIT App Inventor para el mensaje de advertencia de presionar el freno y el botón Start/Stop.	E-4
Figura E.6. Código en MIT App Inventor de la opción Inmovilizar vehículo. ...	E-4
Figura E.7. Código en MIT App Inventor de la opción Localizar vehículo.	E-5
Figura E.8. Código de bloques de la opción Salir de la aplicación.	E-5

Índice de Tablas

Tabla 3.1. División hardware y software del SESA-NiV.....	37
Tabla 3.2. Herramientas para el desarrollo del SESA-NiV.....	37
Tabla 3.3. Comparación de sistemas MCU.....	40
Tabla 3.4. Modos de consumo de energía del módulo ESP32.....	43
Tabla 3.5. Asignación de las terminales del módulo ESP32.	43
Tabla 3.6. Codificación de los bits en el Dip Switch.	59
Tabla B.1. Componentes que integran al SESA-NiV.	B-16

Resumen

Uno de los mayores problemas de seguridad en el país es el número de robos de vehículos, problema que ha ido incrementando, afectando en gran medida a la ciudadanía.

Este trabajo describe el diseño y desarrollo de un prototipo intuitivo y de bajo costo para el sistema electrónico de seguridad automotriz para vehículos Nissan Versa (SESA-NiV) modelos 2017-2021, cuyo propósito es mejorar su sistema de seguridad.

El SESA-NiV es un sistema embebido que controla el arranque, la inmovilización y la localización del vehículo de forma remota. Se implementa en un módulo ESP32 encargado de recopilar, procesar y enviar las señales recibidas por los módulos RFID, GPS y GSM; es controlado mediante una aplicación desarrollada en App MIT Inventor para dispositivos inteligentes con sistema operativo Android.

Capítulo 1. Introducción

La producción de la industria automotriz en las últimas décadas ha ido en aumento; en México se han vendido alrededor de 1,292,004 de vehículos en los últimos dos años.

Debido a diversos factores socio económicos, uno de los mayores problemas en el apartado de seguridad en el país es el número de asaltos y robos de autos. Con el paso de los años, este problema ha ido creciendo, afectando en gran medida a la ciudadanía, provocando el temor de salir a las calles y ser despojados de su unidad vehicular.

A medida que la tecnología avanza, particularmente en lo referente a los sistemas electrónicos que se han implementado en los vehículos, se han desarrollado diversos sistemas de seguridad, sin embargo, la delincuencia ha encontrado maneras de poder evadirlos.

Según datos obtenidos por la Asociación Mexicana de Instituciones y Seguros (AMIS) en lo que va del periodo 2020-2021 ha disminuido el robo de vehículos en un 29.6% con respecto al periodo pico que fue 2017-2018. Esto se debe, en gran medida, a la implementación de sistemas de seguridad en los vehículos de última generación. Aunque cabe mencionar que el robo a los vehículos modernos ha tenido un porcentaje elevado en robo con violencia.

En la Figura 1.1 se puede observar el índice de robos vehiculares que han ocurrido durante los últimos seis años. Es posible observar que los robos han disminuido en un porcentaje aceptable, pero hasta la fecha aún se considera elevado; esto se puede atribuir a que muchos vehículos no cuentan con algún sistema de seguridad, ya que éstos suelen ser costosos y poco confiables. También, se puede observar que el porcentaje de vehículos recuperados es muy bajo.

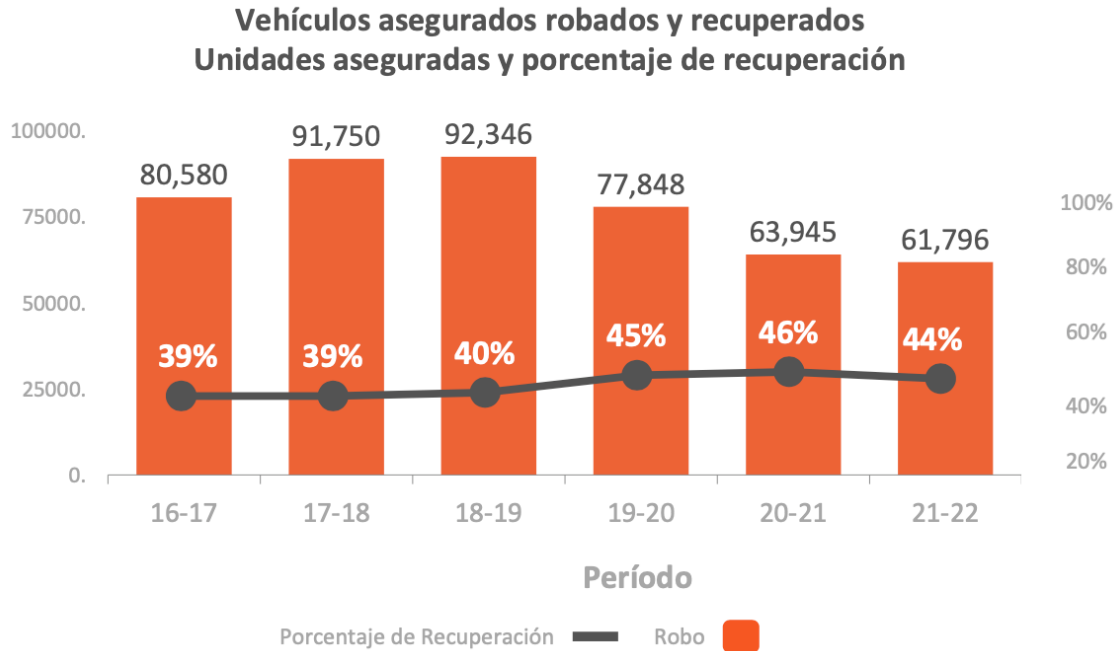


Figura 1.1. Índice de robos vehiculares en los últimos seis años (AMIS, 2022).

Por otro lado, la Figura 1.2 muestra los modelos de vehículos más robados durante los últimos seis años; teniendo como el vehículo con mayor índice de robos el Nissan Versa, seguido de Nissan NP300, General Motors Aveo, Honda CRV y General Motors Beat, respectivamente. La marca Nissan es la más perjudicada en la República Mexicana dada la escasa tecnología de seguridad implementada en sus modelos.

Cabe mencionar que la información de la Figura 1.2 hace referencia a vehículos asegurados, por lo que cabe la posibilidad de que la cifra sea aún más elevada debido a que muchos propietarios de vehículos no adquieren un seguro para sus unidades por su alto costo.

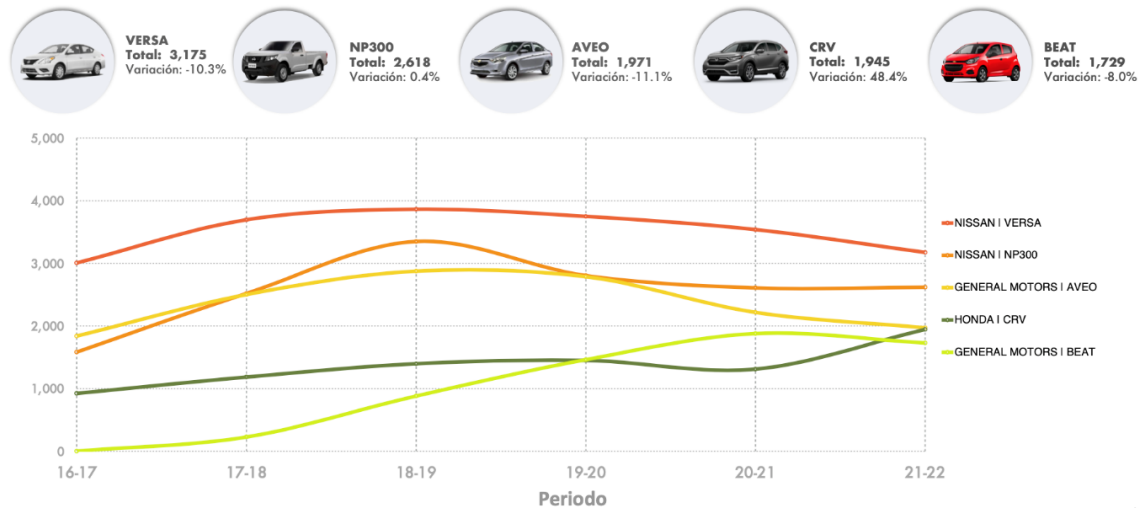


Figura 1.2. Porcentaje de robos por modelos (AMIS, 2022).

1.1. Antecedentes

En el año 2019, Francisco Fernández Morales, estudiante de la Facultad de Ingeniería Mecánica y Eléctrica (FIME) de la Universidad Veracruzana desarrolló un sistema de seguridad antirrobo para vehículo, que incorpora un sistema de control de encendido y apagado, y algunas funciones para control de luces y seguros de las puertas. El dispositivo lleva por nombre “Sisofflock” y se maneja a través de una aplicación instalada en el dispositivo móvil, además, incluye un software de escaneo automotriz que permite identificar las fallas del vehículo para repararlas oportunamente.

En el año 2021, Christian Cárdenas y Diego Villacrés, en su tesis para obtener el título de Ingeniero Mecánico Automotriz, propusieron el desarrollo de un sistema de seguridad antirrobo por inmovilización del motor mediante el corte de combustible, con señal de alerta a un dispositivo móvil y localización GPS (Flores Caballero & Sanchez Muñoz, 2016). En este trabajo se empleó un sistema identificador biométrico de huellas dactilares y se instaló un teclado numérico para la activación de la bomba y de la electroválvula de paso de combustible. Para la implementación de este sistema utilizaron el sistema Arduino MEGA 2560 y el módulo SIM808 GSM, GPRS, GPS, el cual permite conectarse a redes GSM y GPRS para realizar llamadas, enviar y recibir mensajes de texto, conocer

las coordenadas de localización y trabajar con cuatro bandas de frecuencia (850, 900, 1800 y 1900 MHz). Los resultados que se obtuvieron fueron eficientes, el sistema biométrico de huellas tuvo una eficiencia de reconocimiento del 76.66 % y el tiempo de respuesta de la llamada de voz que se emite al propietario del vehículo cuando alguien intenta abrir el vehículo fue de entre 7 a 10 segundos, además, a la tercera llamada se envía un mensaje de alerta. Otra alternativa que se introdujo en este trabajo fue la instalación de un interruptor colocado en lugares estratégicos como la parte del tablero y en la cajuela del vehículo. Al presionar este interruptor, el sistema envía mensajes de alerta cada 10 segundos a un contacto de confianza del usuario, el cual obtendrá las coordenadas de la ubicación del vehículo. Este sistema se implementó como emergencia en caso de un asalto o algún secuestro (Cárdenas Patiño & Villacrés Campoverde, 2021).

En el año 2016, Antonio Flores y Rafel Sánchez, en su tesis para obtener el título de Ingeniero en Comunicaciones y Electrónica desarrollaron un sistema de seguridad vehicular monitoreado vía GSM-GPS. Para este sistema utilizaron el sistema Arduino MEGA 2560 y un módulo GSM SIM 900 para determinar las coordenadas de la ubicación del vehículo. El sistema funciona a través de la comunicación del teléfono del usuario y el microcontrolador, para ello se programaron diversas opciones de seguridad y con un mensaje de texto se habilita o deshabilita la bomba de gasolina para inmovilizar el vehículo. Incluye un modo de seguridad a distancia que, al detectar un movimiento de 100 metros del vehículo respecto a una posición inicial, se activa una señal de alerta y se envía un mensaje al usuario con la ubicación del vehículo.

En el año 2016, Rafael Fabila en su tesis para obtener el título de Ingeniero en Comunicaciones y Electrónica propuso el desarrollo de un sistema de autenticación, monitoreo y detección en tiempo real de vehículos, empleando una interfaz Bluetooth, un transmisor y un receptor de radiofrecuencia NORDIC y un módulo nRF24L (Fabila Ledesma, 2016). Empleó microcontroladores PIC 18F26K22 y PIC16F1938, los cuales reciben los datos del transmisor de

radiofrecuencia NORDIC y del módulo bluetooth clase 2 configurado a una frecuencia de 2.4 GHz. Además, establece un enlace de radiofrecuencia que permite la transmisión de datos y de audio en la banda ISM para la interacción con el usuario.

En el año 2008, Gerardo Mendoza en su tesis para obtener el título de Ingeniero en Robótica Industrial desarrolló de un sistema de seguridad automotriz SAINCO (Sistema Antiarranque codificado), con la ayuda de los microcontroladores AVR MEGA8535 y ARV ATS1200, un control remoto y un teclado colocado en la puerta del vehículo, esto como doble protección del vehículo. Además, cuenta con un sistema inmovilizador, el cual inhabilita tres subsistemas para impedir la marcha del vehículo (Mendoza Contreras, 2008). Cabe mencionar que, si alguien ajeno intenta forzar alguno de los compartimientos, el sistema emite una alarma y activa las luces del vehículo. De igual forma este sistema cuenta con un modo *valet parking*.

1.2. Planteamiento del Problema

En la mayoría de los vehículos se implementa el uso de alarmas como único sistema de seguridad, las cuales al ser sistemas muy sensibles tienden a encenderse al mínimo movimiento del vehículo, lo que ha ocasionado que las personas cercanas al vehículo ignoren la situación y el robo pase desapercibido.

Teniendo en cuenta que la marca con más índices de robos es Nissan, y en particular el modelo Versa, debido principalmente a que son los vehículos más usados como medio de transporte público o familiar; además, presentan un bajo costo y con ello, incorporan poca tecnología en su sistema de seguridad, siendo blancos fáciles para los delincuentes.

Cabe mencionar que la mayoría de las personas que logran comprar un vehículo económico no tienen la posibilidad de poder adquirir un seguro vehicular, cuyos pagos son mensuales y, además, la mayoría de las pólizas de seguros con coberturas básicas no cubren la parte del robo a vehículos, por tal

razón muchas personas optan por no comprar un seguro y corren el riesgo de poder sufrir un atentado en cualquier momento.

Con base en lo anterior, surge la necesidad de contar con un sistema de seguridad para vehículos Nissan Versa modelos 2017-2021, ya que éstos son los vehículos más propensos a ser robados y su recuperación es complicada.

1.3. Justificación

Actualmente, el robo de vehículos sigue en aumento en la mayoría de los estados de la República Mexicana, lo que ha significado un problema para sus propietarios, las empresas de autotransportes y las empresas aseguradoras; particularmente, estas últimas deben pagar un gran porcentaje del valor total de la unidad. Cabe mencionar que la mayoría de los sistemas de inmovilización implementados en el mercado actual no cuentan con un sistema ubicación y rastreo, lo que ocasiona que sea complicado monitorear el vehículo, ya que muchos delincuentes los esconden para posteriormente ser vendidos como piezas.

Por esta razón se propone desarrollar un sistema electrónico de seguridad automotriz para vehículos Nissan Versa modelos 2017-2021 cuyo funcionamiento se base en el diseño de una aplicación móvil que permita autenticar el usuario, conocer la ubicación del vehículo, controlar su encendido/apagado y, si es necesario, inmovilizarlo.

1.4. Limitantes

El presente trabajo de tesis se delimitó a un prototipo demostrativo debido a restricciones presupuestarias, de tiempo y porque no fue factible disponer de un vehículo modelo Nissan Versa a tiempo completo durante el desarrollo de la tesis, además, la fase de potencia implica desarrollar un sistema más complejo capaz de manejar las altas corrientes provenientes de la batería de 12 V del vehículo, realizar esto requeriría un tiempo de investigación más extenso.

La aplicación móvil se ha diseñado como una interfaz simple, interactiva y secuencial centrada en el encendido e inmovilización del vehículo donde se validan los datos del usuario, dado que se trata de un prototipo demostrativo no se consideraron las exigencias que requiere la ingeniería de software en el diseño de la interfaz ni pruebas de usabilidad.

Se propone implementar este sistema siguiendo la misma metodología de forma funcional en futuros trabajos de investigación.

1.5. Hipótesis

Es posible desarrollar con base en un módulo ESP32 un sistema electrónico de seguridad automotriz para vehículos Nissan Versa modelos 2017-2021 cuyo funcionamiento se base en el diseño de una aplicación móvil, en MIT App Inventor, que permita autenticar el usuario, conocer la ubicación del vehículo, controlar su encendido/apagado y, si es necesario, inmovilizarlo.

1.6. Objetivos

1.6.1. Objetivo General

Desarrollar un sistema electrónico de seguridad automotriz para vehículos Nissan Versa modelos 2017-2021 empleando el módulo ESP32.

1.6.2. Objetivos específicos

Para cumplir con el objetivo general de este trabajo de tesis, se establecen los siguientes objetivos específicos:

- Definir los requerimientos del sistema electrónico de seguridad automotriz a desarrollar.
- Desarrollar el sistema de encendido del vehículo Nissan Versa modelos 2017-2021 en el módulo ESP32.
- Desarrollar un sistema inmovilizador para inhabilitar el encendido del vehículo Nissan Versa modelos 2017-2021 en el módulo ESP32.

- Evaluar el funcionamiento del sistema electrónico de seguridad automotriz.

1.7. Metas

Se considera cumplir con las siguientes metas:

- Realizar una investigación documental sobre el funcionamiento de los sistemas de seguridad automotrices.
- Analizar los sistemas de seguridad disponibles en el mercado.
- Estudiar el funcionamiento del sistema de encendido del vehículo Nissan Versa modelos 2017-2021.
- Aprender a programar el módulo ESP32.
- Implementar el sistema de encendido electrónico del vehículo Nissan Versa en el sistema de desarrollo ESP32.
- Realizar pruebas para verificar el funcionamiento del sistema de encendido electrónico.
- Desarrollar una aplicación en MIT App Inventor para autenticar los datos del usuario y habilitar el sistema de encendido.
- Implementar un sistema inmovilizador para habilitar e inhabilitar el encendido del vehículo mediante una señal GPS activada por la aplicación móvil.
- Realizar pruebas al sistema final para verificar su correcto funcionamiento.

1.8. Contenido del Documento de Tesis

El presente documento está organizado en cinco capítulos y tres anexos.

El Capítulo 1 describe la problemática del robo de vehículos y las alternativas existentes para combatirla; se plantea el problema y la propuesta a desarrollar con base en la hipótesis de investigación, así mismo, se presentan los objetivos, metas y metodología de desarrollo.

En el Capítulo 2 se menciona la evolución del sistema de encendido del vehículo, se indican los componentes que han sido integrados al sistema y diversas soluciones que se han llevado a cabo a lo largo del tiempo para hacer más eficiente al motor de combustión interna, con el fin de proveer al usuario un sistema más confiable e interactivo.

En el Capítulo 3 se describe el funcionamiento de los módulos y componentes que componen al SESA-NiV, sus especificaciones, diseño y el procedimiento que se llevó a cabo para la integración e implementación del software y hardware que lo conforman.

En el Capítulo 4 se presentan la integración del software y del hardware, así como los resultados de las pruebas al SESA-NiV, las cuales fueron fundamentales para verificar el correcto funcionamiento del sistema respecto a las especificaciones iniciales.

En el Capítulo 5 se presentan las conclusiones y líneas futuras de investigación. Finalmente, se presentan los anexos con información complementaria sobre la instalación del firmware del módulo ESP32 para su uso en IDE Arduino, el diseño de la placa PCB y el diseño de la caja del SESA-NiV.

Capítulo 2. Marco Teórico

2.1. Evolución del Sistema de Encendido Automotriz

En 1911 se inventó el sistema de ignición denominado sistema Kettering (encendido convencional) y consistía en contactos mecánicos (platinos), un condensador y una bobina (véase Figura 2.1). Este sistema se volvió el estándar en la industria automotriz (Trevor, 1987).

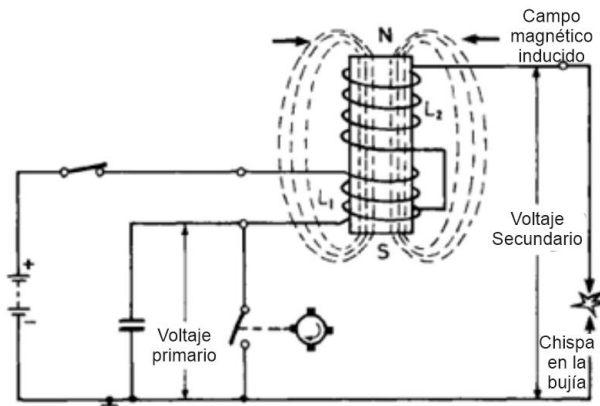


Figura 2.1. Principio básico del sistema de ignición Kettering (Trevor, 1987).

En las primeras décadas del siglo XX se implementaron los sistemas de encendido por interruptor de contactos (platino) y bobina en sincronía con el motor. En 1949, la firma Chrysler implementó la llave convencional de giro, como actualmente se conoce.

A principios de la década de los años 70 surgió el sistema de encendido electrónico para sustituir el sistema de encendido convencional. En los años 90s se implementaron los sistemas de encendido a partir de llaves tipo navaja con sistema de bloqueo remoto integrado, los cuales actualmente comercializan las marcas más importantes.

A comienzos del año 2000 ingresaron al mercado sistemas de encendido por medio de tarjetas integradas y arranque remoto, con la cual se buscó suprimir de alguna forma la llave convencional e incorporar tecnologías de encendido a distancia.

Los avances en la electrónica han permitido eliminar el distribuidor y es una unidad de control electrónico (ECU, *Electronic Control Unit*) la que envía la señal de activación a las bujías. Actualmente, se pueden identificar sistemas integrados de encendido con inyección para la operación del motor a partir del monitoreo de variables como temperatura, combustible, posición del acelerador y revoluciones, entre otras, lo cual busca la optimización de la gestión del motor. También, se encuentran disponibles sistemas de encendido a través de dispositivos móviles llamados *Smart Key*, que además de facilitar la apertura y el cierre de las puertas, permiten controlar el sistema de iluminación exterior, el nivel de combustible y diversas rutinas de mantenimiento.

Los avances tecnológicos parecen no tener límites en esta materia, ya que las marcas premium han implementado sistemas de encendido a través de órdenes de voz y sistemas de reconocimiento facial tanto para el ingreso al vehículo como para su encendido.

2.2. Sistemas de Encendido

En la actualidad existen múltiples sistemas de encendido para los motores de inyección electrónica, los cuales a lo largo del tiempo han ido mejorando. Su funcionamiento consiste en insertar la llave en el interruptor (*switch*) de encendido y girarla a la posición ON, para proporcionar voltaje de alimentación, desde la batería a la bobina de ignición y al módulo de encendido electrónico; al girar el interruptor a la posición Start, se genera un alto voltaje de la chispa que se distribuye a las bujías y se acciona el motor de arranque, el cual es el encargado de vencer la resistencia inicial del cigüeñal y poner a trabajar el motor.

El motor de arranque fue empleado en un vehículo por primera vez por Charles Franklin Kettering en 1912. Su principal descubrimiento fue la ignición eléctrica que libraba la rotación de palancas de hierro y encendían los primeros vehículos de combustión interna. Anteriormente, los autos arrancaban gracias a

la dinámica que imprimía una manivela localizada en su parte delantera. Los conductores tenían que cargar con una palanca que incrustaban bajo la parrilla y con la cual hacían girar el mecanismo que ponía en funcionamiento el sistema de propulsión. A continuación, se describen brevemente los diferentes sistemas de encendido de un vehículo.

2.2.1. Sistema de encendido convencional

El sistema de encendido convencional funciona con una línea primaria de bajo voltaje (12 VCD) y una línea secundaria de alto voltaje (de 10 a 25,000 VCD); la línea primaria conecta la batería del vehículo con el interruptor de encendido, el cual habilita o deshabilita el nivel de voltaje para alimentar la bobina y los platinos; cuando se cierran los platinos se activa el circuito y fluye energía a la bobina, la cual aumenta el voltaje y alimenta la línea secundaria; al abrirse los platinos se corta la corriente y se ocasiona la chispa por inducción en la bobina, la cual se transmite por un cable al distribuidor y de ahí a la bujía donde salta la chispa que hace explotar la mezcla (véase Figura 2.2). Durante el tiempo que el platino está abierto la corriente primaria se almacena en el condensador, además, el distribuidor de platinos tiene muescas en el rotor, una por cada cilindro, que provocan que se abran los platinos para provocar la chispa (Trevor, 1987).

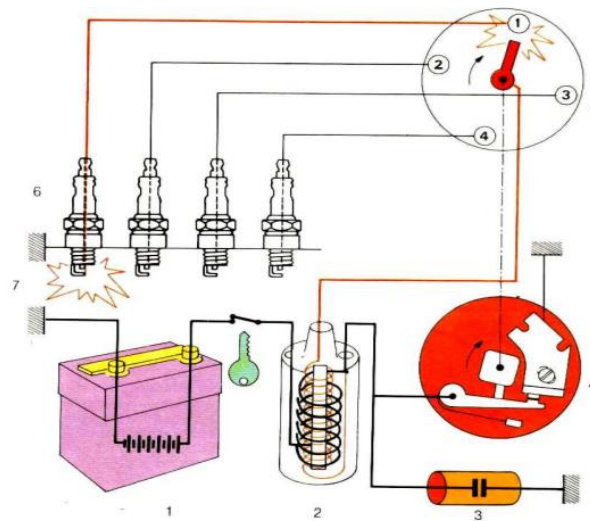


Figura 2.2. Diagrama del sistema de encendido convencional.

Los componentes del sistema de encendido convencional son los siguientes (Mecánica Automotriz Fácil, 2017):

- Llave de contacto: Permite al conductor encender el vehículo por medio del cierre de un circuito eléctrico de encendido al momento de girar la llave. La batería alimenta el circuito primario y el motor de arranque.
- Platino: Conecta o desconecta el circuito primario en la bobina de encendido. Se encarga de interrumpir la corriente en la bobina con la cual se genera una fuerza electromotriz (FEM), la cual se induce por la bobina secundaria y aumenta el voltaje de 12 V a 22,000 V.
- Bobina: Se encarga de generar corriente de alto voltaje usando la que proviene de la batería y pasándola a las bujías.
- Condensador: Absorbe la chispa en los contactos del ruptor, reduce el tiempo de corte de corriente en la bobina y contribuye a que incremente el voltaje.
- Distribuidor: Es el que distribuye la corriente hacia las bujías de acuerdo con el orden de encendido para que se genere la chispa.
- Bujías: Se ubican dentro de los cilindros, donde se genera la chispa que permite la combustión de la mezcla. La chispa se produce gracias a que aumenta el voltaje por la bobina.
- Batería: Dispositivo que almacena energía química que se libera luego de un proceso de reacción en energía eléctrica, se usa por la bobina para enviar la chispa al mismo tiempo que alimenta al motor de arranque.
- Cables: Se utilizan para diversos sistemas del vehículo y en ellos viaja la energía eléctrica; se utilizan según la especificación como calibre y material aislante, entre otros.

2.2.2. Sistema de encendido transistorizado

El sistema de encendido transistorizado añade un transistor entre la bobina y los platinos. El transistor genera una división en la corriente de la batería para minimizar el consumo energético y proporciona a la vez una mayor potencia para la chispa (véase Figura 2.3).

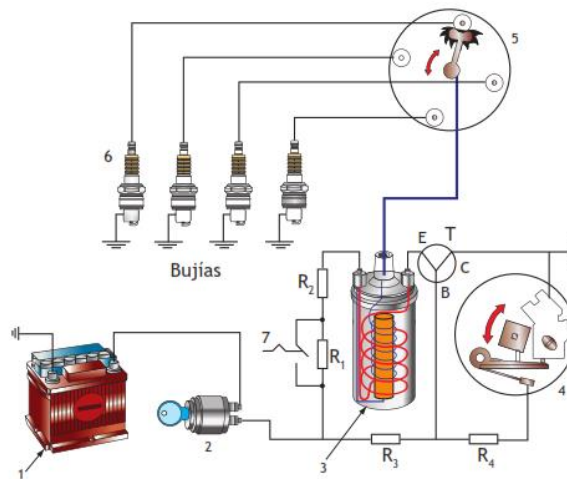


Figura 2.3. Diagrama del sistema de encendido transistorizado.

La división de la corriente tiene dos componentes como destinatarios, los platinos y la bobina, siendo esta última quien recibe la corriente de alto voltaje. Las principales ventajas que tiene este sistema son que genera un voltaje mayor al de un sistema de arranque tradicional y provoca menores pérdidas de energía por arco eléctrico.

2.2.3. Sistema de encendido electrónico

El sistema de encendido electrónico sustituye el platino o ruptor por un generador de impulsos de efecto Hall o de inducción. La pantalla magnética del generador de impulsos va unida al distribuidor, el número de pantallas es igual al número de cilindros, que dirigen el campo magnético hacia la capa Hall cuando se encuentran frente al imán, de esta manera se produce un voltaje positivo en la salida del sensor Hall (Neacsu, 2021). El módulo electrónico toma el voltaje producido por el sensor Hall y cierra el circuito primario de la bobina de encendido (véase Figura 2.4).

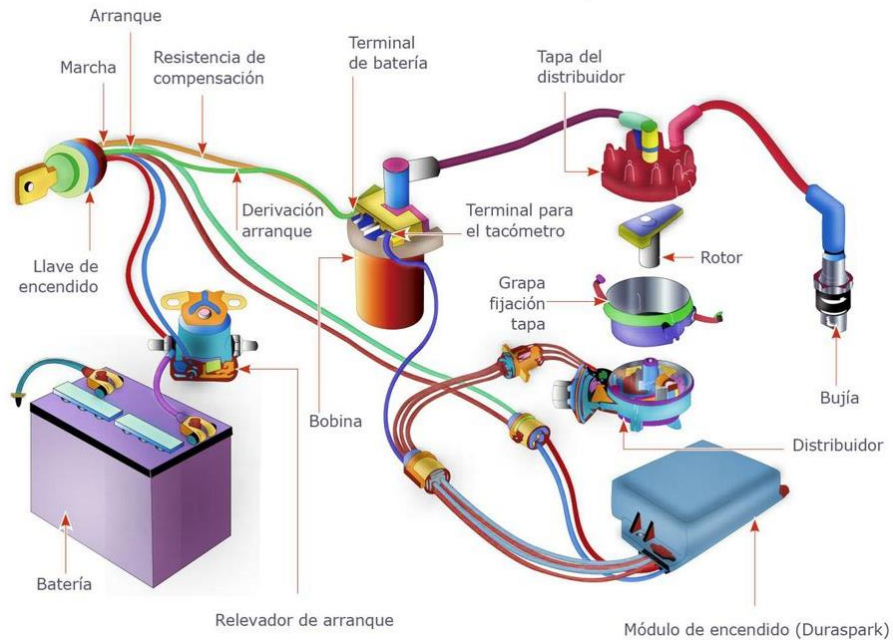


Figura 2.4. Diagrama del sistema de encendido electrónico.

La inducción electromagnética se genera por impulsos eléctricos cuando existe variación del flujo magnético en el interior de la bobina. El entrehierro se ve reducido a medida que el diente del rotor se acerca al núcleo magnético, de manera que aumenta el paso del flujo magnético conforme incrementa la intensidad eléctrica. La variación de intensidad del flujo magnético produce en la bobina un voltaje positivo y justo en ese instante tiene lugar el encendido, que debe coincidir con el salto de chispa en la bujía (Kershaw, 2007).

2.2.4. Encendido controlado por una ECU

Los sistemas de encendido controlados por una ECU¹ activan y controlan la bobina de encendido con base en la información de varios sensores. Este tipo de sistemas de encendido suprimen varios componentes como el distribuidor y el avance de encendido (Denton, 2018). La ECU realiza la distribución de la chispa como muestra la Figura 2.5.

¹ También conocidos como sistemas de encendido electrónico integral.

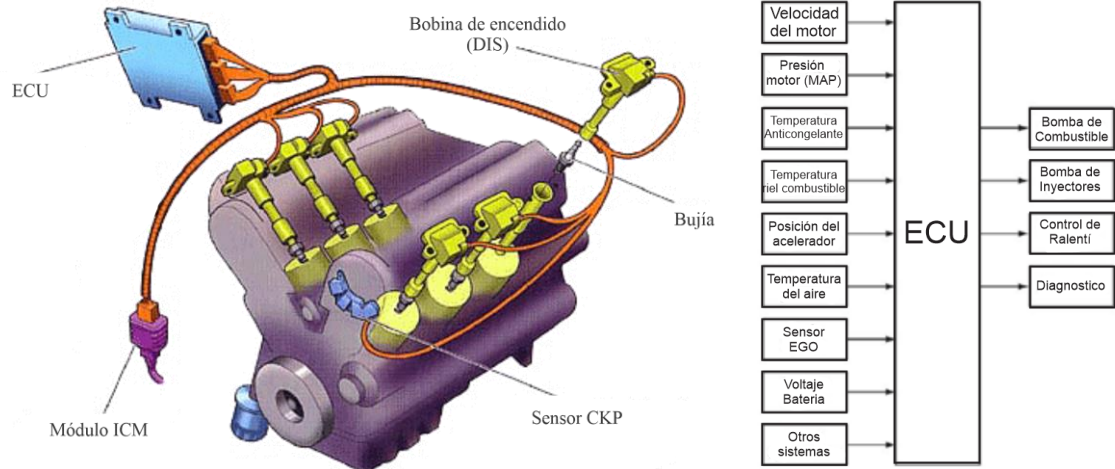


Figura 2.5. Diagrama del sistema de encendido por una ECU (Denton, 2012).

Una variante es el sistema electrónico inteligente de arranque sin llave (*keyless*) que se muestra en la Figura 2.6 . Este sistema sustituye el interruptor de llave por un botón, el cual basa su funcionamiento en un transmisor y un receptor inalámbrico de corto alcance ubicados en la llave y en el tablero del vehículo, capaces de comunicarse entre sí.

El encendido electrónico está incluido en la mayoría de los vehículos, esto debido a que el sistema de encendido convencional tiene algunos problemas (Denton, 2012).



Figura 2.6. Botón de encendido y apagado del motor.

2.3. Unidad de Control Electrónico

La implementación de las primeras ECU se produjo a finales de los años setenta como respuesta de los fabricantes americanos a las exigentes regulaciones de gases de los vehículos. Lo anterior provocó un cambio de mentalidad respecto al paso de lo mecánico a lo electrónico en cuanto a ejecución y regulación de los diversos parámetros con la finalidad de controlar de manera eficaz la combustión del motor.

Las ECU son sistemas electrónicos compuestos por sensores y actuadores, en donde los sensores informan a la ECU y ésta envía la orden necesaria a los actuadores para transformar dicha información inicial. La función de los sensores es registrar diversos parámetros sobre el funcionamiento del vehículo, tales como las revoluciones del motor, temperatura de los sistemas y señal de la posición del acelerador, entre otros. Los sensores actúan como puente hasta la ECU y transforman dichas magnitudes físicas en eléctricas, y a su vez, los actuadores se encargan de convertir las señales eléctricas recibidas en magnitudes mecánicas.

Debido al aumento de funciones y de sistemas electrónicos de los nuevos vehículos, ha surgido la necesidad de diseñar diversas ECU, cada una encargada de una función específica, además de añadirles la capacidad de comunicación entre ellas y una unidad central. La Figura 2.7 muestra el diagrama general de una ECU, cuyos componentes son:

- Interfaz de entradas (sensores): Se encarga de adquirir la información de los sensores, para luego enviarla a la unidad de procesamiento.
- Procesador: Captura la información de los sensores, convirtiendo señales analógicas a señales digitales, enviando la señal de activación a la interfase de salida (realiza cálculos matemáticos).
- Interfase de salidas (actuadores): Hace referencia a los dispositivos que se habilitan de acuerdo con los requerimientos del procesador, y

pueden ser la activación de algún relé o el envío de una señal de activación a algún otro componente electrónico (Halderman, 2012).

- Memoria ROM (*Read Only Memory*): Dispositivo de memoria de sólo lectura, tiene la capacidad de ser una memoria no volátil, es decir, la información obtenida se almacena incluso si el sistema es desconectado.
- Memoria RAM (*Random Access Memory*): Tipo de memoria utilizada para que el procesador tenga acceso a los datos almacenados de forma inmediata.

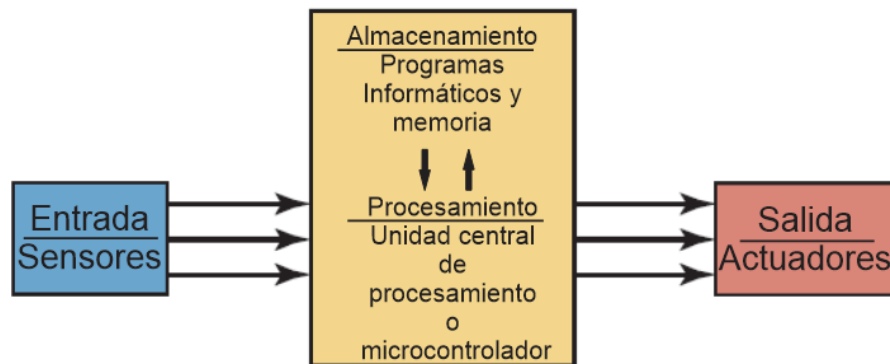


Figura 2.7. Diagrama general de una ECU (Halderman, 2012).

Las ECU modernas tienen la capacidad de ser reprogramadas, con lo cual pueden ser modificadas como consecuencia de un cambio en algún componente, para obtener un adecuado comportamiento o mejorar el rendimiento del vehículo. Los vehículos fabricados a partir de 1996 utilizan ECU que permiten ser reprogramadas mediante el puerto OBDII de manera externa. Para ello se utiliza una computadora portátil que se conecta al vehículo mediante dicho puerto; una vez realizada la configuración correspondiente, se pueden visualizar todas las características de funcionamiento del vehículo, permitiendo modificar la cantidad de combustible que se debe inyectar en el motor, la mezcla correcta de oxígeno y de combustible, y diversos parámetros más. Cabe señalar que en los últimos años ha aumentado el número de componentes electrónicos en los nuevos modelos, con los que existe un mayor nivel de especialización y complejidad tecnológica que se traduce en un mayor número de ECU.

Los niveles de voltaje que alimentan a la ECU deben ser de 12 VCD cuando el vehículo está apagado y hasta 13.8 VCD cuando se encuentra encendido.

A continuación, se describe el funcionamiento de las principales ECU:

- Unidad de control del motor (ECU, *Engine Control Unit*): Es la encargada de determinar la cantidad de combustible que se inyecta basándose en ciertos parámetros. Si el acelerador está presionado a fondo, la ECU se encarga de proporcionar una mayor entrada de aire al motor, además de inyectar más combustible, con base en la cantidad de aire y la presión de la gasolina que esté pasando al motor. Si el motor no alcanza la temperatura ideal de funcionamiento, se inyecta una mayor cantidad de combustible, haciendo que la mezcla sea más rica hasta que el motor logre la temperatura adecuada. Si la ECU detecta que el tiempo de ignición está adelantando en el momento de la fase de compresión, decidirá ralentizar el tiempo en el que se produce la chispa para prevenir el picado en las bielas.
- Control de la distribución variable de válvulas (VVT, *Variable Valve Timing*): Existen motores que poseen distribución de válvulas variable, por lo que la ECU controla el tiempo en el ciclo de motor en el que las válvulas se deben abrir para optimizar el flujo de aire que entra en el cilindro, incrementando la potencia y evitando una inadecuada combustión.
- Módulo de control del tablero de instrumentos (IPC, *Instrument Panel Cluster*): Este módulo se encarga de proporcionar información sobre la condición general y la eficiencia de los principales sistemas de gestión y operación del vehículo, entre los cuales están velocidad, temperatura, presión de aceite del refrigerante del motor y testigos del tablero, entre otros. Dichos sistemas, subsistemas, sensores y actuadores se comunican a través de una red CAN (*Control Area Network*) con el

módulo de control del tren de potencia (PCM, *Powertrain Control Module*).

- Sistema de retención suplementario de airbags (SRS, *Supplemental Restraint System*): Este sistema proporciona una seguridad adicional a los ocupantes en caso de sufrir un impacto. Los sensores con los que cuenta el SRS obtienen información sobre dónde se ha producido el impacto, la velocidad y el ángulo, y la transmiten a las demás ECU. La ECU central procesa y manda el cierre del circuito para activar el SRS de airbag si se cumplen todos los parámetros necesarios. Una vez enviada la señal se activa el detonante que infla las bolsas de aire mediante un proceso de combustión. Por último, se produce el proceso de desinflado inmediatamente después de haber sido infladas por completo, amortiguando el impacto gradualmente y garantizando la visión delantera. Durante este proceso se desprende un polvo que se emplea para la correcta conservación de las bolsas de aire.
- Sistema de frenos antibloqueo (ABS, *Antilock Braking System*): Esta ECU permite al conductor maniobrar durante el proceso de frenado para poder evadir obstáculos, sin perder el control del vehículo. Los componentes del sistema de frenos ABS son los siguientes: sensores de velocidad, unidad electrónica de control, válvulas moduladoras de presión, cilindro de diafragma, válvula de pedal de freno, tanque de aire comprimido y ruedas dentadas. Para el funcionamiento del sistema de frenos ABS todas sus partes se comunican entre sí y brindan un frenado en el cual las ruedas no patinen y el conductor tenga aún más control sobre el vehículo.
- Módulo de control de transmisión (TCM, *Transmission Control Module*): Sistema electrónico que se encarga de recibir la información emitida por componentes instalados en las cajas automáticas y en el módulo de control del motor; se encarga de procesar la información, con

rapidez y precisión, para que el motor haga el cambio de velocidad correspondiente.

- Módulo de control del motor (ECM, *Engine Control Module*): Este módulo controla las funciones de los sistemas eléctricos de un vehículo. Está diseñado con el fin de monitorear las salidas de un sistema y poder controlar las entradas que se envían al mismo. Todos los vehículos cuentan con múltiples sensores en sus sistemas eléctricos, los cuales envían datos al ECM en donde se realizan los cálculos necesarios para poder determinar el tiempo de encendido o el tiempo que debe mantenerse abierto el inyector de combustible para garantizar un rendimiento óptimo del motor.
- Módulo de encendido (ICM, *Ignition Control Module*): Es un dispositivo electrónico que actúa como interruptor de la bobina. Hace uso de un elemento, como puede ser un transistor, para modificar la conducción de la corriente eléctrica por el circuito primario de la bobina de encendido.
- Control electrónico de la carrocería (BCM, *Body Control Module*): Módulo principal de la configuración central del vehículo, controla las funciones de bloqueo de puertas, velocidad del vehículo asociada a la función de bloqueo de puertas (modelo con sistema de bolsa de aire SRS), entrada sin llaves (control de seguro de puertas con mando a distancia), control anti- robo, control de iluminación de cabina, temporizador para ventanas eléctricas, temporizador para desempañar ventana trasera e información de diagnóstico del BCM, entre otras (ISUZU, 2021).
- Módulo de control del tren de potencia (PCM, *Powertrain Control Module*): Es el conjunto de partes que permiten que un auto se mueva e incluye motor, caja de velocidades, ejes y barra de transmisión.
- Sistema de monitoreo de la presión de los neumáticos (TPMS, *Tire Pressure Monitoring System*): Sistema que advierte al conductor del

vehículo sobre un cambio de presión en uno o más neumáticos. Los datos son proporcionados por sensores de presión montados en cada neumático y enviados a la ECU, la cual enciende un testigo de advertencia en el tablero de instrumentos.

2.4. Sistemas de Seguridad Antirrobo

En el transcurso del tiempo se han implementado diversos sistemas de seguridad cuyo objetivo principal es proteger el vehículo de un intento de robo. Las primeras tecnologías surgieron a principios de los años ochenta por General Motors con los sistemas de seguridad antirrobo del vehículo (VATS, *Vehicle Anti-Theft System*). Estos sistemas agregaron un pequeño bloque de carbono, que representa una resistencia eléctrica, a la llave, permitiendo al sistema únicamente operar con la llave indicada.

Años más tarde, las compañías Ford y Toyota implementaron el sistema de transponder, que consiste en un circuito integrado que permite la ignición del vehículo sólo si el decodificador detecta el dispositivo adecuado.

A continuación, se describen los principales sistemas de seguridad antirrobo usados en la actualidad:

- **Bloqueador de volante o bastón:** Este mecanismo de seguridad consiste en bloquear el volante del vehículo mediante la barra que lo inmoviliza (véase Figura 2.8). Es un mecanismo relativamente barato y tiene un uso muy sencillo, ya que sólo se usa una llave para poder abrir el seguro y mover la barra.
- **Alarma y sirena:** Es un sistema que consta de una serie de sensores colocados de forma estratégica dentro del vehículo para poder detectar movimientos o actuaciones anómalas externas o internas (véase Figura 2.9). Al detectar un posible peligro, el sistema emite una alarma o sirena audible para indicar del suceso y tratar de disuadir al asaltante.

- Sistema inmovilizador electrónico: Este sistema incorpora un transponder, con un código único grabado en un circuito integrado dentro de una llave. Este circuito integrado no usa baterías y se alimenta de la señal que proviene de la ECU (véase Figura 2.10). El funcionamiento de este sistema consiste en que, al momento de acercarse a la llave, la ECU envía una señal preguntando si el código es correcto, esta señal se envía a través de la antena que viene alrededor del interruptor de encendido, el cual energiza al transponder para que responda y mande la respuesta de nuevo por la antena al módulo ECU; si el código es correcto, se envía la señal para que el sistema eléctrico encienda y accione el motor de arranque, de otra manera el vehículo no enciende.
- Sistema cortacorriente inalámbrico: Consiste en un bloqueador oculto dentro del vehículo que se conecta de forma remota a un control dentro de un circuito integrado. Al activar dicho control, el auto corta la corriente que alimenta la bomba de gasolina, provocando que el vehículo se quede inmóvil y pueda encenderse hasta que el sistema sea desbloqueado, es decir, cuando se active el relevador que alimenta la bomba de gasolina.
- Sistema Goodlock: La empresa chilena Goodlock desarrolló en 2012 un sistema antirrobo que bloquea al vehículo en tres pasos. Antes de realizar dichos pasos se debe insertar la llave en el interruptor de encendido, poner la llave en ignición, dar contacto para poder activar el sistema y pisar el freno encender el aire acondicionado y ponerse el cinturón de seguridad. Si no se realizan las acciones en el orden programado, el motor no arranca. Lo cual lo hace un sistema invisible, sencillo e inteligente.
- Máquina de humo: En este sistema se instala una máquina de humo en el interior del auto, que se activa cuando alguien logra abrir una de

las puertas del vehículo. La máquina se activará y el humo provocará que el intruso no pueda ver por varios minutos.

- Sistema IGLA 231: Este es un sistema empleado por la empresa AUTHOR ALARM, el cual tiene como objetivo proteger el vehículo de un robo. Este sistema es compatible con la mayoría de las marcas y modelos de vehículos de gasolina, diésel, híbridos y eléctricos. El desbloqueo de la unidad se realiza por medio de un llavero vía radiofrecuencia (*key fob*), un dispositivo inteligente (smartphone) o mediante un código PIN integrado en el vehículo. Para la autorización de su encendido se necesita uno de los tres sistemas mencionados, si la señal no es válida o reconocida, el sistema IGLA bloqueará el motor (AUTHOR ALARM, 2021).



Figura 2.8. Bastón bloqueador de volante.



Figura 2.9. Sistema de alarma ASIN.

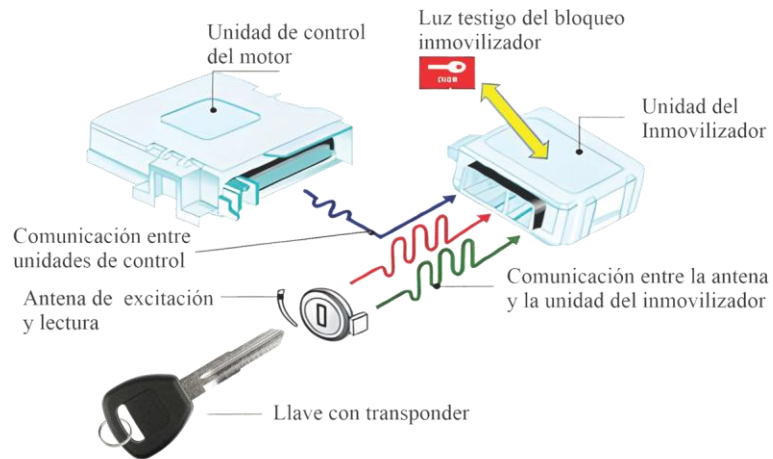


Figura 2.10. Esquema del sistema inmovilizador electrónico.

Capítulo 3. Sistema Electrónico de Seguridad Automotriz

El desarrollo de un sistema empotrado o embebido (*embedded system*) considera el desarrollo en paralelo de software (SW) y de hardware (HW); este proceso contiene funciones específicas y Berger (2002) recomienda realizarlas en siete fases (véase Figura 3.1). Cada una de las fases considera iteraciones para optimizar la solución o producto final. Las siete fases son: Especificación del sistema, División SW y HW, Iteración e implementación, Diseño detallado HW y SW, Integración de componentes HW y SW, Pruebas del sistema y Actualización del sistema. En este trabajo de tesis se consideran las primeras seis fases de dicha metodología de desarrollo, debido a que la actualización del sistema queda fuera de su alcance.

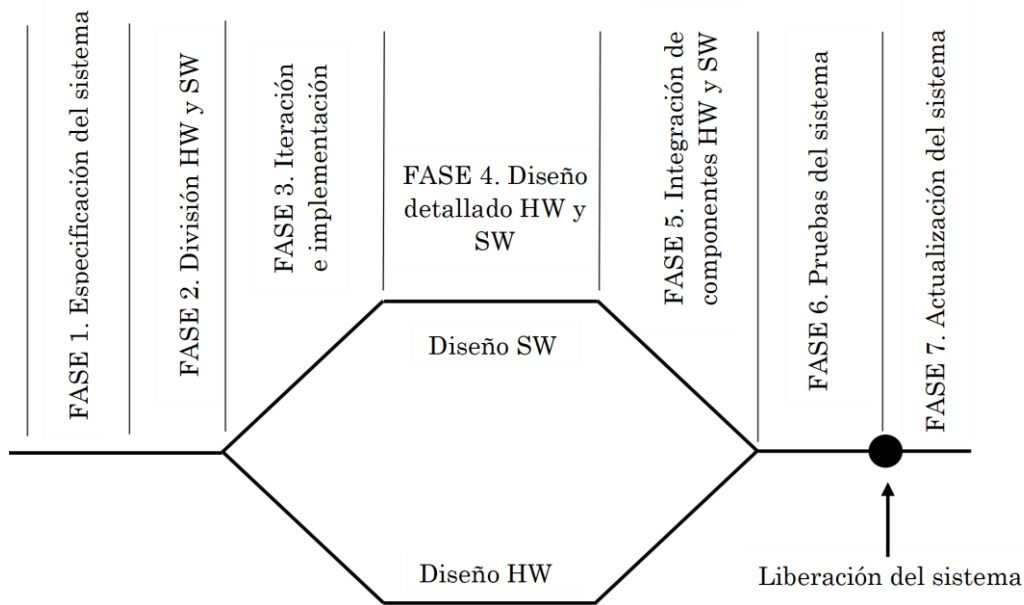


Figura 3.1. Fases del desarrollo de sistemas empotrados.

3.1. Especificación del sistema

El proceso de desarrollo de un sistema empotrado inicia con su especificación, la cual describe *lo que será y lo que hará* el sistema (Berger, 2002) (Chamú Morales, 2005). Para la mayoría de los ingenieros, el desarrollo de un sistema consiste en considerar dichas especificaciones para asegurar un sistema robusto.

3.1.1. Sistema de seguridad de los vehículos Nissan Versa

El sistema electrónico de seguridad automotriz de los vehículos Nissan Versa modelos 2017-2021 proporciona señales de alarma, visuales y audibles, cuando alguien abre una de sus puertas o la cajuela sin autorización. El sistema ayuda a disuadir el robo, pero no lo puede evitar. Además, incorpora un sistema inmovilizador que evita que el vehículo arranque si no se utiliza la llave registrada. Cabe mencionar que el sistema no arranca si es que se presentan las siguientes condiciones: interferencia causada por otra llave registrada, tener cerca un dispositivo de peaje carretero automático o tener un dispositivo de pago automático pegado a la llave.

Los vehículos Nissan Versa incorporan una llave inteligente llamada *Intelligent Key*, la cual está en permanente comunicación con el vehículo, pero únicamente cuando se encuentra en el rango de alcance es cuando a la antena receptora capta la señal del transponder y comienza la comunicación entre la ECU y el transponder, el cual contiene un código único. Una vez que este código es verificado por la ECU, el usuario puede hacer uso del vehículo, es decir puede activar accesorios (ACC) e ignición (ON) y, posteriormente, el encendido del motor (START ENGINE); en caso contrario, el sistema se inmoviliza y no se puede hacer uso de éste. Las funciones de la llave inteligente sólo se pueden usar mientras esté en el rango de operación, el cual es de 80 cm (Nissan, 2021).

La Figura 3.2 muestra el diagrama eléctrico del sistema de encendido del vehículo Nissan Versa modelos 2017-2021, mientras que el diagrama de estados de la Figura 3.3 muestra su funcionamiento.

Inicialmente, la unidad Nissan Versa estará completamente cerrada y tendrá activados los sistemas de alarma e inmovilizador. Una vez que el usuario se encuentre dentro del rango adecuado podrá enviar la señal para desactivar la alarma y los seguros del vehículo, después de esto, el usuario podrá ingresar al vehículo, pero el sistema de encendido aun permanecerá inmovilizado y el interruptor (*switch*) del vehículo estará en posición OFF.

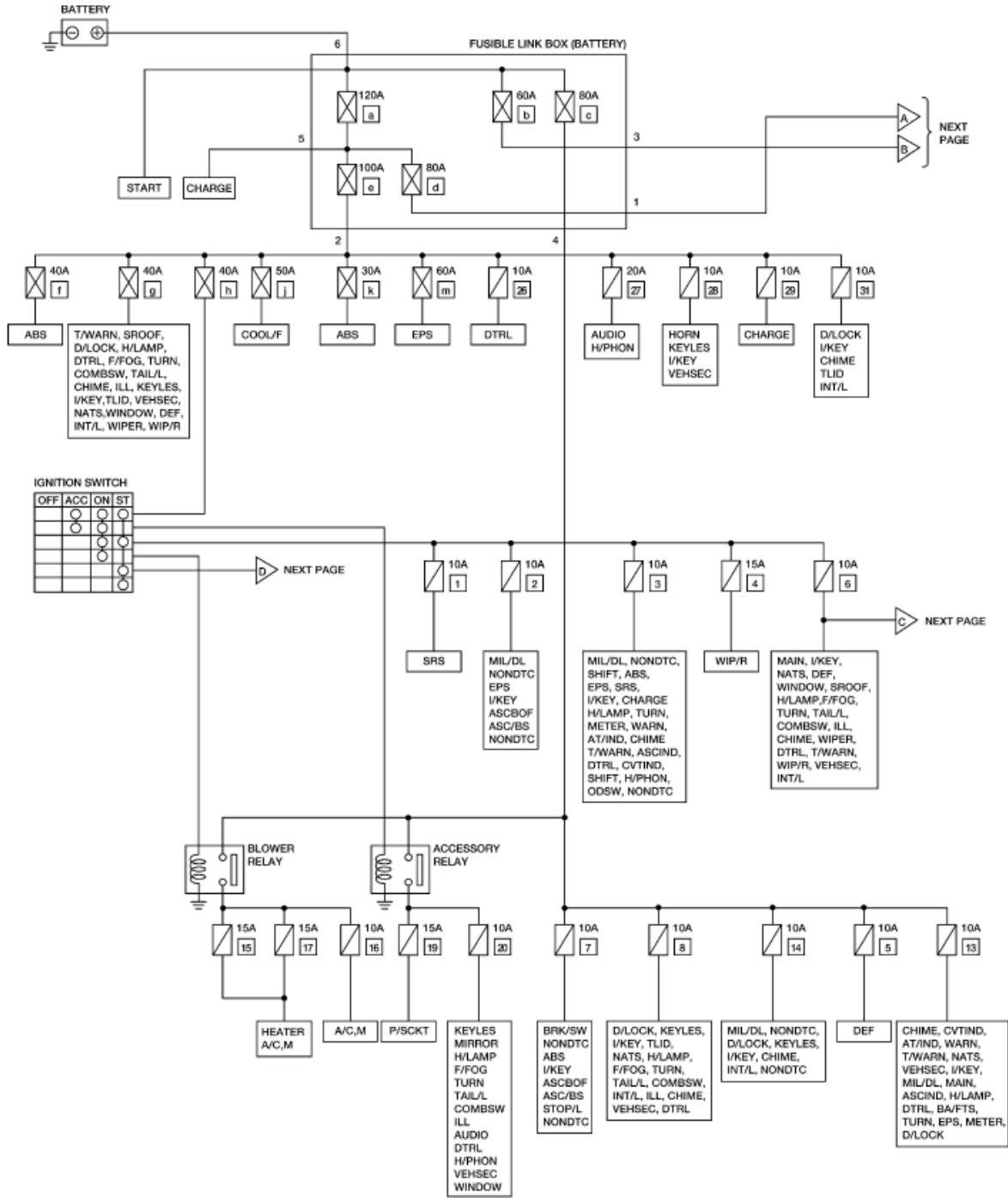


Figura 3.2. Diagrama eléctrico del sistema de encendido electrónico del vehículo Nissan Versa (Nissan, 2007).

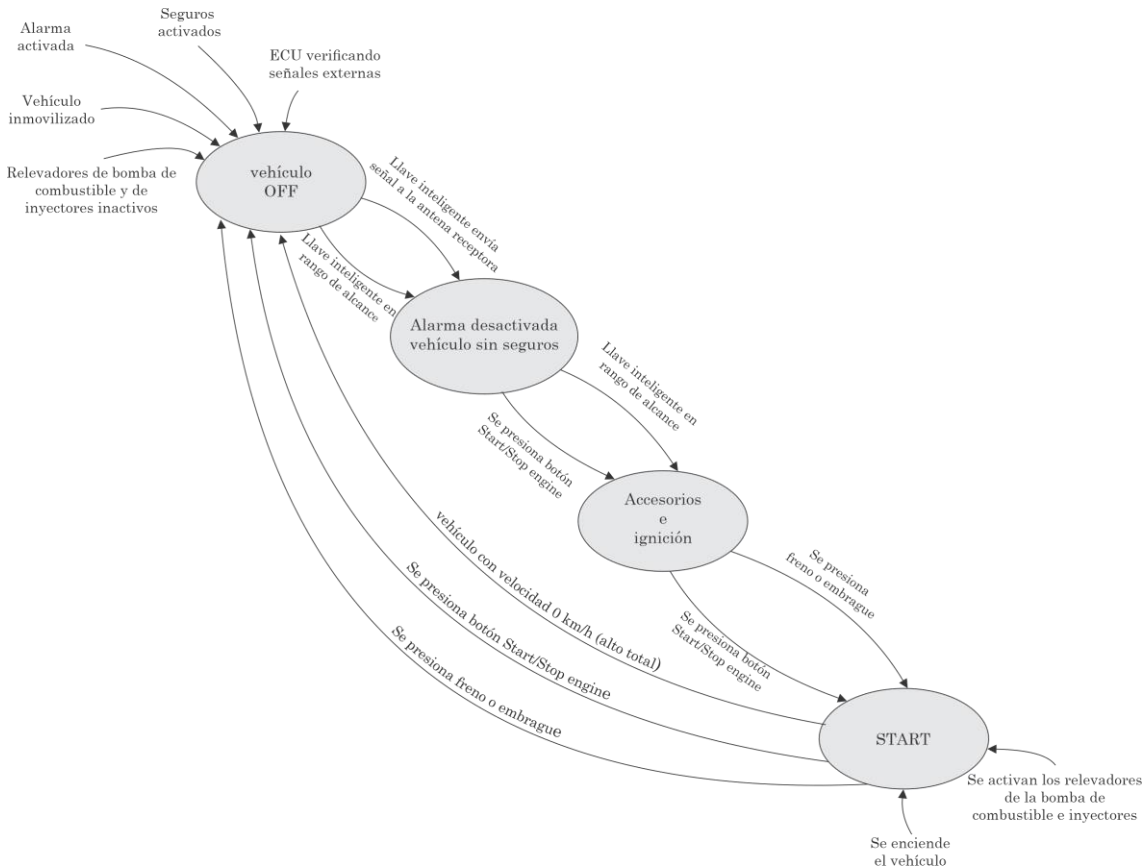


Figura 3.3. Diagrama de estados del sistema de seguridad del vehículo Nissan Versa.

Cuando el sistema detecta el *transponder* que se localiza dentro de la llave, la antena del vehículo envía una señal de radio frecuencia (RF), para excitar el embobinado interno del chip y el *transponder* envía una señal de respuesta; la antena capta dicha señal y la envía a la ECU del vehículo; si la señal es incorrecta, la ECU deja inmóvil la unidad desactivando las líneas de alimentación de la bomba de combustible y de la bobina DIS. Por otro lado, si la señal es correcta, la ECU autoriza el encendido y el interruptor del vehículo pasa a la posición de accesorios (ACC) y, posteriormente, a ignición (ON); solamente podrá ponerse en marcha el motor del vehículo cuando el botón de encendido START/STOP ENGINE se presione junto con el embrague o freno del vehículo (dependiendo el tipo de transmisión, si el vehículo es manual se presiona el embrague y si es automático se presiona el freno). Con ello el interruptor pasa a posición START y el vehículo Nissan Versa podrá ponerse en marcha.

Para apagar el vehículo o motor (STOP ENGINE), éste debe detenerse completamente (0 Km/h) y el usuario debe presionar el botón START/STOP, en ese momento el motor se apaga completamente. Y cuando el usuario salga de la unidad, cierre las puertas y active seguros y alarma, el Nissan Versa quedará completamente cerrado y los sistemas de seguridad e inmovilizador permanecen activos hasta recibir la próxima señal de desactivación correcta (Nissan, 2021).

El principal problema con este tipo de sistema es que sólo cuenta con un sistema inmovilizador y una alarma; debido a esto los delincuentes han encontrado formas de poder vulnerar el sistema, además de poder colocar inhibidores de señales cerca del mismo y con ello robar la unidad con violencia.

3.1.2. Sistema electrónico de seguridad automotriz

Teniendo en cuenta los puntos débiles del sistema de seguridad e inmovilizador del vehículo Nissan Versa, se propone el diseño y desarrollo de un sistema electrónico de seguridad para un vehículo Nissan Versa modelos 2017-2021 (SESA-NiV). La Figura 3.4 muestra el diagrama general del sistema propuesto, mientras que la Figura 3.5 presenta el diagrama de estados para su funcionamiento. Cabe señalar que dicho sistema queda en prototipo. A continuación, se describe cada uno de los subsistemas que lo conforman:

- *Subsistema de encendido electrónico (Starting System):* Este subsistema se implementó en un módulo ESP32 empleando la plataforma de programación de Arduino y considerando el diagrama eléctrico de la Figura 3.2. Se emularon las distintas posiciones del interruptor (*switch*) de encendido mediante las pulsaciones de un *push button* siguiendo el diagrama de flujo de la Figura 3.6. El subsistema de encendido electrónico funciona únicamente cuando el sistema de seguridad autoriza su ejecución (véase Figura 3.7). “Los relés se utilizan para conmutar una corriente alta por medio de corriente de control relativamente baja” (Murugesan, y otros, 2011).

- *Sistema de seguridad automotriz:* Este sistema está conformado por tres funciones elementales, autenticar al usuario para poder habilitar el sistema de encendido electrónico del vehículo, inhabilitar el funcionamiento del vehículo y localizar el vehículo. Para ello se desarrolló una aplicación móvil en MIT App Inventor que considera cuatro procesos: a) Configuración de la aplicación, b) Arranque del vehículo, c) Inmovilizar el vehículo y d) Localización del vehículo. La aplicación autentifica al usuario mediante la introducción de un código de seis dígitos en el dispositivo móvil y, si es correcto, envía la orden, vía Bluetooth, al módulo ESP32 para habilitar al subsistema de encendido electrónico (véase Figura 3.8).
- *Subsistema de inmovilizador automotriz:* Este subsistema basa su funcionamiento en los módulos GPS y GSM. Permite conocer la localización en tiempo real y la inmovilización del vehículo, notificando mediante el envío de mensajes SMS al dispositivo móvil (véase Figura 3.9). Este subsistema habilita o deshabilita los relevadores SPDT (*Single Pole Double Throw*) que alimentan a la bomba de combustible, a la bobina DIS y a los inyectores de combustible, con lo cual se impide que se genere la reacción química (ignición) y con ello el motor no puede completar su ciclo de trabajo (véase Figura 3.10). El relevador SPDT permite cambiar la conexión eléctrica de un circuito entre dos posiciones diferentes cuando el embobinado interno se excita, es decir, pasa de la posición normalmente cerrado (NC) a normalmente abierto (NO) como se muestra en la Figura 3.11.

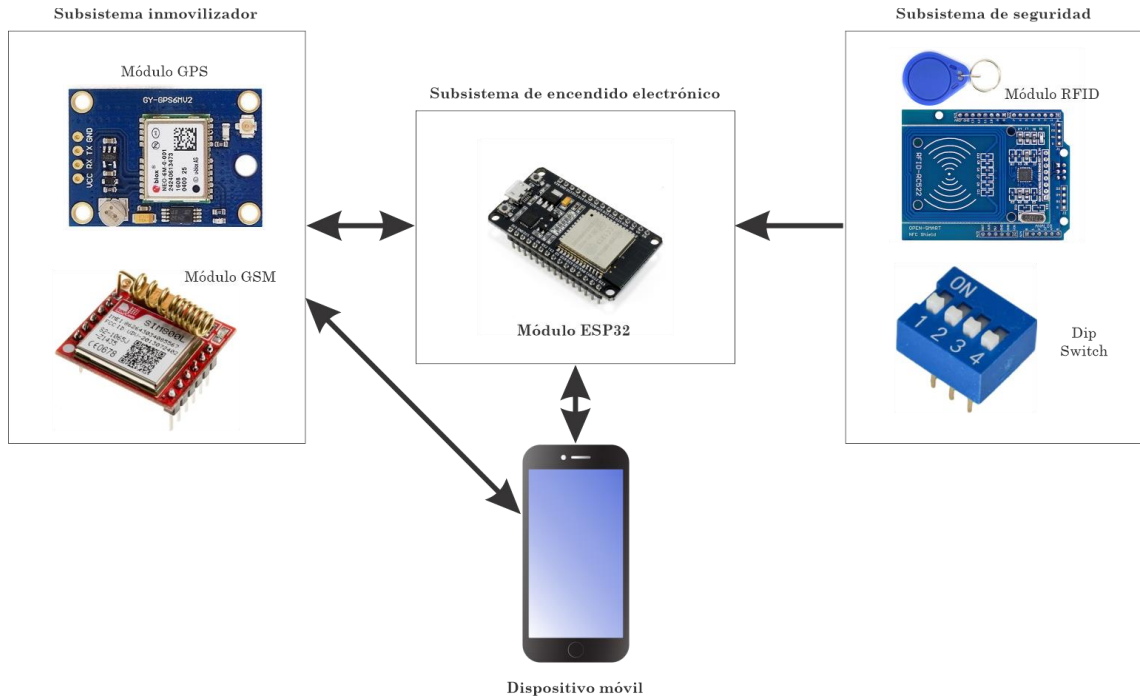


Figura 3.4. Diagrama general del SESA-NiV.

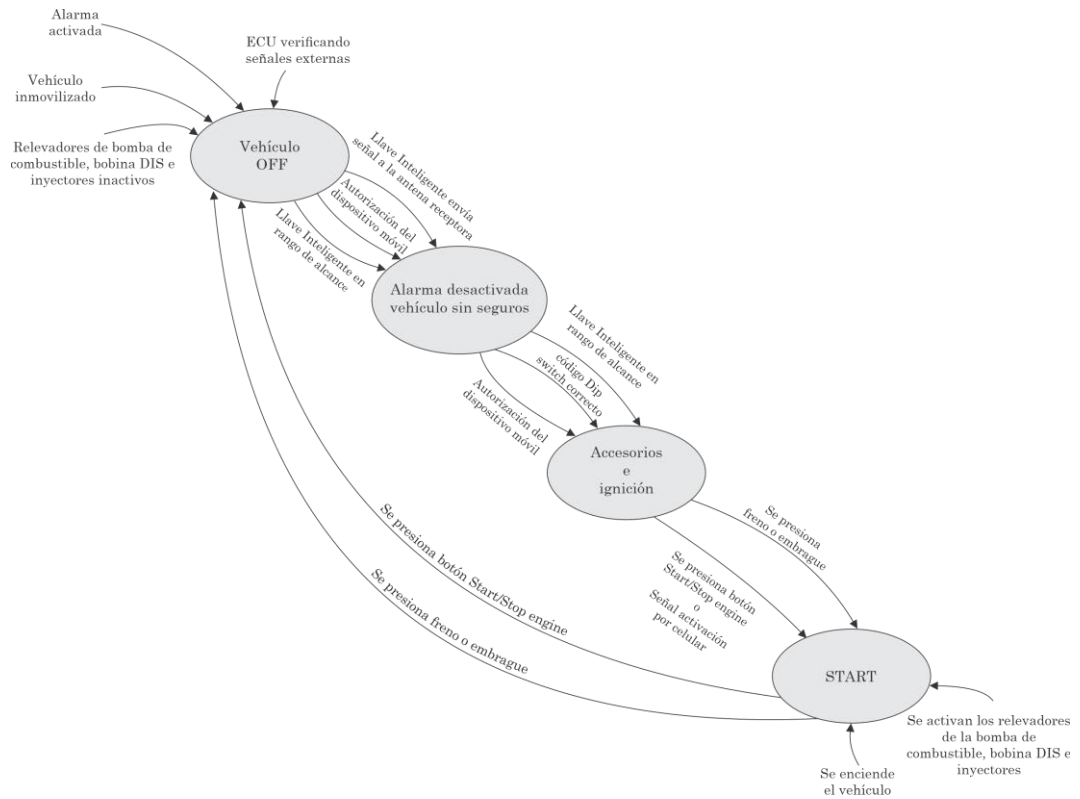


Figura 3.5. Diagrama de estados del SESA-NiV.

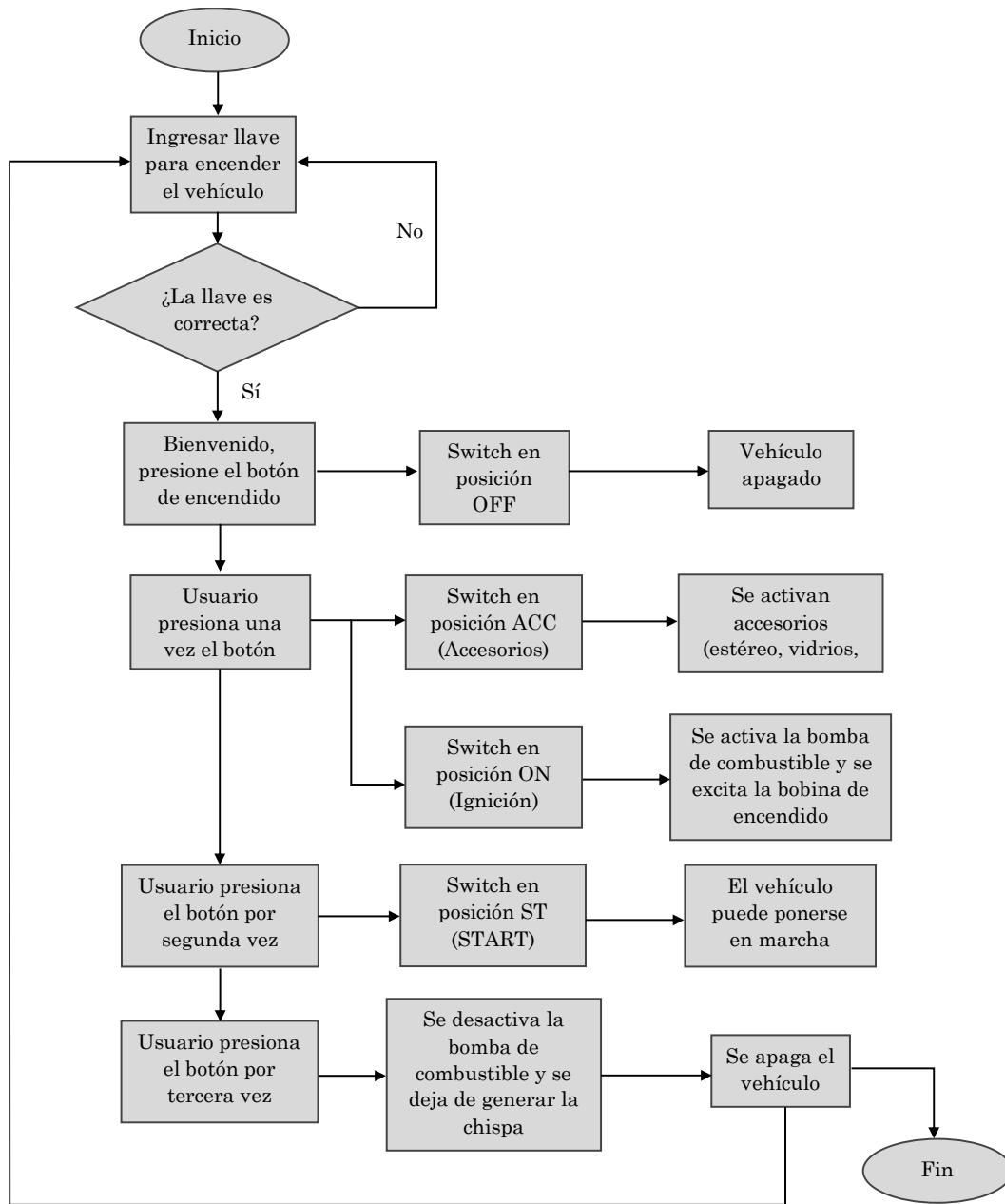


Figura 3.6. Diagrama de flujo del subsistema de encendido electrónico.

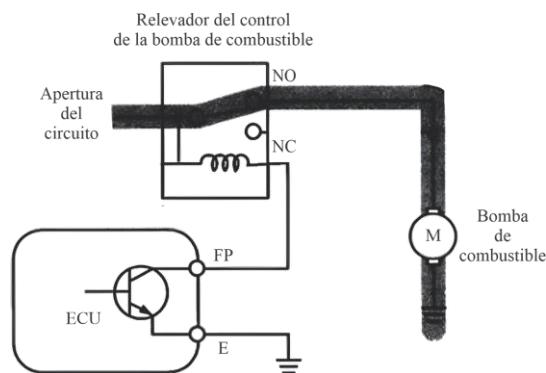


Figura 3.7. Circuito para el accionamiento de la bomba de combustible por un relevador.

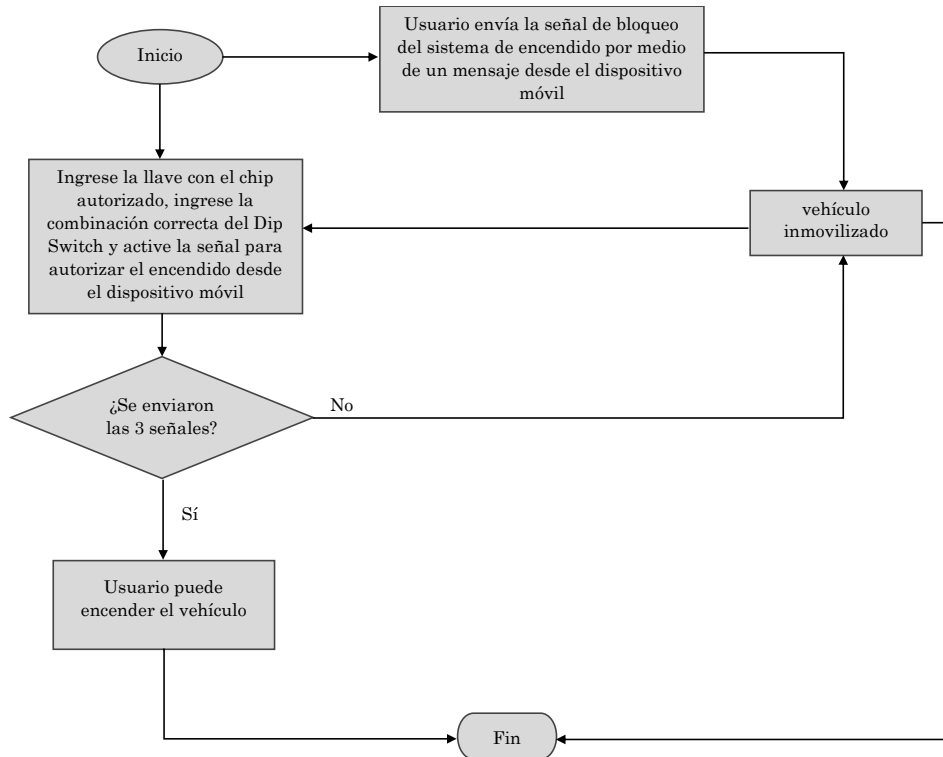


Figura 3.8. Diagrama de flujo del subsistema de seguridad automotriz.

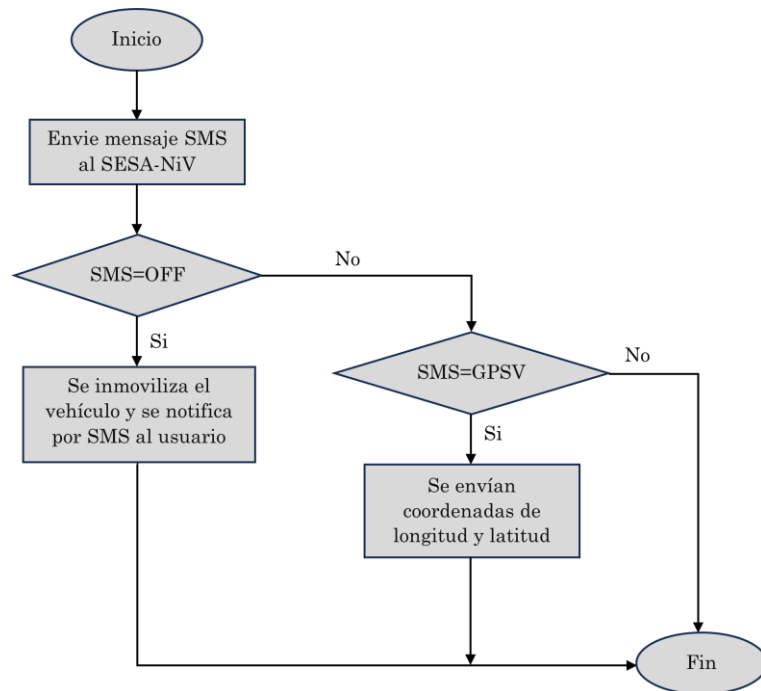


Figura 3.9. Diagrama de flujo del subsistema inmovilizador por SMS.

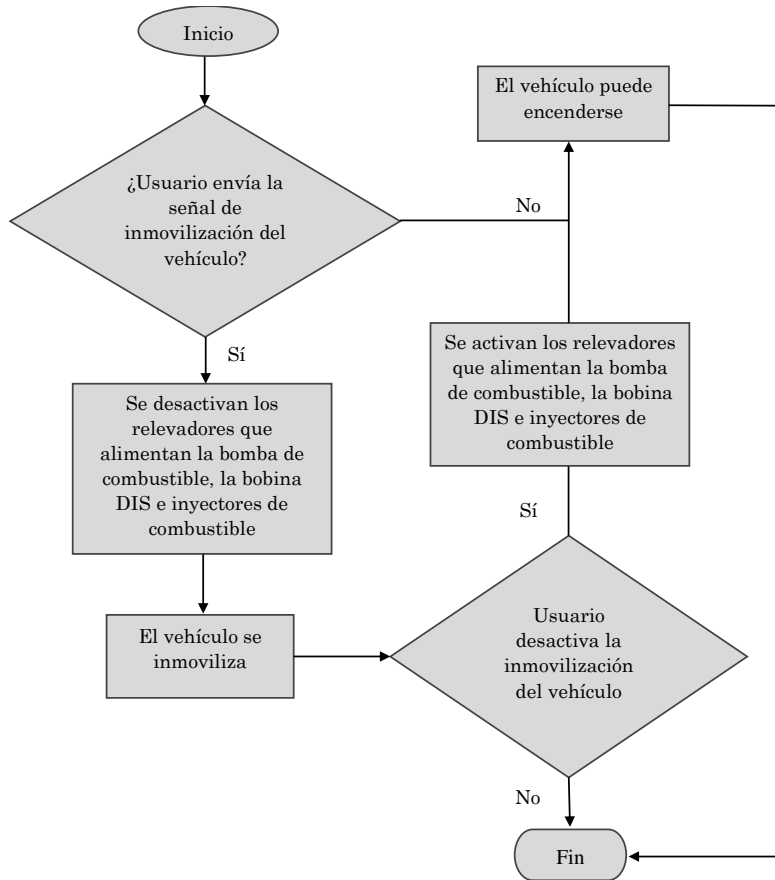


Figura 3.10. Diagrama de flujo del subsistema inmovilizador de relevadores.

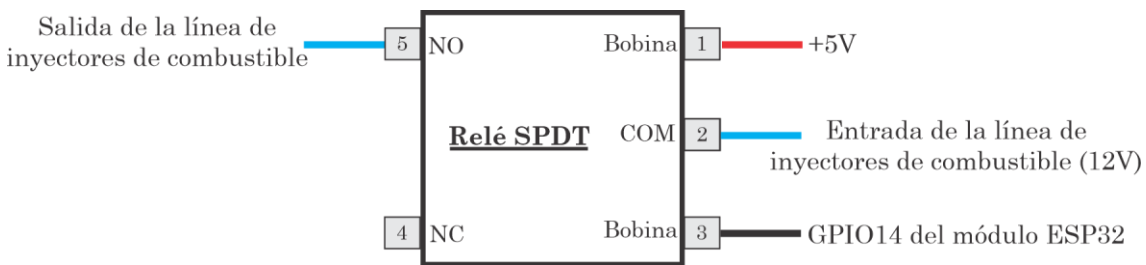


Figura 3.11. Relevador de activación de inyectores de combustible del SESA-NiV.

En este trabajo se desarrolló el SESA-NiV el cual implementa un sistema embebido conectado a diversos sensores que proporcionan ciertas señales digitales provenientes de la ECU (véase Subcapítulo 3.1). Este sistema permite, a través de una aplicación móvil o bien por medio del mismo vehículo, poner en marcha el vehículo una vez que se haya verificado el usuario. Dicha verificación se autentifica por el módulo ESP32.

3.2. División Hardware y Software

En esta fase se dividen las tareas hardware y software que conforman el SESA-NiV, partiendo de la división que lista la Tabla 3.1.

Tabla 3.1. División hardware y software del SESA-NiV.

Tareas SW	Tareas HW
Código del sistema de encendido electrónico	Sistema de alimentación
	Configuración de los sensores
Código del sistema de seguridad	Configuración de los actuadores
	Configuración del relevador
Código del sistema inmovilizador	Configuración de los módulos GPS-GSM

La Tabla 3.2 lista las herramientas utilizadas en el desarrollo del SESA-NiV.

Tabla 3.2. Herramientas para el desarrollo del SESA-NiV.

Herramientas HW	Descripción
Módulo ESP32	Funge como la unidad electrónica que simula la centralita del vehículo, además de ser el sistema intermediario para la comunicación máquina-usuario.
Módulo RFID RC522	Este módulo se implementa como una medida de seguridad. Incorpora un circuito integrado con un TAG único que habilita el encendido electrónico del vehículo.
Módulo SIM800L	Este módulo envía al ESP32 la señal de inmovilizar el vehículo a través de un mensaje de texto por medio del dispositivo móvil.
Relevadores	Dispositivos acoplados al sistema de seguridad que proporcionan la señal de activación o desactivación a la bomba de combustible, inyectores de combustible o excitación a la bobina primaria.
Regulador de voltaje	Tienen la función de regular el voltaje de entrada de 12 a 5V.
L7812CV	Su función es regular la variación de voltaje de entrada de la batería del vehículo y regularla a 12V fijos.
Pantalla Oled	Dispositivo que muestra el estado del sistema de seguridad.
Pulsador (<i>Push button</i>)	Dispositivo que permite dar paso a las distintas posiciones del interruptor de arranque, además de emular un embrague.
Buzzer	Dispositivo que emite una alarma cuando el sistema de seguridad sea vulnerado.
Módulo GPS	Módulo encargado de enviar la ubicación del vehículo a un dispositivo móvil cuando el usuario lo requiera o presione el botón de pánico.
Dip Switch	Contiene cuatro interruptores deslizables que se implementan como una medida de seguridad, funciona como una contraseña y en caso de validar la combinación correcta, envía una bandera de activación.
Herramientas SW	Descripción
Arduino	Entorno de desarrollo integrado (IDE) utilizado para desarrollar el software y la aplicación móvil.
MIT App Inventor	IDE en el cual se desarrolló la aplicación móvil para poder facilitar la interacción entre el usuario y el módulo ESP32.

EAGLE	Software donde se realizó el esquemático de la conexión de los distintos componentes electrónicos, además, para obtener el diseño de la PCB.
SolidWorks	Software utilizado para el diseño de la carcasa del SESA-NiV.

3.3. Iteración e Implementación

Con base a las herramientas SH y HW mencionadas en apartados anteriores, se mencionó el funcionamiento de cada subsistema que conforma el SESA-NiV; cada subsistema cumple una función en específico, y al ser integradas en un sólo sistema, se tendrá un sistema embebido funcional y eficiente.

El diseño HW contempla los siguientes módulos y herramientas:

- Módulos integrados en el microcontrolador ESP32: Módulo RFID RC522, Módulo GPS Neo-6m-7m, Módulo SIM800L y módulo ESP32-WROOM-32.
- Relevadores para la bomba de combustible, la bobina DIS y los inyectores de combustible.
- Pantalla OLED.
- Fuente de alimentación de 12VCD.
- Regulador de voltaje LM7812.
- dispositivo móvil con sistema operativo Android 10.
- Dip Switch.

El diseño del SW contempla lo siguiente:

- Software para comunicar el módulo ESP32 con el dispositivo móvil desarrollado en MIP App Inventor.
- Software de automatización de diseño electrónico y PCB, en EAGLE.
- Entorno de desarrollo integrado (IDE) de Arduino.
- Software SolidWorks en donde se modelará la carcasa que contendrá a la placa PCB.

3.4. Diseño Detallado HW y SW

Una vez seleccionados los elementos que conforman el presente trabajo, se estableció a detalle cada espacio y diseño de cada elemento. Haciendo mención que para el óptimo funcionamiento del sistema principal se debe integrar cada subsistema HW y SW.

Se empleó un módulo ESP32 para implementar el SESA-NiV; el cual se comunica con un módulo RFID para validar al usuario. Así mismo, y para garantizar una mayor seguridad, el SESA-NiV puede activarse desde una aplicación móvil mediante un mensaje SMS para desactivar los relés que alimentan a la bomba de combustible y a la bobina DIS (véase Figura 3.12), y con ello inmovilizar el vehículo. Cabe mencionar que el sistema cuenta con un módulo GPS, el cual envía las coordenadas de localización del vehículo cuando el usuario lo solicite en caso de robo.

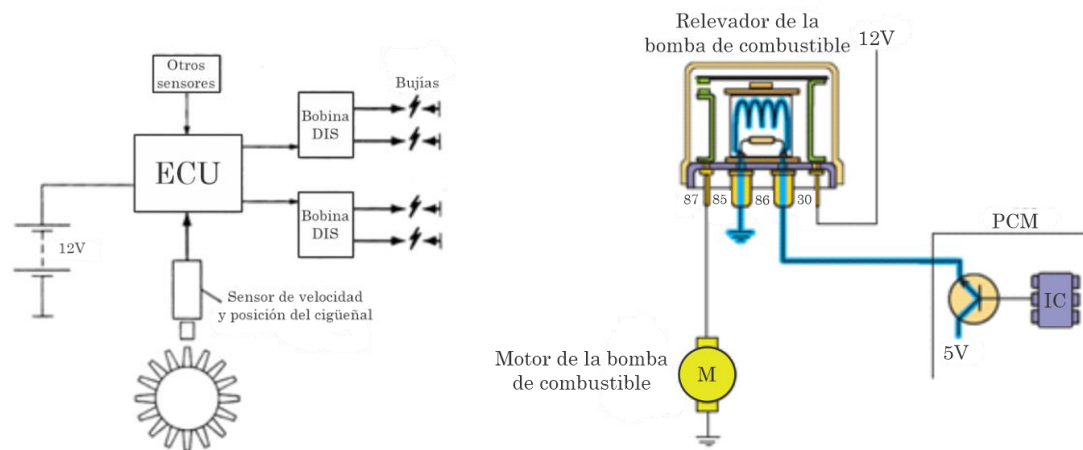


Figura 3.12. Funcionamiento del sistema de encendido de la bobina DIS y del módulo de activación de la bomba de combustible (Denton, 2012) (Halderman, 2012).

El funcionamiento del SESA-NiV se codificó en la plataforma de programación Arduino.

3.4.1. Diseño HW

Para seleccionar el microcontrolador a utilizar se compararon diversas soluciones disponibles en el mercado como muestra la Tabla 3.3 (Raspberry Pi, 2022), (Espressif Systems, 2022), (Arduino, 2022).

Tabla 3.3. Comparación de sistemas MCU.

MCU	ESP8266	ESP32	Arduino Nano	Arduino UNO	Arduino Mega	Raspberry Pi Pico
Voltaje de funcionamiento	3.3 V	3.3 V	5V	5V	5V	3.3V – 5V
Consumo de corriente normal	15 μ A – 400 mA	20 mA - 240 mA	19 mA – 180 mA	45 mA – 80 mA	50 mA – 200 mA	90 mA – 95 mA
Consumo de corriente en modo <i>Deep - Sleep</i>	0.5 μ A	5 μ A	23 μ A	35 μ A	500 μ A	1.3 mA
Corriente de CC para 3.3V	-	40 mA	50 mA	150 mA	150 mA	-
Terminales digitales de E/S	11 -13	36	14	14	54	23
Terminales PWM	11 -13	36	6	6	15	23
Terminales analógicas de entrada	1	15	8	6	16	3
Memoria flash	4 MB	4 MB	32 KB	32 KB	256 KB	2 MB
Velocidad de reloj	52 MHz	80 MHz – 160 MHz	16 MHz	16 MHz	16 MHz	133 MHz
Wi-Fi	Sí	Sí	No	No	No	No
Bluetooth	No	Sí	No	No	No	No
Longitud	48 mm	52 mm	45 mm	69 mm	102 mm	51 mm
Ancho	26 mm	31 mm	18 mm	53 mm	53 mm	21 mm
Precio (MX)	\$120	\$180	\$440	\$480	\$840	\$175

En el presente trabajo de tesis se empleó el módulo ESP32, siendo el encargado de procesar las señales digitales y ejecutar las acciones con base en la lógica de programación que establecen las especificaciones del sistema (véase Subcapítulo 3.1).

3.4.1.1. Módulo ESP32

El módulo ESP32 es la evolución del ESP8266, mejorando sus capacidades de comunicación y procesamiento computacional (véase Figura 3.13). Basa su funcionamiento en un microcontrolador que permite utilizar diversos protocolos de comunicación inalámbrica como Wi-Fi, Bluetooth y BLE (*Bluetooth Low Energy*). En cuanto a procesamiento, cuenta con un CPU de 32 bits con dos núcleos que opera hasta 240 Mhz. Incluye internamente una gran cantidad de periféricos para la conexión con sensores táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD y permite comunicación Ethernet, SPI de alta velocidad, UART, I2S e I2C.

Para el intercambio de información entre la aplicación en el dispositivo móvil y el módulo ESP32 se utilizó el módulo WROOM-32. Su funcionamiento

en el presente trabajo de tesis fue crucial debido a que integra la conectividad Wi-Fi y Bluetooth. Sus principales especificaciones técnicas son las siguientes:

- Microcontrolador ESP32-D0WDQ6 con núcleos de CPU Xtensa de doble núcleo y arquitectura Tensilica.
- Frecuencia de reloj de hasta 240 MHz.
- Memoria RAM de 520 KB tipo SRAM.
- Memoria Flash de 4 MB para almacenamiento de programas y datos.
- Conectividad:
 - Wi-Fi: 802.11 b/g/n (2.4 GHz) con capacidades de estación, acceso punto y modo mixto.
 - Bluetooth 4.2 y BLE.
 - Antenas internas para Wi-Fi y Bluetooth.
- Interfaces:
 - GPIO (terminales de entrada/salida de propósito general) para la interacción con periféricos externos.
 - Interfaces UART, SPI, I2C, I2S, PWM y ADC para la comunicación y el control de dispositivos externos.
 - Interfaz Ethernet.
 - Periféricos integrados:
 - Sensores táctiles capacitivos.
 - Sensor de temperatura interno.
 - Convertidor analógico a digital (ADC) de 12 bits.
- Alimentación:
 - Rango de voltaje de alimentación: 2.2V - 3.6V.
 - Consumo de energía ajustable para un equilibrio entre rendimiento y ahorro de energía.
- Soporte para cifrado WPA/WPA2, así como para encriptación de datos y autenticación.
- Sistema operativo en tiempo real (RTOS) FreeRTOS de Espressif.

El módulo ESP32 ofrece diversas ventajas, ya que cuenta con un mayor número de terminales, permite operar con distintos modos de consumo de energía, cuenta con un módulo Bluetooth y tiene un precio accesible. El módulo ESP32 soporta los siguientes modos de consumo de energía (véase Tabla 3.4) (Espressif Systems, 2022):

- *Active mode*: La señal de radiofrecuencia del chip se encuentra activa, el microcontrolador está listo para ejecutar tareas y responder a eventos.
- *Modem-sleep*: La CPU se encuentra activa y se puede configurar el reloj. Tanto Wi-Fi como Bluetooth se encuentran inhabilitados.
- *Light-sleep*: La CPU se encuentra en pausa, la memoria y los periféricos RTC, así como el coprocesador ULP funcionan normalmente. Cualquier evento de activación (MAC, host, temporizador RTC o interrupciones externas), despertará o activará al ESP32.
- *Deep-sleep*: Sólo la memoria y los periféricos RTC están encendidos. Los datos de las conexiones Wi-Fi y bluetooth se almacenan en la memoria RTC.
- *Hibernation*: El oscilador de 8 MHz y el coprocesador ULP están desactivados. El temporizador RTC en el reloj lento y ciertos GPIO RTC están activos.

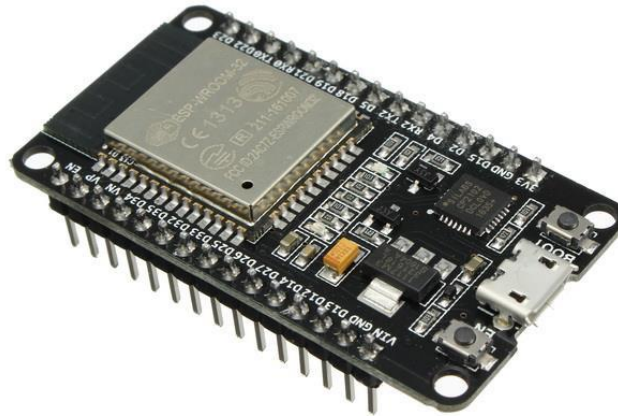


Figura 3.13. Módulo ESP32.

Tabla 3.4. Modos de consumo de energía del módulo ESP32.

Modo	Consumo de energía
<i>Active mode</i>	180 mA – 240 mA
<i>Mode - sleep</i>	20mA – 68 mA
<i>Light - sleep</i>	0.8 mA
<i>Deep - Sleep</i>	10 μ A - 150 μ A
<i>Hibernation</i>	5 μ A
<i>Power off</i>	1 μ A

En el Anexo A se describe la instalación del firmware en el módulo ESP32 para su codificación empleando el IDE Arduino.

3.4.1.1.1. Integración de los componentes en el módulo ESP32

Para la integración de cada módulo y los componentes que integran al SESA-NiV, se declaran las GPIO en el software del programa, teniendo en cuenta sus hojas de datos y diagramas de conexión. Se asignaron los distintos periféricos del microcontrolador ESP32 para el envío de las señales como muestran la Tabla 3.5 y la Figura 3.14.

Tabla 3.5. Asignación de las terminales del módulo ESP32.

Terminales ESP32			
Terminal	Asignación	Terminal	Asignación
EN	No conectada	GPIO23	RFID – MOSI
VP	No conectada	GPIO22	OLED – SCL
VN	No conectada	TX0	No conectada
GPIO34	Dip Switch – Interruptor 4	RX0	No conectada
GPIO35	Dip Switch – Interruptor 3	GPIO21	OLED – SDA
GPIO32	Dip Switch – Interruptor 2	GPIO19	RFID – MISO
GPIO33	Dip Switch – Interruptor 1	GPIO18	RFID – SCK
GPIO25	Push Bottom – Start/Stop	GPIO5	RFID – SDA
GPIO26	Push Bottom – Freno	GPIO17	GPS – RX
GPIO27	Buzzer – Alarma	GPIO16	GPS – TX
GPIO14	Relevador 3 – Inyectores de combustible	GPIO4	SIM800L – TXD
GPIO12	Relevador 2 – Bobina DIS	GPIO2	SIM800L – RXD
GPIO13	Relevador 1 – Bomba de combustible	GPIO15	RFID – RST
GND	GND	GND	GND
VIN	5VDC	3V3	3.3 VCD

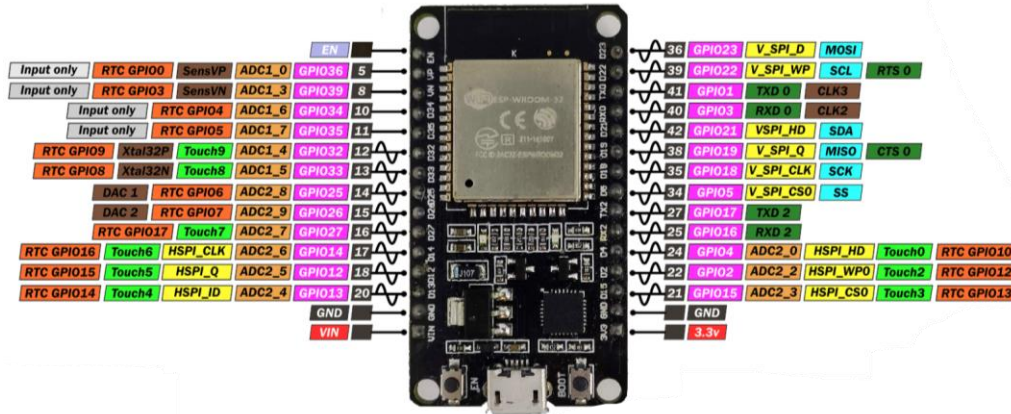


Figura 3.14. GPIO del módulo ESP32.

3.4.1.2. Pantalla OLED

Como dispositivo de visualización de información entre el SESA-NiV y el usuario se empleó una pantalla OLED (*Organic Light-Emitting Diode*), ya que permite mostrar una mayor cantidad de información que un LCD, además, reduce en gran medida el espacio para su colocación en la PCB y requiere un menor número de terminales para su conexión.

Las pantallas OLED están compuestas por láminas de materiales orgánicos que al aplicarles energía emiten luz. La pantalla OLED que se utilizó en este trabajo de tesis tiene una resolución de 128x64 píxeles, utiliza el controlador SSD1306 para comunicarse con el microcontrolador mediante SPI e I2C. En este caso, se utiliza el protocolo de comunicaciones I2C debido a que sólo utiliza dos terminales del módulo ESP32. Además, integra un regulador de voltaje, el cual permite una entrada de voltaje de entre 1.8V a 6V, manteniendo un voltaje de 3.3V para su operación.

Para la visualización de los datos, el controlador SSD1306 contiene una memoria RAM gráfica denominada GDDRAM (*Graphic Display Data RAM*), la cual ocupa 1 KB (1024 bytes) que se distribuye por la pantalla en una matriz de 64 filas (páginas) y 128 columnas (segmentos), y cada bit representa un píxel. En total, la pantalla OLED de 128x64 contiene un total de 8 páginas (véase Figura 3.15); cabe mencionar que cada página contiene 128 segmentos.

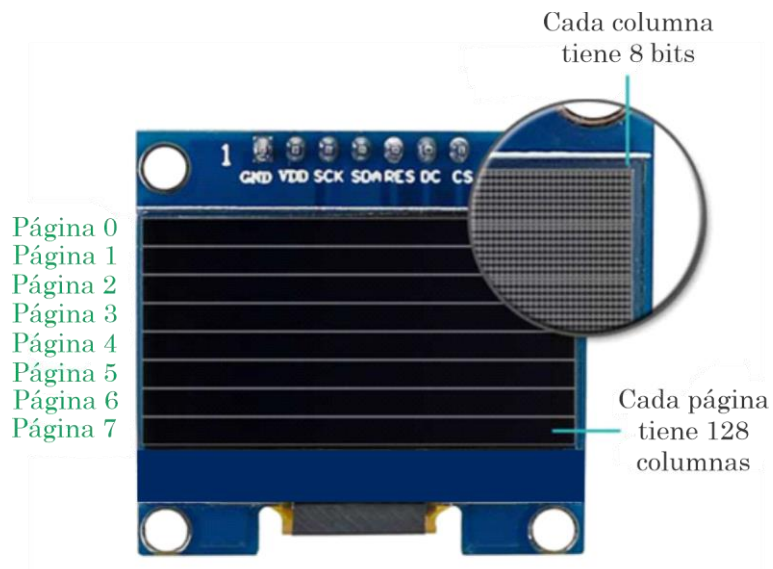


Figura 3.15. Páginas y columnas de la pantalla 128x64 OLED.

La Figura 3.16 muestra las terminales para la conexión I2C de la pantalla OLED:

- GND: Tierra.
- VCC: Alimentación, puede variar entre 1.8 a 6V.
- SCL: Señal de reloj de la interfaz I2C.
- SDA: Señal de datos de la interfaz I2C.

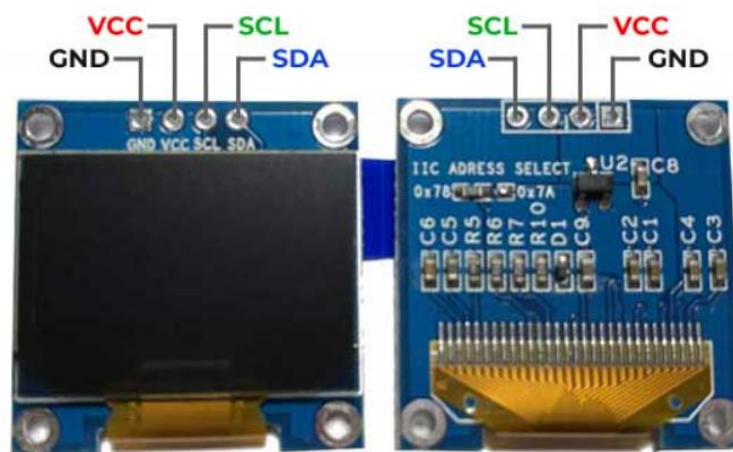


Figura 3.16. Terminales de la pantalla 128x64 OLED.

Para la configuración de la pantalla OLED se asignaron las GPIO (GPIO21 y GPIO22) del módulo ESP32 de la siguiente manera (véase Figura 3.17):

- GPIO21 – SDA.
- GPIO22 – SCL.
- 3.3V ESP32 – VCC.
- GND ESP32 – GND.

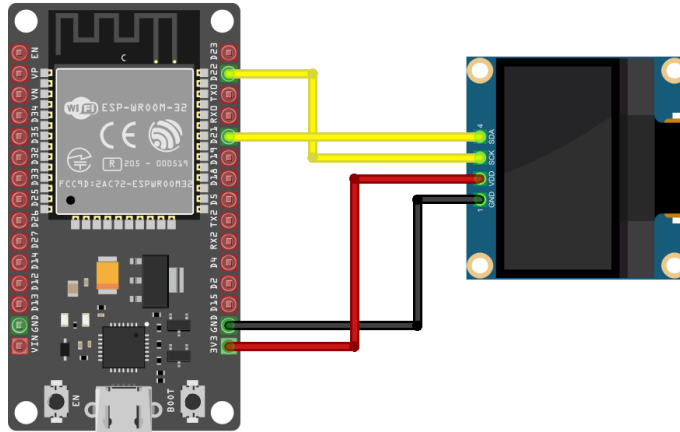


Figura 3.17. Esquema de conexión de los GPIO del ESP32 con la Pantalla 128x64 OLED.

3.4.1.3. Módulo RFID RC522

Cómo elemento de identificación para poder autorizar el encendido del vehículo se seleccionó el módulo RFID RC522; éste consta de dos dispositivos, un módulo lector RFID (*Radio Frequency Identification*) y un microchip, el cual viene integrado en una tarjeta o etiqueta.

La comunicación principal de este sistema es a través de ondas de radiofrecuencia a 13.56 MHz por medio del lector que actúa como transmisor y receptor (véase Figura 3.18). Una vez que se acerca el microchip al lector, éste se energiza por el principio de la ley de Faraday de inducción electromagnética. Es en este momento en que el embobinado interno se excita y activa el microchip integrado, enviando una señal modulada que es capturada por el lector y enviada al microcontrolador.

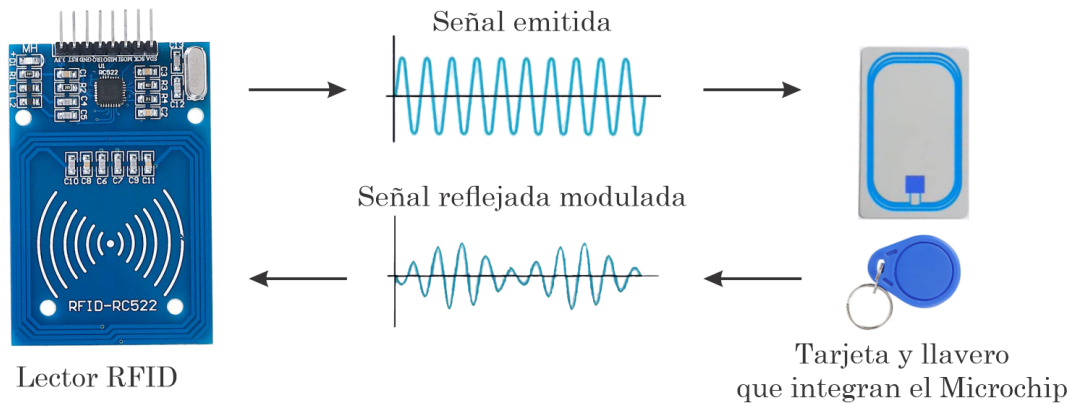


Figura 3.18. Transmisión y recepción de datos del módulo RFID RC522.

Dependiendo de la frecuencia de operación, existen tres tipos de sistemas RFID:

- **Baja frecuencia:** Tienen una longitud de onda larga, sin embargo, su rango de alcance de detección de la etiqueta RFID se encuentra limitado a unos 10 centímetros como máximo. La frecuencia de trabajo oscila entre 125-134 KHz.
- **Alta frecuencia:** Presenta un rango de mayor frecuencia y capacidad de memoria, utilizan protocolos globales como NFC (*Near-Field Communications*), su rango de alcance es de unos 30 cm y su frecuencia de trabajo es de 13.56 MHz.
- **Ultra Frecuencia:** Además de tener un mayor alcance, son capaces de procesar datos de forma más rápida, pero se limitan en su longitud de onda. Su rango de alcance es superior a los 100 metros y su frecuencia de trabajo oscila entre 433 MHz y 960 MHz.

El lector RC522, aunque utiliza la alta frecuencia, tiene un corto alcance, su rango máximo es de unos 5 cm.

Las especificaciones técnicas del módulo RFID RC522 son las siguientes:

- Rango de frecuencias en la banda ISM de 13.56 MHz.
- Interfaz SPI/I2C/UART.
- Voltaje de operación entre 2.5 - 3.3V.
- Corriente máxima de funcionamiento de 13 - 26 mA.

- Corriente mínima de funcionamiento de 10 uA.
- Rango de lectura de hasta 5 cm.

El lector RFID RC522 contiene ocho terminales (véase Figura 3.19):

- VCC: Suministra energía al módulo en un rango de 2.5 a 3.3 Volts.
- RST: Entrada para reinicio y apagado del módulo.
- GND: Tierra.
- IRQ: Interrupción que alerta al microcontrolador cuando una existe una etiqueta RFID cerca.
- MISO/SLC/TX: Realiza tres funciones, actúa como Maestro/Esclavo cuando la interfaz SPI está habilitada, como reloj cuando la interfaz I2C está activada y como salida serie cuando la UART está habilitada.
- MOSI: Entrada en la interfaz SPI.
- SCK: Señal de reloj de la interfaz SPI.
- SS/SDA/RX: Esta terminal actúa como entrada de señal cuando la interfaz SPI está habilitada, cuando la interfaz I2C está activada actúa como entrada de datos y como entrada de datos serie cuando la UART está habilitada.

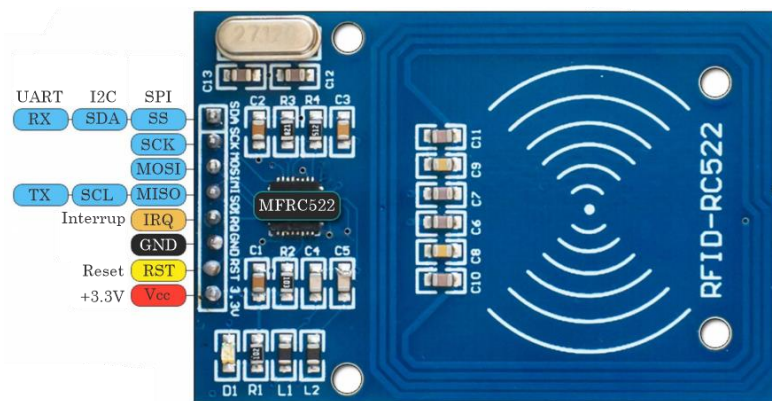


Figura 3.19. Terminales del módulo RFID RC522.

La tarjeta RFID RC522 contiene una memoria de 1KB, la cual está organizada en 16 sectores. Cada sector se divide en cuatro bloques y en cada bloque se pueden almacenar 16 bytes de datos (véase Figura 3.20):

$$16 \text{ sectores} \times 4 \text{ bloques} \times 16 \text{ bytes} = 1024 \text{ bytes} = 1 \text{ kb}$$

Los bloques de 16 bytes se dividen en dos claves, Key A y Key B (véase Figura 3.21), cada uno ocupa 6 bytes, seguido de 3 bytes que establecen las condiciones y permisos de acceso al sector (*Access Conditions*) y un byte restante (U) que no se ocupa.

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
13	55	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]

Figura 3.20. Sectores de la memoria del módulo RFID RC522.

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	62	00	00	Key A	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
13	55	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	52	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
12	51	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
11	47	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	46	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	45	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]

Figura 3.21. Bytes que se almacenan en una línea de un bloque del módulo RFID RC522.

En el presente trabajo se emplea el módulo RFID RC522 como llavero de acceso al SESA-NiV. Una vez que se acerca la tarjeta al lector RFID, se envían

los datos al microcontrolador, el cual habilita o deshabilita los relés de la bomba de gasolina y de la bobina DIS. Si el usuario acerca la tarjeta con el chip adecuado y, además, introduce la clave correcta en el dispositivo móvil, podrá hacer uso de la unidad, de lo contrario el vehículo estará inhabilitado.

Se asignaron las terminales GPIO (GPIO5, GPIO15, GPIO18, GPIO19 y GPIO23) del microcontrolador ESP32 para conectar el módulo RFID RC522 de la siguiente manera (véase Figura 3.22):

- GPIO15 – RST.
- GPIO5 – SDA.
- GPIO18 – SCK.
- GPIO19 – MISO.
- GPIO23 – MOSI.
- 3.3V ESP32 – 3.3V.
- GND ESP32 – GND.

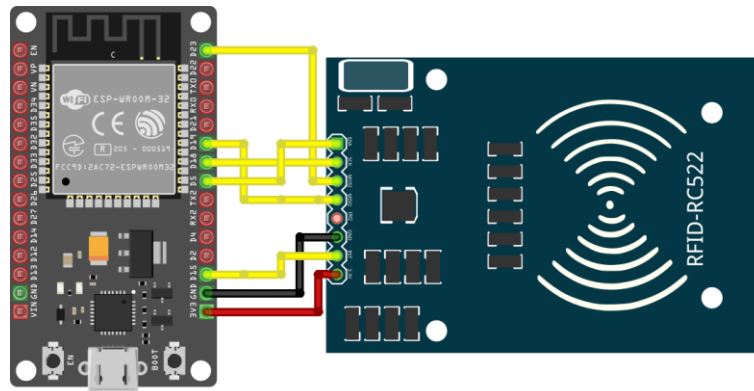


Figura 3.22. Esquema de conexión de GPIO del ESP32 con el módulo RFID RC522.

3.4.1.4. Módulo SIM800L

Debido a diversas problemáticas que se pudieran presentar a la hora en que el usuario se encuentre lejos del rango de la comunicación Bluetooth, el cual tiene un rango de distancia máxima de 10 metros, se emplea el módulo SIM800L para que el usuario tenga un mayor rango de alcance en la comunicación con el SESA-NiV (SIMCom, 2015).

El módulo SIM800L es versátil y ampliamente utilizado para comunicaciones GSM/GPRS, ya que ofrece una gama de características adecuadas para diversas aplicaciones de sistemas integrados e IoT (*Internet of Things*).

El funcionamiento de este sistema se basa en el intercambio de señales de radio analógicas (señales electromagnéticas), las cuales se envían de un transmisor a un receptor. El usuario por medio del dispositivo móvil envía un mensaje de texto, esta señal será captada por una antena de red encargada de dirigir dicha señal a la antena receptora, en este caso a la antena del módulo SIM800L, una vez que capta la señal, la decodifica en una señal digital y la envía al módulo ESP32; posteriormente una vez que se realice la acción se envía una señal de regreso, indicando por mensaje SMS al dispositivo móvil que se ha inmovilizado la unidad y se envían las coordenadas del vehículo (véase Figura 3.23).

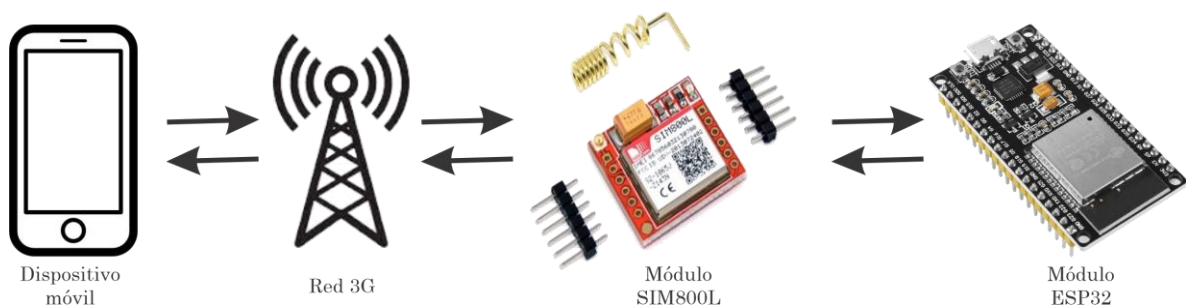


Figura 3.23. Comunicación del SIM800L con el módulo ESP32.

Las principales características del módulo SIM800L son las siguientes:

- Admite comunicaciones GSM (2G) y GPRS (2.5G). Puede operar en cuatro bandas de frecuencia GSM (850 MHz, 900 MHz, 1800 MHz y 1900 MHz) siendo compatible con la mayoría de las redes GSM.
- Proporciona transferencia de datos a través de GPRS (Servicio general de paquetes de radio) con velocidades de datos de hasta 85,6 kbps para carga y 85,6 kbps para descarga. Soporta los protocolos TCP/IP y UDP, que le permiten enviar y recibir datos a través de Internet.

- Permite enviar y recibir mensajes SMS, así como realizar y recibir llamadas de voz, todo ello a través de ordenes AT dedicadas.
- Dispone de varias terminales de GPIO que se pueden usar para interactuar con sensores externos, interruptores u otros dispositivos. Dichas terminales permiten ampliar la funcionalidad del módulo agregando funciones personalizadas.
- Requiere una fuente de alimentación regulada de 3.4 a 4.4 V. Por lo general, consume alrededor de un 1A de corriente durante las operaciones pico, por lo que es importante proporcionar una fuente de alimentación adecuada para garantizar un rendimiento estable.

El módulo contiene un total de doce terminales; la Figura 3.24 muestra la distribución de terminales GPIO del módulo SIM800L:

- IPX ANT: Dedicada a conectar la antena GSM/GPRS IPX/UFL.
- NET: Permite añadir una antena helicoidal para recepción de datos a frecuencias entre 900 a 923 MHz.
- VCC: Terminal para suministro de energía de 3.4 a 4.4V.
- RST: Restablecimiento (*reset*).
- RXD: Receptor de datos para la comunicación serie.
- TXD: Transmisor de datos para la comunicación serie.
- GND: Tierra.
- RING: Interrupción de salida del módulo.
- DTR: Activa/desactiva el modo de espera del módulo.
- MIC: Terminal para establecer la conexión con un micrófono.
- SPK: Interfaz de altavoz.

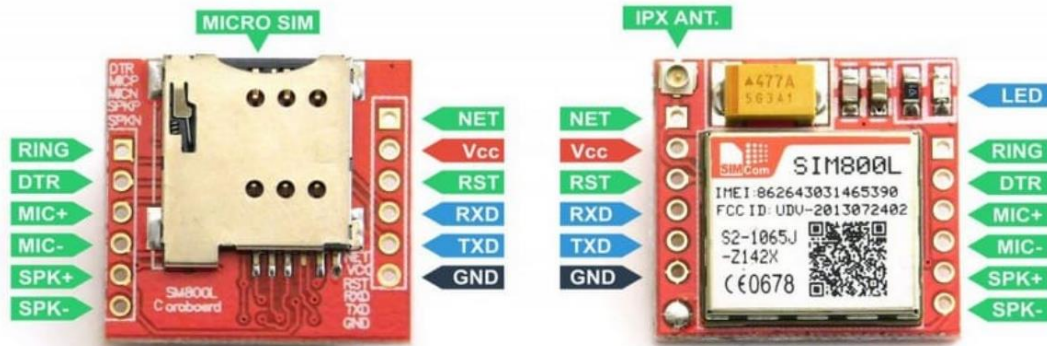


Figura 3.24. Terminales del módulo SIM800L.

Para la configuración del módulo GSM (SIM800L) se asignaron las GPIO (GPIO2 y GPIO4) del módulo ESP32 de la siguiente manera (véase Figura 3.25):

- GPIO2 (TXD) – RXD.
- GPIO4 (RXD) –TXD.
- 3.7V (batería Li-Ion) – VCC.
- GND ESP32 – GND.

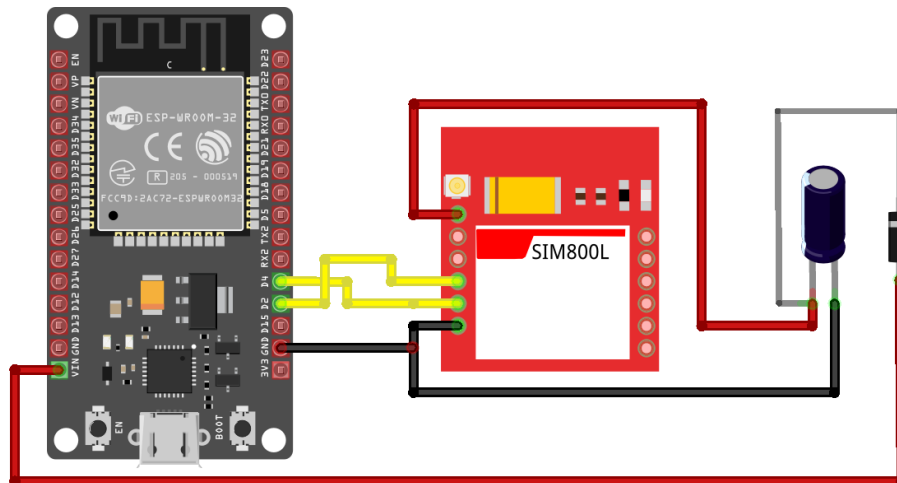


Figura 3.25. Esquema de conexión de GPIO del ESP32 con el módulo SIM800L.

Cabe mencionar que este módulo tiene un rango de funcionamiento entre 3.4V a 4.4V y debido a que los voltajes otorgados por el microcontrolador son de 3.3V y 5V respectivamente, se optó por conectar un diodo y un capacitor para poder atenuar el voltaje y tener un valor adecuado para su correcto funcionamiento.

Se definieron las conexiones de las terminales del receptor (GPIO 4) y del transmisor (GPIO2) para la comunicación entre el módulo SIM800L y el módulo ESP32, así como la velocidad de transferencia en datos en 9,600 baudios. Además, se definió la comunicación RX y TX en el puerto UART1 del ESP32 y el actuador a ser controlado.

3.4.1.5. Módulo GPS Neo-6m-7m

Con el fin de brindar una mayor seguridad al vehículo, se empleó un sistema global de posicionamiento (GPS, *Global Position System*), particularmente el módulo GPS Neo-6m-7m (u-blox, 2014).

El funcionamiento de este módulo se basa en el intercambio de señales electromagnéticas con 4 de los 24 satélites que orbitan la tierra. Una vez que la antena del módulo GPS intercambia señales electromagnéticas con un satélite, calcula la distancia entre el dispositivo GPS y el satélite correspondiente, sin embargo, para tener una mayor precisión de la geolocalización se hace una triangulación con 3 o 4 satélites (véase Figura 3.26), obteniendo las coordenadas de latitud y de longitud del dispositivo.

Cabe mencionar que, si el sistema está en constante movimiento, los satélites realizan varias triangulaciones en cuestión de segundos para obtener la posición del dispositivo GPS en tiempo real con pequeños retrasos en el tiempo.

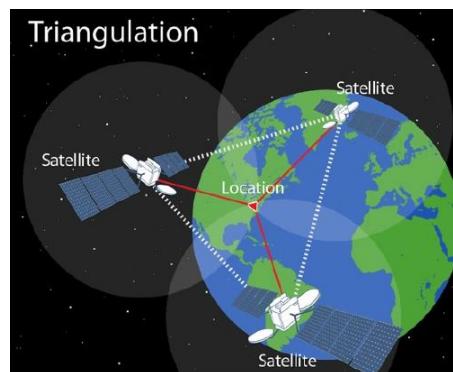


Figura 3.26. Triangulación GPS para calcular las coordenadas del dispositivo GPS.

Las características más importantes del módulo GPS Neo-6m-7m son:

- Voltaje de consumo: 3.3 a 5V.

- Consumo de corriente en búsqueda: 67 mA.
- Velocidad de comunicación serial (*baud rate*): 9,600 baudios.
- Memoria EEPROM.
- Sistema de coordenadas: WGS-84.
- Máxima Altura medible: 18,000.
- Máxima velocidad: 515 m/s.
- Exactitud: 1 microsegundo.
- Frecuencia receptora: L1 1575.42 MHz.
- Tiempo de inicio primera vez: 38 segundos en promedio con buena señal.
- Tiempo de inicio: 35s en promedio.

El GPS Neo-6m-7m cuenta con cuatro terminales (véase Figura 3.27):

- VCC: Suministra la energía al módulo, de 3.3 a 5V.
- RX: Receptor del módulo GPS.
- TX: Transmisor del módulo GPS.
- GND: Tierra.



Figura 3.27. Módulo GPS Neo-6m-7m.

La Figura 3.28 muestra las partes que conforman al módulo GPS, y son las siguientes:

- NEO 6M 7M: Encargado de la obtención de datos a través de canales GPS de radiofrecuencia.

- Batería recargable: Proporciona energía para mantener los datos almacenados en el módulo.
- Memoria EEPROM: Se encarga de almacenar los últimos datos de posicionamiento y configuración del módulo.
- Regulador de voltaje LDO: Eleva o disminuye el voltaje de entrada a 3.3V fijos.
- Conector U.FL: Conecta la antena externa para el correcto funcionamiento del módulo.

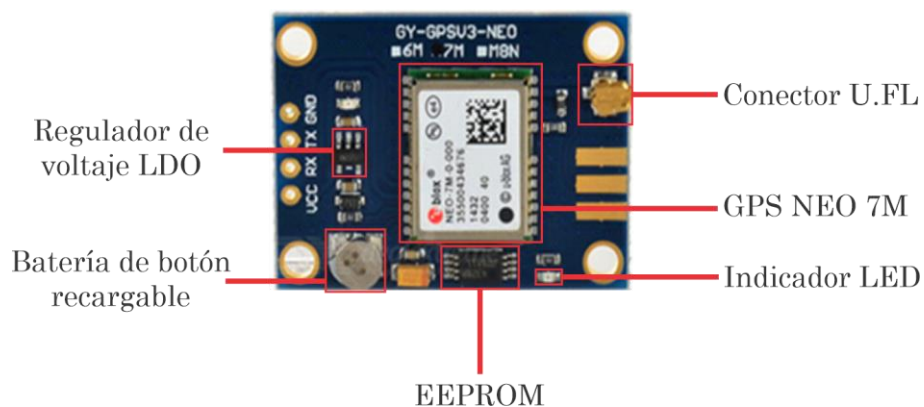


Figura 3.28. Terminales del módulo GPS Neo-6m-7m.

Cuando el usuario selecciona la opción Localizar vehículo por GPS en la aplicación realizada en MIT App Inventor, el teléfono móvil envía un mensaje de texto para comunicarse con el módulo SIM800L que incorpora el SESA-NiV. Al recibir el mensaje, el módulo ESP32 envía una solicitud de localización al módulo GPS Neo-6m-7m, que, una vez que obtiene las coordenadas de localización, las envía mediante un mensaje SMS a la aplicación del teléfono móvil. Con ello se conoce la longitud y la latitud de localización del vehículo con un pequeño retraso (véase Figura 3.29).

Para la configuración del módulo GPS se establecieron los GPIO (GPIO16 y GPIO17) del módulo ESP32, dichas terminales se conectaron al módulo GPS Neo-6m-7m de la siguiente forma (véase Figura 3.30):

- GPIO16 – TX.
- GPIO17 – RX.

- 3.3V ESP32 – VCC.
- GND ESP32 – GND.

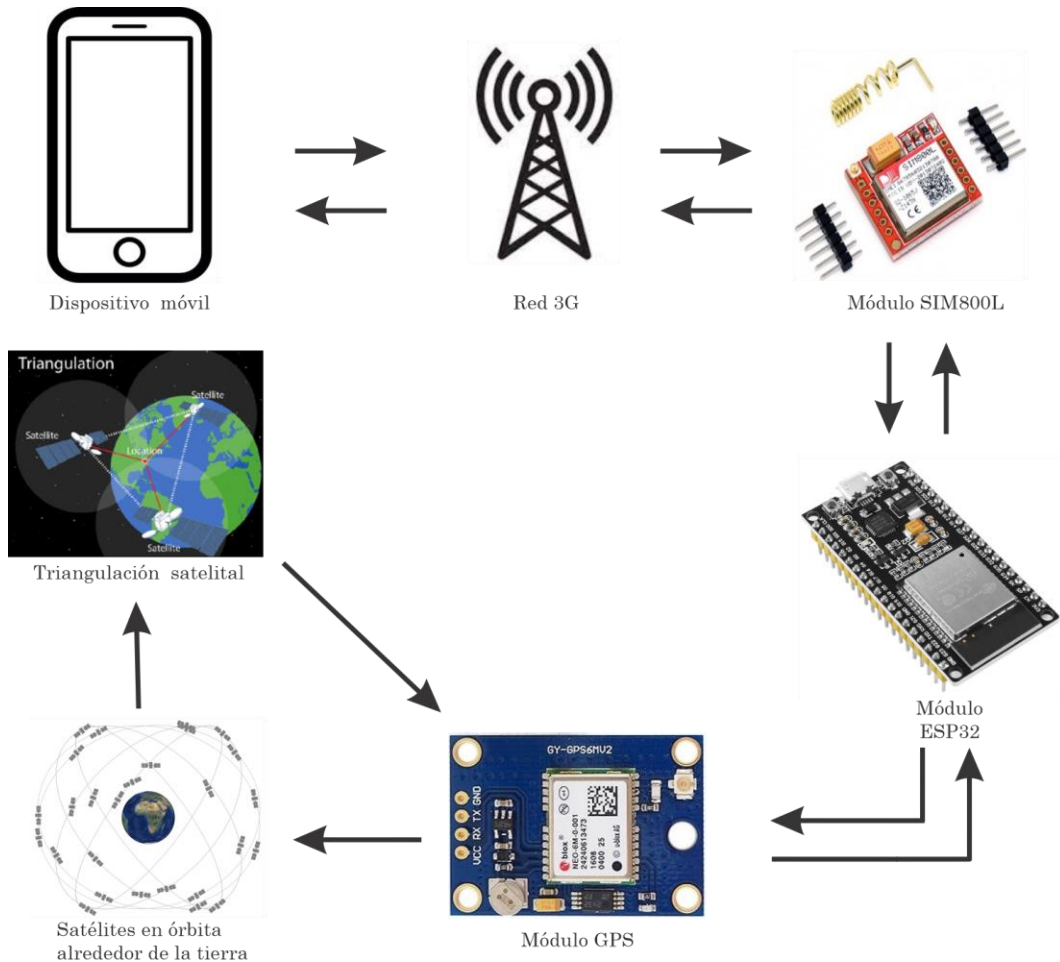


Figura 3.29. Funcionamiento del módulo GPS Neo-6m-7m con el módulo ESP32 y el módulo SIM800L.

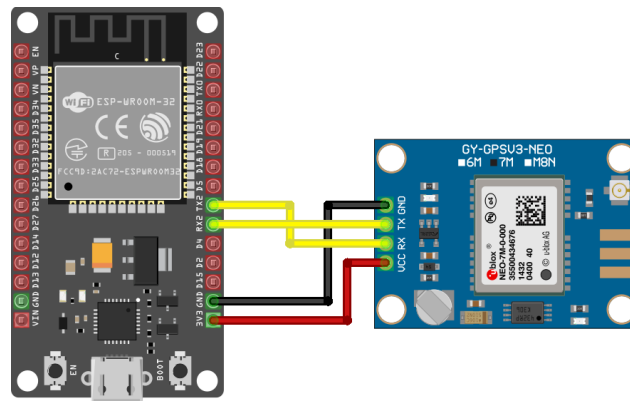


Figura 3.30. Esquema de conexión del ESP32 con el módulo GPS Neo-6m-7m.

3.4.1.6. Dip Switch cuatro posiciones

El DIP switch (*Dual In-line Package switch*) es un dispositivo electrónico que se utiliza para configurar o seleccionar ciertas opciones de funcionamiento en un circuito electrónico. Cada interruptor del DIP switch se puede configurar en una posición “encendido” o “apagado”, lo que permite establecer diferentes configuraciones según la posición de los interruptores.

Las especificaciones técnicas de un DIP switch pueden variar según el fabricante y el modelo específico, pero aquí hay algunas características y especificaciones comunes:

- 4 interruptores.
- 8 pines de conexión.
- Interruptores deslizables.
- Voltaje máximo de 24 VCD.
- Corriente máxima de 100 mA.

El Dip Switch se emplea en el presente trabajo como una alternativa más para el sistema de seguridad, aquí el usuario deberá ingresar la combinación correcta para que ésta envíe una señal de activación (BAN_dip=1) al módulo ESP32. De lo contrario no será posible que el vehículo arranque. Dado que es un interruptor de cuatro posiciones, se tendrán como máximo 16 combinaciones y solo una será la correcta.

Para su configuración se asignaron los GPIO (GPIO33, GPIO32, GPIO35 Y GPIO34) del módulo ESP32 de la siguiente manera (véase Figura 3.31):

- GPIO33 – Interruptor 1 Dip Switch.
- GPIO32 – Interruptor 2 Dip Switch.
- GPIO35 – Interruptor 3 Dip Switch.
- GPIO34 – Interruptor 4 Dip Switch.

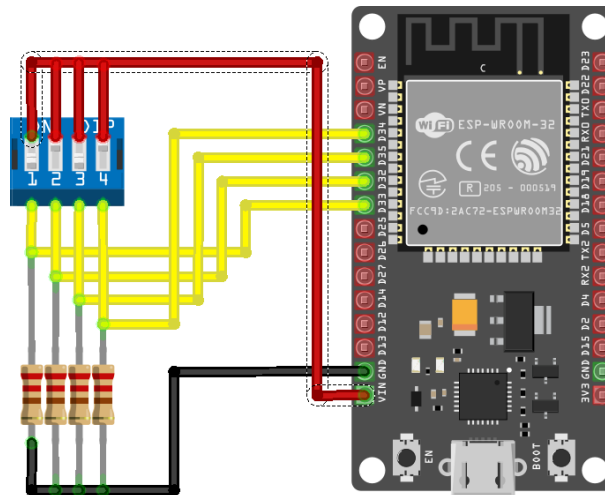


Figura 3.31. Esquema de conexión del ESP32 con el Dip Switch de cuatro posiciones.

Se emplea como una llave digital de seguridad para autorizar el encendido del vehículo, los interruptores se validan leyendo los valores binarios del Dip Switch.

Cada valor está codificado por las posiciones de los interruptores 1, 2, 3 y 4 del Dip Switch (véase Tabla 3.6), por lo que puede tener dieciséis posibles valores ($2^4=16$ combinaciones posibles).

Tabla 3.6. Codificación de los bits en el Dip Switch.

Interruptores del Dip Switch				
Código decimal	Posición 1	Posición 2	Posición 3	Posición 4
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

3.4.1.7. Relevadores

Un relevador es un dispositivo electromagnético conformado por una barra de hierro, denominada núcleo, la cual está rodeada por una bobina de hilo de cobre. Cuando una corriente eléctrica pasa a través de la bobina, el núcleo se magnetiza, convirtiéndose en un imán, por lo que cambia la posición del interruptor normalmente cerrado (NC) a normalmente abierto (NO).

Para este sistema se emplean tres relevadores, el de la bomba de combustible, el de la bobina DIS y el de inyectores de combustible. Para su configuración se asignaron los GPIO (GPIO13, GPIO12 y GPIO14) del módulo ESP32; las terminales asignadas se conectaron a la entrada de cada relevador de la siguiente manera (véase Figura 3.32):

- GPIO13 – Relevador 1: Bomba de combustible.
- GPIO12 – Relevador 2: Bobina DIS.
- GPIO14 – Relevador 3: Inyectores de combustible.

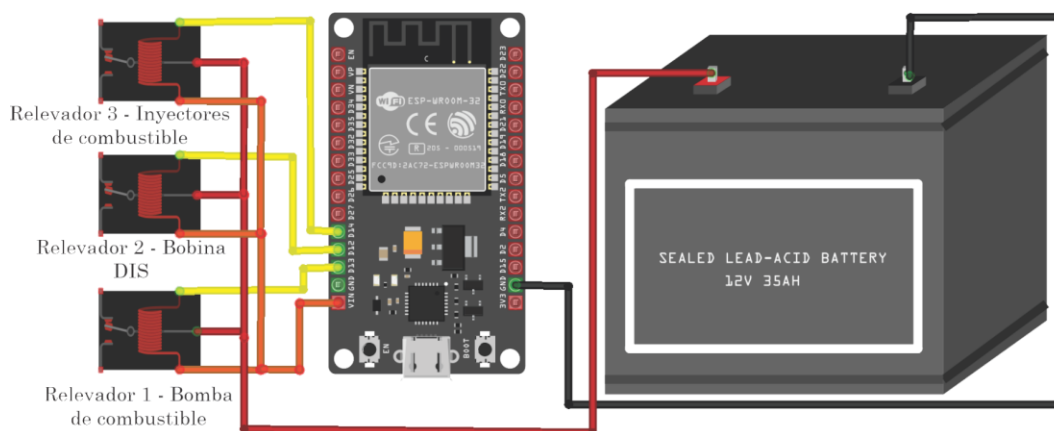


Figura 3.32. Esquema de conexión del ESP32 con los relevadores.

La programación que se realizó en Arduino para el funcionamiento de la pantalla OLED y de los módulos RFID RC522, SIM800L y GPS Neo-6m-7m se presenta en el Apéndice D.

3.4.2. Diseño SW

A continuación, se describen las herramientas software utilizadas en el presente trabajo de tesis.

3.4.2.1. *Arduino IDE*

Entorno de Desarrollo Integrado Arduino (IDE) es un software utilizado para programar y desarrollar proyectos. Proporciona una interfaz gráfica y herramientas para escribir, compilar y cargar código en diversos microcontroladores compatibles, lo que facilita el proceso de desarrollo de proyectos electrónicos y de programación (Arduino, 2023).

Las especificaciones de este sistema son las siguientes:

- Editor de código: Ofrece un editor de texto con funciones de resaltado de sintaxis basado en el lenguaje de programación C/C++.
- Compilación y carga de programas: Permite compilar el código y cargarlo en la placa seleccionada. El IDE se comunica con las placas a través de un puerto USB.
- Gestión de bibliotecas: Proporciona una interfaz para buscar, instalar y administrar bibliotecas de terceros que amplían las capacidades de Arduino.
- Monitor serie: Permite visualizar la comunicación serial entre la placa y la computadora; es una herramienta útil para depurar y visualizar la salida del programa.
- Compatibilidad con múltiples placas: Arduino IDE es compatible con una amplia gama de placas Arduino y placas basadas en microcontroladores AVR y ARM, como Arduino Uno, Arduino Nano y Arduino Mega, entre otras.
- Soporte de extensiones: Permite a los desarrolladores agregar soporte para nuevas placas y sus características a través de la instalación de extensiones (*core* y bibliotecas).

- Sistemas operativos compatibles: MS Windows, MacOS y Linux.

3.4.2.2. MIT App Inventor

Con la finalidad de proveer al usuario una mejor experiencia, en el MIT se diseñó App Inventor como una herramienta de desarrollo de aplicaciones eficiente y de fácil uso para el usuario (Massachusetts Institute of Technology, 2012). En el presente trabajo se utilizó este entorno para desarrollar la aplicación móvil, la cual se comunica con un teléfono móvil y el módulo ESP32 mediante el protocolo de comunicaciones Bluetooth (véase Figura 3.33). El dispositivo móvil funciona como transmisor y el módulo ESP32 actúa como receptor; inicialmente, el usuario ejecuta el software MIT App Inventor y abre la interfaz gráfica de usuario (GUI, *Graphic User Interface*).

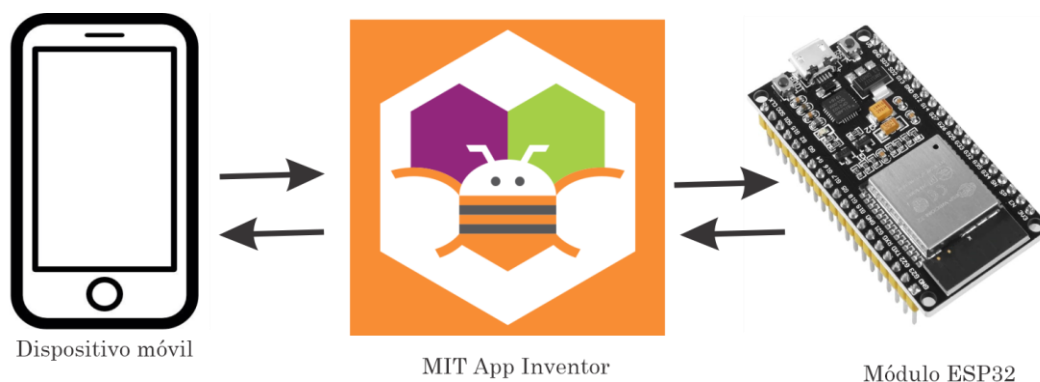


Figura 3.33. Transferencia de datos Bluetooth entre el dispositivo móvil y el módulo ESP32 mediante MIT App Inventor.

App MIT Inventor establece los siguientes requerimientos mínimos para el uso de la aplicación móvil:

- Sistema operativo Android 2.3 Gingerbread, Android 7.0 Nougat, Android 9, Android 10, Android 11 o posterior.
- Bluetooth: bluetooth *classic*, bluetooth BLE, bluetooth 4.0, bluetooth 5.0 o posterior.
- Memoria de 10 a 15 MB.
- Procesador de doble núcleo y un 1 GB de memoria RAM.
- Tipo de archivo Android Package (apk).

Recomienda los siguientes requerimientos:

- Sistema operativo Android 9, Android 10, Android 11 o posterior.
- Bluetooth versiones 4.0, 5.0 o posterior.
- Memoria mayor a 15 MB.
- Procesador de dos núcleos y memoria RAM mayor a 2 GB.

En este proyecto se empleó un dispositivo móvil marca Xiaomi modelo Mi A2 Lite (véase Figura 3.34), cuyas principales características son:

- Pantalla de 5.84 pulgadas con resolución de 1080 x 2280 pixeles.
- Procesador Snapdragon 625 a 2GHz.
- RAM de 4GB.
- Almacenamiento interno de 64GB.
- Expansión microSD.
- Cámara dual de 12MP+5MP.
- Batería de 4000 mAh.
- Sistema operativo Android 8.1.



Figura 3.34. Dispositivo móvil Xiaomi Mi A2 Lite.

Para el diseño de esta aplicación se planteó realizar una interfaz sencilla e intuitiva para el usuario, ya que sólo consta de una pantalla con contraseña y un menú, con base en el diagrama de flujo que muestra la Figura 3.35.

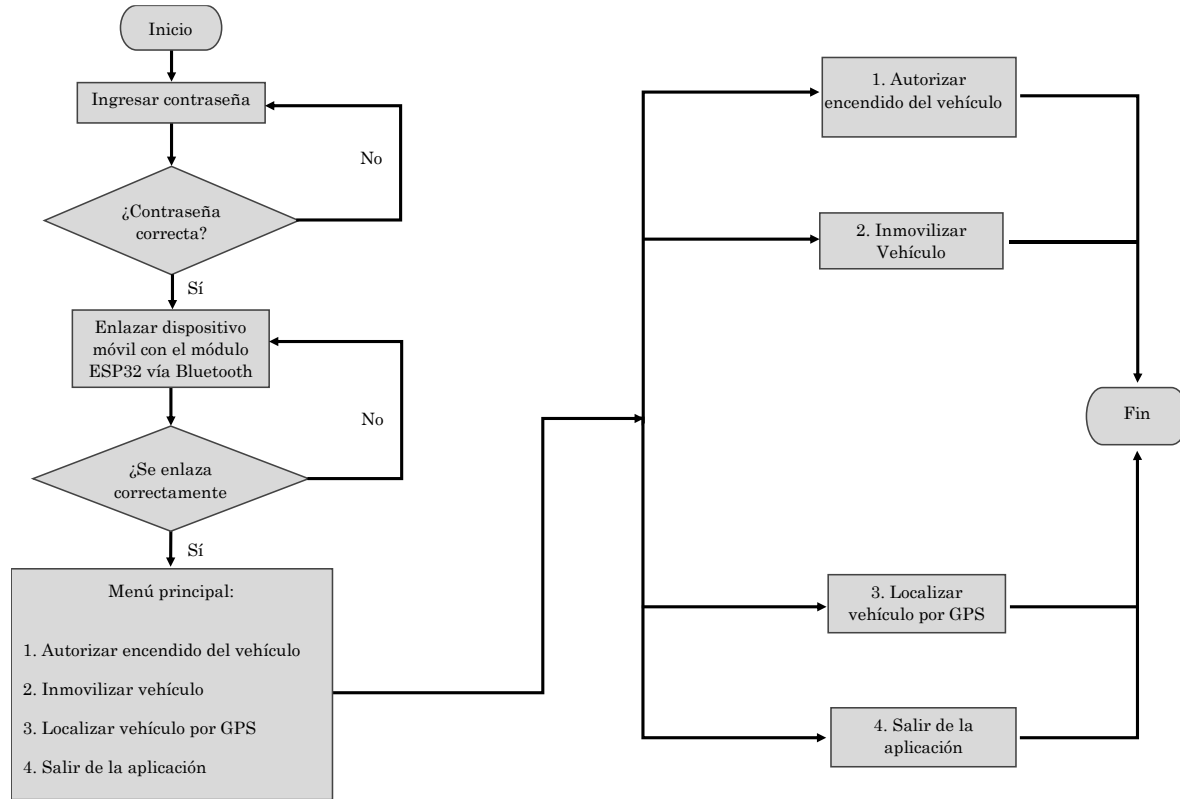


Figura 3.35. Diagrama de flujo que muestra la secuencia de la aplicación.

Lo primero que la aplicación solicita al usuario es que introduzca la contraseña de seis dígitos para poder acceder a la interfaz del menú principal (véase Figura 3.36).



Figura 3.36. Pantalla de ingreso de contraseña.

Posteriormente el usuario debe dar clic al botón Conectar Dispositivo para enlazar el dispositivo móvil con el sistema mediante Bluetooth (véase Figura 3.37); si la conexión es válida, se le notifica al usuario la correcta conexión del sistema de seguridad con el dispositivo móvil.



Figura 3.37. Pantalla para la conexión Bluetooth entre el dispositivo móvil y el módulo ESP32.

La Figura 3.38 muestra el menú principal de la aplicación, en donde el usuario debe dar una pulsación táctil a las opciones que se despliegan en la interfaz de la aplicación. Una vez que el usuario elige una de las tres opciones, la aplicación envía una señal digital por medio de Bluetooth al ESP32, el cual ejecuta una acción con base a la señal recibida.



Figura 3.38. Menú principal de la aplicación del SESA-NiV.

Opción 1: El usuario oprime el botón Autorizar Encendido del vehículo, el módulo ESP32 recibe una “A”, que indica que el vehículo puede encenderse. Se habilita la conexión de los relés que bloquean las señales de activación de la bomba de combustible y de la bobina DIS para impedir la combustión interna.

Cabe mencionar que la pantalla de la aplicación permanece en el menú principal, siempre y cuando se cumplan las tres condiciones de seguridad que se establecieron en el programa principal, el cual está monitoreado por el módulo ESP32, dichas condiciones son las siguientes:

- El usuario mediante la aplicación móvil debe enviar la señal de activación al oprimir la opción Encender vehículo. Una vez que se oprime el botón el sistema escribe $Lec_BT = 'A'$.
- El usuario debe colocar la tarjeta RFID cerca del lector, una vez que se coloca y valida la tarjeta, el sistema escribe $band_rfid_valida = 1$.

Al validar las dos condiciones, el sistema escribe $ban_autorizacion_enc = 1$, por lo que la aplicación recibe ese valor y lo traduce a una señal de confirmación, es decir, la aplicación permite acceder secuencialmente a las opciones Accesorios e Ignición, Encender vehículo y Apagar vehículo (véase Figura 3.39). La opción Apagar vehículo requiere que el usuario presione el freno y botón Start/Stop, para ello la interfaz muestra la advertencia como muestra la Figura 3.40.

La secuencia de funcionamiento del botón Autorizar encendido del vehículo se basó en el diagrama de estados de la Figura 3.41.

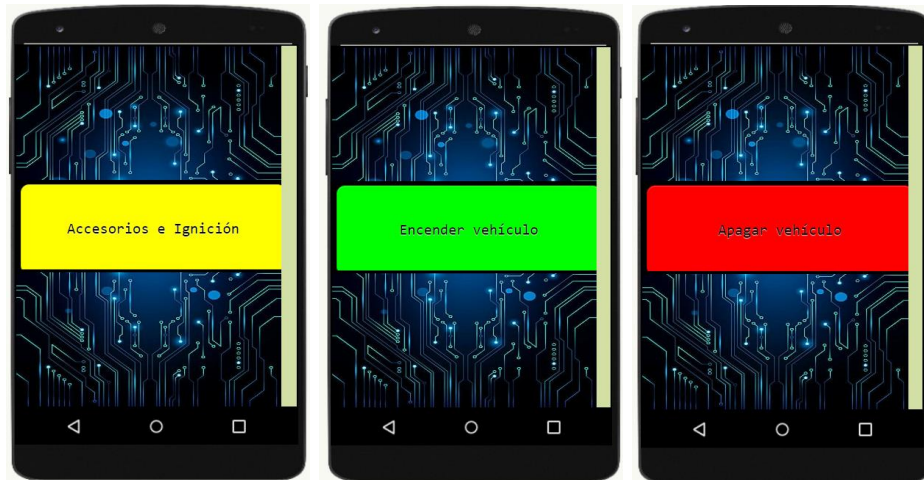


Figura 3.39. Pantallas que se abrirán de manera secuencial al autorizar el encendido del vehículo.



Figura 3.40. Pantalla que indica al usuario presionar el freno para apagar el vehículo.

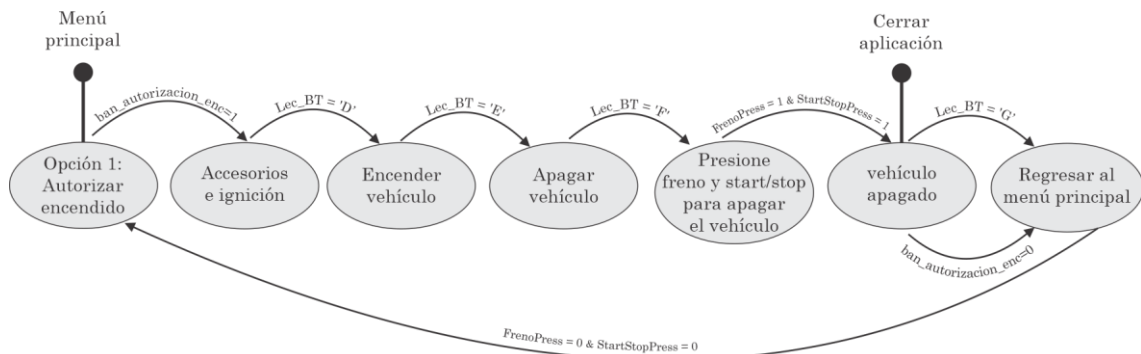


Figura 3.41. Diagrama de estados de la opción 1: Autorizar encendido.

Opción 2: El usuario oprime el botón Inmovilizar vehículo, el sistema de seguridad recibe una “B”, lo que le indica que los relés de alimentación de la

bomba de combustible, la bobina DIS y la bomba de combustible deben permanecer desconectados para evitar el arranque del vehículo o la inmovilización del vehículo (véase Figura 3.42).



Figura 3.42. Pantalla que se abrirá al pulsar el botón Inmovilizar vehículo.

La secuencia de funcionamiento del botón Inmovilizar vehículo se basó en el diagrama de estados de la Figura 3.43.

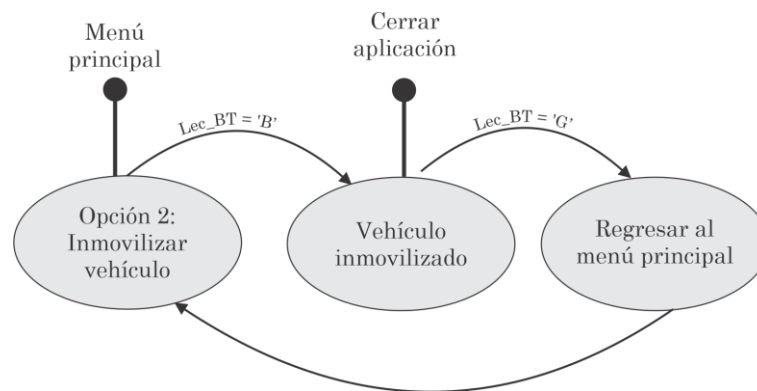


Figura 3.43. Diagrama de estados de la opción 2: Inmovilizar vehículo.

Caso 3: El usuario oprime la opción Localizar vehículo por GPS y el sistema de seguridad recibe una “C” para activar el módulo GPS, este módulo envía las coordenadas de la ubicación de la unidad en tiempo real por medio de un mensaje SMS (véase Figura 3.44).

La secuencia de funcionamiento del botón Localizar vehículo por GPS se basó en el diagrama de estados de la Figura 3.45.



Figura 3.44. Pantalla que indica al usuario el proceso de envío de las coordenadas para la localización del vehículo.

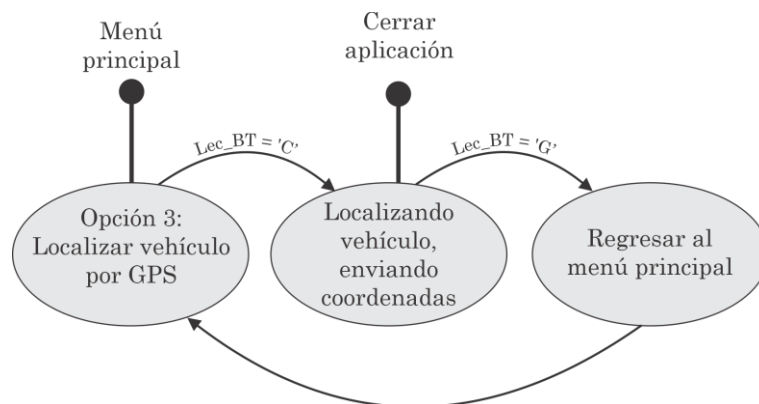


Figura 3.45. Diagrama de estados de la opción 3: Localizar vehículo por GPS.

Opción 4: El usuario oprime el botón Salir de la aplicación, el programa termina y la aplicación se cierra.

En el Apéndice E se describe el Código del SESA-NiV en MIT App Inventor.

3.4.2.3. EAGLE

EAGLE (*Easily Applicable Graphical Layout Editor*) es un software de automatización de diseño electrónico (EDA, *Electronic Design Automation*) que permite diseñar placas de circuito impreso (PCB, *Printed Circuit Board*) a partir del enrutamiento de diagramas esquemáticos con un extenso respaldo de librerías (AUTODESK, 2023).

Contiene un editor para el diseño de diagramas esquemáticos de circuitos electrónicos. Estos esquemas se guardan en archivos con extensión SCH. Mientras que el editor de PCB lo hace con la extensión BRD.

Para el diseño de la placa de circuito impreso se hizo uso del software EAGLE Education en su versión 9.6.2, el cual es un programa optimizado para realizar placas de circuito impreso que cuenta con una amplia lista de componentes electrónicos.

En el Anexo B se describe el diseño de la placa PCB del SESA-NiV.

3.4.2.4. SolidWorks

SolidWorks es un software de diseño asistido por computadora (CAD) 3D, utilizado para el diseño, modelado y análisis de productos y componentes en 3 dimensiones. Es desarrollado por la empresa Dassault Systèmes y ampliamente utilizado en industrias de manufactura, ingeniería mecánica, arquitectura y otras áreas que requieren diseño en 3D (Dassault Systèmes, 2023).

Las características principales de este software son las siguientes:

- **Modelado 3D:** Permite a los usuarios crear modelos 3D de piezas y ensamblajes de manera intuitiva y precisa.
- **Ensamblajes:** Permite la creación de ensamblajes con múltiples piezas que pueden ser colocadas, ajustadas y analizadas entre ellas.
- **Dibujo y documentación:** Facilita la creación de dibujos técnicos 2D a partir de modelos 3D.
- **Análisis de simulación:** Ofrece herramientas para realizar análisis de elementos finitos (FEA, *Finite Element Analysis*), lo que permite evaluar el comportamiento estructural y térmico de los diseños.
- **Animaciones y renderizado:** Permite crear animaciones de productos en 3D y generar representaciones realistas para presentaciones y documentación.

- Integración con otros sistemas: Se integra con una variedad de software de gestión del ciclo de vida del producto (PLM, *Product Lifecycle Management*) y otras herramientas de diseño y fabricación, lo que facilita la colaboración en equipos de diseño.
- Biblioteca de componentes estándar: Incluye una amplia gama de componentes estándar como tornillos, tuercas y rodamientos que pueden ser utilizados en los diseños.

En el Anexo C se describe el diseño de la carcasa del SESA-NiV.

Capítulo 4. Resultados

4.1. Integración de Componentes Hardware y Software

En esta fase se debe contar con múltiples herramientas y métodos para el correcto manejo de la complejidad del sistema. La clave para el diseño de sistemas empotrados es combinar el primer prototipo HW, el SW de aplicación, el código del controlador ECU y el SW del sistema electrónico de seguridad automotriz.

Para la integración del SW y del HW del SESA-NiV se realizaron las siguientes tareas:

- Verificar la correcta conexión de los componentes HW al ESP32.
- Ejecutar programas de prueba al sistema con el fin de valorar posibles fallos.
- Ejecutar el programa principal y verificar su correcto funcionamiento.

4.2. Pruebas del sistema

Las pruebas son necesarias para determinar el correcto funcionamiento del sistema, por lo que se realizaron pruebas de caja gris y de caja negra para verificar que el sistema cumpla con los requerimientos iniciales.

4.2.1. Pruebas de caja gris

A continuación, se describen las pruebas de caja gris que fueron realizadas a los módulos seleccionados para el sistema, con la finalidad de comprender su arquitectura interna y corroborar su correcto funcionamiento.

Para la prueba de funcionamiento de la pantalla OLED fue necesario identificar las terminales de alimentación, transmisión y recepción de datos para establecer la comunicación con el módulo ESP32. Se realizó la conexión de la pantalla OLED con el módulo ESP32 en un protoboard (véase Figura 4.1)

siguiendo el diagrama esquemático de la Figura 3.17. Se compiló el código de prueba y se lograron mostrar datos en la pantalla OLED.

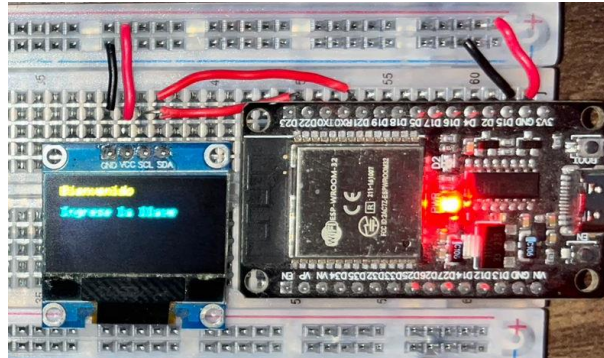


Figura 4.1. Prueba de caja gris de la pantalla OLED.

Para la prueba de entrada de datos y de funcionalidad del módulo RFID fue necesario identificar las terminales de conexión para su correcto funcionamiento y comunicación con el ESP32. Se realizó la conexión de ambos módulos (véase Figura 4.2) siguiendo el diagrama esquemático de conexión de la Figura 3.22.

Se corroboró que el módulo leyera el circuito integrado (*chip*) y devolviera el código de identificación. Una vez realizada la prueba con éxito, se prosiguió a identificar los datos del usuario mediante el reconocimiento de la llave correcta, de esa forma se verificó en el monitor serial de la plataforma Arduino que mostraran las leyendas “llave correcta” o “llave incorrecta”; posteriormente, se imprimieron en una pantalla OLED tal como se muestra en la Figura 4.3.

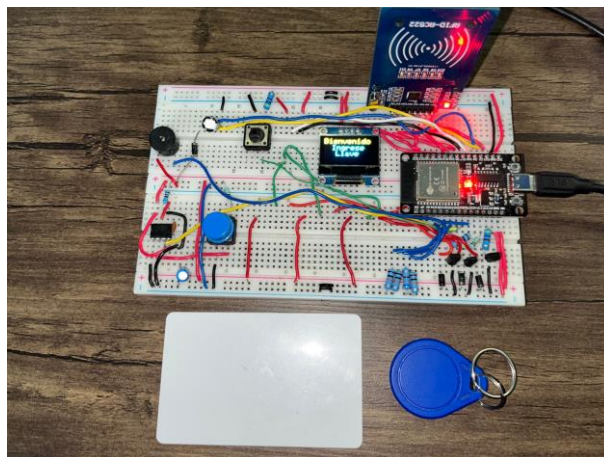


Figura 4.2. Prueba de caja gris del módulo RFID.

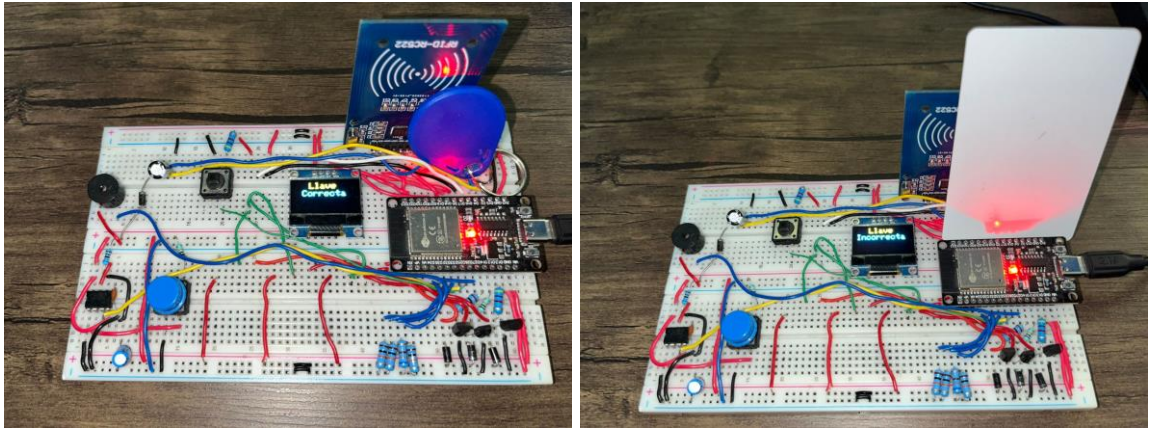


Figura 4.3. Validación de la tarjeta leída por el lector RFID.

Para realizar las pruebas de funcionamiento del módulo GSM SIM800L fue necesario identificar las terminales de alimentación; cabe señalar que este módulo opera con un rango de alimentación de 3.4 a 4.4V, por lo que se tuvo que regular el voltaje con la ayuda de un arreglo de un diodo rectificador y un capacitor de 100uF.

Posteriormente, se identificaron las terminales de transmisión y de recepción de datos, para después conectar el módulo SIM800L con el módulo ESP32 en una protoboard (véase Figura 4.4) de acuerdo con el diagrama esquemático de la Figura 3.25.

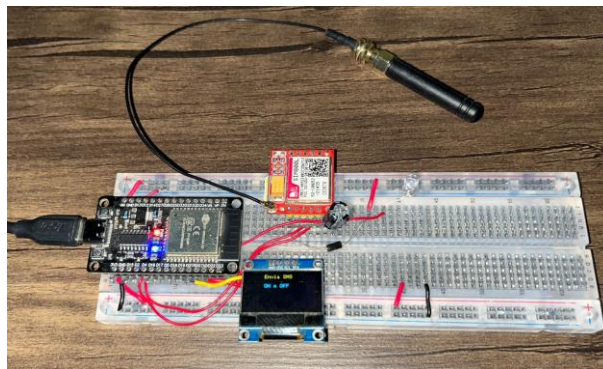


Figura 4.4. Prueba de caja gris del módulo SIM800L.

La primera prueba que se realizó para comprobar la funcionalidad del módulo SIM800L fue el encendido de un actuador e indicar en la pantalla OLED el envío del mensaje con la leyenda ON y OFF por medio de SMS; una vez que el módulo ESP32 recibía dicho mensaje, encendía o apagaba al actuador como se muestra en la Figura 4.5.

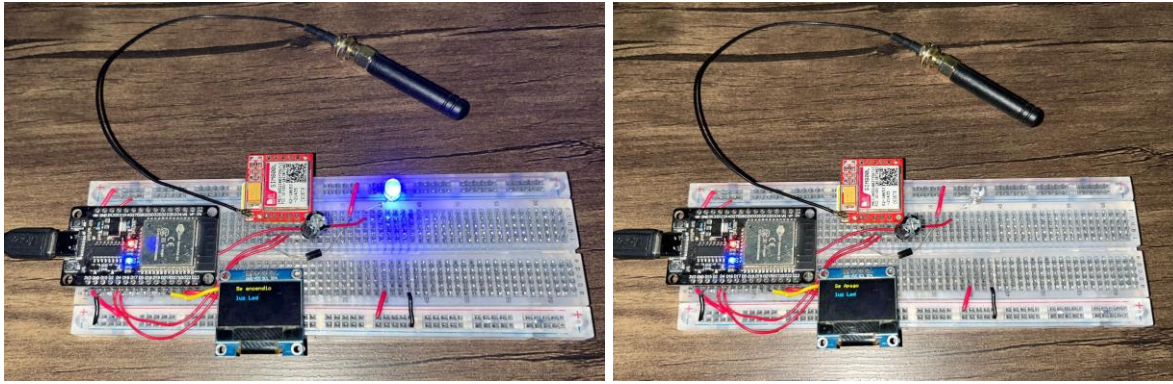


Figura 4.5. Prueba del accionamiento de un actuador al enviar un mensaje SMS de encendido y apagado.

A continuación, se comprobó que el módulo SIM800L retroalimentara la comunicación mediante la notificación al usuario indicando si el actuador se había encendido o apagado mediante el envío de un mensaje SMS al dispositivo móvil (véase Figura 4.6). Se comprobó que el sistema funciona de forma óptima en condiciones de buena recepción de señal.



Figura 4.6. Mensaje SMS de notificación del encendido y apagado del relevador.

Para probar el funcionamiento del módulo GPS fue necesario identificar las terminales de alimentación, de transmisión y de recepción de datos para establecer la comunicación con el módulo ESP32. Se identificó el conector U.FL, el cual establece la comunicación de la antena con el módulo GPS. Posteriormente, se conectó el módulo GPS con el módulo ESP32 en una protoboard (Figura 4.7) siguiendo el diagrama esquemático de conexión de la Figura 3.30.

Se prosiguió con la compilación de un código de prueba, una vez que el módulo lograba realizar la comunicación con los satélites, enviaba las coordenadas de latitud y de longitud al ESP32 y, posteriormente, éste mostraba dicha información en la pantalla OLED.

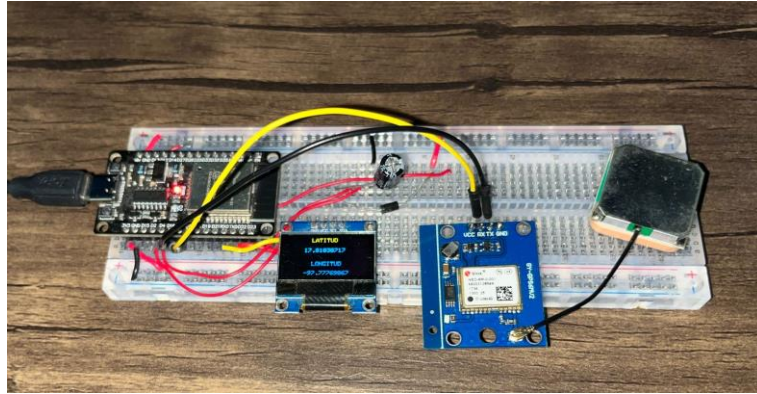


Figura 4.7. Prueba de caja gris del módulo GPS.

4.2.2. Pruebas de caja negra

Las pruebas de caja negra o técnicas de diseño de caja negra son un enfoque de ingeniería de software para el desarrollo de sistemas. Su propósito es verificar que el sistema cumpla con sus especificaciones y requisitos iniciales. Son de gran ayuda para identificar problemas en la interfaz del sistema, errores de inicialización y terminación, errores en las estructuras y posibles brechas de seguridad (véase Figura 4.8).

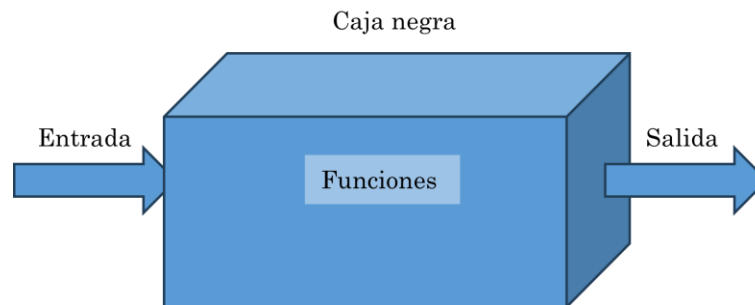


Figura 4.8. Esquema gráfico de las pruebas de caja negra.

Las pruebas de caja negra del SESA-NiV que se consideraron son las siguientes:

- Pruebas de entrada: Consisten en introducir diversos valores de entrada y probar que el sistema muestre los valores en la pantalla OLED.
- Pruebas de funcionalidad: Validan las funciones específicas del sistema, que se realicen las esperadas y no generen errores.
- Pruebas de seguridad: Introducir datos incorrectos o realizar acciones no autorizadas para corroborar que el sistema responde de forma adecuada.
- Pruebas de rendimiento: Se realizan pruebas de velocidad de respuesta del sistema.

Mediante estas pruebas es como se asegura que el sistema cumple con los requisitos para un funcionamiento óptimo sin la necesidad de conocer su diseño y estructura interna.

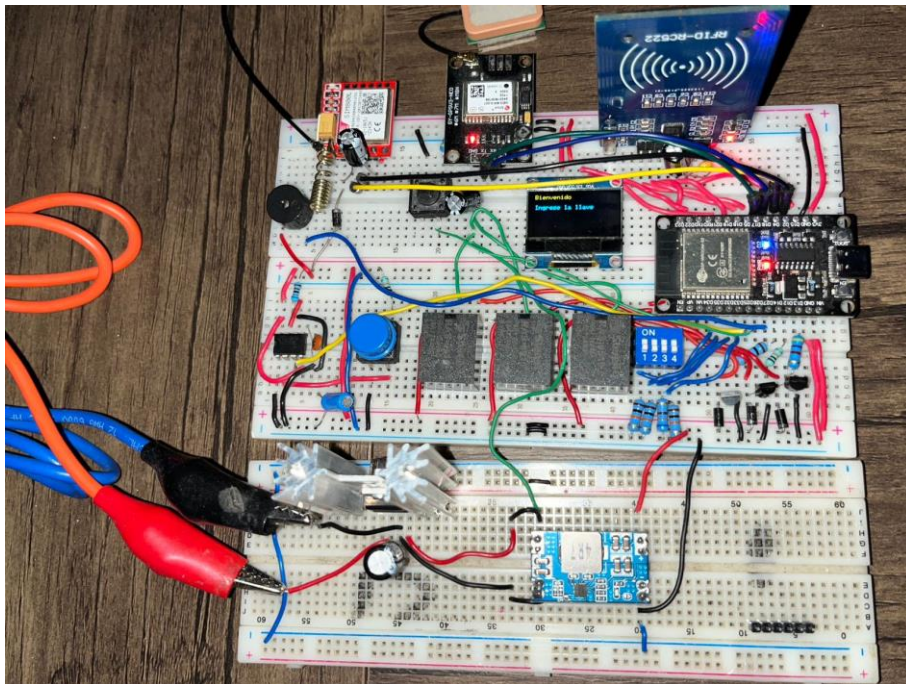


Figura 4.9. Prueba de caja negra del SESA-NiV.

Para las pruebas de caja negra se soldaron los distintos módulos y componentes que conforman el SESA-NiV a la PCB que se muestra en la Figura 4.10.

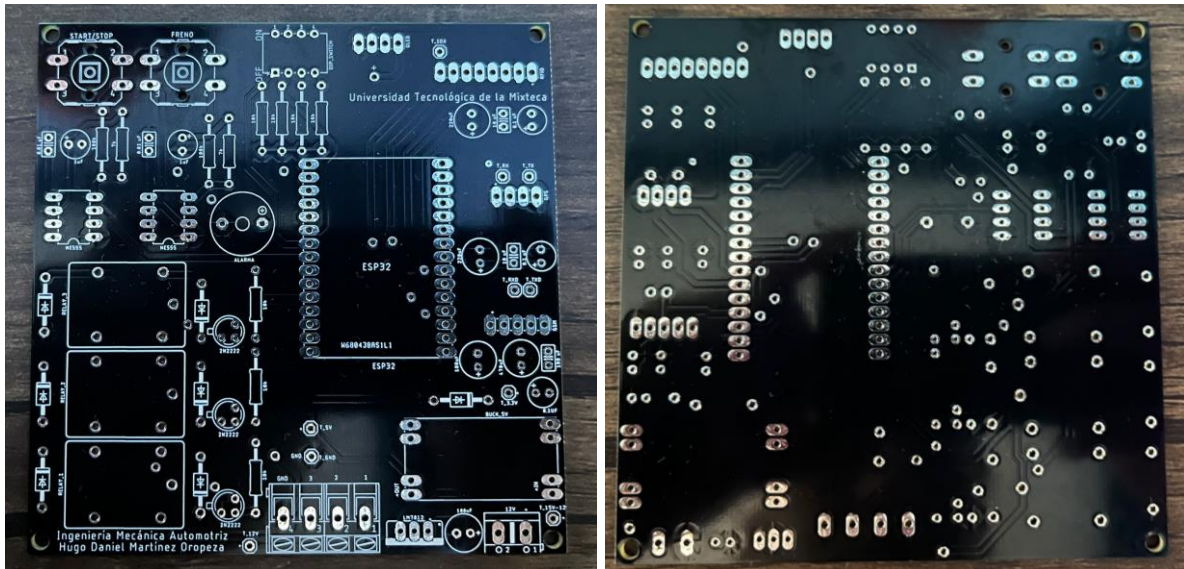


Figura 4.10. Capas Top y Bottom de la PCB del SESA-NiV.

Una vez que todos los componentes de la PCB fueron soldados como se observa en la Figura 4.11, se corroboró la continuidad en todos los puntos de soldadura con ayuda de un multímetro.



Figura 4.11. Placa final del SESA-NiV.

Se alimentó la PCB con una fuente de alimentación de 12V, la cual simula la batería del vehículo, y se realizaron las pruebas de voltaje en distintos puntos del sistema con el fin de comprobar la correcta alimentación en diversos puntos de la PCB (véase Figura 4.12).

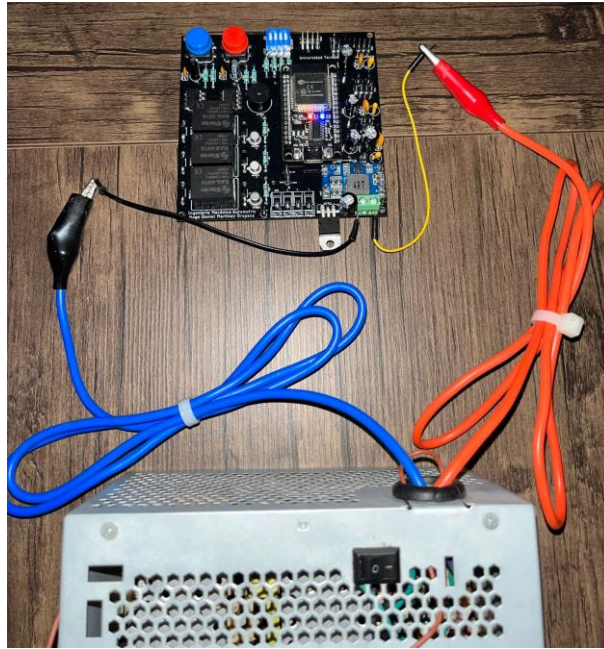


Figura 4.12. SESA-NiV conectado a una fuente de 12V.

Con un multímetro se midieron los voltajes en los puntos de referencia T_3.3V, T_5V, T_12V y T_15V-12V respecto a la terminal T_GND, dando como resultado los voltajes que se muestran en la Figura 4.13. Con lo anterior se comprobó que los módulos del SESA-NiV se alimentan adecuadamente.

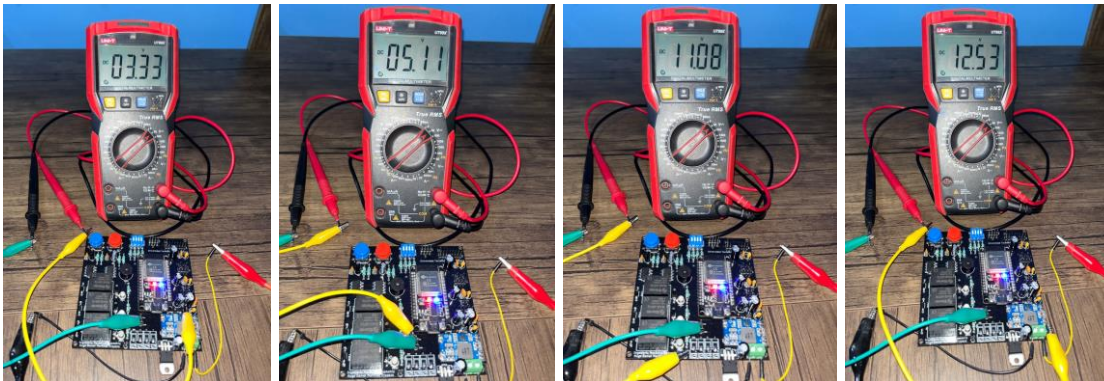


Figura 4.13. Voltajes referenciados a tierra del SESA-NiV.

Después de comprobar que el SESA-NiV recibe los niveles de voltaje adecuados, se integraron los módulos a la PCB para iniciar con las pruebas de funcionamiento del sistema.

En el dispositivo móvil se ejecutó la aplicación y se introdujo la contraseña de acceso (véase Figura 4.14). Sí la contraseña es correcta, se abre una ventana para enlazar el dispositivo móvil con el SESA-NiV vía Bluetooth. Una vez

enlazados, automáticamente se muestra el menú principal de la aplicación como muestra la Figura 4.15.



Figura 4.14. Pantallas para ingresar la contraseña y enlazar el SESA-NiV con el dispositivo móvil mediante Bluetooth.



Figura 4.15. SESA-NiV emparejado al dispositivo móvil y menú principal de la aplicación.

En el menú principal se seleccionó la opción 1. Autorizar encendido del vehículo, el dispositivo móvil envía la señal correspondiente mediante Bluetooth, el SESA-NiV valida la posición de los interruptores del Dip Switch y solicita la validación de la llave RFID. Una vez que se cumplen las condiciones descritas en el diagrama de estados de la Figura 3.5, es posible acceder a las funciones Accesorios e Ignición, Encender vehículo (Figura 4.16) y Apagar vehículo (Figura 4.17).

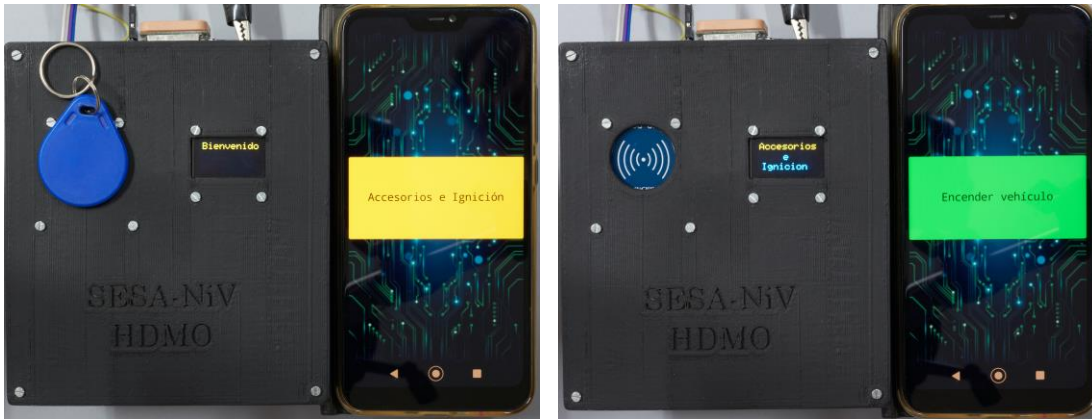


Figura 4.16. Submenús de la aplicación para encender el vehículo.



Figura 4.17. Submenús de la aplicación para apagar el vehículo.

Respecto a la Opción 2. Inmovilizar vehículo, es necesario que el usuario presionó el freno y se envié la de la señal Bluetooth correspondiente, si ambos se cumplen, se procede a inmovilizar el vehículo como se muestra en la Figura 4.18.



Figura 4.18. Condiciones que deben cumplirse para la inmovilización del vehículo.

Después, se verifica el funcionamiento de la Opción 3. Localizar vehículo por GPS enviando la orden mediante Bluetooth al SEESA-NiV, este último ejecuta las

funciones correspondientes para enviar las coordenadas GPS de latitud y de longitud de la ubicación del vehículo (véase Figura 4.19) por mensaje SMS al dispositivo móvil registrado previamente.



Figura 4.19. Envío y recepción de las coordenadas GPS.

Finalmente se verifica el funcionamiento del módulo SIM800L, enviando los mensajes SMS “OFF y GPSV” respectivamente, por lo que este se comunica con el ESP32 regresando la notificación por mensaje SMS de la correcta ejecución de la solicitud como se muestra en la Figura 4.20 . Por lo que el sistema funciona en un rango de largo alcance.



Figura 4.20. Envío y recepción de mensajes SMS al SESA-NiV.

Se realizaron diversas pruebas de seguridad con el fin de corregir vulnerabilidades en la aplicación o en el SESA-NiV, para ello se ingresaron datos del usuario inválidos y posteriormente se validaron. Además, se ingresó una

contraseña aleatoria al iniciar la aplicación y, como respuesta, la aplicación mostró en la pantalla el mensaje de que la contraseña era incorrecta como se observa en la Figura 4.21. Cuando se utilizó una llave RFID inválida, el SESA-NiV emitió la alarma correspondiente.

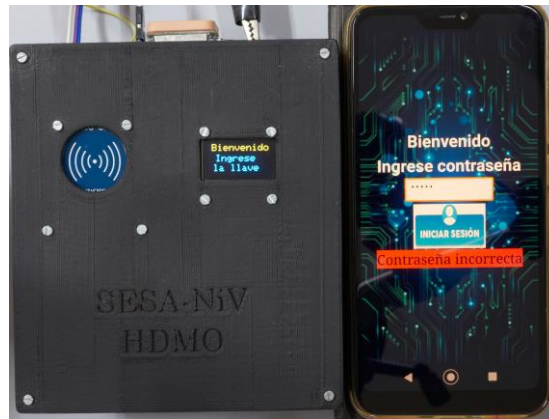


Figura 4.21. Contraseña incorrecta para acceder al SESA- NiV.

En el Dip Switch se configuraron diversas combinaciones inválidas con lo cual el sistema no reconoció la clave y no fue posible continuar con el encendido del vehículo, dando por sentado que el sistema queda protegido a diversos casos.

Finalmente, se analizó el tiempo de respuesta de transmisión y de recepción de datos de los módulos RFID, SIM800L y GPS con el módulo ESP32. Al probar el funcionamiento de módulo RFID por separado, éste presenta un retardo de entre 200 y 500 milisegundos, sin embargo, al integrarse al SESA-NiV dicho tiempo incrementó de entre 5 y 10 segundos en mandar la información al ESP32.

El módulo SIM800L necesita de entre 15 a 20 segundos para enviar un mensaje SMS, una vez que se realiza la petición de envío y es recibido por el dispositivo móvil; además, presenta un retraso de entre 3 a 5 segundos para notificar al usuario por mensaje SMS al dispositivo móvil una vez que se hace la petición correspondiente. Estos tiempos fueron los mismos en el caso de probar su funcionamiento como módulo independiente y como parte del SESA-NiV

Respecto al módulo Micro SIM, éste logra establecer conexión con la red 3G o 4G en un tiempo de 5 a 10 segundos; una vez que se conecta a la red

telefónica, el led testigo de funcionamiento del módulo empieza a parpadear de forma prolongada. Cabe señalar que se requiere de una buena recepción de señal para conectarse a la red.

Una vez que se solicita el emparejamiento Bluetooth entre el dispositivo móvil y el SESA-NiV, el proceso necesita de 1 a 2 segundos para que el dispositivo móvil notifique de un emparejamiento exitoso.

Finalmente, el módulo GPS presenta un tiempo de retraso de 500 milisegundos para comunicarse con el ESP32; cabe señalar que, al inicializarse, el tiempo en que el módulo GPS establece comunicación con las antenas es de aproximadamente de 10 a 20 minutos, pero ya que logra conectarse, el tiempo de respuesta es inmediato.

4.3. Actualización del sistema

Esta fase requiere herramientas adecuadas de HW para la reingeniería. Cabe mencionar que una vez que el sistema funciona correctamente y cumple con los requerimientos iniciales, ya no se llevará a cabo esta fase, pero se propone que futuros trabajos puedan hacer uso de la información y con ello brindar las mejoras que se consideren necesarias mediante la actualización del sistema.

Capítulo 5. Conclusiones

En primer lugar, se realizó una investigación documental, consultado libros, manuales, artículos y páginas de internet, con el propósito de recopilar información sobre el funcionamiento de los sistemas de seguridad automotriz que se encuentran disponibles en el mercado (véase Capítulo 2).

Se analizaron distintos sistemas de seguridad teniendo como resultado la existencia de sistemas de seguridad factibles, pero excesivamente costosos. Por otro lado, la mayoría de los sistemas comerciales accesibles en el mercado basan su funcionamiento en el bloqueo de señales de diversos componentes que impiden el arranque del vehículo e inclusive algunos sistemas sólo funcionan para prevenir de forma manual el uso del vehículo. Se analizaron sus puntos débiles. Se observó que los sistemas de seguridad van de la mano con el sistema de encendido electrónico, para ello fue indispensable estudiar la evolución y el funcionamiento de los sistemas de encendido automotriz en general (véanse los Subcapítulos 2.1 y 2.2).

Por otro lado, el robo de vehículos sigue en aumento; la información consultada de los últimos seis años menciona que, con la salida del Nissan Tsuru del mercado, el modelo Nissan Versa es el vehículo más robado, además de que se ha popularizado como medio de transporte público y privado. Por lo anterior, se propuso el desarrollo de un sistema electrónico de seguridad automotriz para dicho modelo vehicular en concreto.

Con base en lo anterior, se estudió a detalle el diagrama eléctrico del sistema de encendido del Nissan Versa modelos 2017-2021 con la finalidad de comprender su funcionamiento, así como los subsistemas y componentes que lo conforman.

Se delimitó el tema de investigación y se desarrolló el anteproyecto de tesis para describir la hipótesis del proyecto de tesis y establecer los objetivos y las metas para su cumplimiento (véase el Capítulo 1). Se propuso el sistema SESA-

NiV como solución al problema planteado y como metodología de desarrollo la propuesta por Arnold Berger para sistemas empotrados.

Como indica la metodología de desarrollo, la primer fase consiste en establecer los requerimientos del SESA-NiV. Para ello se analizó el sistema de seguridad del Nissan Versa modelos 2017-2021 con la finalidad de comprender la lógica de funcionamiento y plasmarla en un diagrama de estados, que se implementaría en hardware y en un diagrama de flujo para su implementación en software. Una vez identificadas las condiciones y parámetros del sistema original, con base en la investigación documental y en el análisis de los sistemas disponibles en el mercado, se agregaron algunas condiciones para el encendido e inmovilización del vehículo (véase Subcapítulo 3.1).

Para implementar el SESA-NiV se optó como sistema principal el módulo ESP32, cuya selección se basó en la comparación con diversos sistemas MCU, identificando capacidades, especificaciones y modos de consumo de energía como se describe en el Apartado 3.4.1. Dicho módulo cumple con las demandas de requerimientos que establece el SESA-NiV, además de ser compacto, eficiente, disponible a un precio accesible y cuenta con gran compatibilidad para trabajar en diversos entornos de desarrollo y tipos de programación.

Para comenzar a programar en el módulo ESP32, primero fue necesario instalar el firmware para el ESP32 en el entorno de desarrollo de Arduino como se describe en el Inciso 3.4.1.1. Se realizaron pruebas de comunicación entre la PC y el módulo ESP32, además, se instalaron las librerías necesarias para interactuar con los diversos módulos que conforman el SESA-NiV. A continuación, se asignaron las terminales del módulo ESP32 (GPIO) para cada componente y módulos que conforman el sistema.

Establecida la lógica en los diagramas de estados y diagramas de flujo se prosiguió a trabajar en paralelo tanto el HW como el SW como indica la metodología de desarrollo.

Respecto al HW, se pensó en la integración al vehículo, por lo que el SESA-NiV debe ser compacto, considerando el rendimiento y el tamaño de sus componentes.

Para visualizar la información que arroja el SESA-NiV se empleó una pantalla OLED utilizando el protocolo de comunicaciones I2C como se describe en el Inciso 0.

Como medio de validación del usuario para el acceso al SESA-NiV se recurrió al uso un módulo RFID, para ello se realizaron diversas pruebas para corroborar su funcionamiento usando un microchip válido e impidiendo el acceso si se cuenta con uno no válido. Cabe señalar que el rango de lectura del lector RFID-RC522 es de aproximadamente 3 cm, debido a que trabaja en un rango de baja frecuencia.

Dado que el SESA-NiV tiene como objetivo abarcar un rango de funcionamiento de largo alcance por encima de las capacidades de Bluetooth, el cual está limitado a un rango de corto alcance, se empleó un módulo GSM SIM800L, el cual se comunica por medio de la red celular móvil (véase Inciso 3.4.1.4). Para ello, se realizaron diversas pruebas de funcionamiento y eficiencia al módulo, el cual funcionó adecuadamente, presentando un tiempo de espera para la decodificación del mensaje de texto entre el transmisor y el receptor de aproximadamente 3 a 5 segundos.

Para el rastreo y búsqueda del SESA-NiV se hizo uso de un módulo GPS, el cual cumplió correctamente con su finalidad arrojando datos de posicionamiento con las coordenadas de latitud y de longitud de forma precisa, pero con la limitante de contar con un retardo de varios segundos para el envío de la información al SESA-NiV una vez que se enciende el sistema, posteriormente es más rápida la transmisión y recepción de datos. Cabe señalar que el módulo GPS requiere de una buena recepción de la señal satelital para su correcto funcionamiento.

Dado que el SESA-NiV puede ser vulnerable a distintos sucesos como la clonación del código de identificación o tener cerca de su rango de alcance algún sistema inhibidor de señales, se realizó la propuesta de implementar un Dip Switch de cuatro posiciones como mejora de la clave de acceso (véase Inciso 3.4.1.6). Inicialmente se planteó que fuese de cinco o más posiciones para incrementar la cantidad de combinaciones y por ende resultara más difícil de decodificar, pero esta propuesta fue desechada dada las limitaciones de las terminales GPIO que contiene el módulo ESP32, los cuales se ocuparon en su totalidad.

Puesto que para impedir el arranque del Nissan Versa, el sistema principal recurre a mantener desactivados los relevadores de la bomba de combustible y de los inyectores, el SESA-NiV mantiene el mismo procedimiento para evitar el encendido, pero en este caso se mantiene también desactivado el relevador que alimenta a la bobina del sistema DIS. Realizando las pruebas para el encendido del vehículo, se consiguió que los tres relevadores se activen y desactiven correctamente en diversas situaciones en las que el usuario lo requiera.

Una vez que se completó la elección e integración de componentes de forma exitosa, se prosiguió con el desarrollo de la aplicación en MIT App Inventor, la cual cuenta con una GUI interactiva y de fácil uso. Se programó la lógica de la aplicación en el entorno online de MIT App Inventor con base a los diagramas de flujo propuestos en el Apartado 3.1.2.

Se consiguió autenticar los datos del usuario habilitar el sistema de encendido electrónico. Desde la aplicación se envió la señal de activación a distintos relevadores, los cuales hacen que el sistema de encendido funcione correctamente, comprobando que al no cumplirse las condiciones para que el vehículo encienda, éste queda inmovilizado.

Una vez que se comprueba que el sistema puede inhabilitarse mediante el bloqueo de las líneas de los relevadores, se prosiguió a enviar la señal de bloqueo

por la red celular a través de un mensaje GSM, comprobando su correcto funcionamiento.

El proceso de diseño y sectorización de la PCB del SESA-NiV pasó por varias pruebas y correcciones, pero al final se logró un diseño óptimo y compacto, además, debido a su complejidad se optó por su fabricación con una empresa China.

Una que vez que la placa PCB se obtuvo de forma física se realizaron pruebas de continuidad en cada punto de soldadura para verificar que no hubiera algún cortocircuito. Posteriormente, se midieron los niveles de voltaje en cada sector de la placa para asegurar que los componentes no sufrieran daños.

Se realizaron pruebas de caja gris y de caja negra a cada componente y módulo del sistema (véase Subcapítulo 4.2) para verificar su correcto funcionamiento respecto a los requerimientos iniciales.

Cabe señalar que el seguimiento de una metodología facilitó el desarrollo del SESA-NiV brindando una mejor estructura para la implementación de un sistema robusto y confiable respecto a las especificaciones iniciales, así como en la selección de componentes y su integración como sistema empotrado.

5.1. Líneas Futuras de Investigación

De acuerdo con los resultados obtenidos del SESA - NiV se proponen las siguientes líneas de investigación:

- Integración de IoT (*Internet of Things*) al sistema de seguridad e inmovilización, realizando una aplicación que se base en la interconexión del dispositivo móvil con el microcontrolador a través de Wi-Fi.
- Integración de un sistema de reconocimiento de voz como medio de seguridad y de acceso al vehículo.
- Realizar un sistema de reconocimiento facial para validar al usuario y autorizar el encendido del vehículo.

- Realizar un diseño basado en ingeniería de software de la aplicación móvil que considere pruebas de usabilidad.
- Implementar un sistema de Valet, teniendo como medida de seguridad una llave digital la cual permita la activación controlada de ciertas funciones del vehículo, impidiendo su avance cuando éste recorra cierta distancia y envíe un mensaje SMS notificando al usuario.
- Desarrollar un sistema de monitoreo de voltaje de la batería con el propósito de salvaguardar la seguridad del usuario, evitando el apagado del vehículo en situaciones donde el voltaje esté por debajo de los 12.6V. Esta medida garantizará la capacidad de reinicio del vehículo, asegurando un funcionamiento continuo y confiable.
- Se propone llevar a cabo la implementación práctica de este trabajo en un vehículo, aplicando la misma metodología de manera formal y rigurosa.

Referencias Bibliográficas

- AMIS. (2022). *Robo y recuperación de vehículos asegurados*. Ciudad de México: Asociación Mexicana de Instituciones de Seguros.
- Arduino. (2022). *Arduino Documentation*. Ivrea: Arduino.
- Arduino. (10 de Mayo de 2023). *SoftwareSerial Library*. Obtenido de Página web de Arduino: <https://docs.arduino.cc/learn/built-in-libraries/software-serial>
- Arduino. (15 de Marzo de 2023). *SPI*. Obtenido de Página web de Arduino: <https://www.arduino.cc/reference/en/language/functions/communication/spi/>
- Arduino. (11 de Enero de 2023). *What is Arduino?* Obtenido de Página Web de Arduino: <https://www.arduino.cc/en/Guide/Introduction>
- Arduino, N. Z. (09 de Febrero de 2023). *Wire*. Obtenido de Página web de Arduino: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>
- AUTHOR ALARM. (2021). *IGLA Manual de Usuario*. Slovenia: AUTHOR ALARM.
- AUTODESK. (01 de Marzo de 2023). *EAGLE/Fusion 360*. Obtenido de Página Web de Autodesk: <https://www.autodesk.com/products/eagle/overview>
- Aziz, N., Abdul Aziz, N., & Bin Md Tahir, M. (2011). Immobilizer Engine Control Unit (ECU) Using Radio Frequency Module. *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*, 2-6.
- Berger, A. S. (2002). *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques*. Lawrence, Kansas: Publishers Group West.
- Cárdenas Patiño, C. G., & Villacrés Campoverde, D. F. (2021). *Diseño e implementación de un sistema de seguridad antirrobo por inmovilización del motor mediante corte de combustible y señal de alerta por llamada de*

voz a dispositivo móvil, complementado con señal de ubicación del vehículo por GPS. Cuenca: Universidad Politécnica Salesiana.

Chamú Morales, C. (2005). *Desarrollo de un Sistema Educativo para la Enseñanza del Protocolo de Comunicaciones CAN.* Huajuapán de León: Universidad Tecnológica de la Mixteca.

Dassault Systèmes. (2023). *SOLIDWORKS.* Obtenido de Página web de Dassault Systèmes: <https://www.solidworks.com/support/home>

Denton, T. (2012). *Automotive Electrical and Electronic systems.* Abingdon: Taylor & Francis Group.

Denton, T. (2018). *Automobile mechanical and electrical systems.* Abingdon: Taylor & Francis Group.

Dynda, A. (08 de Febrero de 2023). *Adafruit SSD1306.* Obtenido de Página web de Arduino: <https://www.arduino.cc/reference/en/libraries/adafruit-ssd1306/>

Espressif Systems. (2022). *ESP32 Series Datasheet.* Espressif Systems.

Espressif Systems. (2022). *Espressif.* Espressif .

Fabila Ledesma, R. (2016). *Sistema de autenticación, monitoreo y detección en tiempo real de vehículos motorizados.* Ciudad de México: Instituto Politécnico Nacional.

Flores Caballero, A., & Sanchez Muñoz, R. (2016). *Diseño y construcción de un sistema de seguridad vehicular mediante monitoreo vía GSM-GPS.* Ciudad de México: Instituto Politécnico Nacional.

GithubCommunity. (15 de Marzo de 2023). *MFRC522.* Obtenido de Página web de Arduino: <https://www.arduino.cc/reference/en/libraries/mfrc522/>

Halderman, J. (2012). *Diagnosis and Troubleshooting of Automotive Electrical, Electronic, and Computer Systems.* New Jersey: Prentice Hall.

- Hart, M. (10 de Mayo de 2023). *TinyGPS*. Obtenido de Página web de Arduino:
<https://www.arduino.cc/reference/en/libraries/tinygps/>
- ISUZU. (2021). *Módulo de control de Carrocería (BCM)*. Tokio: ISUZU.
- Kershaw, J. (2007). *Automotive Electrical and Electronic Systems*. New Jersey:
Prentice Hall.
- Massachusetts Institute of Technology. (2012). *MIT App Inventor*. Obtenido de
<https://appinventor.mit.edu/explore/ai2/tutorials>
- Mecánica Automotriz Fácil. (2017). *El sistema de encendido electrónico*. CDMX:
Mecánica Automotriz Fácil.
- Mendoza Contreras, G. I. (2008). *Sistema de Seguridad para Automóviles SAINCO*. Ciudad de México: Instituto Politécnico Nacional.
- Mukherjee, K. (2014). Anti-Theft Vehicle Tracking and Immobilization. *2014 International Conference on Power, Control and Embedded Systems (ICPCES)*, 2-4.
- Murugesan, V., Rudramoorthy, R., Chandramohan, G., Kumar, M., Kumar, L., Murthy, K., & Kumar, R. (2011). Development of ECU Based Starting System for Automobiles. *2011 International Conference on Process Automation, Control and Computing*, 1-4.
- Neacsu, D. (2021). *Automotive Power Systems*. Boca Raton: Taylor & Francis Group.
- Nissan. (2007). *Power supply, ground & circuits elements*. Yokohama: Nissan.
- Nissan. (2021). *Manual del propietario Nissan Versa 2021*. Yokohama: Nissan.
- Nissan. (2 de Febrero de 2021). *Nissan Versa*. Yokohama: Nissan.
- SIMCom. (2015). *SIM800H&SIM800L_Hardware Design_V2.02*. Shanghai, China: SIMCom.

Trevor, M. (1987). *Automotive Electronic Systems*. Londres: Butterworth-Heinemann.

u-blox. (2014). *NEO-7 u-blox 7 GNSS modules Data Sheet*. Thalwil, Suiza: u-blox.

UNIT Electronics. (20 de junio de 2023). *Mini 560 Regulador Step Down 99% Eficiencia*. Obtenido de Página web de UNIT Electronics: <https://uelectronics.com/producto/mini-560-regulador-step-down-99-eficiencia/>

Anexo A. Instalación del Firmware en el Módulo ESP32

Al tener diversos componentes electrónicos en el sistema, es necesario que el sistema embebido cuente con un software que los haga funcionar correctamente, para ello se hizo uso de un lenguaje de programación de propósito general y como software de entorno de desarrollo interactivo se hizo uso del IDE Arduino, para permitir la comunicación con cada componente, programando la función que ejecuta cada módulo o componente mediante código una vez que el usuario lo requiera. Para el desarrollo de este trabajo se empleó lenguaje C utilizando el IDE Arduino en su versión 2.1.0 (véase Figura A.1).

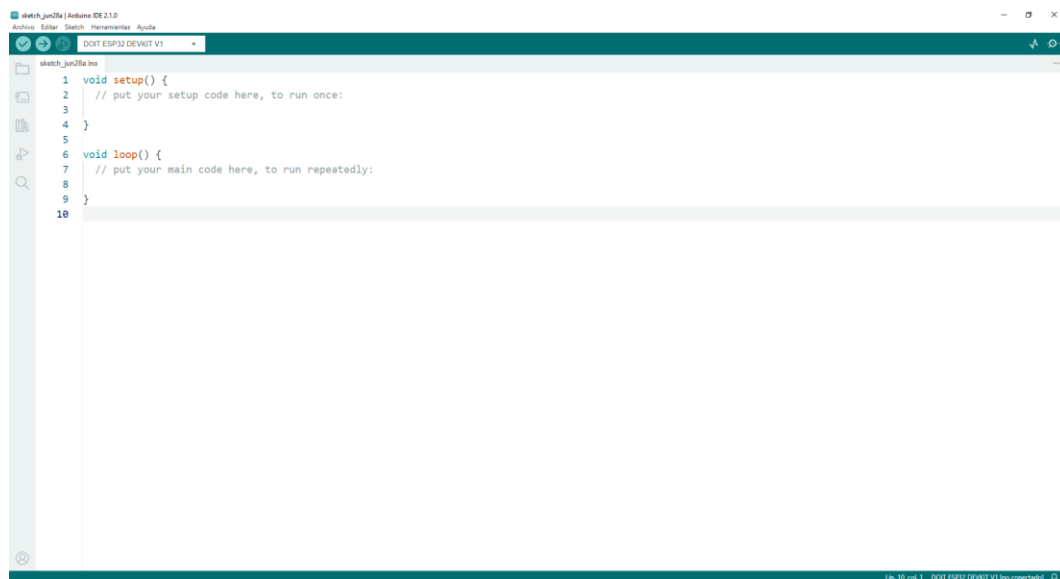


Figura A.1. IDE Arduino.

Una vez instalado el entorno de desarrollo Arduino, es necesario realizar los siguientes pasos. Primeramente, en el menú Archivo seleccionar Preferencias, se abre una ventana como muestra la Figura A.2. En la pestaña Configuración y en donde indica URLs adicionales de gestor de placas se debe escribir el siguiente enlace: https://dl.espressif.com/dl/package_esp32_index.json (véase en la parte inferior izquierda de la Figura A.2) y después aceptar.

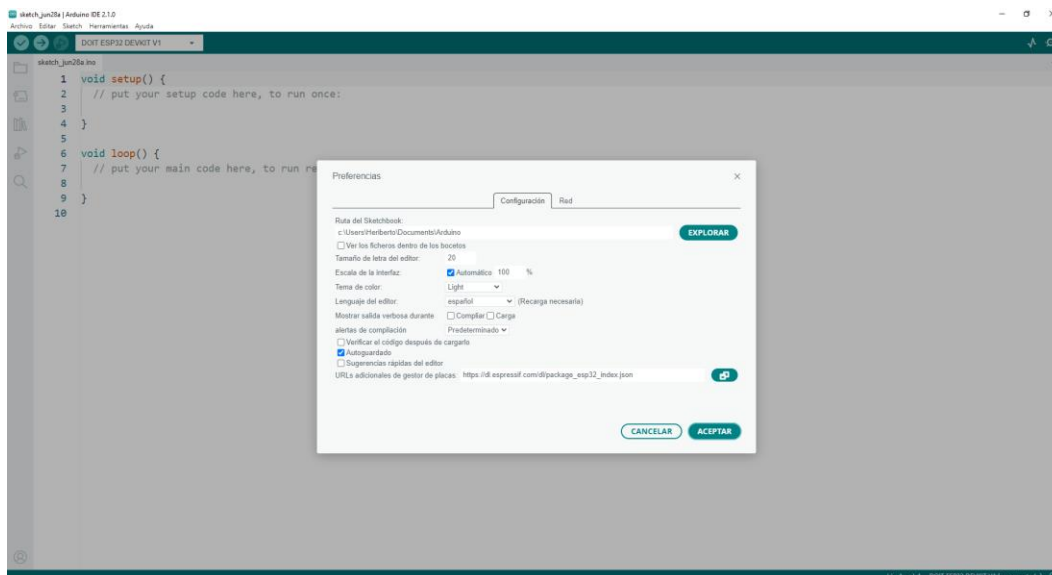


Figura A.2. Pestaña para instalar el URL gestor de placas.

Posteriormente, en la opción Placa del menú Herramientas se selecciona la opción Gestor de placas, se habilita una pestaña en la parte izquierda (véase Figura A.3). En el Filtro de búsqueda se escribe Espressif ESP32 y se da clic en instalar la opción esp32 de Espressif System.

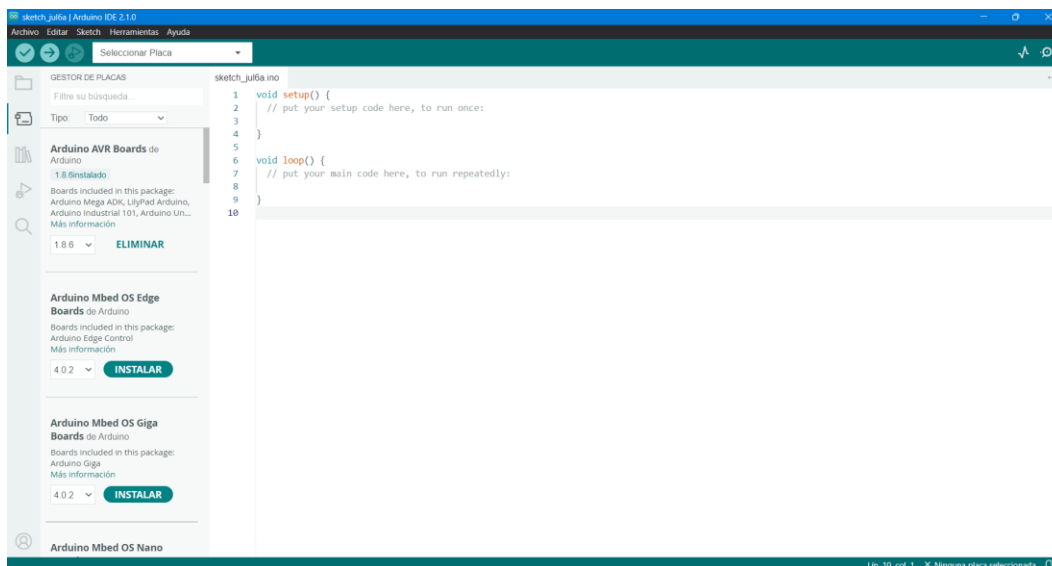


Figura A.3. Menú del gestor de placas.

Una vez que se instaló el gestor de tarjetas, se repiten los pasos anteriores para seleccionar ESP32 Dev Module del submenú esp32 (véase Figura A.4).

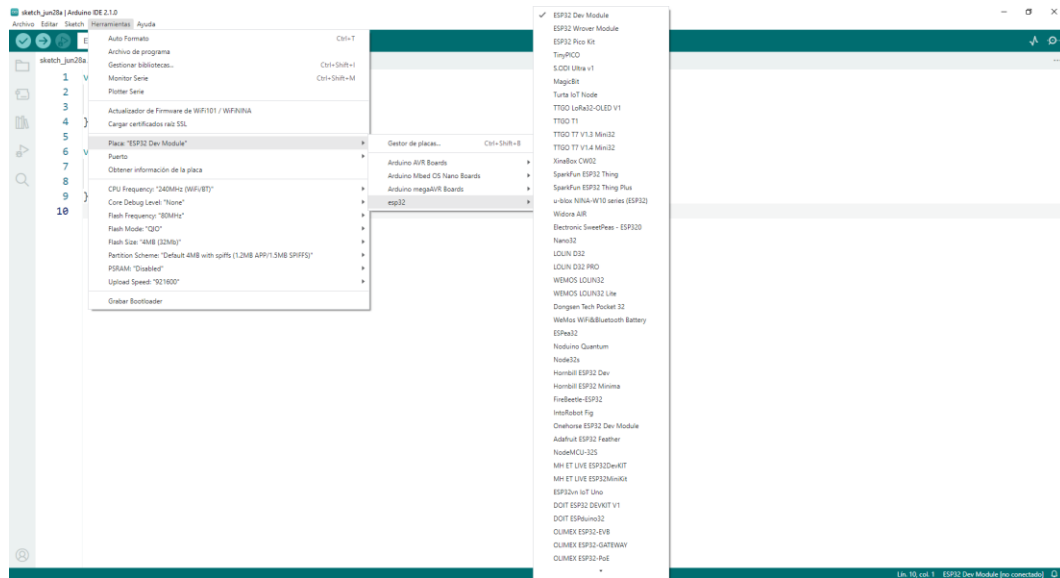


Figura A.4. Menú del tipo de placa al cual se copiará el programa principal.

Esto permite establecer la conexión de la tarjeta ESP32 con el entorno de desarrollo Arduino para su programación.

Anexo B. Diseño de la Placa PCB del SESA-NiV

A continuación, se describe el procedimiento de diseño de la placa PCB del SESA-NiV. Se debe corroborar el correcto funcionamiento de cada componente y de cada módulo como muestra la Figura B.1, ahora bien, se prosigue con el diseño del diagrama esquemático del sistema en EAGLE².

Primeramente, se ejecuta el programa y se selecciona File >New >Schematic para desplegar la ventana principal (véase Figura B.2). En la barra de herramientas, que se localiza en la parte izquierda de la pantalla, se selecciona el icono ADD (véase Figura B.3), en este apartado se localizan los componentes electrónicos que se incluyen en las librerías que incorpora EAGLE.

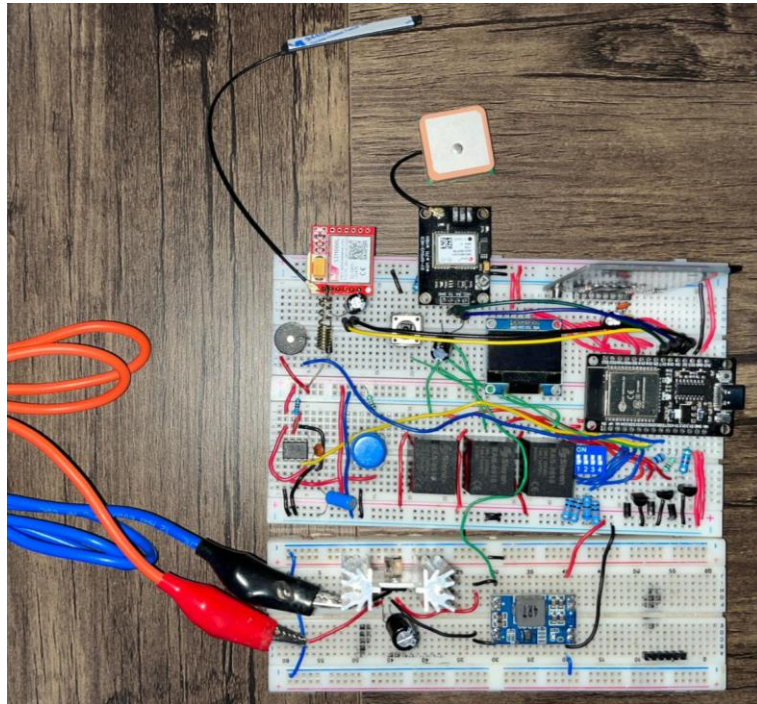


Figura B.1. Prueba de funcionalidad del SESA-NiV.

² Se obtuvo la licencia correspondiente a través de *Autodesk Education Community*.

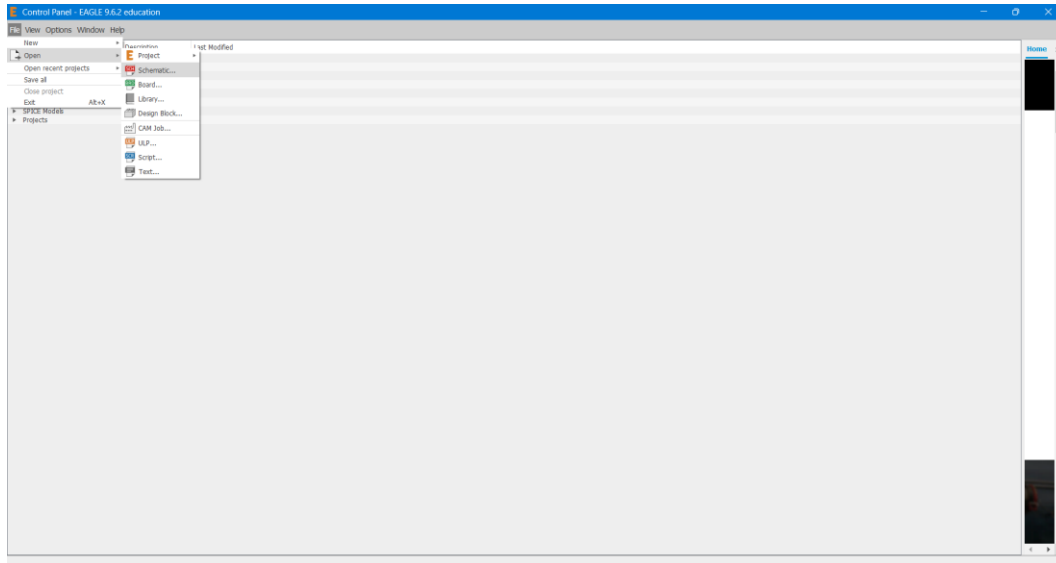


Figura B.2. Pasos para agregar un nuevo esquemático en EAGLE.

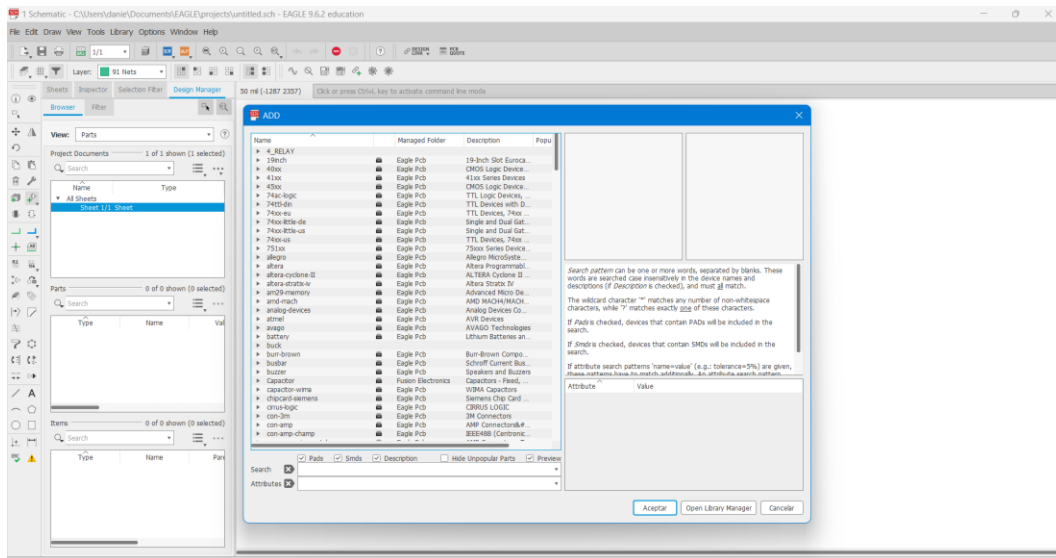


Figura B.3. Secuencia de pasos para agregar un componente de la librería.

Los módulos ESP32, SIM800L, GPS Neo-6m-7m, RFID RC522 y pantalla OLED no se encuentran en dichas librerías, por lo que se procedió a diseñarlos y agregarlos. Para su diseño es necesario contar con las medidas específicas de cada módulo, en este caso se consultaron las hojas de especificaciones y con ayuda de un vernier, se tomó la medida de la distancia de cada terminal (pin) tanto vertical como horizontal. En el menú principal de EAGLE se selecciona File > New > Library como muestra la Figura B.4.

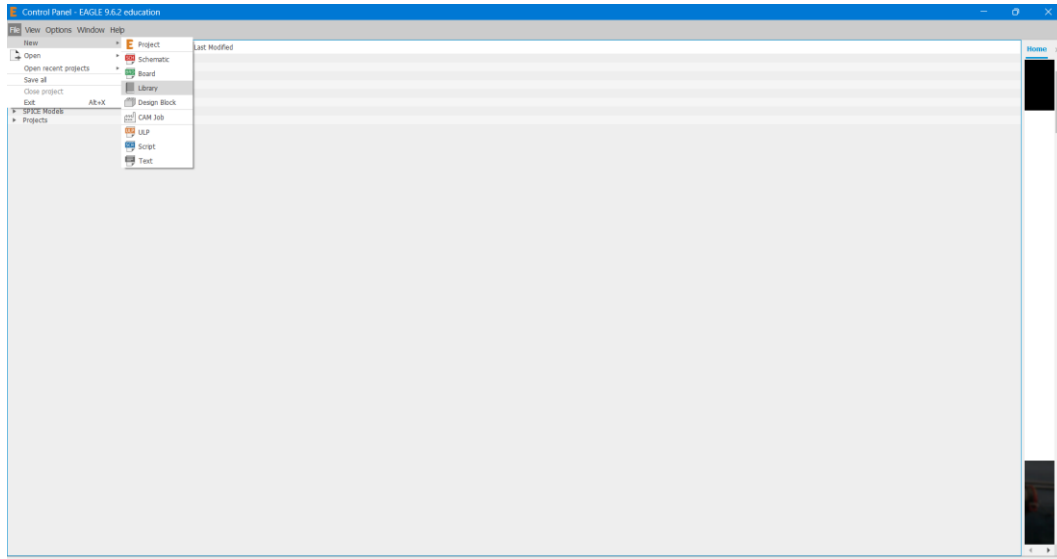


Figura B.4. Pasos para crear una librería.

A continuación, se agrega el símbolo (*symbol*) que será el recuadro que contendrá las terminales (*pads*) para conectar el módulo a diversos sistemas en el diagrama esquemático. Aquí se estableció la cantidad de terminales, su tamaño interno y externo, y la distancia de separación vertical (dY) y horizontal (dX). Se selecciona el botón Add symbol y se declara un nombre que debe coincidir con el asignado al módulo anteriormente. Se abre una ventana nueva, aquí se localiza el icono PAD, del lado izquierdo de la tabla de herramientas, y luego en la barra superior se localiza el icono PAD ARRAY, es aquí donde se abre una ventana en donde se declara el recuadro, el tamaño y cantidad de terminales necesarias para cada módulo en concreto (véase Figura B.5).

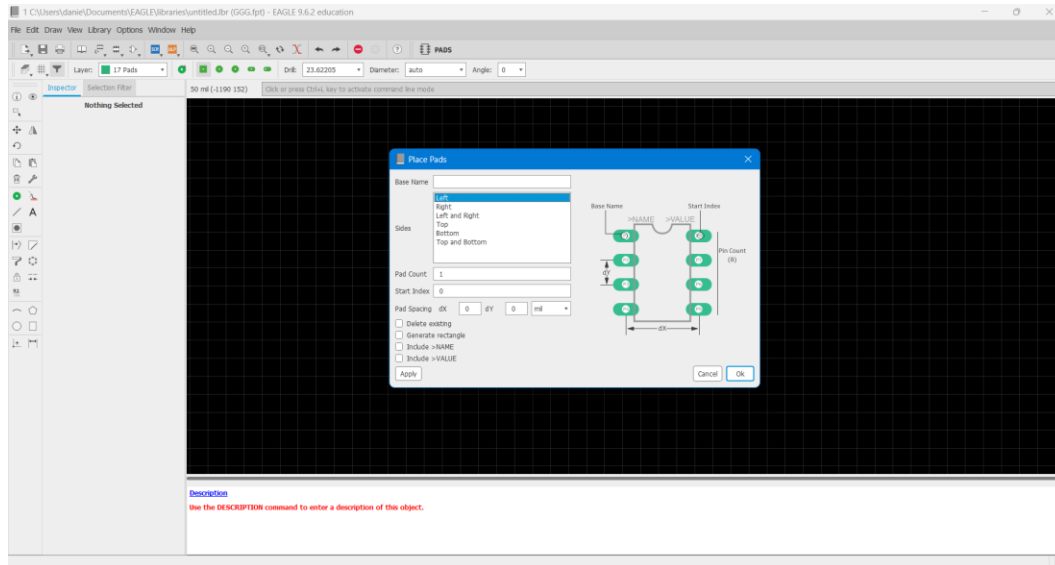


Figura B.5. Crea el recuadro del componente en el esquemático.

Se procede con el diseño de la huella (*Footprint*) para representar el módulo o dispositivo en la pantalla Board en el diseño de la placa PCB. Para ello se selecciona el botón Add Footprint y se asigna un nombre que debe coincidir con el símbolo (*symbol*). En la ventana emergente se sigue el procedimiento descrito anteriormente para diseñar el símbolo (*symbol*) y se vuelve a establecer el tamaño, número y la distancia entre terminales, los cuales deben coincidir con el símbolo.

Una vez que se estableció el símbolo y la huella, se regresa al menú principal mediante el botón Table of contents (véase Figura B.6) y se localiza el botón Add device en la parte inferior izquierda de la pantalla; se selecciona el nombre del dispositivo o módulo (*Device*).

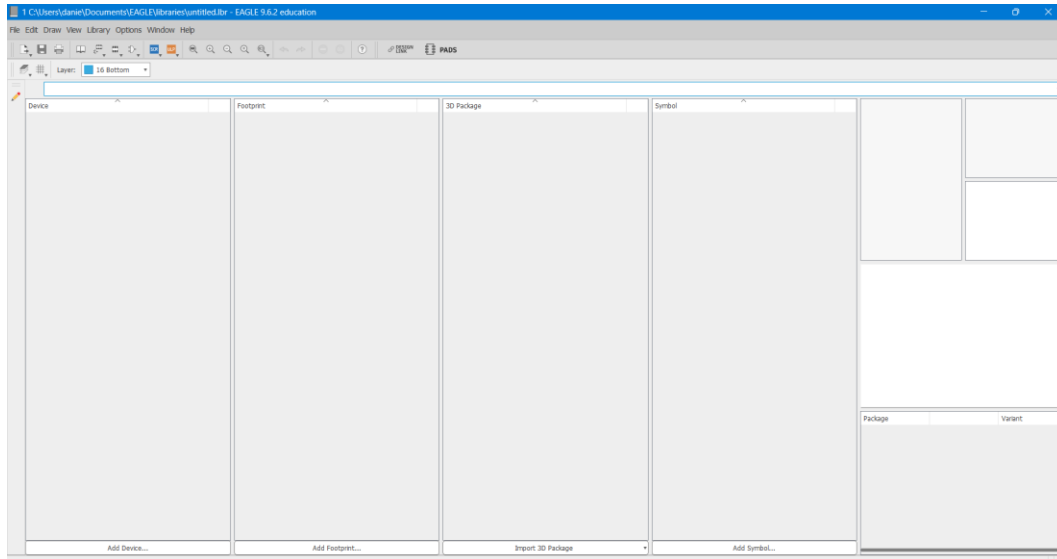


Figura B.6. Menú principal para la creación de un componente.

Una vez que se asigna el nombre al dispositivo, el programa abre una ventana en donde se realiza la búsqueda en la barra de herramientas y se selecciona Add part de la barra de herramientas del lado izquierdo; una vez que aparece el símbolo (symbol), se selecciona y acepta, entonces se abre el cuadro del símbolo junto con sus terminales como muestra la Figura B.7.

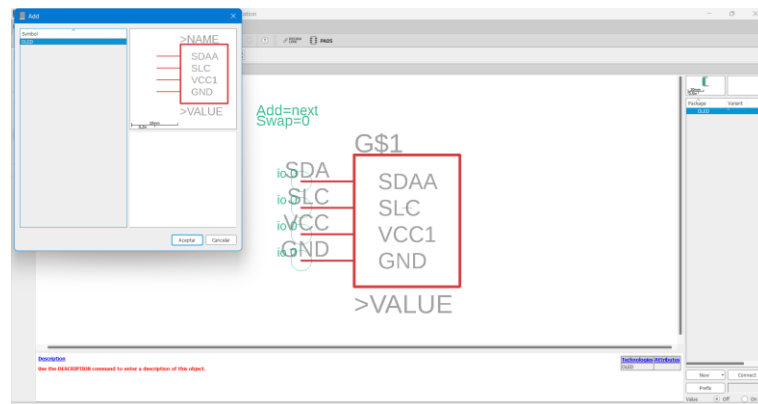


Figura B.7. Símbolo en la librería del componente.

A continuación, en la parte inferior derecha se selecciona new > Add Local Package; se abre a ventana que muestra la Figura B.8 y se elige el Footprint correspondiente.

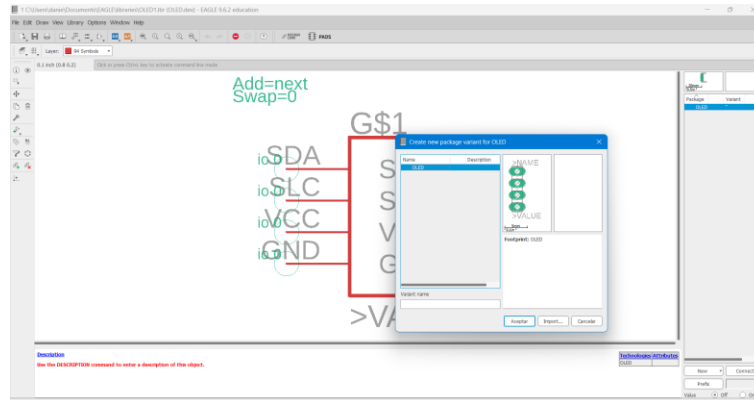


Figura B.8. Pasos para agregar el Footprint a la librería del componente.

Se conectan las terminales del símbolo con las del Footprint como se puede ver en la Figura B.9. Es importante asignar el mismo nombre a cada terminal para evitar confusiones y errores de conexión.

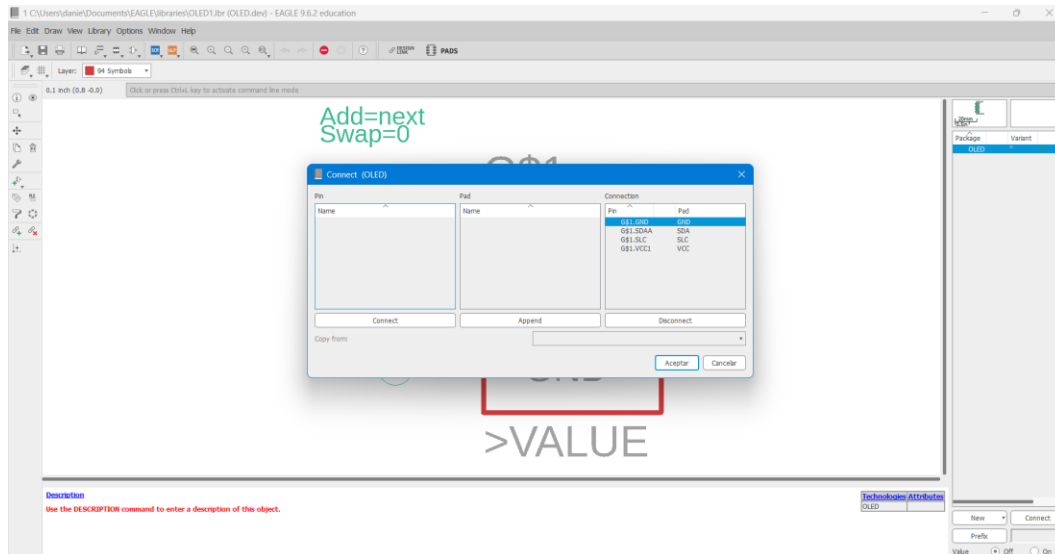


Figura B.9. Conexión de las terminales del símbolo y del Footprint.

Una vez diseñados los componentes y módulos, se realiza el diagrama esquemático del sistema, en donde se utilizaron diversas etiquetas para establecer la conexión entre diferentes módulos y obtener una mejor organización de la placa (véase Figura B.10).

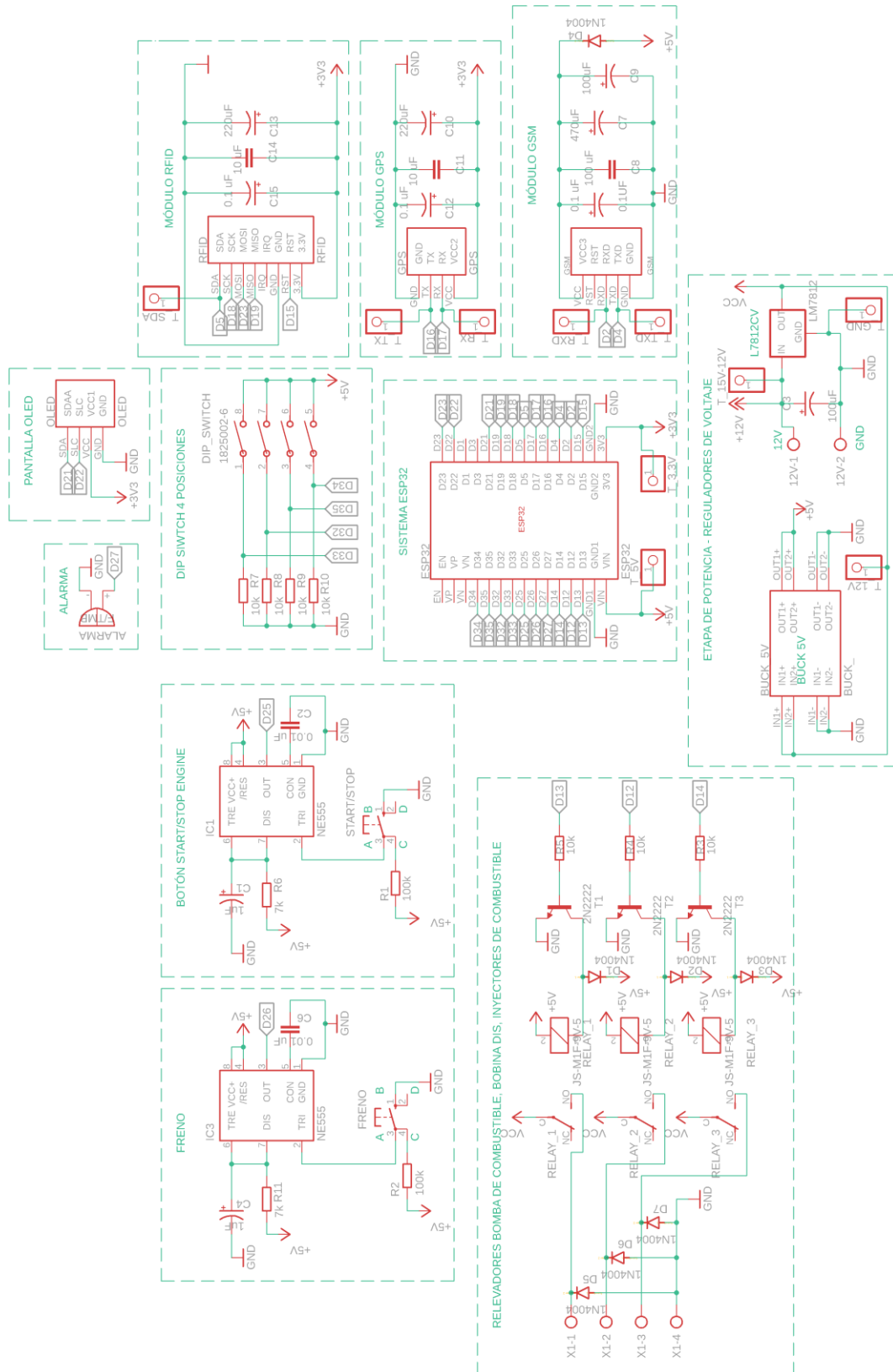


Figura B.10. Esquemático del SESA-NiV diseñado en EAGLE.

Concluido el esquemático, se prosigue con el diseño de la placa PCB, para ello se da clic en Generate/switch to board; el programa genera una placa PCB en 2D virtual mostrando todos los componentes que se establecieron y conectaron en el diagrama esquemático (véase Figura B.11).

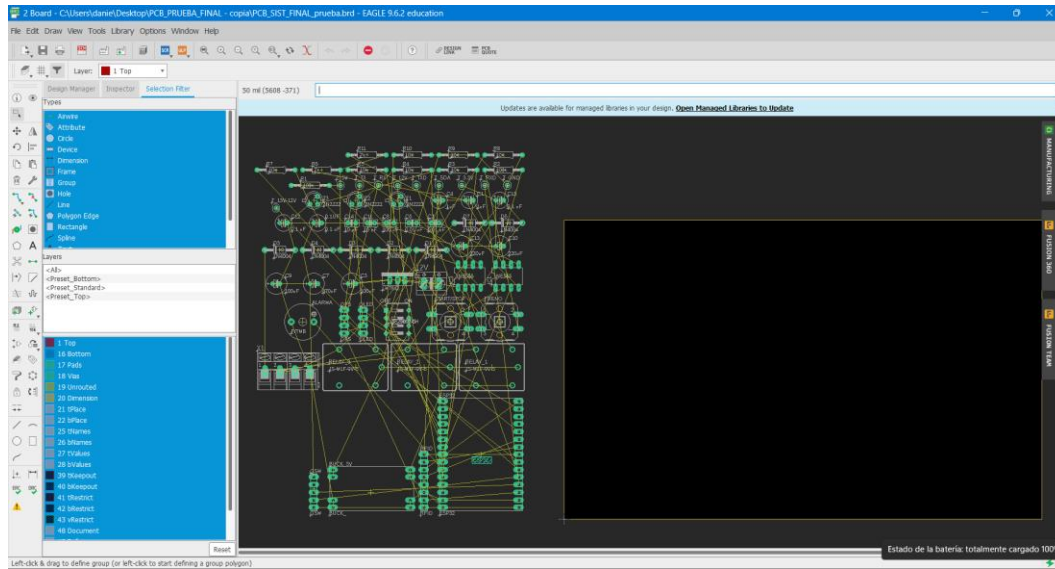


Figura B.11. Generación de la placa PCB.

Para evitar problemas de interferencias electromagnéticas, de señal y tener una mejor disipación térmica, se organizaron los componentes siguiendo las reglas para la colocación de componentes en una PCB. Esto es de suma importancia para tener una buena organización de los componentes, y con ello garantizar un funcionamiento óptimo y mayor durabilidad de la PCB.

Las reglas que se tomaron en cuenta para la fabricación y el diseño de la PCB son las siguientes:

- Agrupación de los componentes por función (sectorizar).
- Utilizar una orientación estándar de los componentes.
- Separar las diferentes áreas entre sí.
- Mantener los componentes alejados de fuentes de calor.
- Crear planos de tierra sólidos.
- Tener en cuenta qué componentes colocar cerca de los bordes del PCB.
- Dejar suficiente espacio para las trazas de cobre.

- Considerar aspectos de accesibilidad y mantenimiento.
- Separar los componentes que trabajen con señales de alta frecuencia.
- Gestión térmica.

También, es necesario que para el diseño de las pistas conductoras de cobre de la PCB se considere la norma para diseño de circuitos impresos IPC-2221, la cual establece los requerimientos para el diseño de PCB e indica la ecuación para calcular el ancho de la pista conductora de cobre. Se realizó el cálculo del área dada la ecuación 1:

$$\text{Área} = \left(\frac{I}{k_1 * \Delta T^{k_2}} \right)^{\frac{1}{k_3}} \quad (1)$$

donde:

I : es la corriente máxima de funcionamiento.

ΔT : es la diferencia de la temperatura máxima de funcionamiento y la temperatura ambiente.

k_1 , k_2 y k_3 : son las constantes para el cálculo.

En este caso al tratarse de una placa externa (doble capa) las constantes adquieren los siguientes valores: $k_1 = 0.0647$, $k_2 = 0.4281$ y $k_3 = 0.6732$.

Se considera una corriente máxima de salida del regulador de 5A, una temperatura máxima de funcionamiento de 85 °C (UNIT Electronics, 2023) y una temperatura ambiente de 25 °C; sustituyendo los datos en la ecuación 1, se obtiene:

$$\text{Área} = \left(\frac{5A}{0.0647 * (85^{\circ}C - 25^{\circ}C)^{0.4281}} \right)^{\frac{1}{0.6732}}$$

$$\text{Área} = 47.190543 \text{ mil}^2$$

Para el cálculo del ancho de la pista, se emplea la ecuación 2:

$$\text{Ancho} = \frac{\text{Área}}{\text{Grosor} * 1.378} \quad (2)$$

En este caso se considera un grosor de la placa de 1 oz/ft², sustituyendo los valores en la ecuación 2 se obtiene que el:

$$\text{Ancho} = \frac{47.190543}{1 * 1.378} = 34.2456 \text{ mil}$$

Sí 1 mil = 0.0254 mm, entonces el ancho es 0.8698 mm.

Para garantizar el correcto funcionamiento de la placa y que ésta no se sobrecaliente, se considera un factor de seguridad un ancho de pista de 0.90 mm.

Una vez que se ubicaron los componentes de manera óptima (véase Figura B.12), se prosiguió con el enrutamiento. Debido a que es una placa de doble capa, es necesario identificar las capas Top y Bottom.

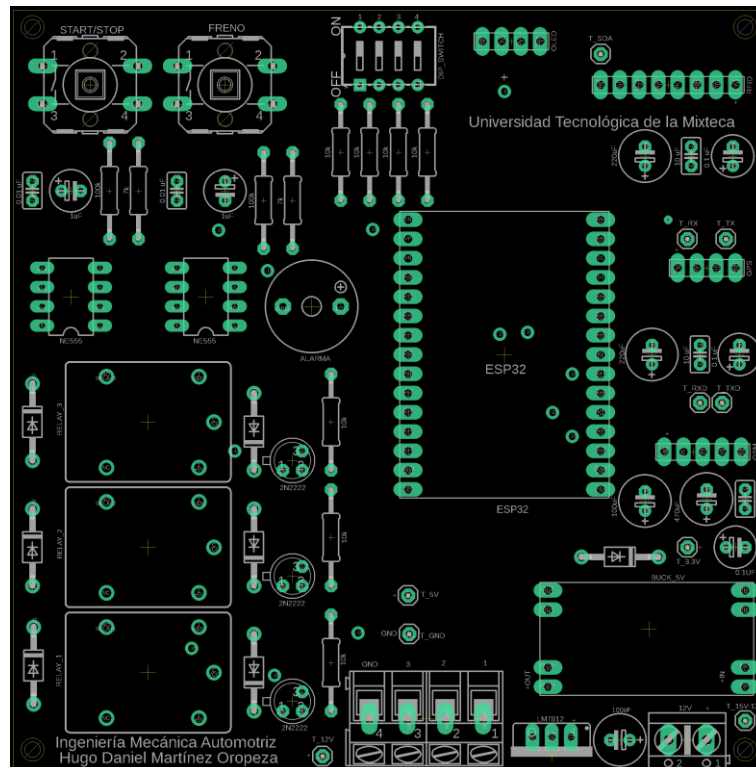


Figura B.12. Prototipo de la PCB sin las pistas.

Una vez que se identificaron las capas, en la barra de herramientas, se selecciona Route Airwire y se prosigue con el enrutamiento componente – componente.

Se inició con el enrutamiento de la capa Top (véase Figura B.13) verificando que las pistas de cobre o rutas no tuvieran ángulos rectos, es decir, que cada nodo no formara un ángulo de 90 grados, debido a que este tipo de ángulo afecta en gran medida la transmisión de señales de altas frecuencia, por lo que se optó por tener líneas con 45 grados en sus nodos.

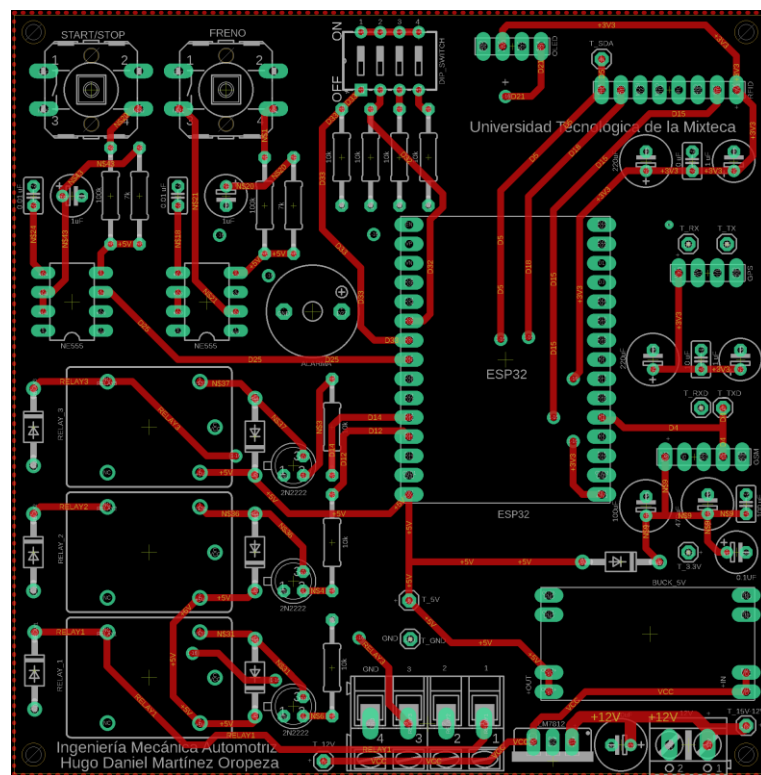


Figura B.13. Enrutamiento de la capa Top.

Posteriormente, se empleó la misma metodología para el enrutamiento de los componentes en la capa Bottom (véase Figura B.14).

Después de haber realizado los enrutamientos, se volvió a corroborar que no hubiera líneas con ángulos rectos. A continuación, se colocó el plano de tierra sólido (GND) en la capa Top, se seleccionó Polygon y se dibujó un recuadro que fuese del mismo tamaño que la placa; posteriormente, se seleccionó la opción Rastnest (véase Figura B.15) y se nombró al plano de tierra sólido como GND; para

ello era necesario nombrar anteriormente las conexiones y líneas GND para que éstas queden en el mismo plano.

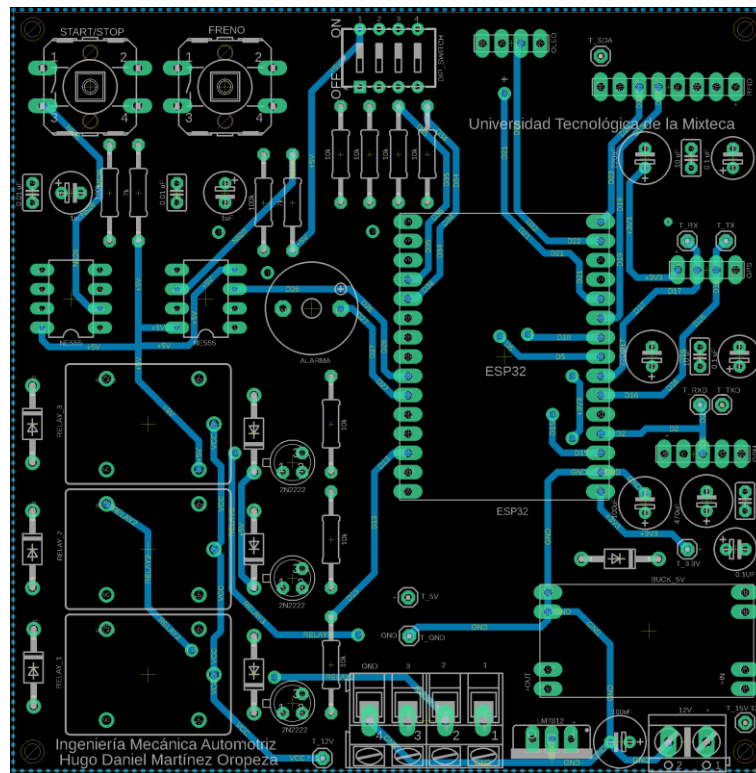


Figura B.14. Enrutamiento de la capa Bottom.

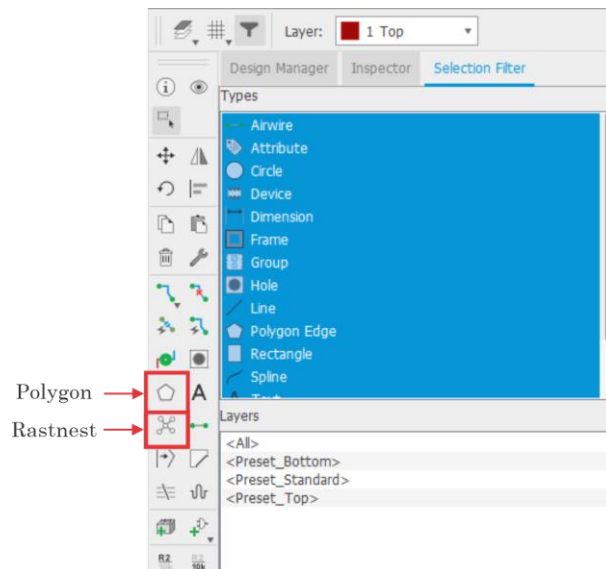


Figura B.15. Iconos Polygon y Rastnest.

Se realizó el mismo procedimiento para la capa Bottom, obteniendo finalmente las dos caras de la placa con planos a tierra sólido como se observa en las Figuras B.16 y B.17.

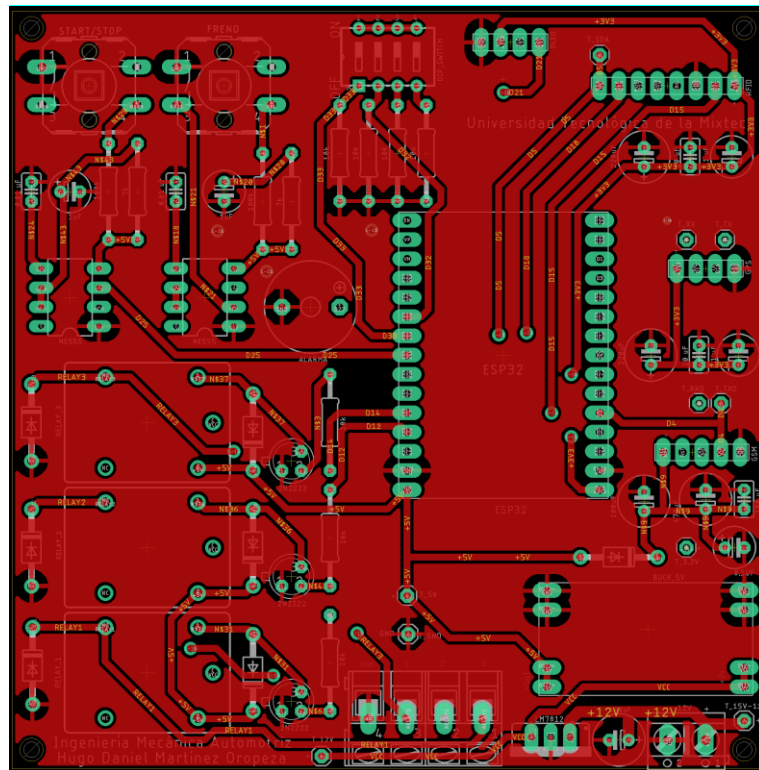


Figura B.16. Colocación del plano GND en la capa Top.

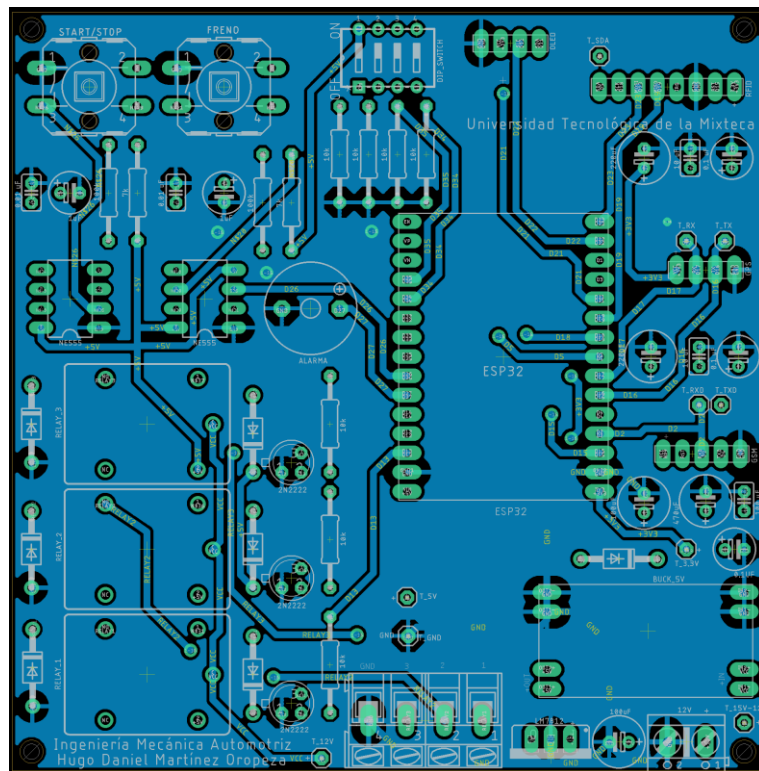


Figura B.17. Colocación del plano GND en la capa Bottom.

Después de haber concluido los enrutamientos, se comprobó la existencia de errores en el diagrama esquemático y se revisaron las reglas de diseño para la PCB; para ello se seleccionaron los iconos que se encuentran en la parte inferior derecha DRC y ERC, como se muestra en la Figura B.18.

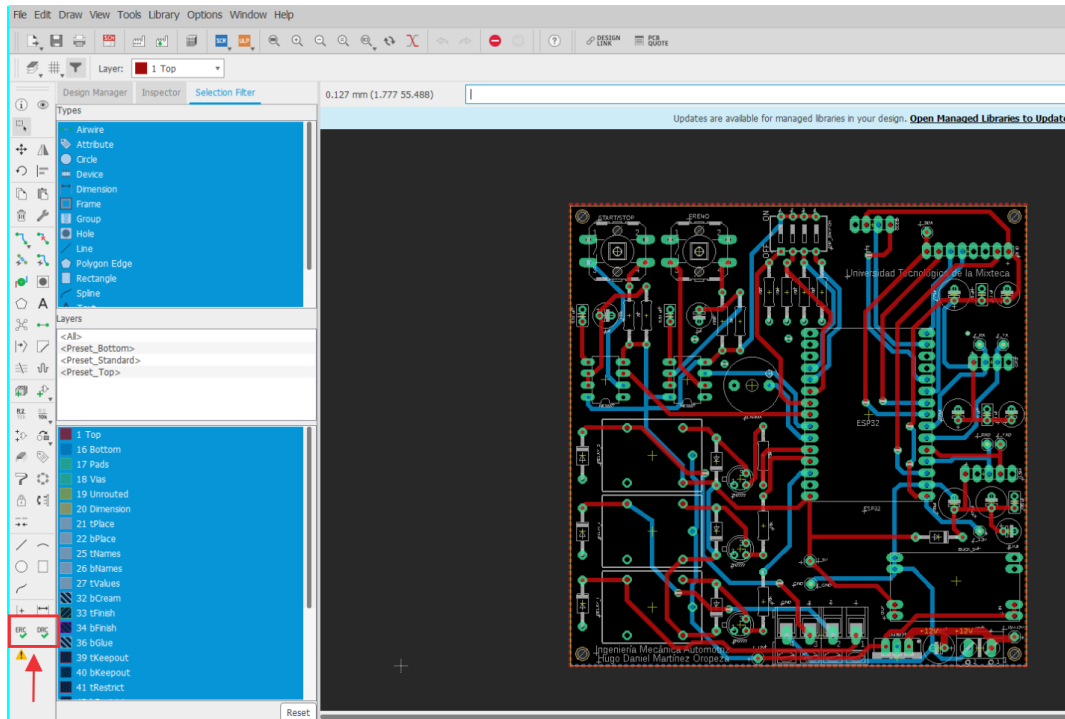


Figura B.18. Iconos DRC y ERC.

Una vez que se comprobaron y corrigieron los errores del diagrama esquemático y de las reglas de diseño de la PCB, se obtuvo el diseño final de la PCB como se muestra en las Figuras B.19 y B.20, procediendo con el envío para su fabricación.

Una vez que se obtuvo la PCB de forma física, se prosiguió a soldar los componentes electrónicos a la PCB, para comenzar con las pruebas del presente proyecto de tesis.

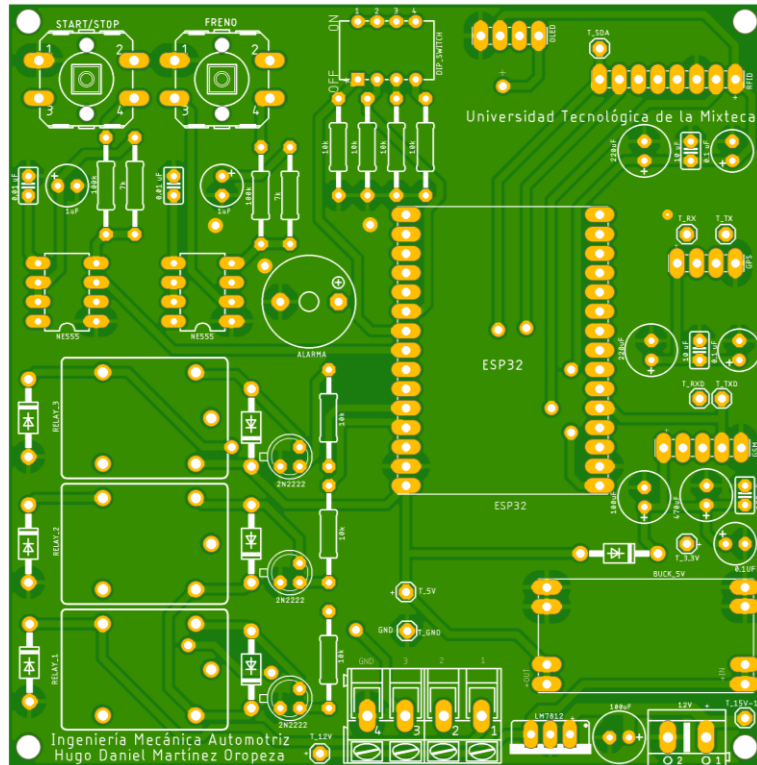


Figura B.19. PCB en la capa Top.

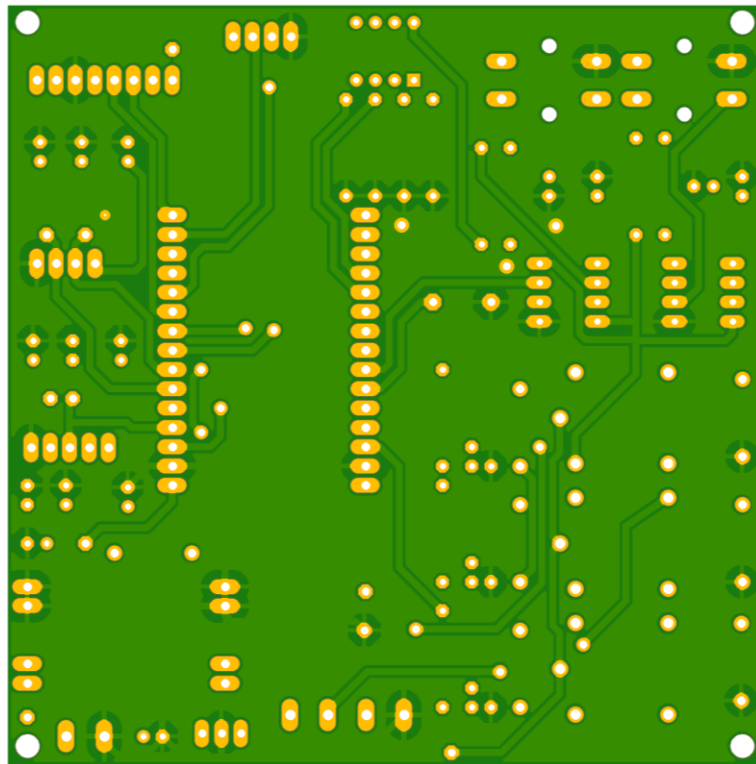


Figura B.20. PCB en la capa Bottom.

La Tabla B.1 muestra la lista de componentes del SESA-NiV.

Tabla B.1. Componentes que integran al SESA-NiV.

Componentes	Cantidad
Módulo ESP32	1
Módulo GPS	1
Módulo GSM	1
Módulo RFID	1
Pantalla OLED	1
Dip Swich de 4 posiciones	1
Push Button	2
Relevador SPDT	3
Convertidor Buck DC/DC 12-5V	1
Regulador L7812CV	1
Buzzer	1
NE555	2
2N2222	3
Bornera 4 terminales	1
Bornera 2 terminales	1
Diodo rectificador	7
Resistencia 10k Ω	7
Resistencia 7k Ω	2
Resistencia 100k Ω	2
Header macho 2.54mm	31
Capacitor electrolítico 100uF/25V	3
Capacitor electrolítico 220uF/25V	1
Capacitor electrolítico 470uF/25V	1
Capacitor cerámico 10nF (103)	2
Capacitor cerámico 100nF (104)	3
Capacitor de tantalio 0.1uF	3
Cable Dupont Hembra-Hembra	16

Anexo C. Diseño de la Carcasa SESA-NiV

A continuación, se describe el procedimiento para el diseño de la caja. En primer lugar, se tomaron las medidas del PCB para poder realizar sus soportes y de los módulos para considerar la altura de la caja. Una vez que tomaron las medidas se analizó la dispersión de los módulos para su integridad y tener así un sistema compacto.

Para el diseño de las piezas de la caja se utilizó SolidWorks, para ello lo primero fue realizar la extrusión saliente de la caja y una extrusión de corte para dejar la caja hueca; posteriormente, se diseñaron los postes con la distancia correcta, de esta manera se atornilló la placa a la base y se realizaron los últimos cortes de apertura para conectar el sistema a una fuente de 12V, manipular los botones de freno y Start/Stop del sistema y empotrar el módulo GPS (véase Figura C.1).

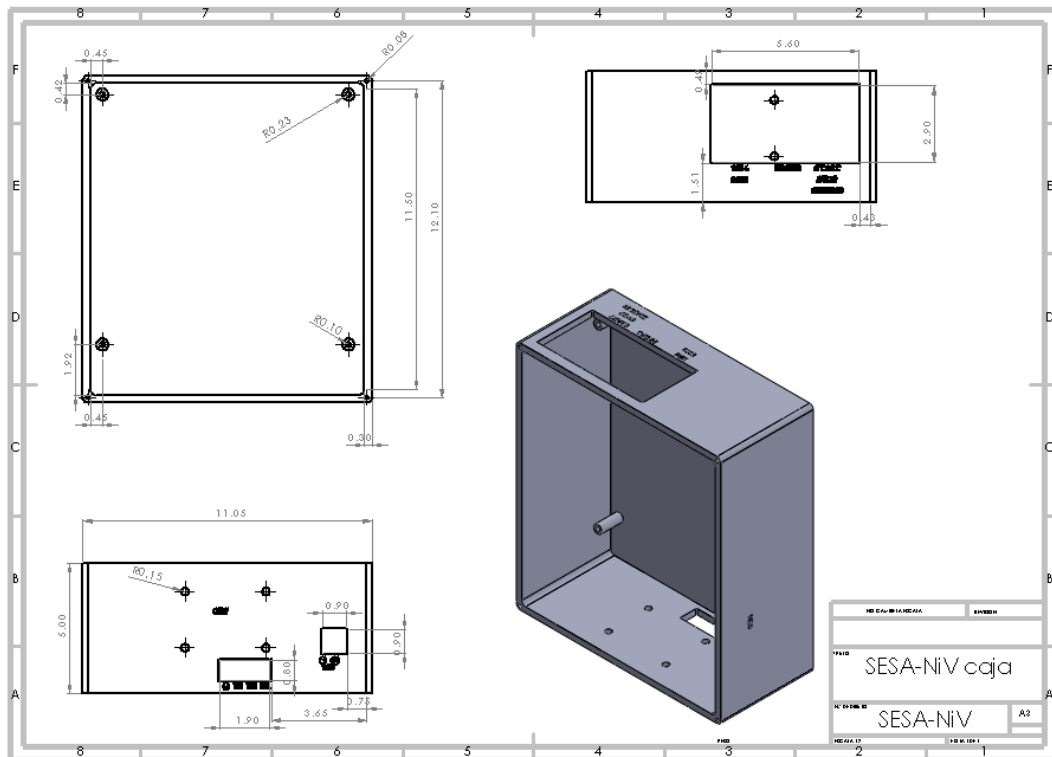


Figura C.1. Dibujo técnico de la carcasa.

Para el diseño de la tapa se rectificaron las medidas teniendo como punto de referencia el perímetro externo de la caja. Se realizó la dispersión adecuada

para contener al módulo RFID y a la pantalla OLED. De esta forma se realiza la extrusión saliente de la tapa y luego la extrusión corte; y al final, se realizan los cortes para el empotramiento de los módulos mencionados (véase Figura C.2).

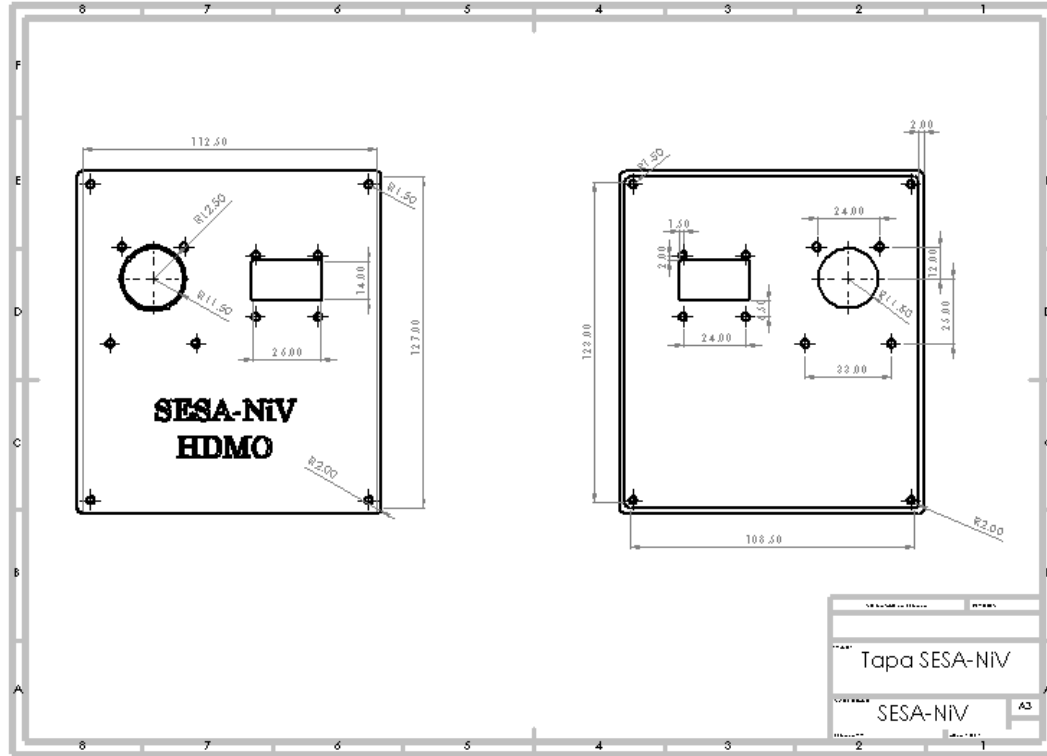


Figura C.2. Dibujo técnico de la tapa.

Se realizó el ensamblaje de ambas piezas para tener una previsualización final de la caja para el SESA-NiV como se observa en la Figura C.3. Una vez que se obtuvo el diseño completo de la pieza, se realizó la impresión en 3D en las instalaciones del Instituto de Electrónica y Mecatrónica de la Universidad Tecnológica de la Mixteca. Se utilizó filamento PLA color café como se observa en la Figura C.4. Una vez que se imprimieron ambas piezas, se lijaron y se pintaron de color negro. Finalmente, se montaron los módulos y la placa base del SESA-NiV (véase Figura C.5).

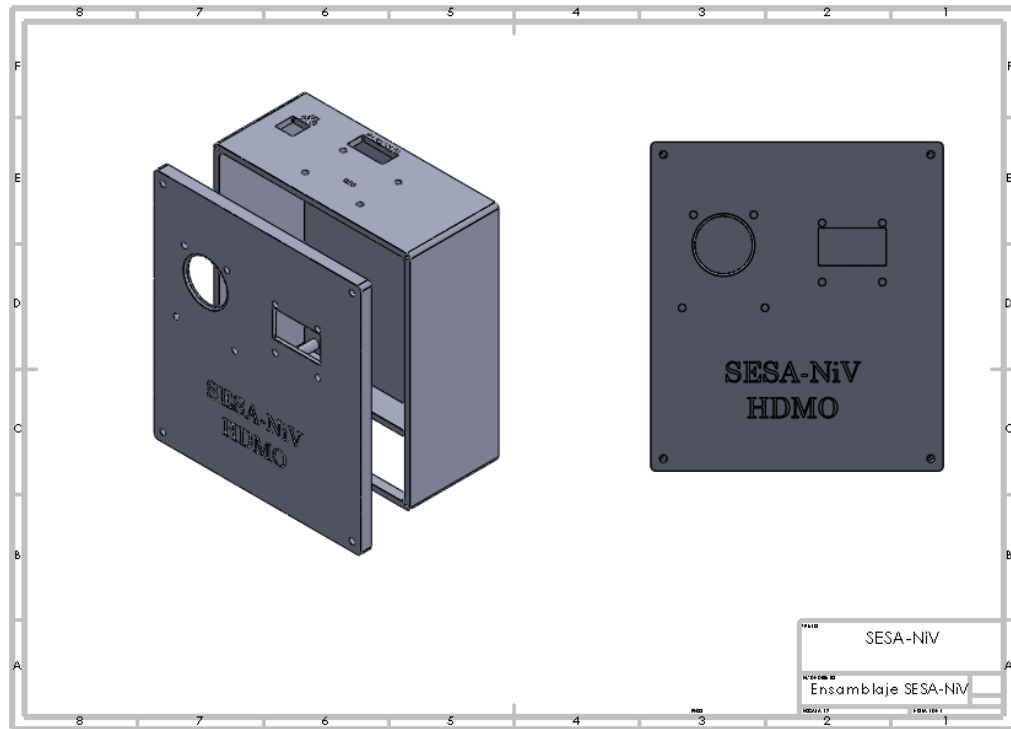


Figura C.3. Ensamblaje de la carcasa.



Figura C.4. Impresión 3D de la caja y tapa.



Figura C.5. Sistema final SESA-NiV.

Anexo D. Código del SESA-NiV en Arduino

Programación de la pantalla OLED

A continuación, se describe el código que se utilizó para establecer la comunicación de la pantalla OLED SSD1306 con el módulo ESP32. Primeramente, se utilizan las siguientes librerías.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

La librería Wire.h permite comunicar al microcontrolador por medio del módulo interno I2C, ya sea como maestro o como esclavo, recibiendo peticiones y respondiendo datos (Arduino N. Z., 2023). La librería Adafruit_GFX.h proporciona la sintaxis y un conjunto de funciones gráficas para pantallas LCD u OLED, además de matrices led (Dynda, 2023). Finalmente, la librería Adafruit_SSD1306.h contiene los controladores oled SSD1306 monocromáticos de 128x64 y 128x32.

Se define el ancho y el largo de la pantalla OLED en pixeles, y se crea la instancia display de la pantalla SDD1306, la cual se conecta por I2C (SDA, SCL).

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

Se inicializa la pantalla y se determina el tamaño del texto teniendo en cuenta que la instrucción SetTextSize(X), donde el valor de X representa 10 pixeles de la pantalla, es decir que cada letra del texto ocupará 10 pixeles; se elige el color del texto con la instrucción display.setTextColor(WHITE). Cabe destacar que el color del texto depende del tipo de pantalla OLED.

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.setTextSize(1);
display.setTextColor(WHITE);
```

Al inicializar la comunicación con el módulo ESP32 se borra el contenido de la pantalla y se establece el posicionamiento inicial del texto; el primer espacio se declara para las filas y el segundo espacio se declara para las columnas una

vez declarado, se establece el texto y se envía la instrucción para que se pueda visualizar en la pantalla.

```
display.clearDisplay();
display.display();
display.setCursor(0, 0);
display.clearDisplay();
display.setCursor(0, 0);
display.println("Bienvenido");
display.setCursor(0, 16);
display.println("Ingrese la llave ");
display.display();
```

El código completo para comunicar la pantalla OLED con el módulo ESP32 es el siguiente.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.setTextSize(1);
display.setTextColor(WHITE);
display.clearDisplay();
display.display();
display.setCursor(0, 0);
display.clearDisplay();
display.setCursor(0, 0);
display.println("Bienvenido");
display.setCursor(0, 16);
display.println("Ingrese la llave ");
display.display();
```

Módulo RFID RC522

A continuación, se describe el código para establecer la correcta comunicación del módulo RFID RC522 con el módulo ESP32, para ello se utilizan las siguientes librerías:

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

La primera librería integra el protocolo de datos en serie síncrono que utilizan los microcontroladores para comunicarse con uno o más dispositivos periféricos en distancias cortas (Arduino, SPI, 2023) y la segunda librería es para leer o escribir una tarjeta o etiqueta RFID, utiliza la interfaz ISO/IEC14443A/MIFARE (GithubCommunity, 2023).

Se definen las conexiones de las terminales RST (GPIO15) y SS (GPIO5) del módulo RFID y se crea la instancia `mfr522` del `MFRC522`.

```
#define RST_PIN 15
#define SS_PIN 5

MFRC522 mfr522(SS_PIN, RST_PIN);
```

Para identificar al usuario se debe realizar un UID (*User Identification*) mediante un arreglo y declararlo, ambos en arreglos de 4 bytes; con ello se define el código de identificación de la tarjeta en hexadecimal (0xFC, 0x6F, 0x17, 0x33).

```
byte lectorUID[4];
byte Usuario1[4] = {0xFC, 0x6F, 0x17, 0x33};
```

Se inicializa la comunicación SPI y el módulo lector `mfr522`.

```
SPI.begin();
mfr522.PCD_Init();
```

Una vez que se inicializa el `mfr522`, el programa inicia la lectura de las tarjetas. Si el lector `MFRC522` no logra detectar alguna tarjeta debido a que se encuentra lejos de su rango de lectura o no es capaz de obtener datos de la tarjeta, retorna a `loop()`.

```
if (!mfr522.PICC_IsNewCardPresent())
    return;
if (!mfr522.PICC_ReadCardSerial())
    return;
```

Si una tarjeta se encuentra cerca del rango de lectura y es detectada por el lector `mfr522`, el programa realiza un ciclo `for` para leer los diez bytes del UID (hexadecimal 0x10), dichos bytes se almacenan en el arreglo llamado `lectorUID[i]`; de lo contrario, no se hace nada. Con esto se lee el código de la tarjeta.

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

```

if (mfr522.uid.uidByte[i] < 0x10) { } else { }
lectorUID[i] = mfr522.uid.uidByte[i];
}

```

Para validar que la tarjeta tenga el código correcto, la función `comparaUID` se encarga de validar al usuario, para ello, compara los bytes leídos del UID (arreglo `lectura[]`) con los bytes del UID autorizado (arreglo `usuario[]`). Se realiza un ciclo `for` para recorrer los bytes y comparar los arreglos; si los arreglos son iguales se regresa un valor verdadero y si, por el contrario, son diferentes se regresa un valor falso.

```

boolean comparaUID(byte lectura[], byte usuario[])
{
for (byte i = 0; i < mfr522.uid.size; i++) {
if (lectura[i] != usuario[i])
return (false);
}
return (true);
}

```

Si se valida al usuario, se asigna un valor 1 a la variable `band_rfid_valida` para la autorización, si no es así, se asigna un valor 1 a la variable `band_rfid_No_valida`.

```

if (comparaUID(lectorUID, Usuario1)) {
band_rfid_valida = 1;
} else {
band_rfid_No_valida = 1;
}
mfr522.PICC_HaltA(); // detiene la comunicación con tarjeta
}

```

El código completo para controlar el módulo RFID RC522 con el módulo ESP32 se presenta a continuación.

```

#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 15
#define SS_PIN 5
MFRC522 mfr522(SS_PIN, RST_PIN);
byte lectorUID[4];
byte Usuario1[4] = { 0xFC, 0x6F, 0x17, 0x33 };

void setup() {
SPI.begin();

```

```

mfr522.PCD_Init();
}

void loop() {
  if (!mfr522.PICC_IsNewCardPresent())
    return;
  if (!mfr522.PICC_ReadCardSerial())
    return;
  for (byte i = 0; i < mfr522.uid.size; i++) {
    if (mfr522.uid.uidByte[i] < 0x10) {
    } else {
    }
    lectorUID[i] = mfr522.uid.uidByte[i];
  }
  if (comparaUID(lectorUID, Usuario1)) {
    band_rfid_valida = 1;
  } else {
    band_rfid_No_valida = 1;
  }
  mfr522.PICC_HaltA();
}

boolean comparaUID(byte lectura[], byte usuario[]) {
  for (byte i = 0; i < mfr522.uid.size; i++) {
    if (lectura[i] != usuario[i])
      return (false);
  }
  return (true);
}

```

Módulo SIM800L

El código que se utilizó para establecer la comunicación del módulo SIM800L con el módulo ESP32, es el siguiente:

```

#define rxPin 4
#define txPin 2
#define BAUD_RATE 9600
#define rele 13

```

Se declara la variable `mensaje_sms` de tipo `string` para almacenar los datos en el ESP32.

```

String mensaje_sms;

```

Se inicializa la comunicación y se configura la velocidad serial del módulo SIM800L especificando las terminales Rx y Tx. Se habilita el envío de SMS de texto enviando la cadena “AT+CMGF=1”.

```
pinMode(rele, OUTPUT);  
Serial.begin(9600);  
sim800.begin(115200, SERIAL_8N1, rxPin, txPin);  
sim800.println("AT+CMGF=1");
```

Para controlar al actuador se emplean las siguientes instrucciones.

```
if (sim800.available()) {  
  mensaje_sms = sim800.readString();  
  if (mensaje_sms.indexOf("ON") >= 0)  
    digitalWrite(led, HIGH);  
  if (mensaje_sms.indexOf("OFF") >= 0)  
    digitalWrite(led, LOW);  
}
```

El código completo para establecer la comunicación del módulo SIM800L con el módulo ESP32 es el siguiente.

```
#define rxPin 4  
#define txPin 2  
#define BAUD_RATE 9600  
#define led 23  
String mensaje_sms;  
  
void setup() {  
  pinMode(led, OUTPUT);  
  Serial.begin(9600);  
  sim800.begin(115200, SERIAL_8N1, rxPin, txPin);  
  delay(1000);  
  sim800.println("AT+CMGF=1");  
  delay(100);  
}  
  
void loop() {  
  if (sim800.available()) {  
    mensaje_sms = sim800.readString();  
    if (mensaje_sms.indexOf("ON") >= 0) {  
      digitalWrite(led, HIGH);  
      delay(1000); }  
    if (mensaje_sms.indexOf("OFF") >= 0) {  
      digitalWrite(led, LOW);  
      delay(1000); }  
  }  
}
```

Módulo GPS Neo-6m-7m

Para establecer la comunicación del módulo GPS Neo-6m-7m con el módulo ESP32 se emplearon dos librerías.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
```

La primera permite analizar flujos de datos NMEA y proporciona métodos compactos para extraer la posición, la fecha, la hora, la altitud, la velocidad y el rumbo de los dispositivos GPS (Hart, 2023), mientras que la segunda librería permite la comunicación serie con otros dispositivos o sistemas (Arduino, 2023).

Se declaran las terminales del receptor (GPIO17) y del transmisor (GPIO16) para la comunicación entre el módulo ESP32 y el módulo GPS, así como la velocidad de transferencia en datos, en 9,600 baudios.

```
static const int RXPin = 16, TXPin = 17;
static const uint32_t GPSBaud = 9600;
```

Se inicializa el puerto serie a la velocidad de transferencia de datos definida por GPSBaud.

```
ss.begin(GPSBaud);
```

Para leer la información del módulo GPS se emplean las siguientes instrucciones.

```
while (ss.available() > 0) {
  gps.encode(ss.read());
  display.clearDisplay();
  display.display();
  display.setCursor(40, 0);
  display.println("LATITUD");
  display.setCursor(28, 16);
  display.println(gps.location.lat(), 9);
  display.setCursor(40, 40);
  display.println("LONGITUD");
  display.setCursor(28, 56);
  display.println(gps.location.lng(), 8);
  display.display();
  delay(10000); }
```

A continuación, se presenta el código completo para establecer la comunicación del módulo GPS Neo-6m-7m con el módulo ESP32.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const int RXPin = 16, TXPin = 17;
static const uint32_t GPSBaud = 9600;

void setup() {
  ss.begin(GPSBaud);
}

void loop() {
  while (ss.available() > 0) {
    gps.encode(ss.read());
    if (gps.location.isUpdated()) {
      display.clearDisplay();
      display.display();
      display.setCursor(40, 0);
      display.println("LATITUD");
      display.setCursor(28, 16);
      display.println(gps.location.lat(), 9);
      display.setCursor(40, 40);
      display.println("LONGITUD");
      display.setCursor(28, 56);
      display.println(gps.location.lng(), 8);
      display.display();
      delay(10000);
    }
  }
}
```

A continuación, se presenta el código completo del SESA-NiV con el módulo ESP32.

```
#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <BluetoothSerial.h>

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <SPI.h>
```



```
#include <MFRC522.h>

#define RST_PIN 15

#define SS_PIN 5

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define rxPin 4

#define txPin 2

#define BAUD_RATE 9600

int rele1 = 13;

int rele2 = 12;

int rele3 = 14;

int contador = 0;

float sinVal;

int toneVal;

int BAN = 0;

int ban_autorizacion_enc = 0;

int DIP1 = 33;

int DIP2 = 32;

int DIP3 = 35;

int DIP4 = 34;

int BAN_dip = 0;

int BAN_interruptores = 0;

int BAN_inte = 0;

int band_rfid_valida = 0;

int band_rfid_No_valida = 0;

int bandera_gps = 0;

bool COD1, COD2, COD3, COD4;

bool StartStopPress = 0;

bool FrenoPress = 0;

int opc = 0;
```

```
static const int RXPin = 16, TXPin = 17;

static const uint32_t GPSBaud = 9600;

String smsStatus, senderNumber, receivedDate, msg;

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

char Lec_BT = 0;

MFRC522 mfrc522(SS_PIN, RST_PIN);

byte lectorUID[4];

byte Usuario1[4] = { 0xFC, 0x6F, 0x17, 0x33 };

void Freno() {

  FrenoPress = digitalRead(26) == 1;

}

void StartStop() {

  StartStopPress = digitalRead(25) == 1;

  opc = contador;

}

void setup() {

  delay(3000);

  GSM_1();

  delay(1000);

  attachInterrupt(digitalPinToInterrupt(26), Freno, CHANGE);

  attachInterrupt(digitalPinToInterrupt(25), StartStop, CHANGE);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

  display.setTextSize(2);

  display.setTextColor(WHITE);

  pinMode(rele1, OUTPUT);

  pinMode(rele2, OUTPUT);

  pinMode(rele3, OUTPUT);

  pinMode(27, OUTPUT);

  pinMode(33, INPUT);

  pinMode(32, INPUT);
```

```
pinMode(35, INPUT);

pinMode(34, INPUT);

digitalWrite(rele1, LOW);

digitalWrite(rele2, LOW);

digitalWrite(rele3, LOW);

SPI.begin();

mfr522.PCD_Init();

sim800.begin(9600, SERIAL_8N1, rxPin, txPin); //inicializamos sim800

delay(1000);

sim800.println("AT+CMGF=1");

delay(100);

sim800.println("AT+CNMI=1,2,0,0,0");

SerialBT.begin("SESA-NiV HUGO");

Pan_Inicio();

GPS_1();

esp_sleep_enable_ext0_wakeup(GPIO_NUM_25, 1);

}

void loop() {

  Bluetooth_Lectura_2();

  Bluetooth_Lectura();

  tarjeta_rfid();

  dip_switch();

  GPS_1();

  GSM_1();

  if (band_rfid_valida == 1 && Lec_BT == 'A' && BAN_dip == 1) {

    ban_autorizacion_enc = 1;

    SerialBT.write(ban_autorizacion_enc);

    tarjeta_valida();

    GSM_1();

    BAN = 0;

  }

}
```

```
    }  
    if (band_rfid_No_valida == 1) {  
        tarjeta_No_Valida();  
        GSM_1();  
        BAN = 1;  
    }  
}  
  
void Pan_Inicio() {  
    display.clearDisplay();  
    display.setCursor(7, 0);  
    display.println("Bienvenido");  
    display.setCursor(15, 20);  
    display.println("Ingrese ");  
    display.setCursor(10, 40);  
    display.println("la llave ");  
    display.display();  
    GPS_1();  
    GSM_1();  
}  
  
void Bluetooth_Lectura() {  
    if (SerialBT.available()) {  
        char c = SerialBT.read();  
        Lec_BT = c;  
        GSM_1();  
    }  
}  
  
void Bluetooth_Lectura_2() {  
    if (SerialBT.available()) {  
        char c = SerialBT.read();  
        Lec_BT = c;
```

```
    opciones_BT();

    GPS_1();

    GSM_1();

}

}

void opciones_BT() {

    GSM_1();

    if (Lec_BT == 'B' && FrenoPress == 1) {

        digitalWrite(rele1, LOW);

        digitalWrite(rele2, LOW);

        digitalWrite(rele3, LOW);

        display.clearDisplay();

        display.display();

        display.setCursor(50, 0);

        display.println("Se");

        display.setCursor(0, 16);

        display.println("inmovilizo");

        display.setCursor(50, 32);

        display.println("el");

        display.setCursor(5, 48);

        display.println("Vehiculo");

        display.display();

        ini();

        esp_deep_sleep_start();

        GPS_1();

        GSM_1();

    }

    if (Lec_BT == 'C') {

        GPS_1();

        sim800.println("AT+CMGS=\"+529535384418\"");
```

```
delay(1000);
sim800.println("LATITUD Y LONGITUD");
sim800.print("");
sim800.print(gps.location.lat(), 6);
sim800.print(" ");
sim800.print(gps.location.lng(), 6);
sim800.print("");
sim800.write(0x1A);
display.clearDisplay();
display.display();
display.setCursor(15, 0);
display.println("LATITUD");
display.setCursor(0, 16);
display.println(gps.location.lat(), 7);
display.setCursor(10, 32);
display.println("LONGITUD");
display.setCursor(0, 48);
display.println(gps.location.lng(), 8);
display.display();
}
if (Lec_BT == 'D' && band_rfid_valida == 1) {
  do {
    Accesorios();
    GPS_1();
    GSM_1();
  } while (Lec_BT == 'D' && band_rfid_valida == 1);
}
if (Lec_BT == 'E' && band_rfid_valida == 1) {
  do {
    v_on();
```

```
    GPS_1();

    GSM_1();

    } while (Lec_BT == 'E' && band_rfid_valida == 1);

}

if (Lec_BT == 'F' && band_rfid_valida == 1) {

    do {

        v_off();

        GPS_1();

        GSM_1();

    } while (Lec_BT == 'F' && band_rfid_valida == 1);

}

if (Lec_BT == 'G') {

    setup();

    GPS_1();

    GSM_1();

    band_rfid_valida == 0;

    contador = 0;

}

if (Lec_BT == 'H') {

    ESP.restart();

}

}

void tarjeta_valida() {

    alarma_No_activada();

    mensaje("Bienvenido", 1);

    delay(1000);

    GSM_1();

    do {

        eleccion(opc);

        GPS_1();
```

```
GSM_1();
} while (band_rfid_valida == 1);
}
void tarjeta_No_Valida() {
    alarma_activada();
    mensaje("Tarjeta No Valida", 0);
    delay(1000);
    Pan_Inicio();
    GPS_1();
    GSM_1();
}
void alarma_activada() {
    for (int x = 0; x < 180; x++) {
        sinVal = (sin(x * (3.1412 / 180)));
        toneVal = 2000 + (int(sinVal * 1000));
        tone(27, toneVal);
        delay(2);
    }
}
void alarma_No_activada() {
    for (int x = 0; x < 180; x++) {
        sinVal = (sin(x * (3.1412 / 180)));
        toneVal = 2000 + (int(sinVal * 1000));
        tone(27, LOW);
        delay(2);
    }
}
void eleccion(int op) {
    GSM_1();
    switch (op) {
```



```
    case 0:
        GPS_1();
        GSM_1();
        Bienvenida();
        break;
    case 1:
        GPS_1();
        GSM_1();
        Accesorios();
        break;
    case 2:
        GPS_1();
        GSM_1();
        v_on();
        break;
    case 3:
        GPS_1();
        GSM_1();
        v_off();
        break;
    }
}

void ini(void) {
    display.clearDisplay();
    display.display();
    GSM_1();
}

void Bienvenida(void) {
    Lec_BT = 0;
    delay(1000);
```

```
    mensaje("Bienvenido", 0);  
  
    Bluetooth_Lectura_2();  
  
    GSM_1();  
  
    contador = 1;  
}  
  
void Accesorios(void) {  
  
    delay(1000);  
  
    digitalWrite(rele1, HIGH);  
  
    digitalWrite(rele2, HIGH);  
  
    digitalWrite(rele3, LOW);  
  
    display.clearDisplay();  
  
    display.setCursor(5, 0);  
  
    display.println("Accesorios");  
  
    display.setCursor(55, 20);  
  
    display.println("e ");  
  
    display.setCursor(10, 40);  
  
    display.println("Ignicion ");  
  
    display.display();  
  
    Bluetooth_Lectura_2();  
  
    GSM_1();  
  
    contador = 2;  
}  
  
void v_on(void) {  
  
    delay(1000);  
  
    digitalWrite(rele1, HIGH);  
  
    digitalWrite(rele2, HIGH);  
  
    digitalWrite(rele3, HIGH);  
  
    display.clearDisplay();  
  
    display.setCursor(10, 0);  
  
    display.println("Vehiculo");
```

```
display.setCursor(10, 30);

display.println("encendido");

display.display();

Bluetooth_Lectura_2();

GSM_1();

contador = 3;

}

void v_off(void) {

    delay(1000);

    display.clearDisplay();

    display.setCursor(15, 0);

    display.println("Apagando");

    display.setCursor(10, 30);

    display.println("vehiculo");

    display.display();

    Bluetooth_Lectura_2();

    GSM_1();

    delay(1000);

    ini();

    if (FrenoPress == 1 && StartStopPress == 1) {

        digitalWrite(rele1, LOW);

        digitalWrite(rele2, LOW);

        digitalWrite(rele3, LOW);

        StartStopPress = 0;

        contador = 0;

        delay(1000);

        esp_deep_sleep_start();

    }

}

void mensaje(String ms, int ast) {
```

```
display.clearDisplay();

display.setCursor(7, 0);

display.println(ms);

if (ast == 1) {

    display.setCursor(55, 20);

    display.println("**");

}

display.display();

}

void GPS_1() {

    while (ss.available() > 0) {

        gps.encode(ss.read());

        if (gps.location.isUpdated()) {

            gps.location.lat();

            gps.location.lng();

            delay(10000);

        }

    }

}

void tarjeta_rfid(void) {

    if (!mfrc522.PICC_IsNewCardPresent())

        return;

    if (!mfrc522.PICC_ReadCardSerial())

        return;

    for (byte i = 0; i < mfrc522.uid.size; i++) {

        if (mfrc522.uid.uidByte[i] < 0x10) {

        } else {

        }

        lectorUID[i] = mfrc522.uid.uidByte[i];

    }

}
```

```
    if (comparaUID(lectorUID, Usuario1)) {
        band_rfid_valida = 1;
    }
    else {
        band_rfid_No_valida = 1;
    }
    mfrc522.PICC_HaltA();
}

boolean comparaUID(byte lectura[], byte usuario[]) {
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        if (lectura[i] != usuario[i])
            return (false);
    }
    return (true);
}

void dip_switch(void) {
    COD1 = digitalRead(33);
    COD2 = digitalRead(32);
    COD3 = digitalRead(35);
    COD4 = digitalRead(34);

    if (COD1 == HIGH && COD2 == HIGH && COD3 == HIGH && COD4 == HIGH) {
        BAN_dip = 1;
    } else {
        BAN_dip = 0;
    }
}

void GSM_1(void) {
    if (sim800.available()) {
        mensaje_sms = sim800.readString();

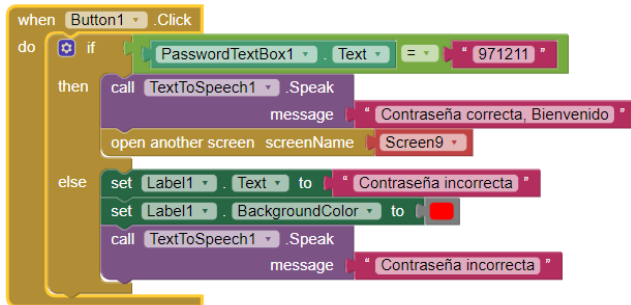
        GPS_1();
    }
}
```

```
if (mensaje_sms.indexOf("OFF") >= 0) {  
    digitalWrite(rele1, LOW);  
    digitalWrite(rele2, LOW);  
    digitalWrite(rele3, LOW);  
    delay(1000);  
    sim800.println("AT+CMGS="+529535384418);  
    delay(1000);  
    sim800.println("El vehiculo se inmovilizo");  
    sim800.write(0x1A);  
    delay(1000);  
    display.clearDisplay();  
    display.display();  
    display.setCursor(50, 0);  
    display.println("Se");  
    display.setCursor(0, 16);  
    display.println("inmovilizo");  
    display.setCursor(50, 32);  
    display.println("el");  
    display.setCursor(5, 48);  
    display.println("vehiculo");  
    display.display();  
    delay(7000);  
    ini();  
    esp_deep_sleep_start();  
}  
  
if (mensaje_sms.indexOf("GPSV") >= 0) {  
    sim800.println("AT+CMGS="+529535384418);  
    delay(1000);  
    sim800.println("LATITUD Y LONGITUD");  
    sim800.print("(");
```

```
sim800.print(gps.location.lat(), 6);  
  
sim800.print(" ");  
  
sim800.print(gps.location.lng(), 6);  
  
sim800.print(" ");  
  
sim800.write(0x1A);  
  
display.clearDisplay();  
  
display.display();  
  
display.setCursor(15, 0);  
  
display.println("LATITUD");  
  
display.setCursor(0, 16);  
  
display.println(gps.location.lat(), 7);  
  
display.setCursor(10, 32);  
  
display.println("LONGITUD");  
  
display.setCursor(0, 48);  
  
display.println(gps.location.lng(), 8);  
  
display.display();  
  
delay(1000);  
  
}  
  
}
```


Anexo E. Código del SESA-NiV en MIT App Inventor

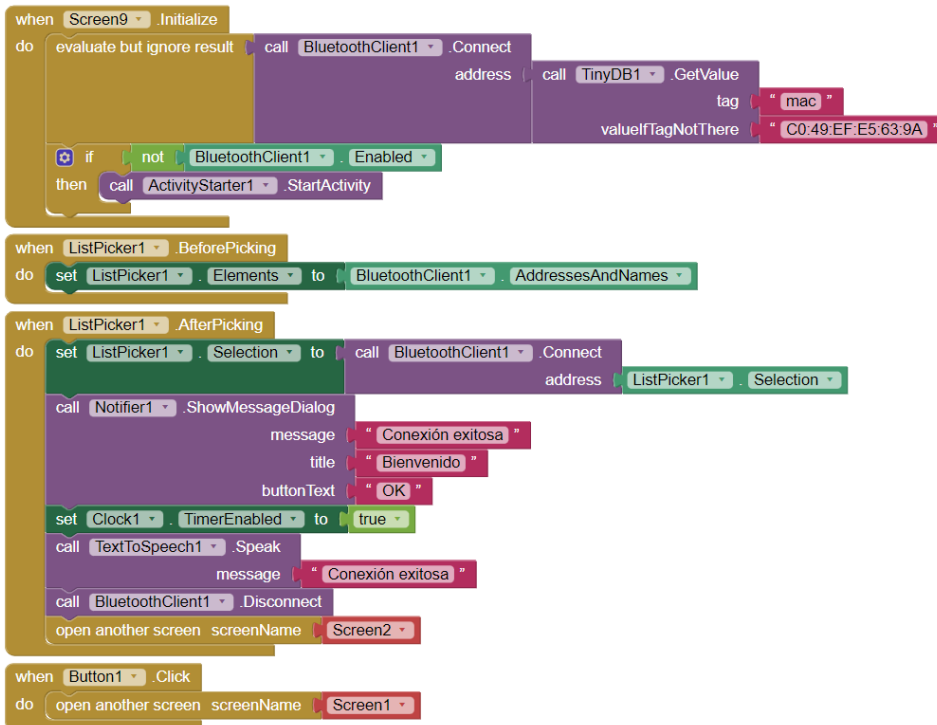
En este anexo se describe la codificación del SESA-NiV en MIT App Inventor. La Figura E.1 muestra el código de bloques de la pantalla Contraseña en MIT App Inventor.



```
when Button1 . Click
do
  if PasswordTextBox1 . Text = "971211"
  then
    call TextToSpeech1 . Speak
      message "Contraseña correcta. Bienvenido"
    open another screen screenName Screen9
  else
    set Label1 . Text to "Contraseña incorrecta"
    set Label1 . BackgroundColor to red
    call TextToSpeech1 . Speak
      message "Contraseña incorrecta"
```

Figura E.1. Código en MIT App Inventor para la pantalla Contraseña.

La Figura E.2 muestra el código en MIT App Inventor para la conexión Bluetooth entre el dispositivo móvil y el módulo ESP32.



```
when Screen9 . Initialize
do
  evaluate but ignore result call BluetoothClient1 . Connect
    address call TinyDB1 . GetValue
      tag "mac"
      valueIfTagNotThere "C0:49:EF:E5:63:9A"
  if not BluetoothClient1 . Enabled
  then call ActivityStarter1 . StartActivity

when ListPicker1 . BeforePicking
do
  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 . AfterPicking
do
  set ListPicker1 . Selection to call BluetoothClient1 . Connect
    address ListPicker1 . Selection
  call Notifier1 . ShowMessageDialog
    message "Conexión exitosa"
    title "Bienvenido"
    buttonText "OK"
  set Clock1 . TimerEnabled to true
  call TextToSpeech1 . Speak
    message "Conexión exitosa"
  call BluetoothClient1 . Disconnect
  open another screen screenName Screen2

when Button1 . Click
do
  open another screen screenName Screen1
```

Figura E.2. Código en MIT App Inventor para la Conexión Bluetooth.

La Figura E.3 muestra el código correspondiente en MIT App Inventor del menú principal.

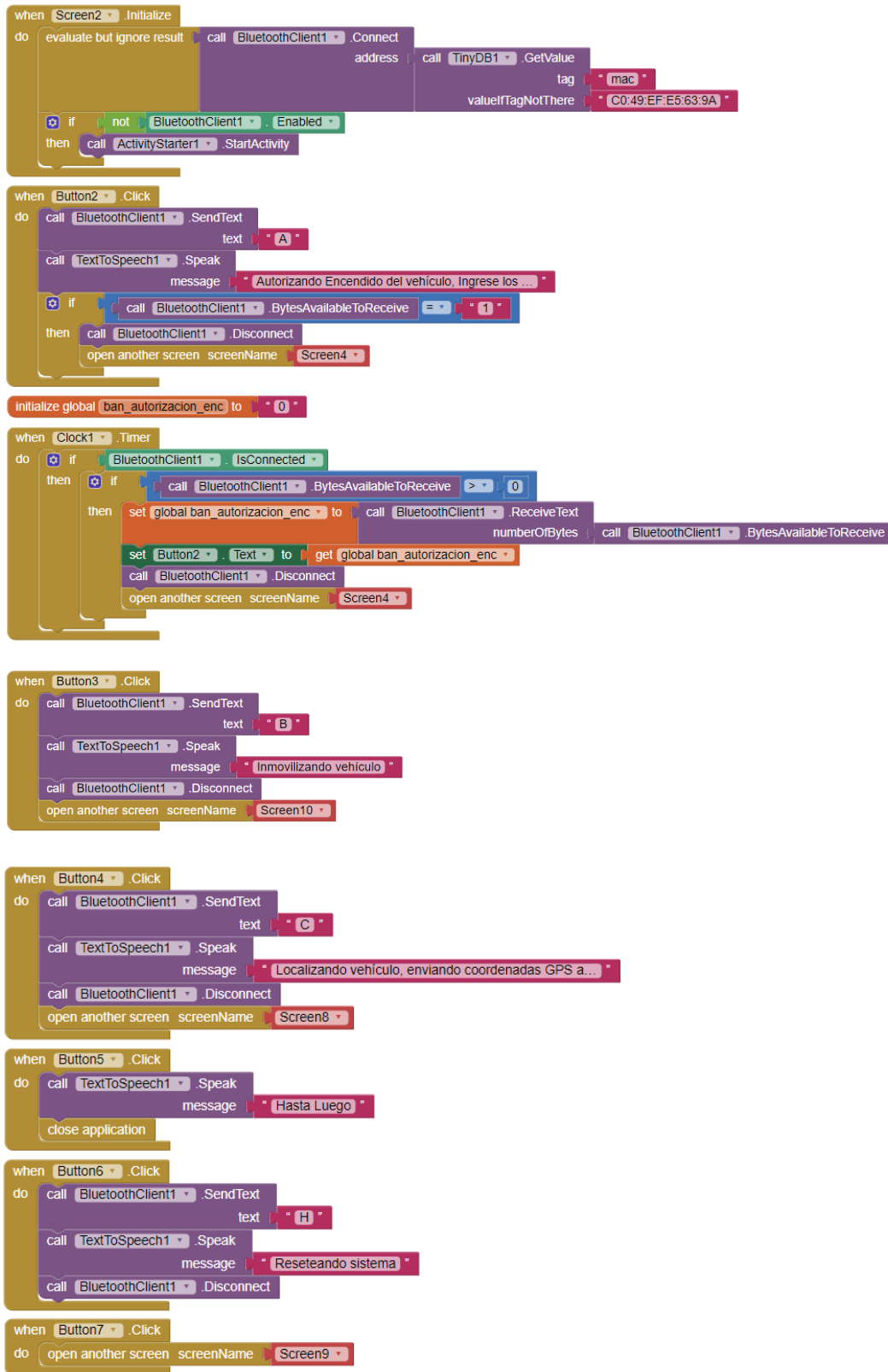


Figura E.3. Código en MIT App Inventor del Menú principal.

La Figura E.4 muestra el código correspondiente en MIT App Inventor de las opciones Accesorios e ignición, Encender vehículo y Apagar vehículo.

```
when Screen4.Initialize
do
  evaluate but ignore result call BluetoothClient1.Connect
  address call TinyDB1.GetValue
  tag "mac"
  valueIfTagNotThere "C0:49:EF:E5:63:9A"

when Button1.Click
do
  call BluetoothClient1.SendText
  text "D"
  call TextToSpeech1.Speak
  message "Accesorios e ignición"
  call BluetoothClient1.Disconnect
  open another screen screenName Screen5

when Screen5.Initialize
do
  evaluate but ignore result call BluetoothClient1.Connect
  address call TinyDB1.GetValue
  tag "mac"
  valueIfTagNotThere "C0:49:EF:E5:63:9A"

when Button1.Click
do
  call BluetoothClient1.SendText
  text "E"
  call TextToSpeech1.Speak
  message "Encendiendo vehículo"
  call BluetoothClient1.Disconnect
  open another screen screenName Screen6

when Screen6.Initialize
do
  evaluate but ignore result call BluetoothClient1.Connect
  address call TinyDB1.GetValue
  tag "mac"
  valueIfTagNotThere "C0:49:EF:E5:63:9A"

when Button1.Click
do
  call BluetoothClient1.SendText
  text "F"
  call TextToSpeech1.Speak
  message "Apagando vehículo"
  call BluetoothClient1.Disconnect
  open another screen screenName Screen7
```

Figura E.4. Código en MIT App Inventor de las opciones Accesorios e ignición, Encender vehículo y Apagar vehículo.

La Figura E.5 muestra el Código en MIT App Inventor para el mensaje de advertencia de presionar el freno y el botón Start/Stop.

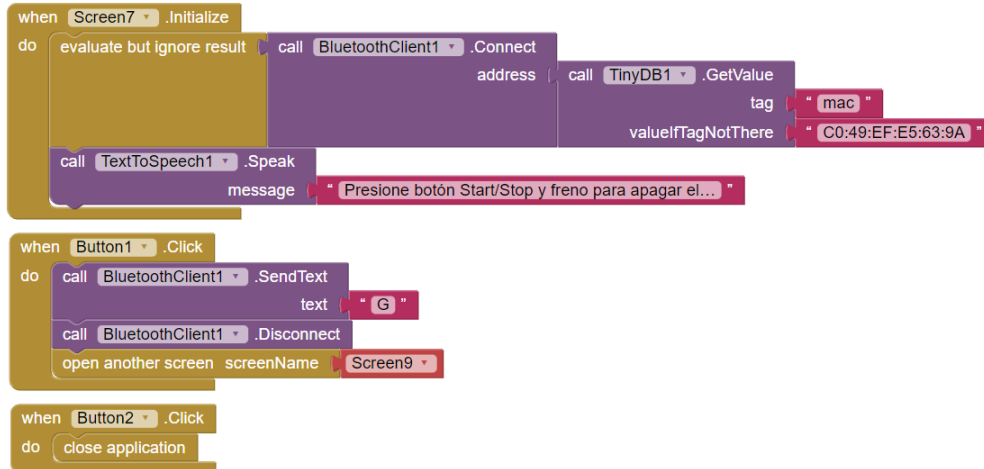


Figura E.5. Código en MIT App Inventor para el mensaje de advertencia de presionar el freno y el botón Start/Stop.

La Figura E.6 muestra la codificación en MIT App Inventor MIT App de la opción Inmovilizar vehículo.

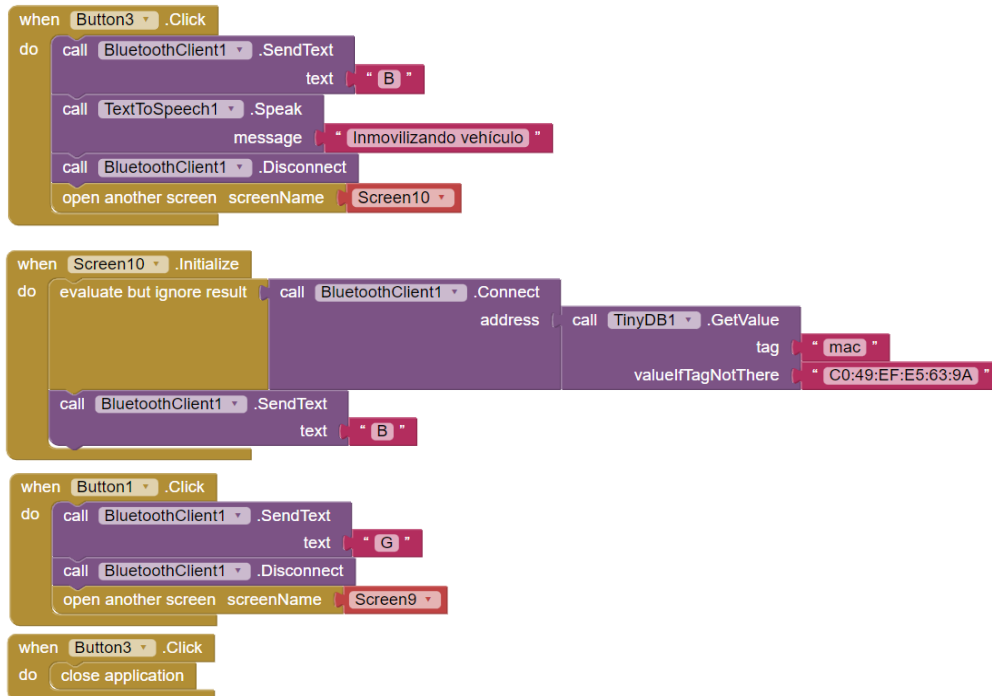


Figura E.6. Código en MIT App Inventor de la opción Inmovilizar vehículo.

La Figura E.7 muestra el código en MIT App Inventor de la opción Localizar vehículo.

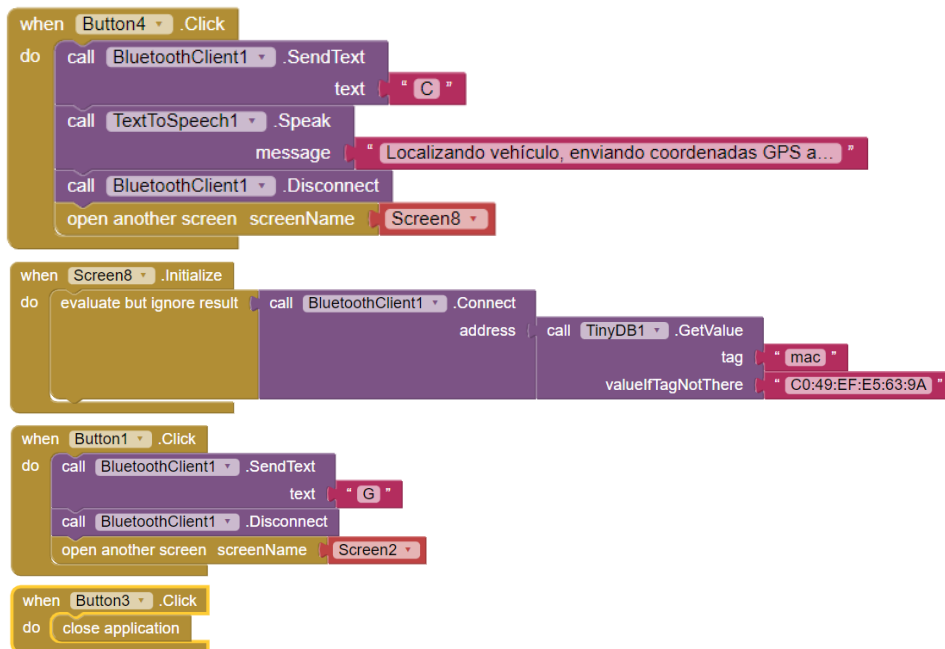


Figura E.7. Código en MIT App Inventor de la opción Localizar vehículo.

Finalmente, la Figura E.8 muestra la codificación de la opción Salir de la aplicación en MIT App Inventor.

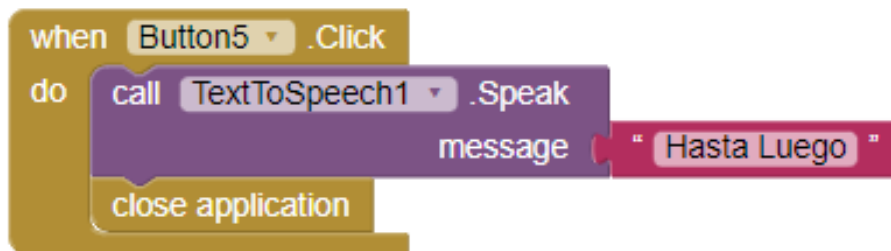


Figura E.8. Código de bloques de la opción Salir de la aplicación.

