



**Proyecto Previo a la Obtención del Título de Ingeniero
Automotriz**

INGENIERÍA AUTOMOTRIZ

Autor: Denys Miguel Velasco Uquillas

Tutor: Ing. Fernando Gómez Berrezueta, M.Sc.

**Metodología para Programación de un Módulo Inmovilizador
Usando el Programador UPA V1.3**

Certificado de Autoría

Yo, Denys Miguel Velasco Uquillas, declaro bajo juramento, que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada. Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.

Denys Miguel Velasco Uquillas

C.I.: 0925681587

Aprobación del Tutor

Yo, Fernando Gómez Berrezueta certifico que conozco al autor del presente trabajo siendo responsable exclusivo tanto de su originalidad y autenticidad, como de su contenido.

Ing. Fernando Gómez Berrezueta, M.Sc.

C.I.: 0103441846

Director de Proyecto

Dedicatoria

A Dios, por darme fortaleza en los momentos difíciles y por iluminar mi camino.

Dedico este trabajo a mis padres, por su apoyo incondicional, ejemplo de esfuerzo y valores que me impulsan a superarme día a día. También dedico este logro a mis docentes, quienes me transmitieron no solo conocimientos técnicos, sino también la pasión por la ingeniería automotriz.

Denys Velasco

Agradecimiento

Agradezco en primer lugar a Dios, por darme la fortaleza y perseverancia necesarias para culminar este proyecto. Extiendo mi gratitud a mi familia, que con su respaldo y motivación

constante me permitió seguir adelante a pesar de las dificultades.

Mi reconocimiento especial a mi tutor Fernando Gómez Berrezueta, cuyas orientaciones y sugerencias fueron fundamentales para alcanzar los objetivos planteados. Agradezco además

a los compañeros y amigos que colaboraron con ideas, datos y valiosas experiencias,

enriqueciendo el desarrollo de esta investigación.

Finalmente, gracias a la institución educativa que me brindó los recursos y el espacio para

crecer profesionalmente.

Denys Velasco

Índice General

Certificado de Autoría.....	iii
Aprobación del Tutor.....	iv
Dedicatoria.....	v
Agradecimiento.....	vi
Índice General.....	vii
Índice de Figuras.....	xi
Índice de Tablas	xiii
Resumen.....	xiv
Abstract	xv
Capítulo I	1
Antecedentes	1
1.1 Tema de Investigación	1
1.2 Planteamiento, Formulación y Sistematización del Problema.....	1
1.2.1 Planteamiento del Problema.....	1
1.2.2 Formulación del Problema	4
1.2.3 Sistematización del Problema.....	5
1.3 Objetivos de la Investigación.....	5
1.3.1 Objetivo General.....	5
1.3.2 Objetivos Específicos	5
1.4 Justificación y Delimitación de la Investigación	5
1.4.1 Justificación Teórica.....	6
1.4.2 Justificación Metodológica.....	7
1.4.3 Justificación Práctica	7
1.4.4 Delimitación Temporal	8

1.4.5	<i>Delimitación Geográfica</i>	8
1.4.6	<i>Delimitación del Contenido</i>	8
	Capítulo II	10
	Marco Referencial.....	10
2.1	Marco Teórico.....	10
2.1.1	<i>Conceptos Preliminares</i>	10
2.1.2	<i>Importancia de los Inmovilizadores de un Vehículo</i>	13
2.1.3	<i>Sistema Inmovilizador de Vehículos</i>	13
2.1.4	<i>Usos de los Scripts de Programación</i>	14
2.1.5	<i>Eficiencia y Programación de los Inmovilizadores de Motor</i>	15
2.2	Marco Conceptual.....	17
2.2.1	<i>Sistemas Automáticos de Inmovilizadores</i>	17
2.2.2	<i>Tipos de Inmovilizadores</i>	18
2.2.3	<i>Sistemas Inmovilizadores en Vehículos</i>	19
2.2.4	<i>Programador UPA (Universal Programmer Adapter)</i>	20
2.2.5	<i>Microcontroladores en el Sector de la Automoción</i>	22
2.2.6	<i>Programación Antirrobo</i>	22
2.2.7	<i>Principio del Programador Inmovilizador</i>	23
2.2.8	<i>Diferencia entre EPROM y EEPROM</i>	23
2.2.9	<i>Script</i>	24
2.2.10	<i>Editor Hexadecimal</i>	25
2.2.11	<i>Programador de Memorias y Microcontroladores (UPA)</i>	25
2.2.12	<i>Ventajas del UPA</i>	26
2.2.13	<i>Desventajas del UPA</i>	26
	Capítulo III.....	27

Conexión y Configuración Inicial del Programador UPA V1.3	27
3.1 Métodos.....	27
3.2 Tipo de Estudio	27
3.2.1 <i>Estudio Experimental</i>	27
3.2.2 <i>Estudio Aplicado</i>	28
3.3 Guía de Instalación	28
3.4 Conexión – Programador UPA V1.3	30
3.4.1 <i>Conexión Física</i>	30
3.4.3 <i>Verificación de Conexión</i>	31
3.4.4 <i>Preparación para Programación</i>	31
3.4.5 <i>Lectura / Escritura</i>	32
3.4.6 <i>Desconexión Segura</i>	33
3.4.7 <i>Configuración Inicial del Programador UPA V1.3</i>	33
3.5 Lectura y Programación de un Inmovilizador Chevrolet Sail con EEPROM 25C020	35
3.5.1 <i>Identificación y Retiro del Inmovilizador</i>	36
3.5.2 <i>Ubicación del Chip EEPROM 25C020</i>	36
3.5.3 <i>Conexión del Programador UPA V1.3 a la EEPROM</i>	37
3.5.4 <i>Lectura de la EEPROM</i>	38
3.5.5 <i>Edición del Archivo</i>	39
3.5.6 <i>Programación de la EEPROM</i>	39
3.5.7 <i>Reinstalación del Módulo</i>	39
3.5.8 <i>Prueba de Arranque del Vehículo</i>	40
Capítulo IV.....	41
Programación de un Módulo Inmovilizador	41
4.1 Descripción	41

4.2	Errores y Soluciones Comunes en la Programación del Módulo Inmovilizador con UPA V1.3.....	41
4.3	Proceso Detallado de Programación	41
4.3.1	<i>Preparación</i>	41
4.3.2	<i>Conexión</i>	42
4.3.3	<i>Configuración del Programador</i>	43
4.3.4	<i>Programación</i>	43
4.3.5	<i>Verificación</i>	44
4.4	Errores Comunes y Soluciones	44
4.5	Consejos para Evitar Errores	45
4.6	Recomendaciones Finales	45
	Conclusiones	46
	Recomendaciones	47
	Bibliografía	48

Índice de Figuras

Figura 1 <i>ECU de Automóviles</i>	2
Figura 2 <i>Redes en Vehículos y Pruebas de ECU</i>	4
Figura 3 <i>Programador UPA V1.3</i>	12
Figura 4 <i>Inmovilizador de un Vehículo</i>	13
Figura 5 <i>UPA – Software</i>	15
Figura 6 <i>Partes de un Inmovilizador</i>	18
Figura 7 <i>Funcionamiento del Transpondedor</i>	20
Figura 8 <i>Coberturas y Funciones Clave del Programador UPA</i>	21
Figura 9 <i>Tipos de Memoria</i>	24
Figura 10 <i>Programador de Memorias y Microcontroladores (UPA)</i>	26
Figura 11 <i>Instalación de Archivos del Programa</i>	29
Figura 12 <i>Activación de UUSP</i>	29
Figura 13 <i>Software y Drivers</i>	30
Figura 14 <i>Conexión Física</i>	31
Figura 15 <i>Verificación de Conexión</i>	31
Figura 16 <i>Verificación de Conexión</i>	32
Figura 17 <i>Zócalo ZIF</i>	32
Figura 18 <i>Lectura – Escritura</i>	33
Figura 19 <i>Memoria EEPROM 25C020</i>	35
Figura 20 <i>Chip 25C020 en la Placa del Inmovilizador</i>	36
Figura 21 <i>Elementos – Programación y Escritura</i>	37
Figura 22 <i>Armado – Realizar Pruebas</i>	37
Figura 23 <i>Selección del Tipo de Memoria</i>	38
Figura 24 <i>Lectura del Contenido de la Memoria</i>	39

Figura 25 <i>Cable USB Compatible</i>	42
Figura 26 <i>OBD-II Ubicación en el Vehículo</i>	42
Figura 27 <i>Ejecución del Programador</i>	43

Índice de Tablas

Tabla 1 <i>Características del UPA VI.3</i>	34
Tabla 2 <i>Características del UPA VI.3- Continuación</i>	35
Tabla 3 <i>Mapeo de Pines</i>	38
Tabla 4 <i>Errores y Soluciones en la Programación del UPA VI.3</i>	44

Resumen

La programación de un módulo inmovilizador automotriz mediante el programador UPA V1.3 constituye un procedimiento esencial para restablecer la operatividad del sistema de arranque y garantizar la correcta autenticación electrónica entre la unidad de control del motor (ECU) y la llave codificada. La metodología incluye la extracción y lectura de la memoria EEPROM del inmovilizador, la edición segura de sus registros binarios, y la posterior escritura del archivo calibrado para sincronizar códigos criptográficos. Este proceso requiere identificar la arquitectura de comunicación serial del módulo (SPI o I2C), así como aplicar procedimientos de verificación de checksum para asegurar la integridad de los datos. La utilización del programador UPA V1.3 optimiza estas tareas gracias a su compatibilidad con múltiples microcontroladores y su interfaz gráfica de usuario intuitiva. De esta forma, se contribuye a una reparación electrónica eficiente, evitando reemplazos innecesarios y reduciendo costos en el sistema inmovilizador automotriz.

Palabras Clave: inmovilizador, programación, EEPROM, UPA V1.3, sincronización.

Abstract

The programming of an automotive immobilizer module using the UPA V1.3 programmer is an essential procedure to restore the starting system's functionality and ensure correct electronic authentication between the engine control unit (ECU) and the coded key. The methodology includes extracting and reading the immobilizer's EEPROM memory, securely editing its binary records, and subsequently writing the calibrated file to synchronize cryptographic codes. This process requires identifying the serial communication architecture of the module (SPI or I2C), as well as applying checksum verification procedures to guarantee data integrity. The use of the UPA V1.3 programmer optimizes these tasks thanks to its compatibility with multiple microcontrollers and its intuitive graphical user interface. In this way, it contributes to efficient electronic repair, avoiding unnecessary replacements and reducing costs in the automotive immobilizer system.

Keywords: immobilizer, programming, EEPROM, UPA V1.3, synchronization.

Capítulo I

Antecedentes

1.1 Tema de Investigación

Metodología para programación de un módulo inmovilizador usando el programador UPA V1.3.

1.2 Planteamiento, Formulación y Sistematización del Problema

En la industria automotriz actual, la seguridad es una prioridad fundamental. Los módulos inmovilizadores son componentes cruciales que impiden el arranque del vehículo sin una llave adecuada, protegiendo así contra el robo. Sin embargo, la programación de estos módulos puede ser un desafío técnico significativo, especialmente para técnicos y profesionales que utilizan herramientas específicas como el programador UPA V1.3. La falta de una metodología clara y sistemática para la programación de módulos inmovilizadores puede llevar a errores, tiempos de reparación prolongados y costos adicionales.

1.2.1 Planteamiento del Problema

En la actualidad, la seguridad vehicular es un aspecto fundamental para prevenir el robo de automóviles. Los sistemas inmovilizadores electrónicos han surgido como una solución eficaz, utilizando tecnología avanzada para impedir el arranque del vehículo sin la llave correcta. Estos sistemas se basan en la programación de módulos inmovilizadores que interactúan con la ECU (Unidad de Control del Motor) y otros componentes electrónicos del vehículo. La programación de módulos inmovilizadores es una tarea compleja que requiere conocimientos especializados y herramientas específicas (Sandoval, 2022).

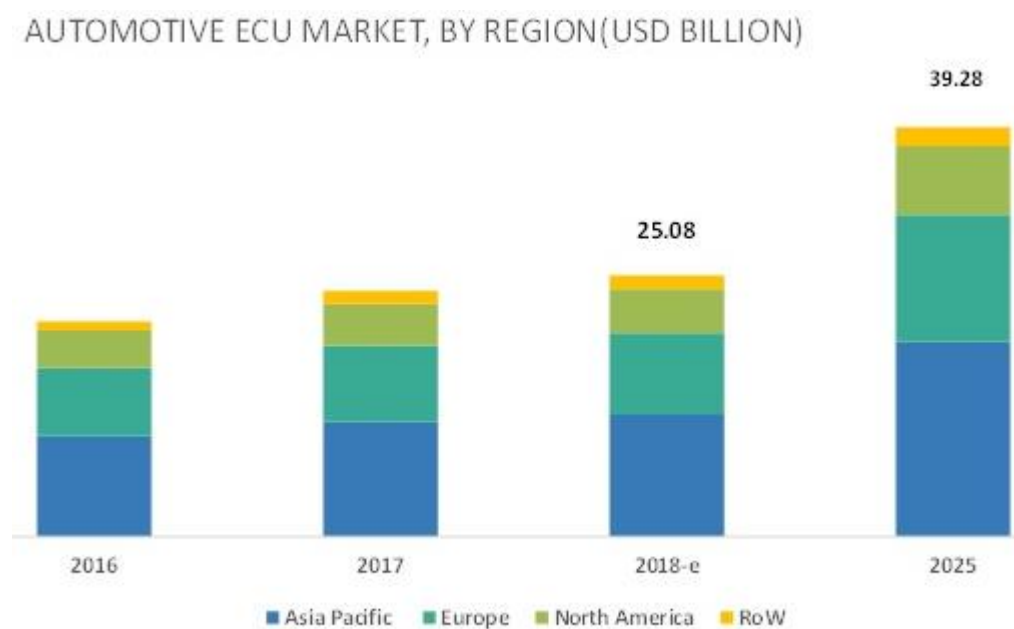
En muchos casos, los técnicos automotrices enfrentan desafíos significativos debido a la falta de una metodología clara y detallada para llevar a cabo este proceso de manera eficiente y segura.

El programador UPA V1.3 es una herramienta comúnmente utilizada para la programación de estos módulos. Sin embargo, la falta de una guía metodológica específica puede llevar a errores durante el proceso de programación, resultando en fallas en el sistema inmovilizador, posibles daños al vehículo, y pérdida de tiempo y recursos.

Se proyecta que el mercado de ECU automotrices (Unidad de Control Electrónico) crezca a una CAGR del 5,77 % entre 2018 y 2025. Se estima que el mercado de ECU automotrices fue de USD 25.080 millones en 2018 y se proyecta que alcance los USD 39.280 millones para 2025. Los principales impulsores del crecimiento de este mercado son la mayor incorporación de sistemas avanzados de seguridad, conveniencia y confort en los vehículos y el aumento de la demanda de vehículos eléctricos híbridos (HEV) y vehículos eléctricos de batería (BEV) (Figura 1).

Figura 1

ECU de Automóviles



Source: MarketsandMarkets Analysis

Tomadode: <https://www.marketsandmarkets.com/Market-Reports/automotive-ecu-market-34863602.html>

En el contexto de la seguridad automotriz moderna, el sistema inmovilizador electrónico constituye un elemento esencial para la prevención de robos, al impedir el arranque del motor si no se detecta un código electrónico válido desde la llave del vehículo. Este sistema, que se comunica con la unidad de control del motor (ECU), requiere un correcto funcionamiento y sincronización para garantizar la protección del vehículo (Bosch, 2022).

Sin embargo, en el Ecuador y en otros países de América Latina, es común encontrar vehículos que presentan fallos en sus módulos inmovilizadores debido a diversas causas como daños por humedad, fluctuaciones eléctricas o intentos de manipulación no autorizada. Estos fallos suelen generar la imposibilidad de arranque, bloqueando completamente el uso del vehículo. Ante esta situación, la reprogramación del módulo inmovilizador se convierte en una solución técnica eficaz, particularmente en entornos donde el reemplazo del módulo representa un alto costo o la pieza no está fácilmente disponible (Avendaño et al., 2020).

La falta de una metodología estandarizada y detallada para la programación de módulos inmovilizadores utilizando el programador UPA V1.3 presenta varios problemas críticos:

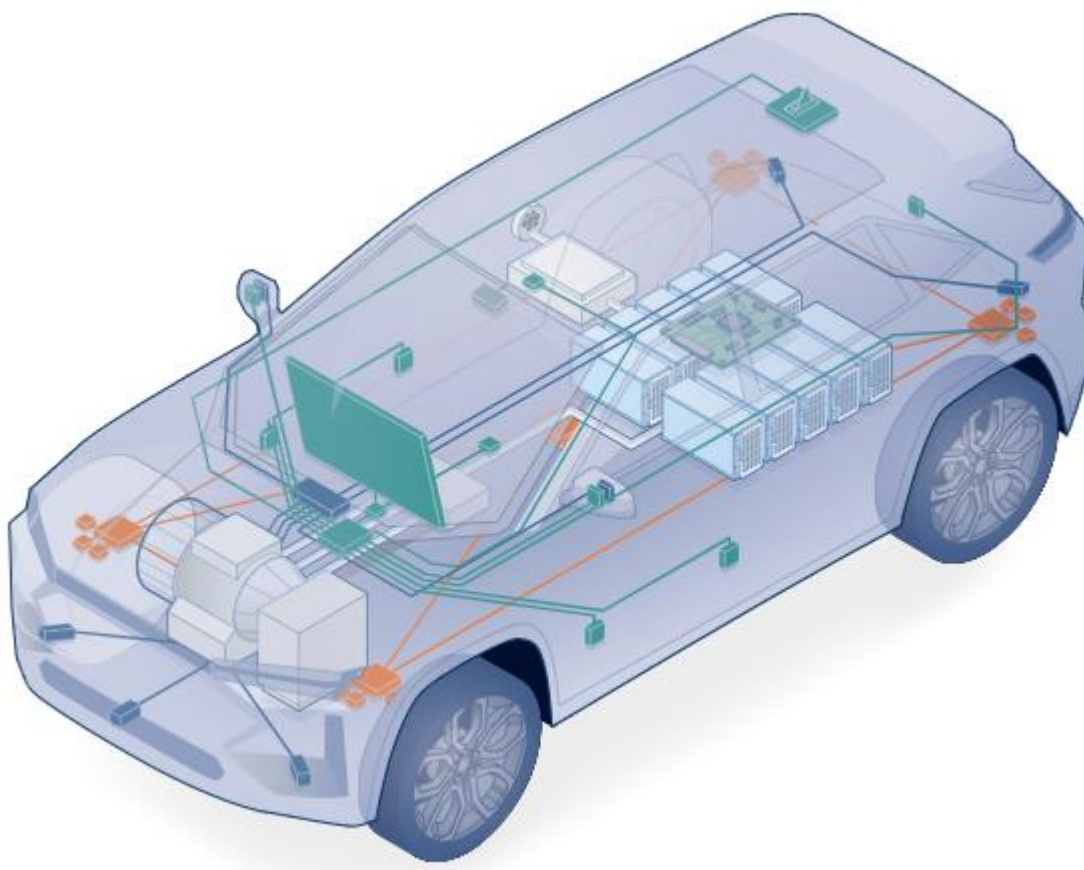
- Errores en la programación: Sin una guía clara, los técnicos pueden cometer errores durante el proceso de programación, lo que puede causar que el inmovilizador no funcione correctamente.
- Ineficiencia: La ausencia de una metodología estructurada puede hacer que el proceso de programación sea más largo y complicado de lo necesario.
- Riesgos de seguridad: Un sistema inmovilizador mal programado puede dejar el vehículo vulnerable a robos.
- Daños potenciales: Procedimientos incorrectos pueden resultar en daños a los componentes electrónicos del vehículo, aumentando los costos de reparación.

El desarrollo de niveles más altos de conducción autónoma y los nuevos servicios que ofrece la conectividad automotriz requieren computadoras de alto rendimiento conectadas a

redes a bordo de vehículos con velocidades de datos cada vez mayores. Al mismo tiempo, los proveedores de componentes automotrices y los fabricantes de vehículos deben garantizar el rendimiento, la interoperabilidad, la integridad de los datos y la seguridad de las redes de vehículos intravehiculares (IVN) (Figura 2).

Figura 2

Redes en Vehículos y Pruebas de ECU



Tomado de: https://www.rohde-schwarz.com/cz/solutions/automotive-testing/in-vehicle-networks-and-ecu-testing/in-vehicle-networks-and-ecu-testing_231834.html

1.2.2 Formulación del Problema

¿Cómo desarrollar una metodología detallada y sistemática para la programación de un módulo inmovilizador utilizando el programador UPA V 1.3 que sea eficiente, replicable y accesible para técnicos automotrices, con el fin de mejorar la seguridad vehicular y reducir los tiempos y costos asociados a este proceso?

1.2.3 Sistematización del Problema

- ¿Cómo influyen los tipos de inmovilizadores usados en un vehículo?
- ¿Es posible el uso de secuencias de comandos diseñados para automatizar ciertas funciones y procesos de programación con la ayuda de scripts específicos?
- ¿Existen la posibilidad de proporcionar recomendaciones para un proceso de lectura de memorias EEPROM que incluya los pasos específicos para la programación del módulo inmovilizador?

1.3 Objetivos de la Investigación

1.3.1 Objetivo General

- Crear una guía detallada y práctica para la programación de módulos inmovilizadores utilizando el programador UPA V1.3.

1.3.2 Objetivos Específicos

- Investigar los tipos de inmovilizadores usados en un vehículo.
- Describir los procedimientos de conexión y configuración inicial del programador UPA V1.3 para la correcta programación del módulo inmovilizador.
- Documentar los posibles errores y soluciones comunes que pueden surgir durante la programación del módulo inmovilizador con el UPA V1.3.

1.4 Justificación y Delimitación de la Investigación

Desarrollar una metodología detallada y estandarizada para la programación de módulos inmovilizadores utilizando el programador UPA V1.3 es crucial para:

- Mejorar la precisión: Reducir los errores durante la programación y asegurar que el sistema inmovilizador funcione correctamente.
- Aumentar la eficiencia: Proveer a los técnicos una guía clara que les permita realizar la programación de manera rápida y efectiva.

- Garantizar la seguridad: Asegurar que el vehículo esté adecuadamente protegido contra robos.
- Reducir riesgos de daño: Minimizar los riesgos de daños a los componentes electrónicos del vehículo.

Esta metodología será una herramienta valiosa para los técnicos automotrices, proporcionando un recurso confiable y eficiente para la programación de sistemas inmovilizadores, contribuyendo a mejorar la seguridad y operatividad de los vehículos.

1.4.1 Justificación Teórica

La implementación de una metodología clara y detallada para la programación de módulos inmovilizadores usando el programador UPA V1.3 no solo mejora la seguridad y funcionalidad del sistema inmovilizador, sino que también estandariza el proceso técnico, reduciendo el margen de error y facilitando la formación de nuevos técnicos en el área. Este proyecto, por lo tanto, tiene una relevancia significativa tanto para la industria automotriz como para los profesionales encargados de la seguridad vehicular. En el contexto de la seguridad vehicular, los sistemas inmovilizadores juegan un papel crucial al prevenir el robo de vehículos. Estos sistemas, integrados en la electrónica del vehículo, impiden el arranque del motor sin la presencia de la llave correcta, que está sincronizada electrónicamente con el módulo inmovilizador. La programación y reprogramación de estos módulos es una tarea técnica especializada que requiere herramientas y conocimientos específicos.

El programador UPA V1.3 es una herramienta ampliamente utilizada en la programación de módulos inmovilizadores debido a su precisión, fiabilidad y compatibilidad con una variedad de microcontroladores y EEPROMs.

Este proyecto propone una metodología detallada para la programación de un módulo inmovilizador usando el UPA V1.3, con el objetivo de optimizar y estandarizar el proceso, garantizando la seguridad y funcionalidad del sistema.

1.4.2 Justificación Metodológica

La metodología propuesta para la programación del módulo inmovilizador usando el programador UPA V1.3 se justifica por los siguientes motivos:

- Estandarización del Proceso: Crear un procedimiento estandarizado que pueda ser seguido por técnicos y mecánicos, garantizando uniformidad y calidad en la programación de los módulos inmovilizadores.
- Minimización de Errores: Reducir la probabilidad de errores durante la programación mediante el uso de un protocolo detallado y bien documentado.
- Eficiencia en el Tiempo: Optimizar el tiempo necesario para programar un módulo inmovilizador, permitiendo a los talleres automotrices atender más vehículos en menos tiempo.
- Facilidad de Aprendizaje: Facilitar el aprendizaje del procedimiento para técnicos nuevos mediante una guía clara y detallada.
- Compatibilidad y Actualización: Asegurar que la metodología sea compatible con el programador UPA V1.3 y esté actualizada con las últimas versiones del software y firmware del programador.

1.4.3 Justificación Práctica

En el mundo automotriz actual, los sistemas de seguridad y antirrobo son esenciales para proteger tanto los vehículos como sus componentes. El inmovilizador es una de las principales tecnologías utilizadas para este fin, ya que impide que el vehículo arranque si no se detecta una llave o transpondedor autorizado. Con la evolución de la tecnología, la reprogramación y reparación de estos módulos se ha vuelto crucial para talleres mecánicos y técnicos automotrices.

La importancia práctica radica en:

- **Mejora en la Seguridad Automotriz:** La capacidad de reprogramar módulos inmovilizadores garantiza que los sistemas antirrobo se mantengan actualizados y funcionales, reduciendo el riesgo de robo de vehículos.
- **Reducción de Costos:** La reprogramación y reparación de inmovilizadores en lugar de su reemplazo completo reduce significativamente los costos para los propietarios de vehículos y los talleres.
- **Actualización de Conocimientos Técnicos:** Proveer a los técnicos automotrices con una metodología clara y concisa para el uso del Programador UPA V1.3 contribuye a la formación y actualización de conocimientos, permitiéndoles manejar tecnologías modernas con mayor confianza.
- **Incremento en la Eficiencia de los Talleres:** Una metodología bien definida agiliza el proceso de reprogramación, reduciendo tiempos de reparación y mejorando la eficiencia operativa de los talleres mecánicos.

1.4.4 Delimitación Temporal

El proyecto se desarrolla durante un período de 6 meses, comenzando en abril de 2025 y finalizando en septiembre de 2025.

1.4.5 Delimitación Geográfica

Para el proyecto de investigación se realiza en un vehículo que se encuentra en Guayaquil-Ecuador.

1.4.6 Delimitación del Contenido

Para delimitar el contenido del proyecto "Metodología para Programación de un Módulo Inmovilizador Usando el Programador UPA V1.3", se deben definir los límites y el alcance de la investigación.

A continuación, se presenta una delimitación detallada del contenido del proyecto:

- Vehículos cubiertos: Especificar las marcas y modelos de vehículos para los cuales se aplica esta metodología.
- Componentes y herramientas: Incluir detalles sobre el programador UPA V1.3 y cualquier otro equipo necesario.
- Limitaciones: Describir las limitaciones técnicas y de alcance del proyecto, como las versiones del inmovilizador soportadas.
- Estudios y manuales previos: Revisión de documentación existente sobre la programación de inmovilizadores y el uso del programador UPA V1.3.
- Tendencias actuales en la programación de inmovilizadores: Análisis de las metodologías actuales y tecnologías emergentes en este campo.
- Descripción del Programador UPA V1.3.
- Características técnicas: Descripción detallada del hardware y software del UPA V1.3.
- Funciones principales: Enumerar y describir las principales funciones del programador.
- Compatibilidad: Tipos de módulos inmovilizadores y vehículos compatibles con el UPA V1.3.

Esta delimitación del contenido busca proporcionar una guía estructurada y completa para la programación de un módulo inmovilizador usando el programador UPA V1.3, garantizando que todos los aspectos relevantes sean cubiertos y documentados de manera sistemática.

Capítulo II

Marco Referencial

2.1 Marco Teórico

En el ámbito de la automoción, los sistemas inmovilizadores son una de las principales medidas de seguridad empleadas para prevenir el robo de vehículos. Estos sistemas impiden el arranque del motor a menos que se utilice una llave con un transpondedor que transmita un código válido al módulo inmovilizador del vehículo. La programación de estos módulos es una tarea crítica que requiere herramientas especializadas y una metodología precisa para garantizar la correcta sincronización entre la llave y el inmovilizador. El Programador UPA V1.3 es una herramienta popularmente utilizada para este propósito debido a su versatilidad y eficiencia.

2.1.1 *Conceptos Preliminares*

Para desarrollar una metodología para la programación de un módulo inmovilizador usando el Programador UPA V1.3, es importante comprender algunos conceptos preliminares, dentro de los aspectos más importantes están:

Inmovilizador de Vehículos

- **Definición:** Un inmovilizador es un dispositivo de seguridad electrónica instalado en un vehículo para prevenir el arranque del motor a menos que la llave correcta (transponder) esté presente (Sandoval, 2022).
- **Funcionamiento:** Al introducir la llave en el encendido, el inmovilizador verifica el código de la llave. Si el código es correcto, el motor puede arrancar.

Programador UPA V1.3

- **Descripción:** El programador UPA V1.3 es una herramienta utilizada para leer, escribir y programar chips de microcontroladores y memorias EEPROM.
- **Características:** Ofrece una interfaz de usuario amigable, soporte para múltiples dispositivos y una conexión USB para interactuar con una computadora.

Microcontroladores y EEPROM

- Microcontroladores: Son circuitos integrados que contienen un procesador, memoria y periféricos. Son comunes en sistemas electrónicos de automóviles.
- EEPROM: Es una memoria no volátil utilizada para almacenar información de configuración, como los códigos de inmovilizadores.

Transpondedores

- Definición: Son dispositivos pequeños incorporados en las llaves de los vehículos que emiten un código único al ser interrogados por el inmovilizador.
- Tipos: Hay transpondedores pasivos, que no tienen batería y son energizados por el campo magnético del inmovilizador, y transpondedores activos, que tienen su propia fuente de energía.

Interfaz de Usuario del Programador UPA V1.3

- Conexión y Configuración: Implica conectar el programador a la computadora, instalar el software y seleccionar el tipo de microcontrolador o EEPROM a programar.
- Lectura y Escritura de Datos: El software permite leer los datos del dispositivo conectado, modificarlos según sea necesario y escribir los datos modificados de vuelta al dispositivo.

Procedimientos de Programación

- Lectura del Inmovilizador Existente: Implica conectar el UPA V1.3 al inmovilizador y leer los datos actuales almacenados en la EEPROM o microcontrolador.
- Modificación de Datos: Ajustar los datos leídos para programar una nueva llave o reiniciar el inmovilizador.
- Escritura de Datos Nuevos: Grabar los datos modificados de vuelta al inmovilizador para actualizar su configuración.

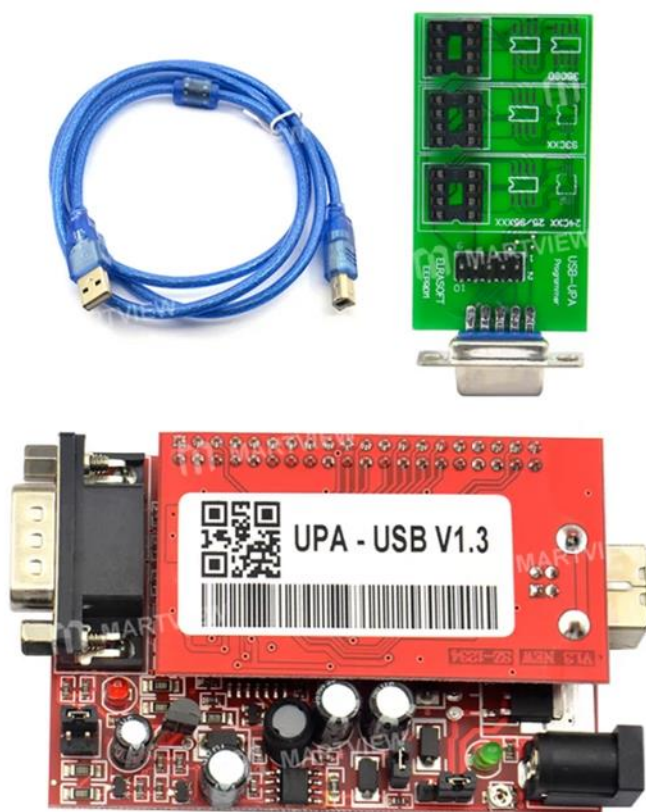
Seguridad y Precauciones

- Backup de Datos: Siempre realizar una copia de seguridad de los datos originales antes de hacer cualquier modificación.
- Verificación de Conexiones: Asegurarse de que todas las conexiones estén bien establecidas para evitar daños a los dispositivos.
- Uso Correcto del Software: Familiarizarse con la interfaz del software del UPA V1.3 para evitar errores durante la programación.

Con estos conceptos preliminares se puede definir la metodología detallada para la programación del módulo inmovilizador usando el Programador UPA V1.3 (Figura 3).

Figura 3

Programador UPA V1.3



Tomado de: <https://mcecursoautomotrices.jimdo.com/venta-de-equipos/upa/>

2.1.2 Importancia de los Inmovilizadores de un Vehículo

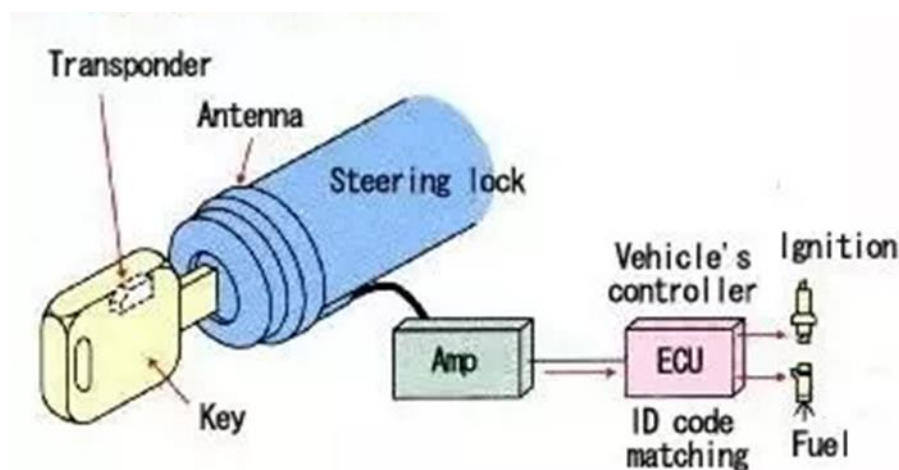
El inmovilizador impide que el automóvil funcione a menos que esté presente una llave autorizada y haya sido correctamente adaptada a varios componentes del vehículo. Se han utilizado varios esquemas de inmovilizador diferentes para diferentes años y diferentes modelos de vehículos con diferentes procedimientos que se les aplican (Berrezueta, 2023).

Los componentes del sistema inmovilizador que requieren procedimientos especiales de reemplazo incluyen las llaves, el controlador del motor (ECU) y el grupo de instrumentos (o los módulos inmovilizadores separados en vehículos así equipados).

En algunos casos, otros módulos o piezas también pueden estar relacionados con el inmovilizador; por ejemplo, los modelos Audi más nuevos también tienen la transmisión como parte del inmovilizador (Figura 4).

Figura 4

Inmovilizador de un Vehículo



Tomado de: <https://www.quora.com/What-is-an-immobilizer-How-does-it-work-in-an-automobile>

2.1.3 Sistema Inmovilizador de Vehículos

Los vehículos automotores modernos se están convirtiendo en una colección de subsistemas integrados interconectados, donde las partes mecánicas son controladas por partes electrónicas y el vehículo se transforma en un sistema de información móvil. Sin embargo, los

estándares de la industria para la comunicación en el vehículo no siguen las políticas de seguridad informática establecidas desde hace mucho tiempo. Esta tendencia no solo hace que los vehículos sean propensos a robos y ataques automatizados, sino que también pone en peligro la seguridad de los pasajeros.

Un sistema inmovilizador de vehículos (también conocido como inmovilizador de motor) es un elemento de tecnología de seguridad electrónica diseñado para disuadir e inhibir el robo de vehículos. Su objetivo es evitar que alguien que no tenga la llave para arrancar el motor pueda manipular el coche y robarlo. Las estadísticas del gobierno del Reino Unido muestran una reducción significativa del robo de vehículos después de 1992, cuando los inmovilizadores comenzaron a convertirse en algo habitual.

2.1.4 Usos de los Scripts de Programación

Los scripts de programación en el contexto del programador UPA (Universal Programmer Adapter) son secuencias de comandos o instrucciones diseñadas para llevar a cabo tareas específicas de programación en dispositivos electrónicos, como chips de memoria, microcontroladores, unidades de control del motor (ECU), tableros de instrumentos y otros componentes automotrices (Thoughtworks, 2025).

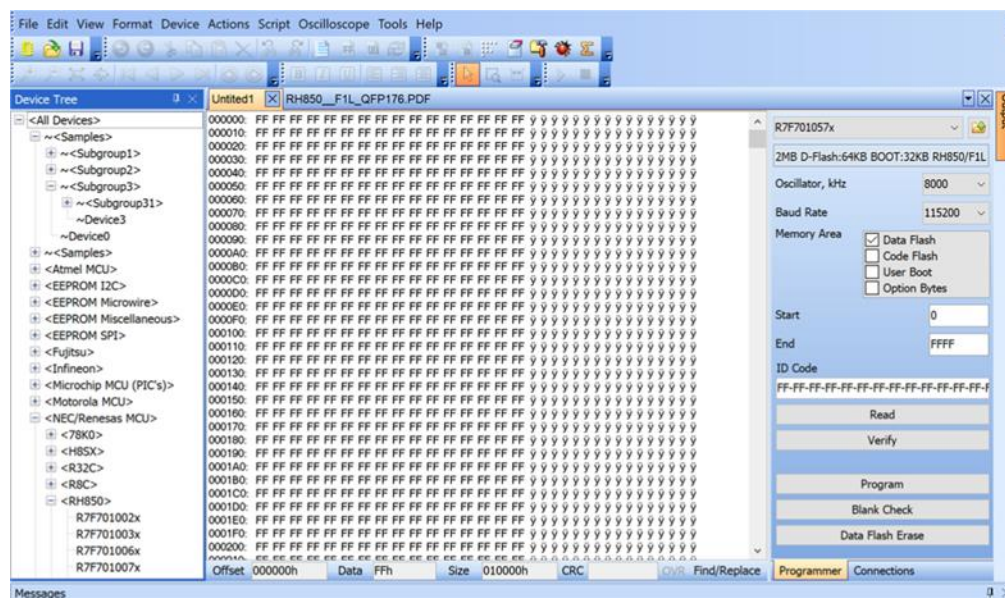
Estos scripts son programas informáticos que interactúan con el hardware del programador UPA, permitiendo la lectura, escritura y modificación de datos en los dispositivos objetivo. Están escritos en un lenguaje específico y son compatibles con una amplia variedad de dispositivos electrónicos y fabricantes de automóviles.

Los scripts de programación del UPA pueden realizar operaciones como leer y escribir datos en la memoria flash, programar claves de acceso para sistemas de seguridad de automóviles, reprogramar módulos de control del motor para mejorar el rendimiento o corregir errores, entre otras funciones. El Programador Serie UPA-USB-S (UUSP-S) es un programador diseñado para funcionar con odómetros, inmovilizadores, airbags y otras unidades de control

de vehículos. Además, el dispositivo se utiliza para programar procesadores AVR y chips de memoria EEPROM de STMicroelectronics, Motorola, Atmel y NEC, incluyendo BDM con protección contra lectura (Figura 5).

Figura 5

UPA – Software



Tomado de: <https://datatronik-automotive.com/software/upa-software-download/>

2.1.5 Eficiencia y Programación de los Inmovilizadores de Motor

Un inmovilizador de motor es un dispositivo electrónico que actúa para controlar el funcionamiento del sistema de encendido de un vehículo, el sistema de combustible, el motor de arranque o una combinación de estos tres sistemas. El inmovilizador no permitirá que el vehículo arranque sin recibir primero la señal correcta de la persona que intenta arrancarlo. Esta señal se puede comunicar a la unidad inmovilizadora de varias maneras. Estas incluyen el uso de un transpondedor (dispositivo electrónico de proximidad que convierte señales codificadas electrónicamente en señales de radio); una llave remota; una llave electrónica; un teclado codificado; o un interruptor de llave. El sistema de transpondedor es el preferido por los fabricantes de los inmovilizadores más nuevos y generalmente se percibe como el más inmune al robo (Potter, 2001).

Con el inicio del nuevo milenio, los inmovilizadores de automóviles evolucionaron hasta convertirse en sofisticados sistemas integrados con el ordenador de a bordo. Los inmovilizadores inteligentes entraron en escena, ofreciendo funciones como cierre remoto y sistemas de alarma que se comunicaban con el inmovilizador. Los primeros modelos utilizaban un código estático en la llave, mientras que los posteriores empleaban códigos variables o criptografía avanzada. Estos avances no solo mejoraron la seguridad, sino que también ofrecieron comodidad a los usuarios, ofreciendo una combinación armoniosa de seguridad y facilidad de uso (Smarttechlocksmiths.co.uk, 2025).

La evolución de los sistemas electrónicos automotrices ha llevado a la implementación de tecnologías avanzadas orientadas a la seguridad del vehículo. Entre estos dispositivos, el módulo inmovilizador cumple una función crítica: prevenir el encendido del motor si no se detecta un código electrónico válido, impidiendo así el robo del vehículo mediante métodos tradicionales (Bosch, 2022). Este módulo trabaja en conjunto con la unidad de control del motor (ECU), la antena del transpondedor y la llave codificada, estableciendo una red de validación que garantiza la autenticidad del arranque.

La programación de un módulo inmovilizador en un automovil es una técnica necesaria cuando se reemplazan componentes electrónicos, se pierden las llaves o se presenta una falla del sistema. Dicha programación implica leer, modificar o escribir datos en memorias EEPROM o microcontroladores específicos, generalmente mediante el uso de programadores como el UPA-USB, el Xprog o el VVDI PROG, que permiten acceso directo al hardware (Salcedo & Méndez, 2021).

En muchos casos, especialmente en países en vías de desarrollo, la programación del inmovilizador representa una solución más viable y económica frente al reemplazo total del sistema. No obstante, esta tarea requiere un conocimiento técnico profundo sobre estructuras de datos en la memoria, protocolos de comunicación y medidas de seguridad implementadas

por los fabricantes (García & Herrera, 2020). El uso de scripts automatizados ha simplificado procesos antes exclusivos de técnicos certificados, aunque aún implica riesgos como corrupción de datos, daño a la ECU o desincronización de componentes si no se realiza adecuadamente.

Por estas razones, resulta indispensable desarrollar metodologías estandarizadas para la programación de módulos inmovilizadores, que sean seguras, reproducibles y técnicamente fundamentadas, permitiendo a técnicos y especialistas una intervención precisa sin comprometer la integridad del sistema electrónico del vehículo.

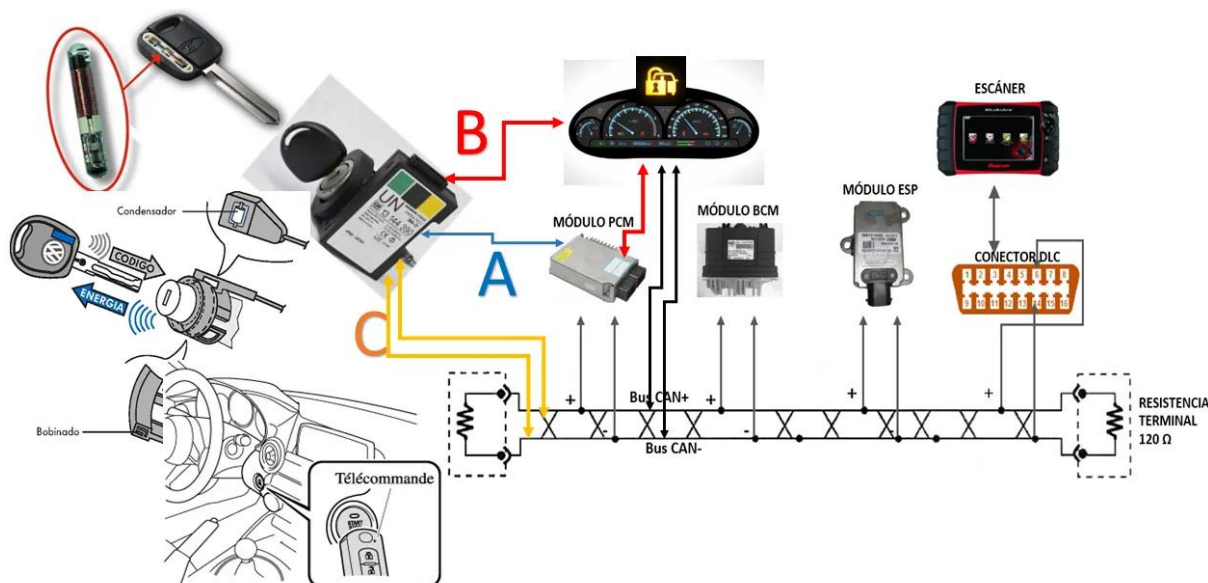
2.2 Marco Conceptual

2.2.1 Sistemas Automáticos de Inmovilizadores

Hoy en día, los sistemas son automáticos (Figura 6), lo que significa que el propietario del coche no tiene que hacer nada para que funcionen. Para evitar que se copien los códigos de la llave, estos sistemas utilizan códigos de cambio o criptografía más compleja. Un campo electromagnético activa un microcircuito dentro de la llave. Este envía un código binario único que la ECU del coche puede leer (Coello-Torres, 2014).

Aquí hay una explicación sencilla de cómo funcionan:

- El llavero o llave: tiene un chip rastreador en su interior que tiene un código único.
- El lector: La señal del transpondedor es captada por una antena o sensor colocado alrededor del área de encendido.
- La unidad de control: el lector y la unidad de control del motor (ECU) pueden comunicarse con esta parte del sistema informático del automóvil.
- Verificación de código: la unidad de control obtiene el código del transpondedor y lo compara con los datos que ha guardado.
- Control del motor: si los códigos son los mismos, la unidad de control permite que la ECU vuelva a la normalidad, lo que arranca el motor.

Figura 6*Partes de un Inmovilizador*

Tomado: <https://www.taaetcom.app.exur.com/news-taaet-blog/inmovilizadores-electronicos>

2.2.2 Tipos de Inmovilizadores

Aunque hacen el mismo trabajo básico, los sistemas inmovilizadores son diferentes en la forma en que el usuario los maneja (Sandoval, 2022).

- **Inmovilizador de transpondedor (más común):** Esta es la tecnología que incorporan la mayoría de los coches modernos. El llavero contiene un chip transpondedor inactivo que no necesita batería. El campo eléctrico del lector carga el chip, lo que le permite enviar su código único.
- **Inmovilizador con llave táctil:** En este caso, el chip de rastreo está integrado en la propia llave. Para verificar el código, el usuario debe tocar la llave con un área específica del lector dentro del vehículo.
- **Inmovilizador de control remoto:** Este dispositivo menos común cuenta con un llavero independiente con botones que permiten activar y desactivar el inmovilizador a distancia. Para mayor comodidad, se puede conectar al sistema de cierre centralizado del coche.

- **Inmovilizador electrónico:** El primer tipo era un inmovilizador electrónico que utilizaba una señal estática. Aunque funcionaban, los ladrones con herramientas especiales podían copiar las señales que enviaban.
- **Herramientas de bloqueo criptográfico:** Los sistemas criptográficos utilizan métodos avanzados para generar un nuevo código cada vez que se usa la clave. Esto dificulta enormemente su copia.
- **Inmovilizadores adicionales:** Hay sistemas adicionales disponibles para coches antiguos o que no traían inmovilizadores de fábrica. Estos deben desactivar al menos dos circuitos, como la bomba de combustible y el encendido, para que la compañía de seguros los apruebe. **Conectividad con otras funciones de seguridad:** Algunos inmovilizadores vienen con alarmas integradas que pueden ser activas, pasivas o silenciosas, y algunos pueden conectarse a una aplicación para rastreo GPS (Cárdenas, 2021).

2.2.3 *Sistemas Inmovilizadores en Vehículos*

- **Definición y Función:**

Un sistema inmovilizador es un dispositivo antirrobo electrónico que impide que el motor del vehículo se encienda sin una llave autorizada. Su principal función es añadir una capa adicional de seguridad que complementa otras medidas físicas y electrónicas del vehículo (Gómez, 2023).

- **Componentes:**

Llave con transpondedor: Contiene un chip que envía un código único al inmovilizador.

Módulo inmovilizador: Recibe y verifica el código de la llave.

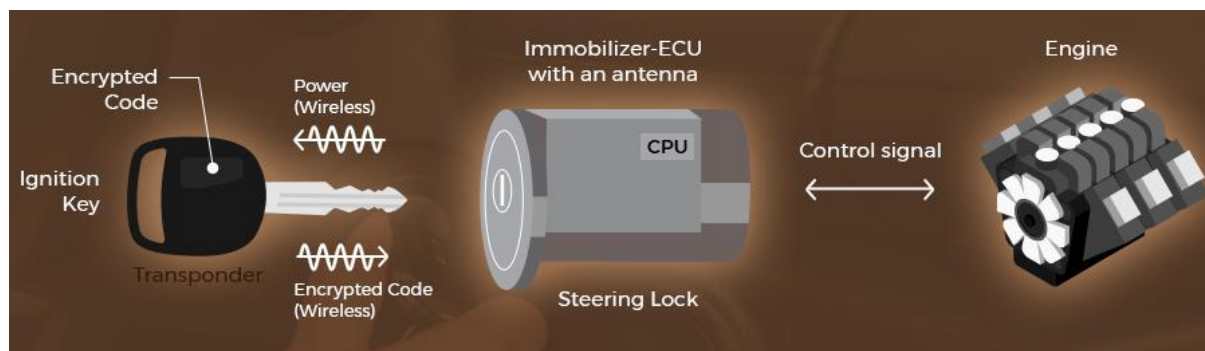
ECU (Unidad de Control del Motor): En muchos sistemas, la ECU se comunica con el módulo inmovilizador para permitir o impedir el arranque del motor.

- **Proceso de Comunicación:**

Cuando se inserta la llave en el encendido, el transpondedor envía un código al módulo inmovilizador. Si el código es correcto, el módulo envía una señal a la ECU para permitir el arranque del motor (Figura 7).

Figura 7

Funcionamiento del Transpondedor



Tomado: <https://trackobit.com/blog/what-is-engine-immobilizer-working-and-benefits>

2.2.4 Programador UPA (Universal Programmer Adapter)

El UPA es una herramienta de programación universal ampliamente utilizada en la industria automotriz para leer y escribir datos en diversos módulos electrónicos de control (ECU) y otros dispositivos electrónicos. Una característica distintiva del UPA es su capacidad para utilizar scripts, que son pequeños programas o secuencias de comandos diseñados para automatizar ciertas funciones y procesos de programación. A continuación, se describen algunas coberturas y funciones clave del programador UPA (Figura 8):

- Cobertura de vehículos: El UPA es compatible con una amplia gama de vehículos, incluidos automóviles, camiones, motocicletas y otros vehículos equipados con ECUs y módulos electrónicos.
- Funciones de lectura y escritura: El UPA puede leer y escribir datos en la memoria de la ECU, lo que permite la modificación y reprogramación de parámetros y configuraciones del vehículo.

complementos instalados en el programador UPA. Se recomienda verificar la compatibilidad y recibir la capacitación adecuada antes de utilizar el UPA con scripts en vehículos específicos.

2.2.5 *Microcontroladores en el Sector de la Automoción*

Un microcontrolador es un microordenador que, además de su procesador, combina varios otros componentes importantes y elementos periféricos en un chip, es decir, un sistema de un solo chip, pero que no debe confundirse con un sistema en un chip. Por lo general, se seleccionan de manera adaptada a sus respectivas aplicaciones especiales y, en el mejor de los casos, deberían cumplir sus tareas con la menor cantidad posible de componentes externos y pueden fabricarse en grandes cantidades. A menudo realizan tareas de control y comunicación de todo tipo.

El requisito más importante de las familias de microcontroladores para automoción es sin duda la seguridad técnica que deben aportar, que difiere claramente de la de los dispositivos de consumo o de las aplicaciones de comunicación. Además, las memorias flash y RAM integradas con sus mecanismos de seguridad garantizan la robustez durante todo el ciclo de vida.

Los dispositivos de control de dominio y de zona integran funciones lógicas de software que antes se implementaban en sistemas distribuidos. De este modo, la complejidad del desarrollo se traslada de la red del vehículo con muchos dispositivos de control individuales a la arquitectura de hardware/software de los dispositivos de control de dominio/zonas centrales. Además de una potencia de cálculo escalable hasta 15 kDMIPS, esto requiere flexibilidad para la integración de funciones de varios ASIL (Nivel de integridad de seguridad automotriz) según ISO26262 (Elektrobit, 2025).

2.2.6 *Programación Antirrobo*

En los automóviles modernos, se utilizan inmovilizadores electrónicos para evitar el acceso no autorizado y el robo. Estos sistemas se basan en pequeños chips electrónicos llamados

transpondedores integrados en la llave. Cuando se inserta la llave en el encendido, el transpondedor se comunica con el sistema inmovilizador del automóvil y envía un código único que hace que el motor arranque.

Si las llaves del automóvil se pierden o son robadas, se requiere un nuevo juego de llaves por cualquier motivo. La nueva llave o chip transpondedor debe programarse para que funcione con el sistema antirrobo del automóvil. Por eso los automóviles necesitan una programación antirrobo.

2.2.7 Principio del Programador Inmovilizador

El principio de un programador de llave inmovilizadora se basa en la capacidad de comunicarse con el sistema inmovilizador del vehículo y extraer la información necesaria para programar una nueva llave. Esta información incluye el identificador único del sistema antirrobo, los códigos de llave y cualquier otro dato necesario (Berrezueta, 2023).

2.2.8 Diferencia entre EPROM y EEPROM

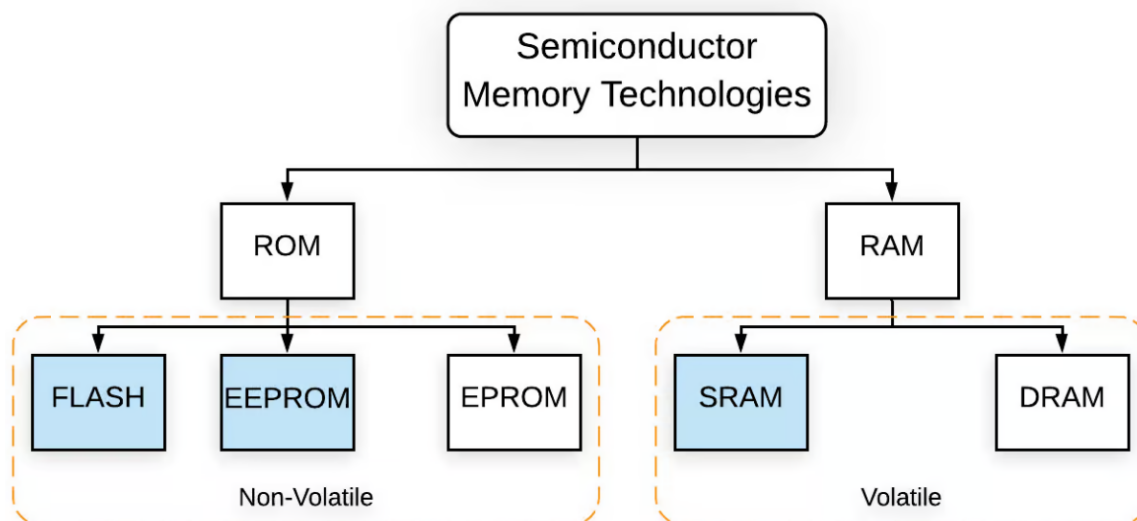
EPROM (Erasable Programmable Read Only Memory) es también el tipo de ROM que se puede leer y escribir ópticamente. Para escribir una EPROM asociada, sus celdas de almacenamiento deben permanecer dentro del mismo estado inicial. La EPROM ofrece una permanencia de almacenamiento reducida en comparación con la PROM como resultado de que la EPROM es receptiva a la radiación y al ruido eléctrico. En la construcción de EPROM, se utilizan transistores MOS.

En la EEPROM (memoria de solo lectura programable y borrrable eléctricamente) se utiliza una señal eléctrica para borrar el contenido de la EEPROM. La EEPROM es una memoria no volátil que conserva su contenido a pesar de que se cambia la capacidad. Almacena la pequeña cantidad de información que se utiliza para el BIOS de la computadora. Fue un reemplazo de la PROM y la EPROM. La principal diferencia entre la EPROM y la EEPROM es que el contenido de la EPROM se borra mediante rayos UV (Salas, 2024).

Por otro lado, el contenido de la EEPROM se borra mediante señales eléctricas (Figura 9).

Figura 9

Tipos de Memoria



Tomado de: <https://www.circuitbasics.com/types-of-memory-on-the-arduino/>

2.2.9 Script

En programación informática, un script es un programa o secuencia de instrucciones que es interpretada o ejecutada por otro programa en lugar de por el procesador de la computadora (como un programa compilado).

Un script del UPA en automoción es un archivo de instrucciones automatizadas que se ejecutan dentro del software del programador UPA-USB (Universal Programmer Adapter) para facilitar tareas repetitivas sobre memorias EEPROM o microcontroladores de módulos electrónicos automotrices.

Estos scripts están escritos en un lenguaje similar a Basic y permiten automatizar procesos como:

- Leer datos de la memoria.

- Interpretar y mostrar información como el PIN, código IMMO, número VIN, kilometraje, etc.
- Calcular checksums.
- Escribir datos corregidos o nuevos en la memoria.
- Clonar o sincronizar llaves con inmovilizadores o ECUs.

2.2.10 Editor Hexadecimal

El editor hexadecimal (HexEdit) permite editar archivos binarios para programar una memoria o un microcontrolador. El tamaño máximo del archivo es teóricamente de 2 GB, pero en realidad depende de la memoria virtual disponible en el ordenador. El editor funciona en modo de sobrescritura o inserción, que se alterna con la tecla Insertar o el botón Insertar/Sobrescribir situado en la parte inferior de la barra de estado de la ventana. HexEdit permite abrir varios archivos y realizar diversas operaciones con ellos. HexEdit tiene tres áreas: desplazamiento, numérico y texto.

- 000000: 61 62 73 64 65 66 67 68 absdefgh
- 000008: 6A 6B 6C 6D 6E 6B 70 71 jklmnkpq

2.2.11 Programador de Memorias y Microcontroladores (UPA)

El equipo programador UPA (Figura 10) es uno de los más utilizados en el mercado de electrónica automotriz, dado que posibilita realizar funciones especiales como, el ajuste de kilometraje, reinicio de airbags, habilitación de módulos, emparejamientos de módulos, recolección del código de seguridad del inmovilizador y otras funciones, que garantiza un trabajo completo.

De esta manera el equipo ejecuta con diferentes adaptadores que contribuyen al trabajo seguro y cómodo con los clips de soldadura. Para la configuración del dispositivo se efectúa mediante teclas especiales de ajuste que nos deslinda de toda forma manual (MCE, 2021).

Figura 10

Programador de Memorias y Microcontroladores (UPA)



2.2.12 Ventajas del UPA

- Posee una base de datos extensa para leer y escribir memorias.
- El precio del equipo es accesible para aquellos que están empezando en este campo.
- Los adaptadores adicionales para leer y escribir memorias están incluidos con la adquisición del equipo.
- La conectividad del equipo con el ordenador es muy versátil, ya que se realiza a través del puerto USB.

2.2.13 Desventajas del UPA

- En ciertos módulos de inmovilizador, es necesario realizar una configuración adicional para leer la memoria correctamente.
- No dispone de soporte técnico ni actualizaciones, lo cual representa una desventaja para el usuario.
- Los recursos de apoyo para este equipo se encuentran muy limitada.
- Carece de diagramas de conexión para leer la memoria del inmovilizador.

Capítulo III

Conexión y Configuración Inicial del Programador UPA V1.3

En este capítulo se describe la metodología empleada para la conexión y configuración inicial del programador UPA V1.3, herramienta utilizada para la lectura, escritura y programación de memorias EEPROM y microcontroladores.

El procedimiento se detalla paso a paso, abarcando tanto la instalación del software correspondiente como la correcta vinculación del hardware con el equipo de cómputo. Esta etapa es fundamental para garantizar una comunicación estable y segura entre el programador y el dispositivo a intervenir, asegurando la precisión en la manipulación de los datos electrónicos involucrados.

3.1 Métodos

El propósito de este proyecto es desarrollar un procedimiento detallado y práctico para la metodología para programación de un módulo inmovilizador usando el programador UPA V1.3.

3.2 Tipo de Estudio

El tipo de estudio aplicado en este proyecto es experimental y aplicado (Estrella, 2025).

3.2.1 Estudio Experimental

Para el proyecto de metodología para programación de un módulo inmovilizador usando el programador UPA V1.3, se lleva a cabo un estudio experimental.

El proceso incluye:

- Preparación inicial: Pasos preliminares antes de comenzar la programación, incluyendo la configuración del programador y la preparación del vehículo.
- Procedimiento de conexión: Detallar el proceso de conexión del UPA V1.3 al módulo inmovilizador.

- Proceso de programación: Lectura de datos: Instrucciones para leer los datos del inmovilizador.

Modificación de parámetros: Procedimientos para modificar los parámetros necesarios.

Escritura de datos: Instrucciones para grabar los datos modificados en el inmovilizador.

3.2.2 Estudio Aplicado

En este estudio práctico, se busca proporcionar una guía estructurada y completa para la programación de un módulo inmovilizador usando el programador UPA V1.3, garantizando que todos los aspectos relevantes sean cubiertos y documentados de manera sistemática, para un correcto funcionamiento.

3.3 Guía de Instalación

El programador de dispositivos USB UPA V1.3.0.14 que es compatible para varios vehículos se lanzó hace años. A continuación, se incluye una guía de instalación paso a paso:

- La versión 1.3 de UPA-USB incorpora varios chips ECU nuevos y corrige errores como "el código no está firmado".
- Es compatible con Windows.
- Si se instala el software uuprog por primera vez en su computadora, debe instalar el controlador USB.
- También se debe activar UUSP antes de usar.
- Insertar el CD del software en su computadora.
- Luego se obtiene UPA-USB Device Programmer.zip, se debe descomprimir, luego acceder a la carpeta UPA 1.3, abrir la carpeta UPA 1.3 y se verán los archivos (Figura 11).

Figura 11*Instalación de Archivos del Programa*

- Haga clic en "Configuración" para UPA-USB Device Programmer, luego continuar con la guía del sistema hasta que se finalice la configuración.
- Abrir esta carpeta.

Luego se debe copiar el archivo uuprog.lic en la carpeta del software uuprog, y si el software uuprog se instala por primera vez en esta computadora, también debe instalar el controlador usb.

- Después de terminar, se debe activar UUSP antes de usarlo, cuando se muestre el cuadro de diálogo de activación, se debe dar clic en el botón "Obtener activación" y copiar/pegar la clave de activación obtenida y el nombre de usuario en el cuadro de diálogo de activación de UUPROG y hacer clic en el botón "Activar" (Figura 12).

Figura 12*Activación de UUSP*

- Después de terminar, puede usar V1.3 UPA-USB Serial Programmer sin ningún problema.

3.4 Conexión – Programador UPA V1.3

La conexión del programador UPA V1.3 se realiza mediante un cable USB que enlaza el dispositivo con el computador, permitiendo la comunicación entre ambos.

Es fundamental asegurarse de que el programador esté correctamente conectado y reconocido por el sistema operativo antes de iniciar cualquier operación, ya que una conexión inadecuada puede provocar errores en la lectura o escritura de datos.

Figura 13

Software y Drivers



3.4.1 Conexión Física

- Conectar el cable USB del programador al PC.
- El LED del UPA debe encenderse (color verde o rojo, según el modelo).

Figura 14

Conexión Física



3.4.3 Verificación de Conexión

- Abrir el software UPA-USB.
- Ir a menú “Hardware Setup” o sección de estado.
- Confirmar que indique: “Device Connected” o “UPA Connected”.

Figura 15

Verificación de Conexión



3.4.4 Preparación para Programación

- Seleccionar el tipo de dispositivo (EEPROM o MCU).
- Consultar el diagrama de conexión del chip en el software.

- Conectar el chip usando:

Zócalo ZIF (fuera de circuito) (Figura 17).

Pines adaptadores o cables jumper (en-circuit).

Figura 16

Verificación de Conexión

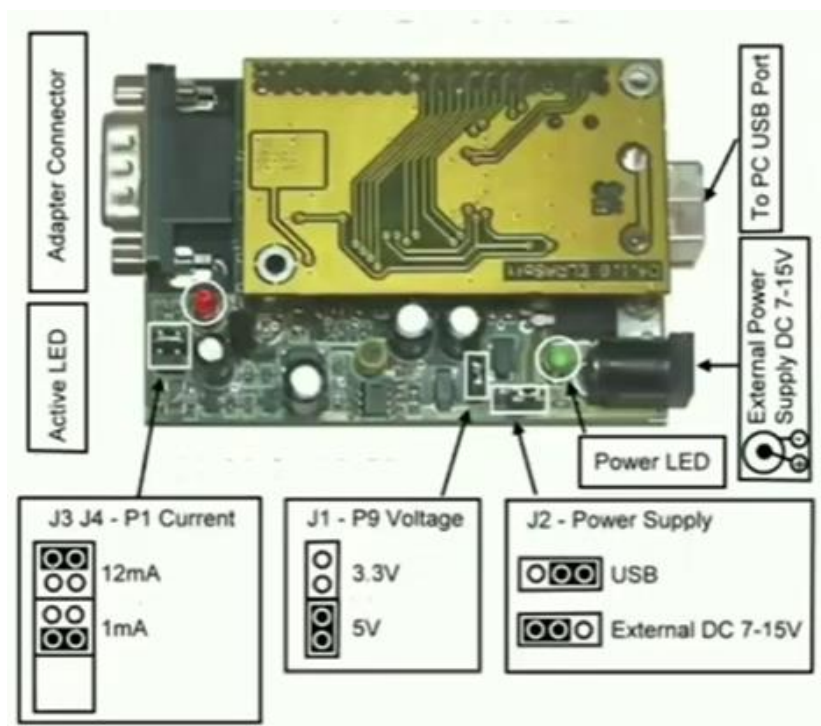


Figura 17

Zócalo ZIF



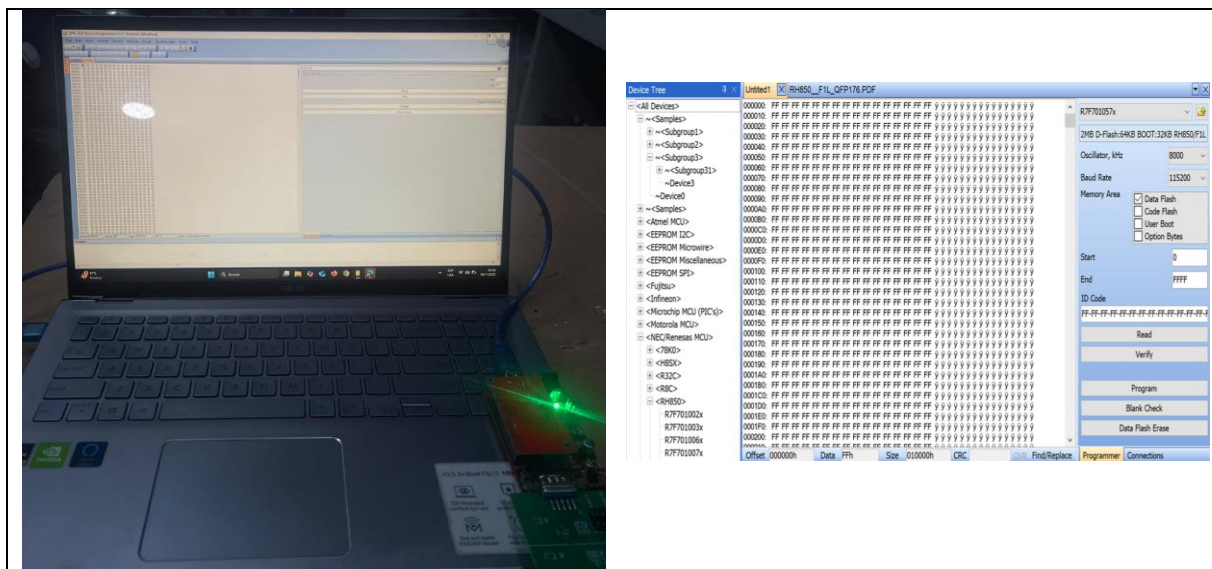
3.4.5 Lectura / Escritura

- Elige "Read" para extraer datos.

- Guarda un backup del archivo leído.
- Modifica o carga un archivo nuevo si es necesario.
- Usa “Write” para grabar el dispositivo.
- Espera confirmación de operación exitosa.

Figura 18

Lectura – Escritura



3.4.6 Desconexión Segura

- Cierra el software.
- Desconecta el programador del USB.
- Retira el chip del adaptador o cables.

3.4.7 Configuración Inicial del Programador UPA V1.3

En la Tabla 1 se muestra la conexión y configuración inicial del programador UPA V1.3, usada típicamente para programación de EEPROMs y microcontroladores automotrices.

Tabla 1*Características del UPA V1.3*

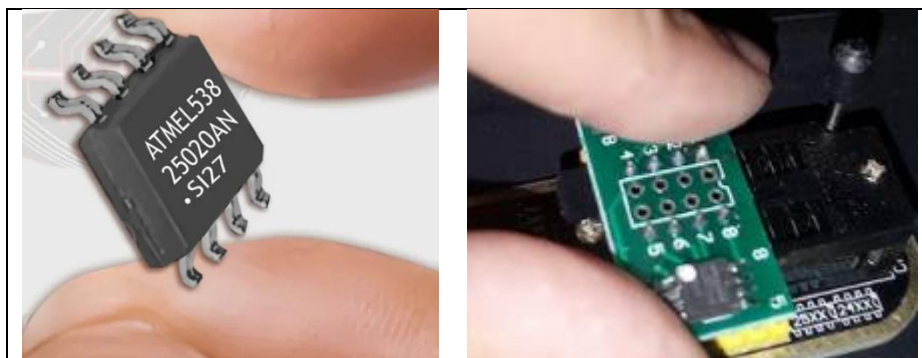
Elemento	Descripción / Detalle
Nombre del equipo	Programador UPA-USB V1.3
Tipo de conexión	USB 2.0 al PC (cable USB estándar)
Alimentación	Alimentación vía USB (5V)
Software requerido	UPA-USB Device Programmer Software (versión compatible con V1.3)
Instalación del software	Instalar software UPA-USB desde el CD incluido o página oficial; ejecutar como administrador.
Drivers necesarios	UPA-USB Driver (instalación automática o manual desde carpeta 'Drivers')
Sistema operativo compatible	Windows XP, 7, 8, 10 (preferiblemente 32 bits, aunque versiones recientes soportan 64 bits)
Configuración inicial	1. Conectar UPA al puerto USB del PC. 2. Instalar driver. 3. Abrir software y verificar conexión en "Hardware Setup".
Detección del dispositivo	En el software, verificar que el programador aparezca como "Connected".
Tipos de dispositivos soportados	EEPROMs (I2C, SPI, Microwire), MCU's (Motorola, ST, etc.)
Modo de programación	Circuito en-circuit (ICP) o fuera de circuito, según el chip.

Tabla 2*Características del UPA V1.3- Continuación*

Elemento	Descripción / Detalle
Precauciones	Verificar correcta orientación del adaptador ZIF; evitar desconexión durante la escritura.
Actualizaciones firmware	Usualmente no necesarias; seguir indicaciones del fabricante si es el caso.
Indicadores LED	LED encendido (alimentación), LED parpadeante (lectura/escritura).

3.5 Lectura y Programación de un Inmovilizador Chevrolet Sail con EEPROM 25C020

En el ámbito automotriz, los sistemas de inmovilizador electrónico juegan un papel fundamental en la seguridad del vehículo, impidiendo el encendido sin la llave original. El Chevrolet Sail, como muchos vehículos modernos, emplea un módulo inmovilizador controlado por una memoria EEPROM 25C020, donde se almacena la información clave del reconocimiento del transponder, claves de acceso y configuración del sistema.

Figura 19*Memoria EEPROM 25C020*

La lectura y programación de esta EEPROM es un procedimiento habitual en talleres especializados cuando es necesario:

- Realizar copias de seguridad del archivo original.
- Adaptar nuevas llaves.
- Realizar una clonación del módulo.
- Restablecer el sistema tras una pérdida de llaves.

3.5.1 Identificación y Retiro del Inmovilizador

- Ubicar el inmovilizador en el vehículo Chevrolet Sail. En este modelo, usualmente se encuentra próximo a la columna de dirección o integrado en la BCM.
- Extraer cuidadosamente el módulo, asegurando que no se dañen las pistas o componentes del circuito impreso.

3.5.2 Ubicación del Chip EEPROM 25C020

- Localizar el chip 25C020 en la placa del inmovilizador. Este es un circuito integrado de 8 pines tipo SOIC o DIP.
- Tomar nota de la posición del pin 1 (marcado con un punto o muesca en el encapsulado).

Figura 20

Chip 25C020 en la Placa del Inmovilizador



3.5.3 Conexión del Programador UPA V1.3 a la EEPROM

- Si se trabaja en circuito, se debe utilizar una pinza SOIC8 para evitar desoldar el chip. Asegurarse de que los pines estén firmemente conectados.
- Si se trabaja fuera de circuito, se debe desoldar la EEPROM y conectarla en una placa adaptadora DIP8 o zócalo correspondiente del UPA.

Figura 21

Elementos – Programación y Escritura

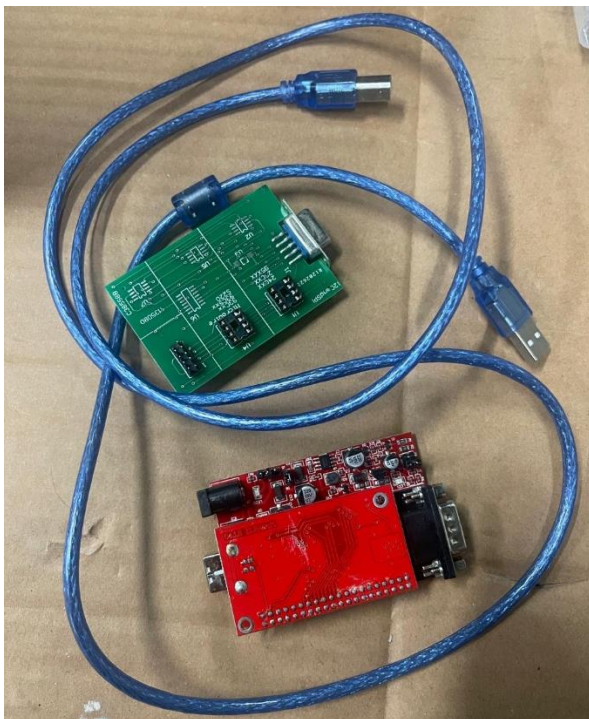
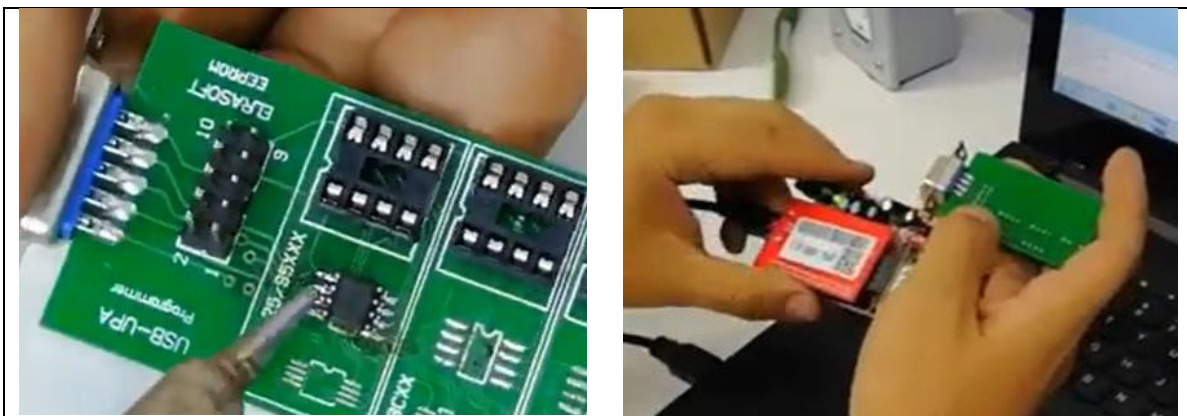


Figura 22

Armado – Realizar Pruebas



- Realizar las conexiones según el mapeo de pines de la Tabla 3.

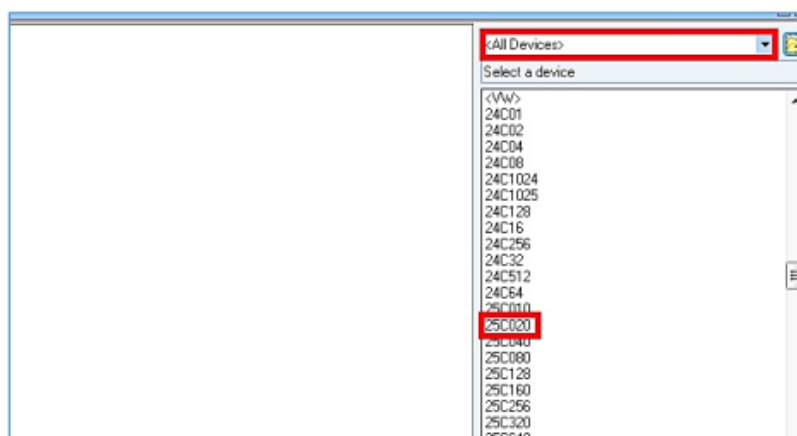
Tabla 3*Mapeo de Pines*

EEPROM 25C020	UPA V1.3 Pin
Pin 1 (CS)	Chip Select (CS)
Pin 2 (DO)	MISO
Pin 3 (WP)	GND
Pin 4 (GND)	GND
Pin 5 (DI)	MOSI
Pin 6 (CLK)	CLK
Pin 7 (HOLD)	GND
Pin 8 (VCC)	3.3V – 5V

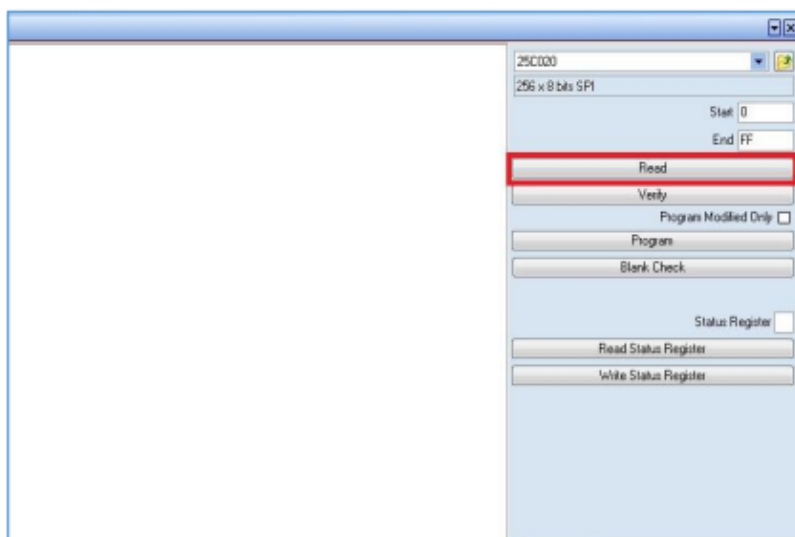
Se recomienda utilizar 3.3V para este tipo de EEPROM.

3.5.4 Lectura de la EEPROM

- Abrir el software del UPA y seleccionar el tipo de memoria: EEPROM → SPI → 25C020.

Figura 23*Selección del Tipo de Memoria*

- Hacer clic en “Read” (Leer) para obtener el contenido de la memoria.

Figura 24*Lectura del Contenido de la Memoria*

- Guardar el archivo leído en formato .bin con un nombre identificativo (por ejemplo: `sail_ori_25C020.bin`).

3.5.5 Edición del Archivo

- Utilizar un editor hexadecimal como HxD o WinHex para modificar valores específicos (por ejemplo, eliminar claves o cargar archivo virgen).
- También puedes usar un archivo virgen ya preparado para el modelo de inmovilizador del Sail y grabarlo directamente.

3.5.6 Programación de la EEPROM

- Cargar el archivo modificado o virgen desde el software del UPA.
- Hacer clic en “Write” (Grabar).
- Luego, realizar una verificación mediante la opción “Verify” para confirmar que los datos se grabaron correctamente.

3.5.7 Reinstalación del Módulo

- Si la EEPROM fue desoldada, volver a soldarla asegurando una correcta orientación del pin 1.

- Reinstalar el inmovilizador en el vehículo y conectar todos los cables.

3.5.8 Prueba de Arranque del Vehículo

- Intentar arrancar el motor con la llave original.
- Si se cargó un archivo virgen, puede ser necesario sincronizar la ECU con el inmovilizador mediante escáner automotriz o procedimiento de emparejamiento.

Capítulo IV

Programación de un Módulo Inmovilizador

4.1 Descripción

En este capítulo se expone el proceso de programación de un módulo inmovilizador, componente esencial del sistema de seguridad electrónica de un vehículo. Se detallan los pasos necesarios para acceder, modificar y reprogramar la información contenida en la memoria del módulo, utilizando herramientas especializadas como el programador UPA V1.3. Esta intervención permite restablecer o adaptar el sistema inmovilizador, especialmente en casos de pérdida de llaves, reemplazo de módulos o fallos en la configuración original.

4.2 Errores y Soluciones Comunes en la Programación del Módulo Inmovilizador con UPA V1.3

En este capítulo se detalla el proceso de programación del módulo inmovilizador en un vehículo Chevrolet Sail utilizando el programador UPA V1.3. Además, se describe los errores más frecuentes que pueden surgir durante el procedimiento y cómo solucionarlos, para asegurar un proceso exitoso y seguro.

4.3 Proceso Detallado de Programación

4.3.1 Preparación

- Verificar que la batería del vehículo esté cargada y en buen estado.
- Contar con el programador UPA V1.3 actualizado.
- Tener el cable OBD-II compatible y en buen estado.
- Apagar todos los dispositivos eléctricos del vehículo.
- Manual técnico del módulo inmovilizador para el Chevrolet Sail 2018.

Figura 25*Cable USB Compatible*

4.3.2 Conexión

- Ubicar el puerto OBD-II del Chevrolet Sail, ubicado bajo el tablero, a la izquierda de la columna de dirección.

Figura 26*OBD-II Ubicación en el Vehículo*

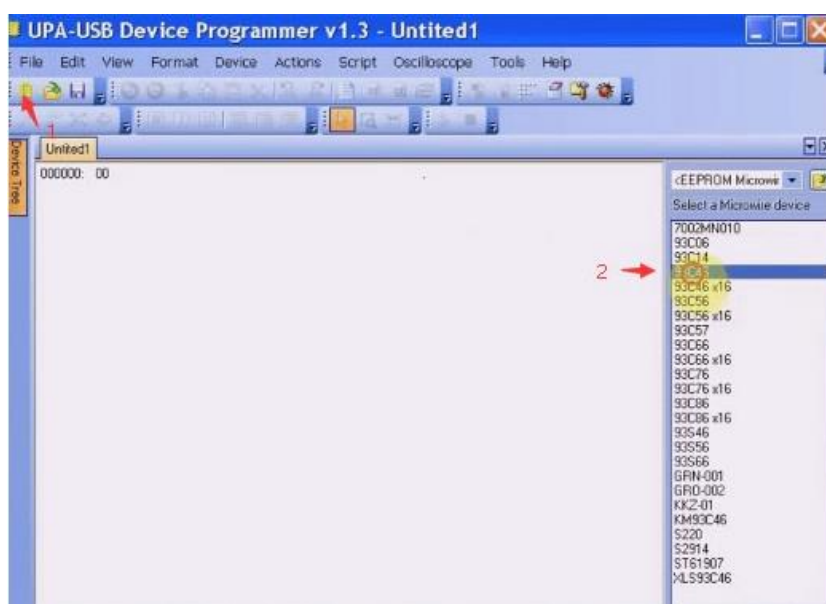
- Conectar el cable OBD-II del UPA V1.3 al puerto, asegurando un buen contacto.
- Encender el programador UPA V1.3.
- Colocar el vehículo en modo "Accesorios" (encendido sin arrancar motor).

4.3.3 Configuración del Programador

- En el menú principal del UPA V1.3, seleccionar la marca Chevrolet.
- Seleccionar modelo Sail 2018.
- Elegir el sistema módulo inmovilizador.
- Confirmar la selección y proceder a la lectura del módulo para verificar el estado actual.

Figura 27

Ejecución del Programador



4.3.4 Programación

- Realizar una copia de seguridad de la configuración original.
- Seleccionar la opción para programar llaves nuevas o actualizar firmware, según sea el caso.
- Seguir las indicaciones del UPA V1.3, que incluyen tiempos estimados y pasos a seguir.
- Al finalizar, el dispositivo mostrará un mensaje de éxito o error.

4.3.5 Verificación

- Apagar y encender el vehículo para comprobar el correcto funcionamiento del inmovilizador.
- Probar con las llaves programadas si el motor arranca sin inconvenientes.

4.4 Errores Comunes y Soluciones

En la práctica, los usuarios (especialmente principiantes) suelen enfrentarse a ciertos errores recurrentes que afectan la programación.

Tabla 4

Errores y Soluciones en la Programación del UPA V1.3

Error	Causa Probable	Solución Recomendada
No se detecta el módulo	Cable mal conectado, puerto OBD-II sucio o dañado	Verificar conexión, limpiar puerto, usar otro cable si es posible
Error de comunicación	Firmware desactualizado o incompatibilidad de software	Actualizar firmware del UPA V1.3 y software en PC
Interrupción de energía	Batería baja o desconexión accidental durante programación	Asegurar batería cargada, evitar mover cables durante proceso
Datos corruptos / Checksum	Interferencia eléctrica o desconexión durante escritura	Repetir procedimiento en ambiente libre de interferencias, asegurar conexión estable
Módulo incompatible	Selección errónea del modelo o tipo de módulo en el programador	Confirmar datos exactos y modelo en manual, corregir selección
Programador congelado / bloqueo	Problemas de software interno o hardware	Reiniciar el UPA V1.3, si persiste actualizar o consultar soporte técnico
Llave no reconocida después de programación	Fallo en la sincronización entre módulo y llave	Repetir programación de llave, verificar que llave sea compatible con el vehículo

4.5 Consejos para Evitar Errores

- Siempre mantener el software y firmware del UPA V1.3 actualizados.
- Evitar mover los cables o desconectar el programador durante la programación.
- Realizar la programación en un ambiente con poca interferencia electromagnética.
- Utilizar batería externa o cargador para mantener energía estable en el vehículo si es necesario.
- Consultar el manual del vehículo y del programador para detalles específicos.

4.6 Recomendaciones Finales

- Actualizar regularmente el software del UPA V1.3.
- Realizar copias de seguridad antes de modificaciones.
- Realizar la programación en un entorno con buena iluminación y sin interferencias eléctricas.
- Seguir estrictamente el manual del fabricante para evitar daños.

Conclusiones

La investigación permitió identificar que los inmovilizadores utilizados en vehículos modernos pueden clasificarse en sistemas transponder, RFID, inmovilizadores integrados en la ECU, y módulos independientes con memorias EEPROM o Flash. En el caso del Chevrolet Sail, se utiliza un sistema con módulo separado que incorpora una EEPROM 25C020, lo cual permite su reprogramación directa.

Se determinó que el uso del programador UPA V1.3 para intervenir un módulo inmovilizador requiere una correcta identificación del tipo de memoria, la orientación de los pines, y el voltaje de operación (3.3V en el caso de la EEPROM 25C020). El software del programador permite realizar operaciones de lectura, edición y grabación de manera segura y precisa, siempre que se respeten los protocolos establecidos y la correcta conexión del hardware.

Durante el proceso de programación se identificaron errores frecuentes como: mala conexión de los pines, uso incorrecto del voltaje, detección fallida del chip o verificación inconsistente después de la escritura. Dichos errores se solucionan revisando soldaduras, ajustando la pinza SOIC, eligiendo correctamente el chip en el software, volviendo a leer el componente. La documentación de estos errores permite una rápida solución de fallos futuros y mejora la eficiencia del procedimiento.

Recomendaciones

Se recomienda utilizar un escáner automotriz antes y después del proceso de programación para verificar códigos de error relacionados con el inmovilizador y confirmar el correcto funcionamiento del sistema tras la reprogramación.

Mantener actualizado el software del UPA V1.3 garantiza una mayor compatibilidad con diversos chips y mejora la detección y escritura de memorias EEPROM modernas.

Se sugiere que los técnicos automotrices se capaciten constantemente en el uso de programadores y lectura de diagramas electrónicos, ya que la programación de inmovilizadores requiere conocimientos técnicos especializados.

Bibliografía

- Aguirre Guevara, C. P., & Estévez Andrade, E. D. (2022). Procesos de codificación en módulos del sistema UDS de grupo VAG (Bachelor's thesis).
- Amaguaña Pupiales, L. C., & Salazar Yandun, J. E. (2023). Proceso de sincronización y programación de Computadores ME17 de sail mediante protocolo j2534 y modificación de archivos de memoria (Bachelor's thesis).
- Arciniega García, F. R., & Fueltan Fueltan, J. A. (2020). Reprogramación de funciones especiales en vehículos modernos aplicando Ingeniería inversa a registros de memorias y microcontroladores de módulos automotrices (Bachelor's thesis).
- Berrezueta, M. F. G., Mena, A. F. L., Torres, P. W. M., & Estrella-Guayasamín, M. X. (2023). Programación del transponder en sistemas inmovilizadores para automóviles tipo M1. *South Florida Journal of Development*, 4(2), 857-866.
- Buitrón Mejía, S. E. (2021). Desarrollo de metodología para decodificación, programación y hermanación de módulos inmovilizadores automotrices (Bachelor's thesis).
- Chalacán Erazo, E. F., & Fuelagan, J. J. (2021). Procesos de programación en banco de sistemas inmovilizadores mediante modificación de archivos en módulos de vehículos de procedencia China (Bachelor's thesis).
- Chávez Calderón, L. F., & Flores Arteaga, J. D. (2014). Construcción e implementación de un banco didáctico de un sistema de inmovilizador con transponder para la Escuela de Ingeniería Automotriz (Bachelor's thesis, Escuela Superior Politécnica de Chimborazo)
- Estrella-Guayasamín, M., Vivar Quiroz, V., Delgado Quinto, A., & Gomez Berrezueta, F. (29 de Abril de 2025). Springer. https://link.springer.com/chapter/10.1007/978-3-031-87065-1_21.

- Guerrero Guevara, J. C. (2014). Análisis de la programación del sistema inmovilizador mediante el protocolo J2534 para vehículos Hyundai Accent (Bachelor's thesis, Universidad del Azuay).
- Landhäußer, F., Heinzelmann, M., Michalske, A., Susaeta, M. L., Grosser, M., Feger, J., ... & Walther, M. (2014). Electronic Control Unit (ECU). In Diesel Engine Management: Systems and Components (pp. 272-277). Wiesbaden: Springer Fachmedien Wiesbaden.
- Mircea, M., Croitorescu, V., & Oprean, M. (2016). Implementing New IT Security Solutions in Automotive Immobilizer Systems. *Applied Mechanics and Materials*, 822, 354-359.
- Sandoval Villamar, E. D., & Gómez Berrezueta, F. T. (2022). *Diagnóstico de Principales Averías del Sistema Inmovilizador Automotriz (INMO)* (Doctoral dissertation, GUAYAQUIL/UIDE/2022).
- Villca Renfijo, W. Implementación de un sistema de seguridad y monitoreo permanente mediante control automatizado a través de una aplicación en Android para el motor Toyota 2AZ-FE (Doctoral dissertation).
- Wouters, L., Van den Herrewegen, J., D Garcia, F., Oswald, D., Gierlichs, B., & Preneel, B. (2020). Dismantling DST80-based immobiliser systems. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(2), 99-127.



