Resolución de una Problemática

Actividad Integradora:

***Alumno***: Mariscal Rodríguez Omar Jesús

***Código de Estudiante:*** 220858478

***Maestra:*** Sabrina Lizeth Vega

***Fecha de Elaboración:*** 11 de Mayo de 2025

***Programación orientada a Objetos***

***Clave: I5289 Grupo: 195114***

***Martes y Jueves: 11:00am a 12:55pm***

Contenido

[Delimitación del Problema 3](#_Toc196606500)

[Título del Problema 3](#_Toc196606501)

[Delimitación del Problema 3](#_Toc196606502)

[Planteamiento del Problema 3](#_Toc196606503)

[Planificación con el Diseño de Diagramas UML 5](#_Toc196606504)

[Diagrama de Clases 5](#_Toc196606505)

[Diccionario de Clases 6](#_Toc196606506)

[Diagrama de Casos de Uso 9](#_Toc196606507)

[Diagrama de Comunicación 10](#_Toc196606508)

[Diagrama de Actividades 11](#_Toc196606509)

[Conclusiones de la Actividad 12](#_Toc196606510)

[Qué fue lo más relevante que aprendió en esta actividad 13](#_Toc196606511)

# Delimitación del Problema

Para este Producto Integrador, se pondrá en práctica ***todos los conocimientos*** y el viaje a lo largo de este curso, pasando por los ***pilares de la Programación Orientada a Objetos*** y todas las ***herramientas que hemos analizado*** para darle ***solución a una problemática determinada***; a continuación, se delimita la problemática que se resolverá y la implementación de la misma.

## Título del Problema

"Falta de un ***sistema integral*** para la ***gestión segura y eficiente*** de ***usuarios, vehículos y registros de mantenimiento*** automotriz"

## Delimitación del Problema

Actualmente, muchos talleres automotrices, flotas vehiculares y pequeñas empresas de mantenimiento ***carecen de herramientas digitales que les permitan gestionar de manera segura, eficiente y estructurada la información crítica*** relacionada con sus operaciones. En particular, los sistemas tradicionales (o la ausencia de ellos) presentan dificultades para:

* ***Registrar usuarios y distinguir sus roles*** (por ejemplo, administradores, técnicos o usuarios generales).
* ***Controlar los permisos de acceso y edición de información sensible***, como datos de vehículos o historial de mantenimientos.
* ***Mantener la trazabilidad de los servicios realizados*** a los vehículos.
* Organizar y consultar el historial de mantenimientos de manera rápida y segura.
* ***Generar reportes de manera rápida*** sobre los distintos ***mantenimientos que se le han realizado*** a sus vehículos.

Estas limitaciones conllevan una ***falta de control, errores humanos frecuentes, duplicidad o pérdida de información*** y, en general, ***ineficiencias*** que afectan directamente la calidad del servicio, la administración de los recursos y la satisfacción de los clientes.

## Planteamiento del Problema Y Nombre del Programa

La ausencia de un ***sistema de gestión especializado*** con control de usuarios, y en consecuencia, la realización de estas tareas de forma manual provoca una serie de problemáticas específicas:

* ***No se puede restringir adecuadamente quién tiene acceso a información crítica***, permitiendo que cualquier usuario pueda ***modificar, eliminar o introducir datos importantes*** sin supervisión.
* El ***registro y seguimiento de mantenimientos*** de los vehículos resulta ***disperso***, ***inconsistente o vulnerable a manipulaciones***, dificultando la construcción de un historial confiable de servicios.
* El proceso de ***generación de reportes sobre mantenimientos***, vehículos o usuarios es ***manual, lento e ineficiente***, complicando la toma de decisiones estratégicas.
* La ***falta de estandarización en la entrada y almacenamiento*** de datos ***incrementa los errores humanos y disminuye la capacidad de escalar el servicio*** de forma ordenada.

Ante este contexto, surge la necesidad de utilizar la tecnología como una herramienta para automatizar y facilitar todos estos procesos manuales, ante los cuales, se ha desarrollado el siguiente programa de nombre:

***GIVAM***: Sistema de Gestión Inteligente de Vehículos y Administración de Mantenimiento

Este sistema se concibe como una solución para los conflictos antes mencionados y proporcionar una base sólida para una escalabilidad del negocio

* ***Proporcionar una plataforma digital estructurada*** para el registro, consulta y edición controlada de ***usuarios, vehículos y mantenimientos.***
* ***Distinguir niveles de acceso entre administradores y mecánicos*** mediante autenticación y validación de roles.
* ***Registrar y encriptar las contraseñas*** con las que se accedan al programa.
* ***Llevar el registro de los datos en formato .csv*** para facilitar su gestión y ***complementar la herramienta con programas como Excel*** para la visualización del almacenamiento del programa.
* Facilitar la ***generación de reportes y la exportación de información en formato .txt***, promoviendo la trazabilidad y el análisis de datos.
* Brindar un ***menú amigable*** que favorezca su uso incluso para ***personal con conocimientos básicos en informática.***

En resumen, ***GIVAM busca resolver la falta de control, seguridad y eficiencia en la administración de datos relacionados con usuarios, vehículos y mantenimientos dentro del contexto automotriz,*** optimizando así los procesos de gestión y mejorando la calidad operativa de los negocios que lo implementen.

# Planificación con el Diseño del Programa

El primer paso para la solución tecnológica es identificar el problema y delimitarlo para qué es puntualmente lo que queremos solucionar y/o automatizar, esta información está anteriormente en este documento, ahora, el desarrollo de GIVAM se dividió en 4 fases principales; las primeras fases del desarrollo corresponden al núcleo del programa, lo que esencialmente necesita para su primer puesta en marcha, y las demás son las funcionalidades extra de interés que aportan nutritivamente a los focos que GIVAM pretende solucionar:

|  |  |  |  |
| --- | --- | --- | --- |
| Fase | Nombre de la Fase | Resumen | Puntos a Desarrollar |
| 1 | ***Proceso de Abstracción*** | Capturar los objetos principales del problema y cómo interaccionarán entre ellos | ***Abstracción Primaria***   * Delimitar qué objetos intervienen en cada parte de la solución * Pensar en qué métodos y cómo se comunicarán entre los objetos * Realizar una primera versión del diagrama de clases enfocada en la estructura principal |
| 2 | ***Desarrollo de la Estructura Central de Funcionamiento de GIVAM*** | Estas funcionalidades forman el núcleo del sistema y deben estar presentes sí o sí para que funcione correctamente: | ***Gestión de Usuarios***   * Registro de Usuarios con Roles (Administrados, Mecánico) * Login con Validación de Credenciales * Diferenciación de permisos por tipo de usuario   ***Gestión de Vehículos***   * Alta de Vehículos * Consulta General e Individual * Modificación de Vehículos * Baja de Vehículos   ***Gestión de Mantenimientos***   * Registrar Mantenimiento (preventivo y correctivo) * Asignar mantenimiento a un vehículo * Consultar mantenimiento por Vehículo   ***Persistencia de Datos***   * Guardar y cargar datos de vehículos, usuarios y mantenimientos en archivos .csv o .txt   ***Menú Interactivo***   * Menús dinámicos según el tipo de usuario * Validaciones básicas de entrada * Flujo lógico entre las funcionalidades del sistema |
| 3 | ***Lógica y edición refinada*** | Estas funcionalidades mejoran la experiencia y aportan más potencia al sistema, pero pueden desarrollarse después del núcleo | ***Historial y Seguimiento***   * Historial completo de mantenimiento por vehículo * Consulta de mantenimientos por fecha * Consulta de mantenimientos por tipo   ***Sistema de Búsqueda y Edición***   * Búsqueda vehículos por placa, tipo, marca * Buscar mantenimientos por ID o por los realizados a un Vehículo * Implementar la edición del propio perfil * Administradores sean capaces de editar datos de Vehículos y Mantenimientos Registrados |
| 4 | ***Seguridad y Reportes*** | Estas funcionalidades dan un plus profesional y podrían desarrollarse al final o como mejora posterior si el tiempo lo permite: | ***Seguridad***   * Encriptación de contraseñas mediante un Hash simple   ***Reportes y Exportación***   * Generar reporte de mantenimientos de un periodo específico * Reporte por vehículo específico * Exportar historial de mantenimiento a .txt |

Tener una planificación por fases mejora enormemente la organización de un proyecto grande, además de esto, el organizarlo de tal manera, que las funcionalidades primarias sean las primeras en desarrollarse, nos da un seguro contra el tiempo, esto es, si el proyecto tuviera que ser entregado por distintas circunstancias antes de tiempo sin desarrollar aún todas las funcionalidades que estaban previstas, el proyecto después de un punto crítico en el desarrollo (en el caso de GIVAM es la segunda fase), estaría funcionando, por lo que podría ser entregado aún si el tiempo no dio para planificar .

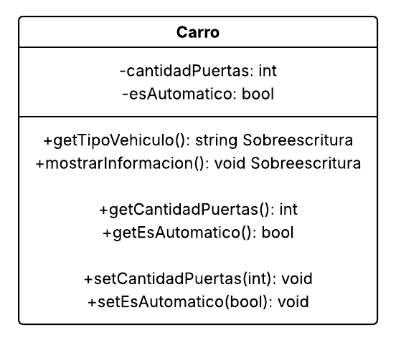
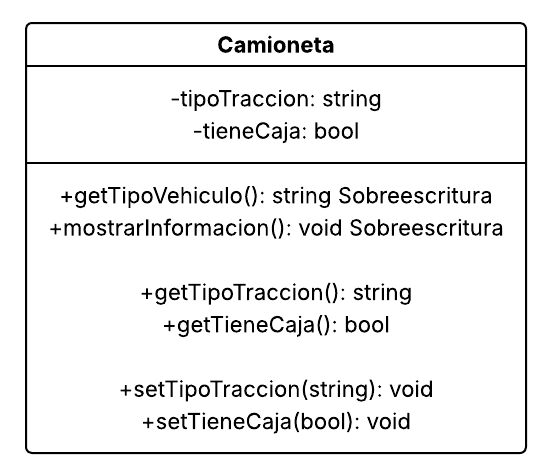
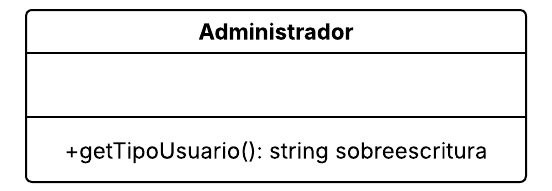
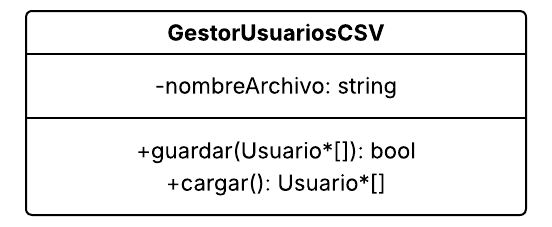
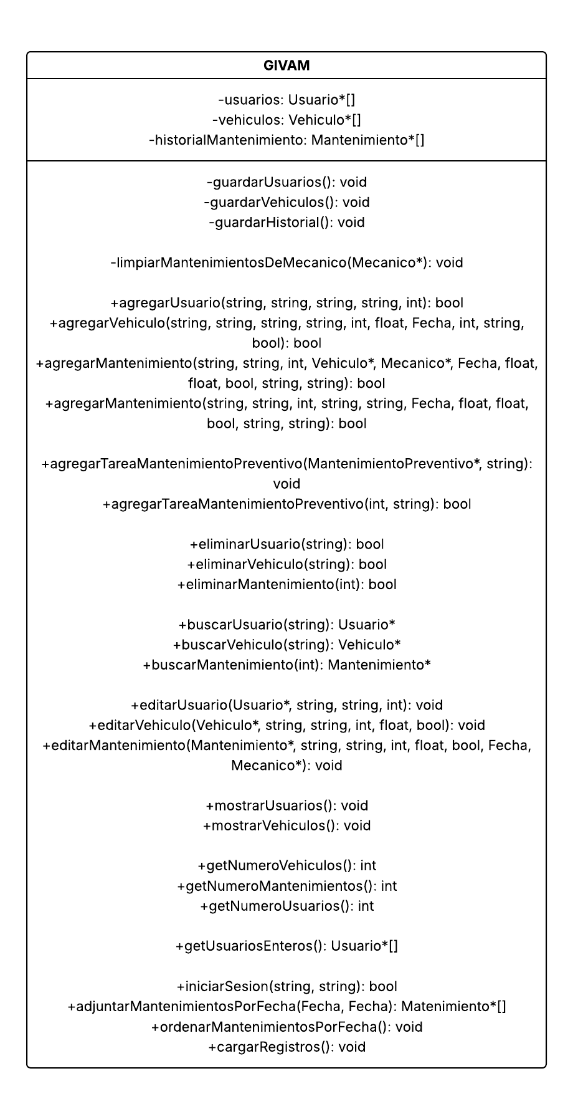
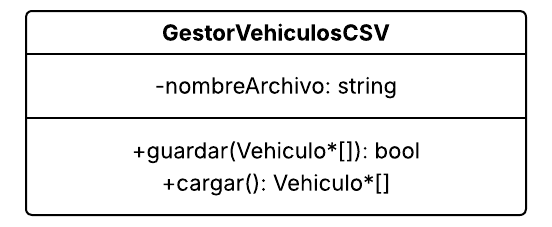
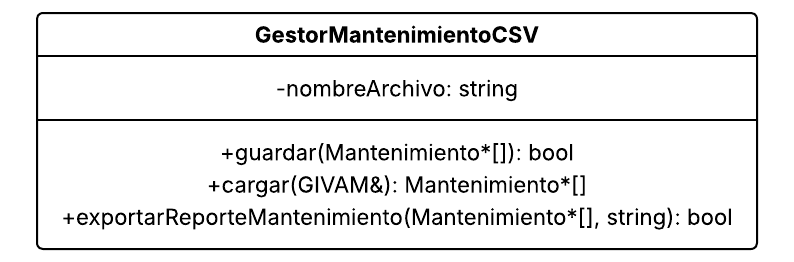
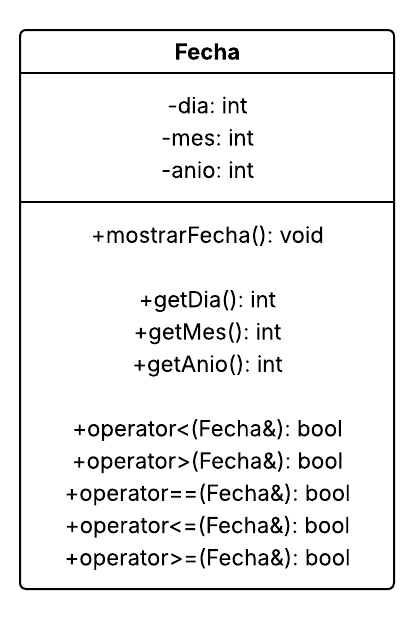
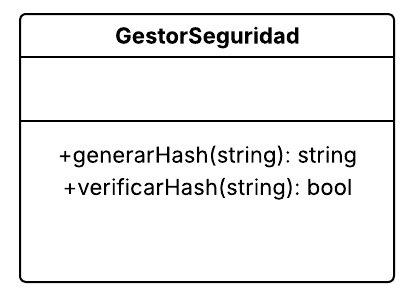
A continuación, se anexa un link al repositorio de GitHub donde, a partir de la primera versión del producto integrador, se fueron anexando los cambios registrados, llevando un control de las versiones, no solo del programa, sino de la documentación en general, incluyendo los diagramas y viendo con comentarios los cambios que ha sufrido el proyecto:

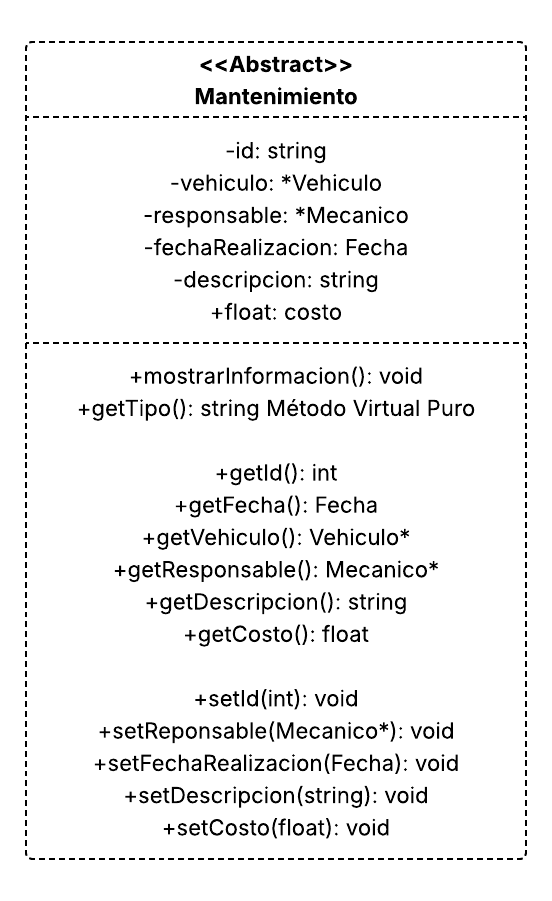
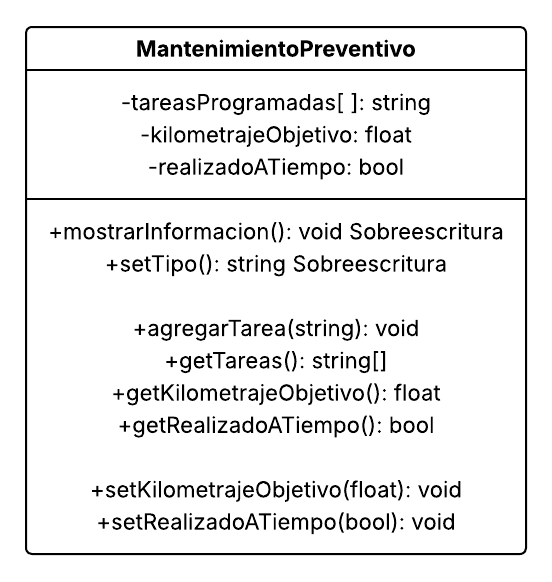
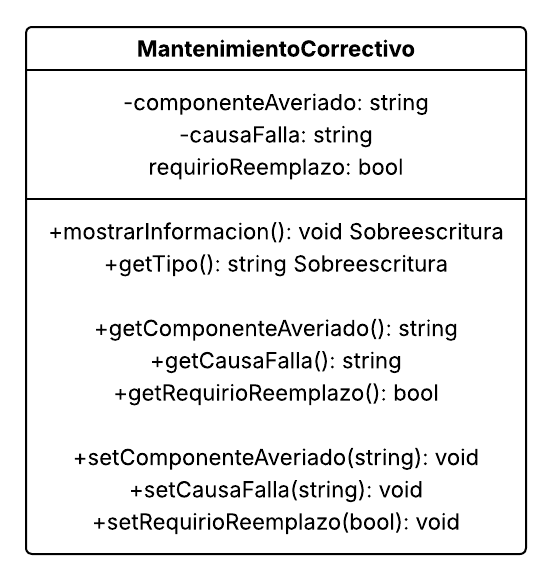
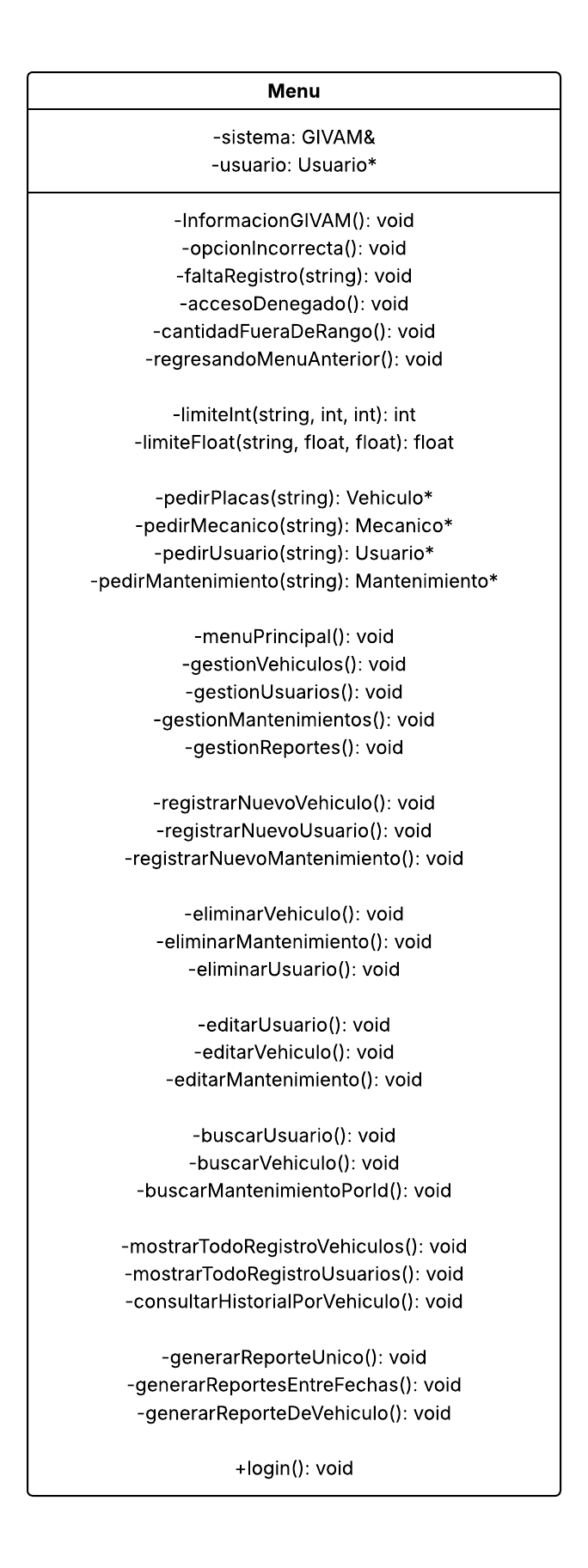
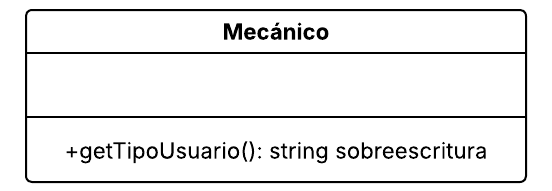
<https://github.com/OmarMariscal/Programaci-n-Orientada-a-Objetos>

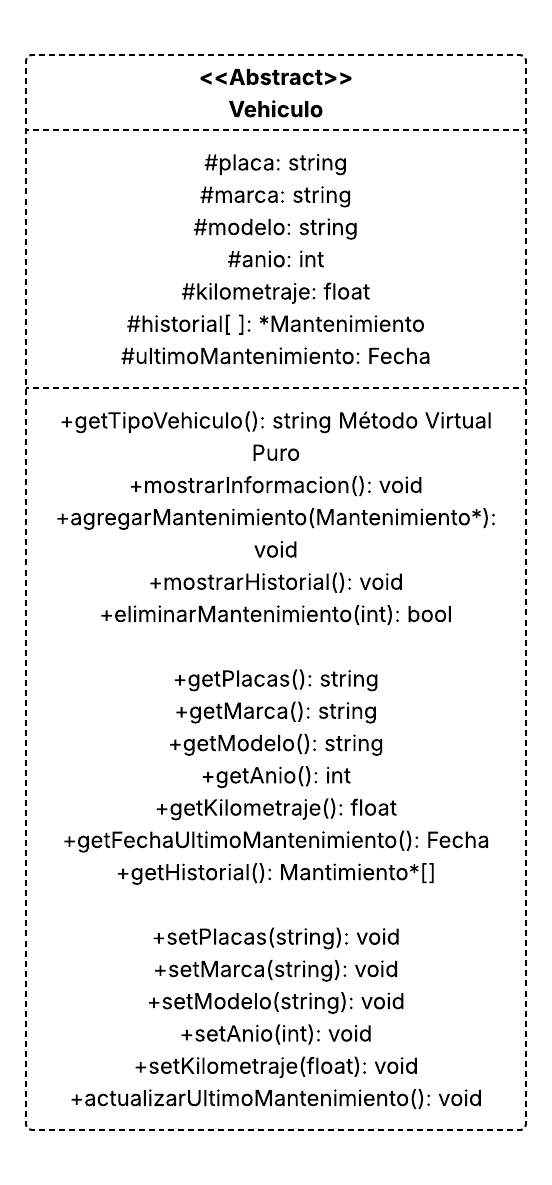
En suma a todo este proceso de planificación del desarrollo, entran los diagramas UML, una convención de modelado que nos ayuda mucho a clarificar y previsualizar cómo será el programa en distintos niveles y a través de distintas vistas, ayudando tanto a los clientes que verán que se supone que el programa cubre y que no, y a los desarrolladores para guiar su proyecto en el desarrollo, trabajar de manera colaborativa teniendo todos ya una idea más cerca de lo que globalmente el programa será y para que el funcionamiento del mismo sea más claro a otros programadores al explicar funciones específicas a nivel de comunicación, interacción etc, para su mantenimiento y expansión.

## Diagrama de Clases

## Diccionario de Clases

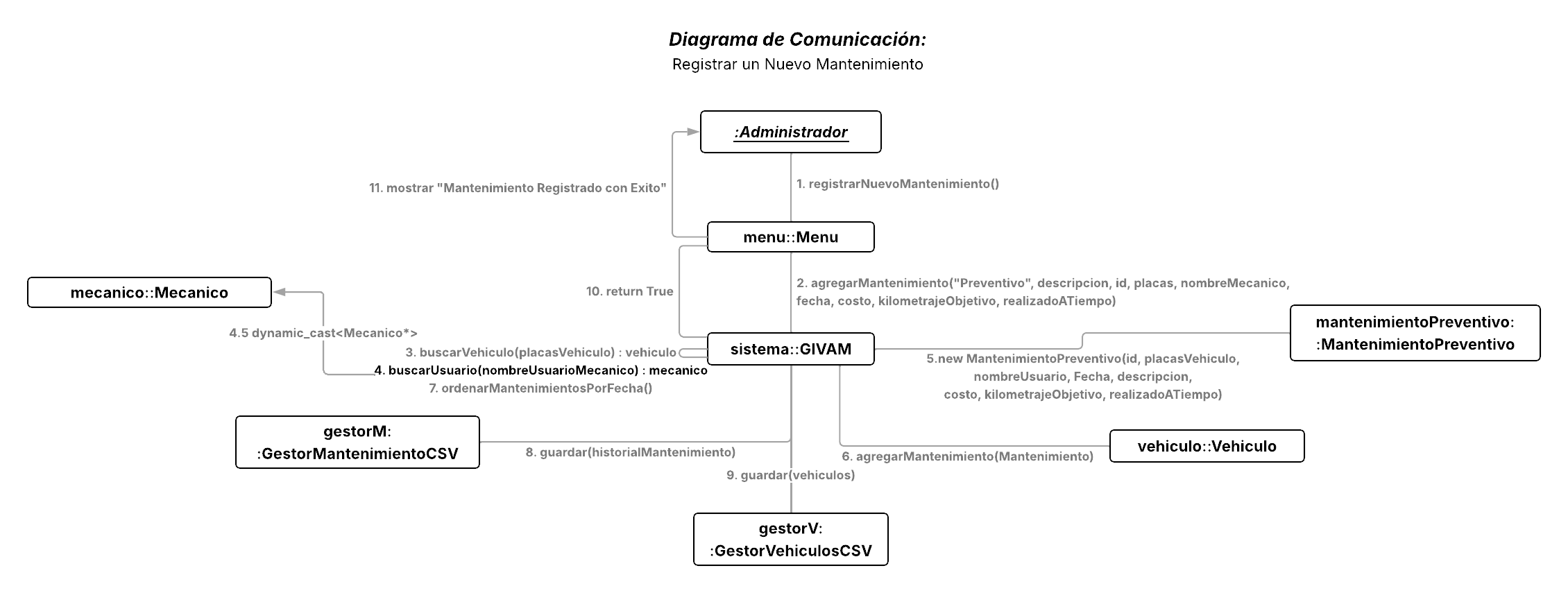






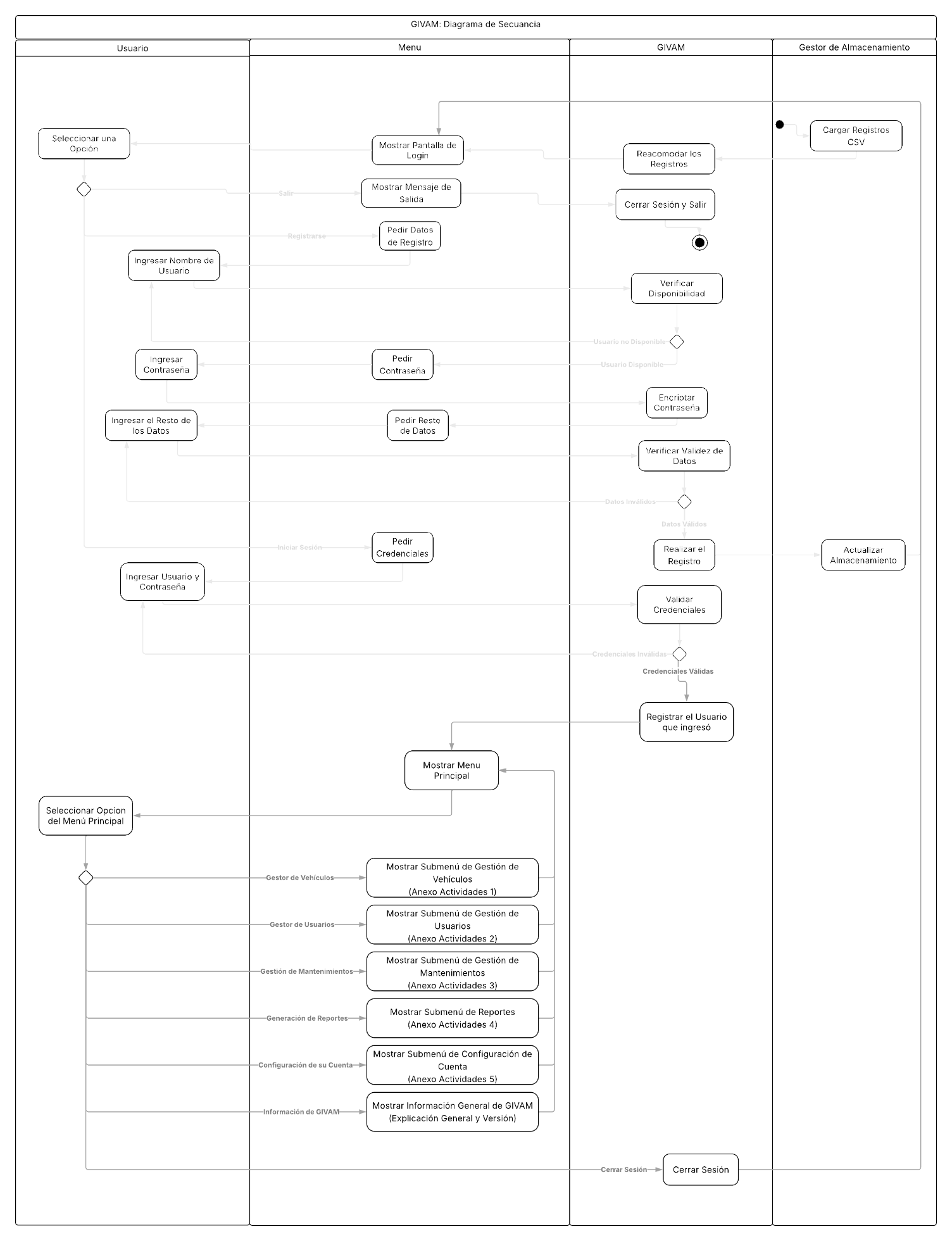
## Diagrama de Casos de Uso

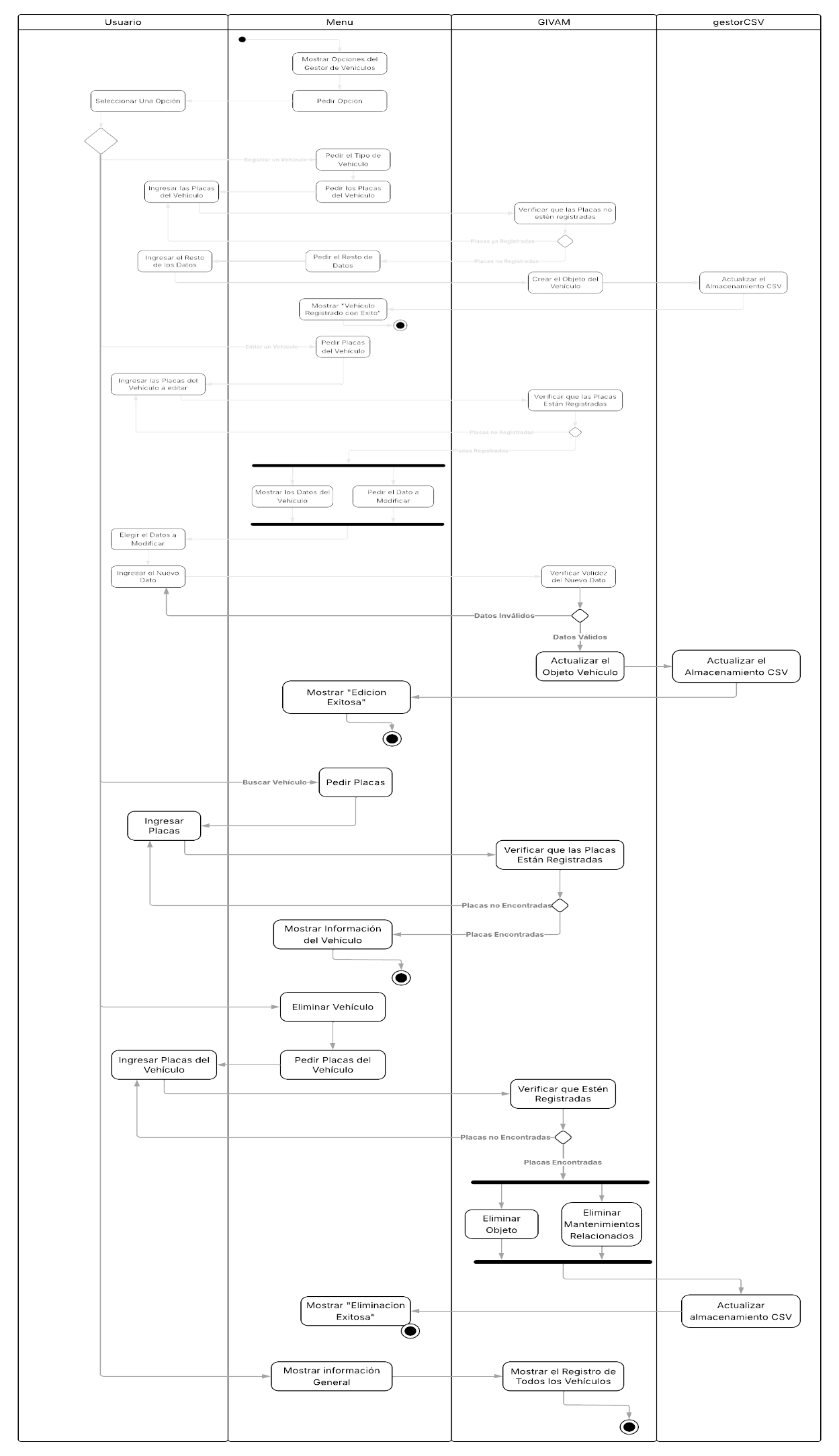
## Diagrama de Comunicación



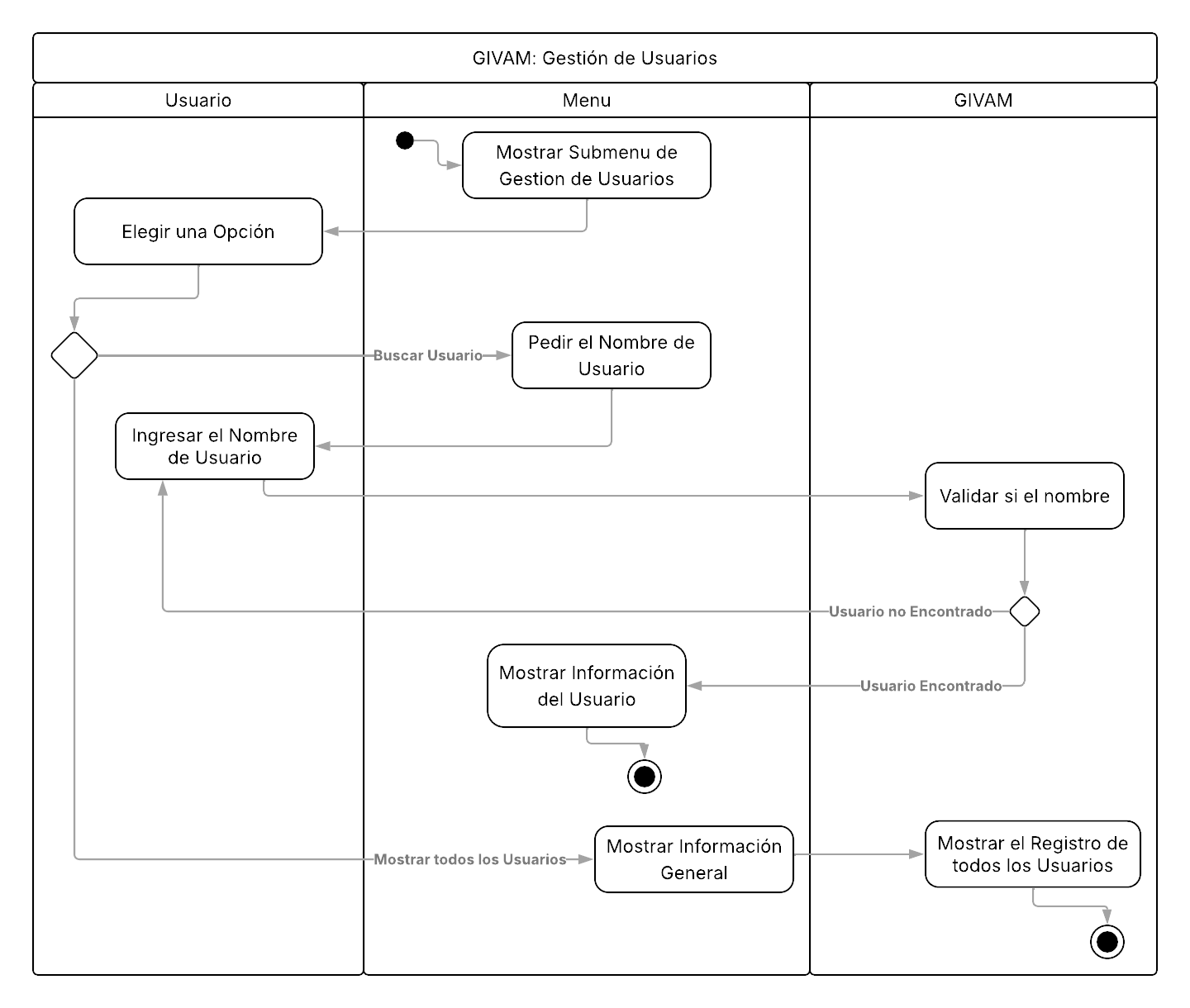
El diagrama organiza como es que los distintos objetos se comunican entre sí y qué mensajes se mandan para llevar a cabo la acción de registrar un nuevo mantenimiento e intervienen varias veces para que esta acción se lleve correctamente a cabo.

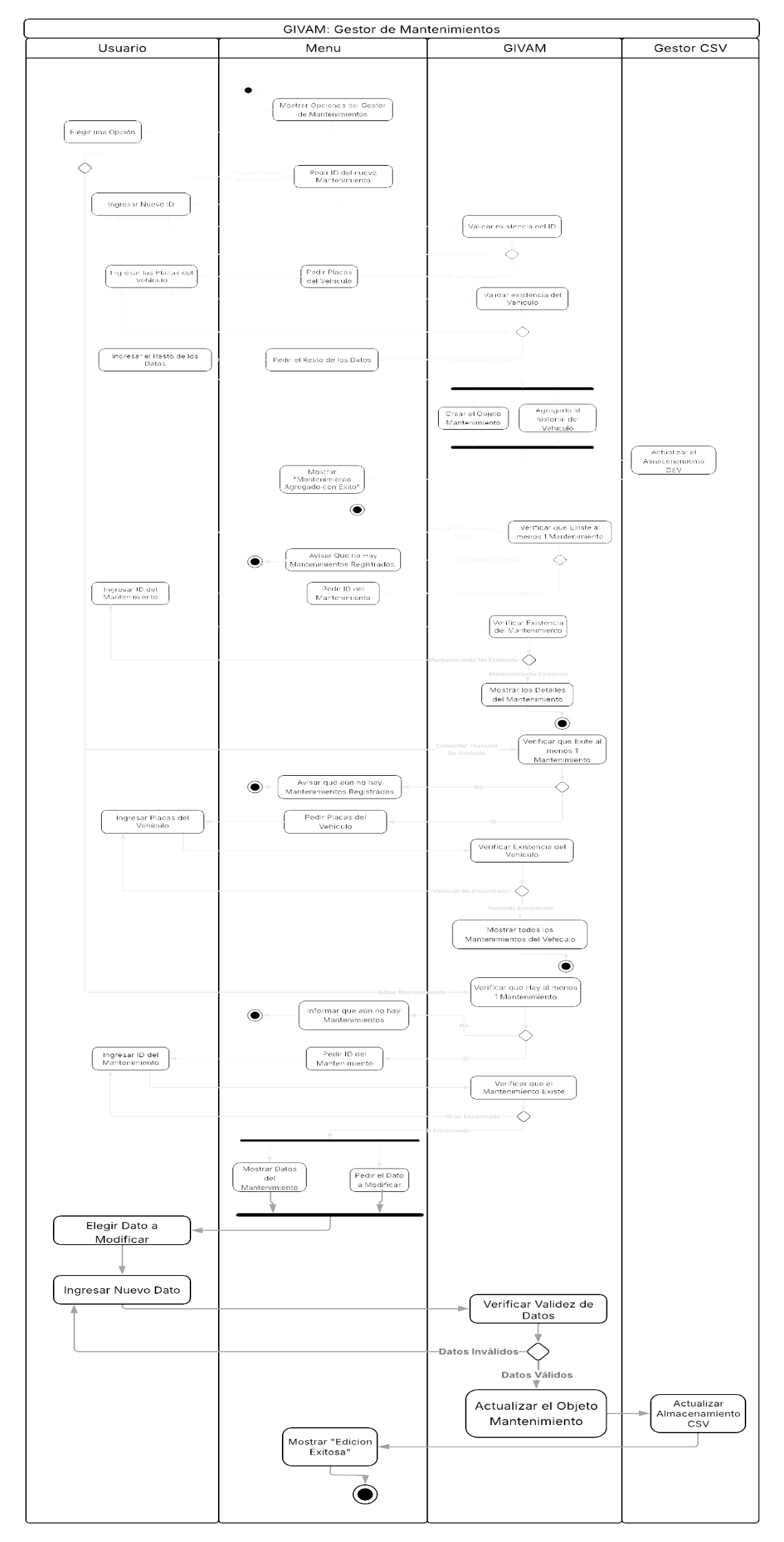
## Diagrama de Actividades

El diagrama de Actividades que a continuación se presenta, representa el flujo sintetizado del funcionamiento de todo el programa de GIVAM, por la extensión del mismo, el diagrama de actividades se ha segmentado en el diagrama principal que abarca desde la validación de credenciales, la elección del submenú hasta la salida del programa, los demás diagramas complementan las acciones de los distintos submenús:

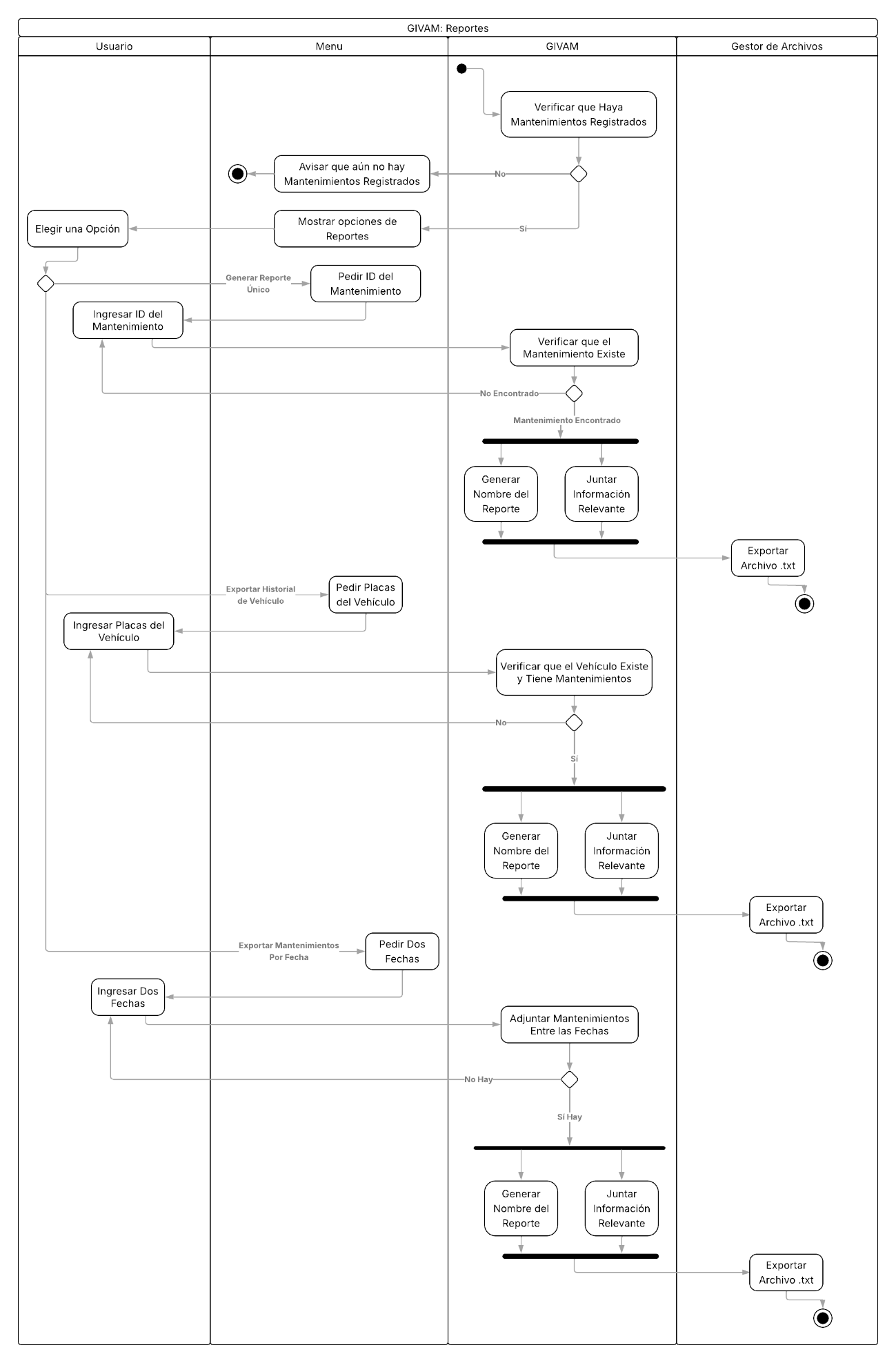
***Anexo Actividades 1:*** “Gestor de Vehículos”

Anexo Actividades 2: “Gestor de Usuarios”

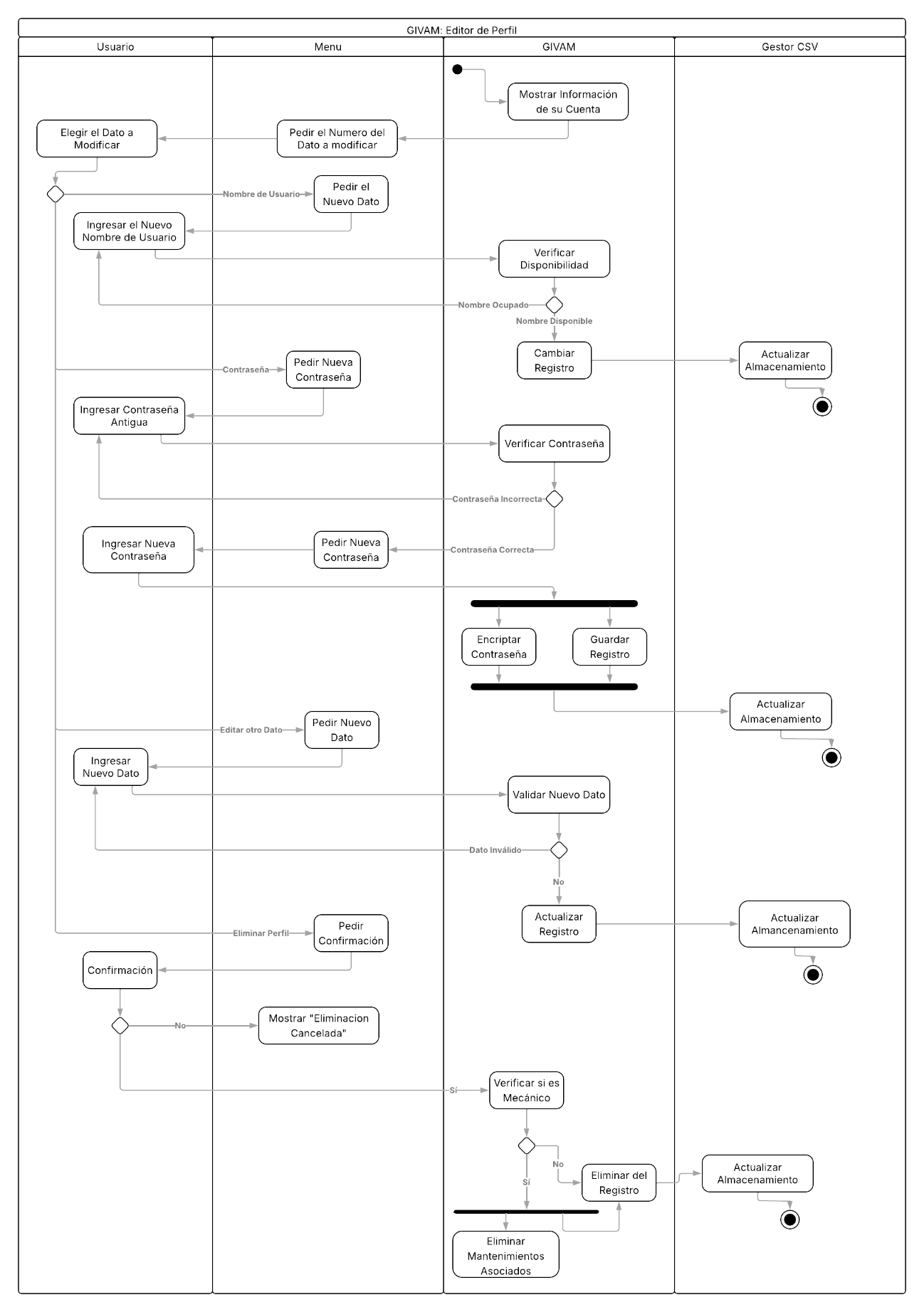


***Anexo Actividades 3:*** “Gestor de Mantenimientos”

Anexo Actividades 4: “Actividades de Reportes”



Anexo Actividades 5: “Configuración de la Cuenta”



# Conclusiones de la Actividad

# Qué fue lo más relevante que aprendió en esta actividad