

# Actividad Integradora:

## Resolución de una Problemática

*Programación orientada a Objetos*

*Clave: I5289 Grupo: 195114*

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



**Alumno:** Mariscal Rodríguez Omar Jesús

**Código de Estudiante:** 220858478

**Maestra:** Sabrina Lizeth Vega

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

# Contenido

---

Delimitación del Problema .....	3
Título del Problema.....	3
Delimitación del Problema.....	3
Planteamiento del Problema Y Nombre del Programa.....	3
Planificación con el Diseño del Programa.....	5
Diagrama de Clases.....	7
Diccionario de Clases.....	8
Diagrama de Casos de Uso .....	11
Diagrama de Comunicación.....	12
Diagrama de Actividades .....	13
Implementación de la Solución .....	19
Login y Registro de GIVAM .....	19
Gestión de Vehículos .....	22
Registrar un Vehículo .....	22
Editar un Vehículo .....	24
Buscar un Vehículo.....	25
Eliminar un Vehículo.....	25
Mostrar el Registro de Todos los Vehículos .....	26
Gestión de Usuarios .....	26
Buscar un Usuario Registrado .....	26
Mostrar Registro de Usuarios Registrados .....	27
Gestor de Mantenimientos .....	27
Registrar un Nuevo Mantenimiento.....	27
Buscar Mantenimiento por ID .....	29
Consultar Historial por Vehículo .....	29
Editar un Mantenimiento .....	30
Reportes .....	31
Exportar un Reporte Determinado .....	31
Exportar Reportes de un Vehículo .....	31
Exportar Mantenimientos Entre Fechas .....	32
Configuración de su Cuenta .....	33
Información de GIVAM.....	34
Cerrar Sesión.....	34
Control de Almacenamiento .....	34
Conclusiones de la Actividad .....	36

# 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	<b>Proceso de Abstracción</b>	Capturar los objetos principales del problema y cómo interaccionarán entre ellos	<b>Abstracción Primaria</b> <ul style="list-style-type: none"> <li>Delimitar qué objetos intervienen en cada parte de la solución</li> <li>Pensar en qué métodos y cómo se comunicarán entre los objetos</li> <li>Realizar una primera versión del diagrama de clases enfocada en la estructura principal</li> </ul>
2	<b>Desarrollo de la Estructura Central de Funcionamiento de GIVAM</b>	Estas funcionalidades forman el núcleo del sistema y deben estar presentes sí o sí para que funcione correctamente:	<b>Gestión de Usuarios</b> <ul style="list-style-type: none"> <li>Registro de Usuarios con Roles (Administrados, Mecánico)</li> <li>Login con Validación de Credenciales</li> <li>Diferenciación de permisos por tipo de usuario</li> </ul> <b>Gestión de Vehículos</b> <ul style="list-style-type: none"> <li>Alta de Vehículos</li> <li>Consulta General e Individual</li> <li>Modificación de Vehículos</li> <li>Baja de Vehículos</li> </ul> <b>Gestión de Mantenimientos</b> <ul style="list-style-type: none"> <li>Registrar Mantenimiento (preventivo y correctivo)</li> <li>Asignar mantenimiento a un vehículo</li> <li>Consultar mantenimiento por Vehículo</li> </ul> <b>Persistencia de Datos</b> <ul style="list-style-type: none"> <li>Guardar y cargar datos de vehículos, usuarios y mantenimientos en archivos .csv o .txt</li> </ul> <b>Menú Interactivo</b> <ul style="list-style-type: none"> <li>Menús dinámicos según el tipo de usuario</li> <li>Validaciones básicas de entrada</li> <li>Flujo lógico entre las funcionalidades del sistema</li> </ul>
3	<b>Lógica y edición refinada</b>	Estas funcionalidades mejoran la experiencia y aportan más potencia al sistema, pero pueden desarrollarse después del núcleo	<b>Historial y Seguimiento</b> <ul style="list-style-type: none"> <li>Historial completo de mantenimiento por vehículo</li> <li>Consulta de mantenimientos por fecha</li> <li>Consulta de mantenimientos por tipo</li> </ul> <b>Sistema de Búsqueda y Edición</b> <ul style="list-style-type: none"> <li>Búsqueda vehículos por placa, tipo, marca</li> <li>Buscar mantenimientos por ID o por los realizados a un Vehículo</li> <li>Implementar la edición del propio perfil</li> <li>Administradores sean capaces de editar datos de Vehículos y Mantenimientos Registrados</li> </ul>
4	<b>Seguridad y Reportes</b>	Estas funcionalidades dan un plus profesional y podrían desarrollarse al final o como mejora posterior si el tiempo lo permite:	<b>Seguridad</b> <ul style="list-style-type: none"> <li>Encriptación de contraseñas mediante un Hash simple</li> </ul> <b>Reportes y Exportación</b> <ul style="list-style-type: none"> <li>Generar reporte de mantenimientos de un periodo específico</li> <li>Reporte por vehículo específico</li> </ul>

- |  |  |  |
|--|--|--|
|  |  | <ul style="list-style-type: none"> <li>• Exportar historial de mantenimiento a .txt</li> </ul> |
|--|--|--|

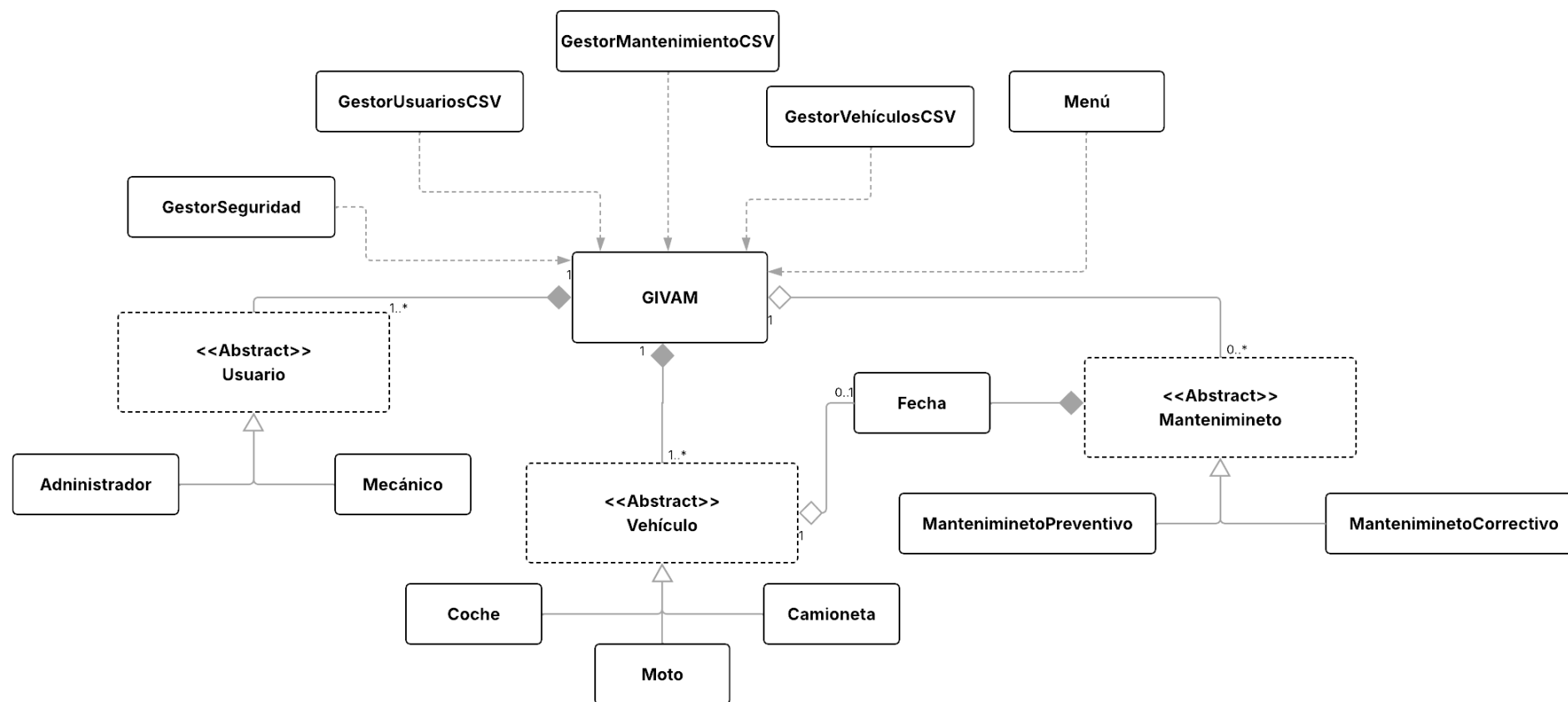
Tener una planificación por fases mejora enormemente la organización de un proyecto grande, además de esto, el organizarlo de tal manera, que las funcionalidades primarias sean las primeras en desarrollarse, nos da un seguro contra el tiempo, esto es, si el proyecto tuviera que ser entregado por distintas 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

Administrador
+getTipoUsuario(): string sobreescritura

Camioneta
-tipoTraccion: string -tieneCaja: bool
+getTipoVehiculo(): string Sobreescritura +mostrarInformacion(): void Sobreescritura
+getTipoTraccion(): string +getTieneCaja(): bool
+setTipoTraccion(string): void +setTieneCaja(bool): void

Carro
-cantidadPuertas: int -esAutomatico: bool
+getTipoVehiculo(): string Sobreescritura +mostrarInformacion(): void Sobreescritura
+getCantidadPuertas(): int +getEsAutomatico(): bool
+setCantidadPuertas(int): void +setEsAutomatico(bool): void

Fecha
-dia: int -mes: int -anio: int
+mostrarFecha(): void
+getDia(): int +getMes(): int +getAnio(): int
+operator<(Fecha&): bool +operator>(Fecha&): bool +operator==(Fecha&): bool +operator<=(Fecha&): bool +operator>=(Fecha&): bool

GestorMantenimientoCSV
-nombreArchivo: string
+guardar(Mantenimiento*[]): bool +cargar(GIVAM&): Mantenimiento*[] +exportarReporteMantenimiento(Mantenimiento*[], string): bool

GestorSeguridad
+generarHash(string): string +verificarHash(string): bool

GestorUsuariosCSV
-nombreArchivo: string
+guardar(Usuario*[]): bool +cargar(): Usuario*[]

GestorVehiculosCSV
-nombreArchivo: string
+guardar(Vehiculo*[]): bool +cargar(): Vehiculo*[]

GIVAM
-usuarios: Usuario*[] -vehiculos: Vehiculo*[] -historialMantenimiento: Mantenimiento*[]
-guardarUsuarios(): void -guardarVehiculos(): void -guardarHistorial(): void
-limpiarMantenimientosDeMecanico(Mecanico*): void
+agregarUsuario(string, string, string, string, int): bool +agregarVehiculo(string, string, string, string, int, float, Fecha, int, string, bool): bool +agregarMantenimiento(string, string, int, Vehiculo*, Mecanico*, Fecha, float, float, bool, string, string): bool +agregarMantenimiento(string, string, int, string, string, Fecha, float, float, bool, string, string): bool
+agregarTareaMantenimientoPreventivo(MantenimientoPreventivo*, string): void +agregarTareaMantenimientoPreventivo(int, string): bool
+eliminarUsuario(string): bool +eliminarVehiculo(string): bool +eliminarMantenimiento(int): bool
+buscarUsuario(string): Usuario* +buscarVehiculo(string): Vehiculo* +buscarMantenimiento(int): Mantenimiento*
+editarUsuario(Usuario*, string, string, int): void +editarVehiculo(Vehiculo*, string, string, int, float, bool): void +editarMantenimiento(Mantenimiento*, string, string, int, float, bool, Fecha, Mecanico*): void
+mostrarUsuarios(): void +mostrarVehiculos(): void
+getNumeroVehiculos(): int +getNumeroMantenimientos(): int +getNumeroUsuarios(): int
+getUsuariosEnteros(): Usuario*[]
+iniciarSesion(string, string): bool +adjuntarMantenimientosPorFecha(Fecha, Fecha): Mantenimiento*[] +ordenarMantenimientosPorFecha(): void +cargarRegistros(): void



<<Abstract>> Mantenimiento	
-id: string -vehiculo: *Vehiculo -responsable: *Mecanico -fechaRealizacion: Fecha -descripcion: string +float: costo	
+mostrarInformacion(): void +getTipo(): string Método Virtual Puro	
+getId(): int +getFecha(): Fecha +getVehiculo(): Vehiculo* +getResponsable(): Mecanico* +getDescripcion(): string +getCosto(): float	
+setId(int): void +setReponsable(Mecanico*): void +setFechaRealizacion(Fecha): void +setDescripcion(string): void +setCosto(float): void	

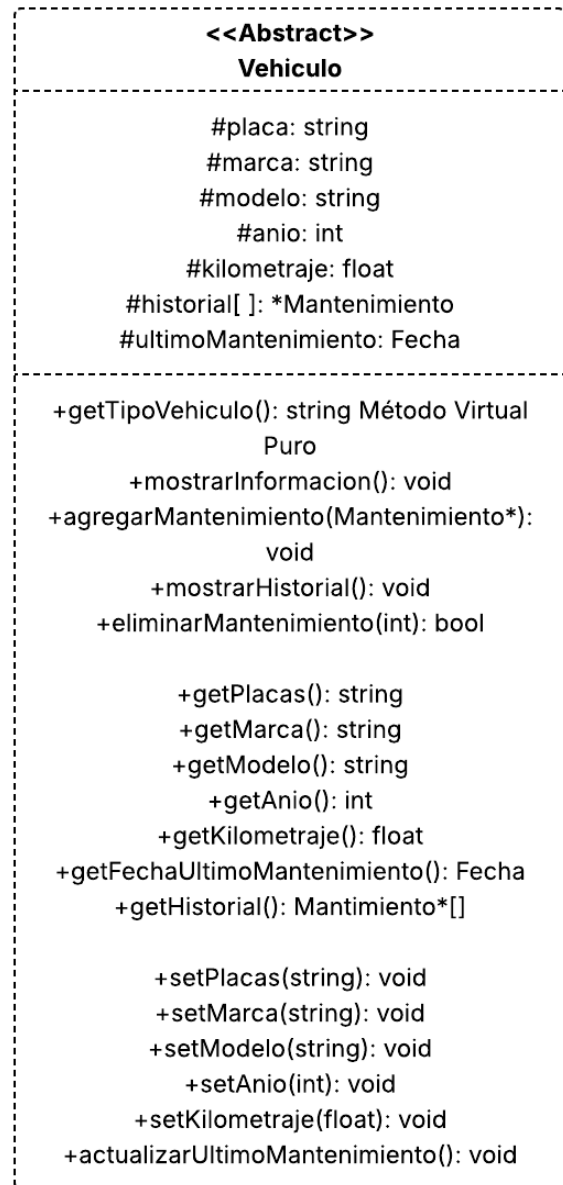
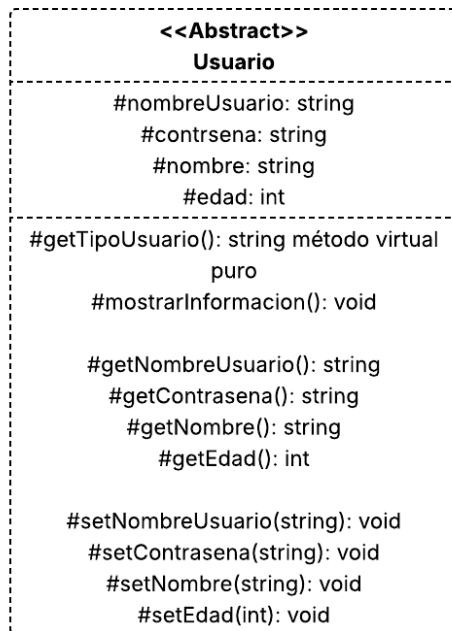
MantenimientoCorrectivo
-componenteAveriado: string -causaFalla: string requirioReemplazo: bool
+mostrarInformacion(): void Sobreescritura +getTipo(): string Sobreescritura
+getComponenteAveriado(): string +getCausaFalla(): string +getRequirioReemplazo(): bool
+setComponenteAveriado(string): void +setCausaFalla(string): void +setRequirioReemplazo(bool): void

MantenimientoPreventivo
-tareasProgramadas[ ]: string -kilometrajeObjetivo: float -realizadoATiempo: bool
+mostrarInformacion(): void Sobreescritura +setTipo(): string Sobreescritura
+agregarTarea(string): void +getTareas(): string[] +getKilometrajeObjetivo(): float +getRealizadoATiempo(): bool
+setKilometrajeObjetivo(float): void +setRealizadoATiempo(bool): void

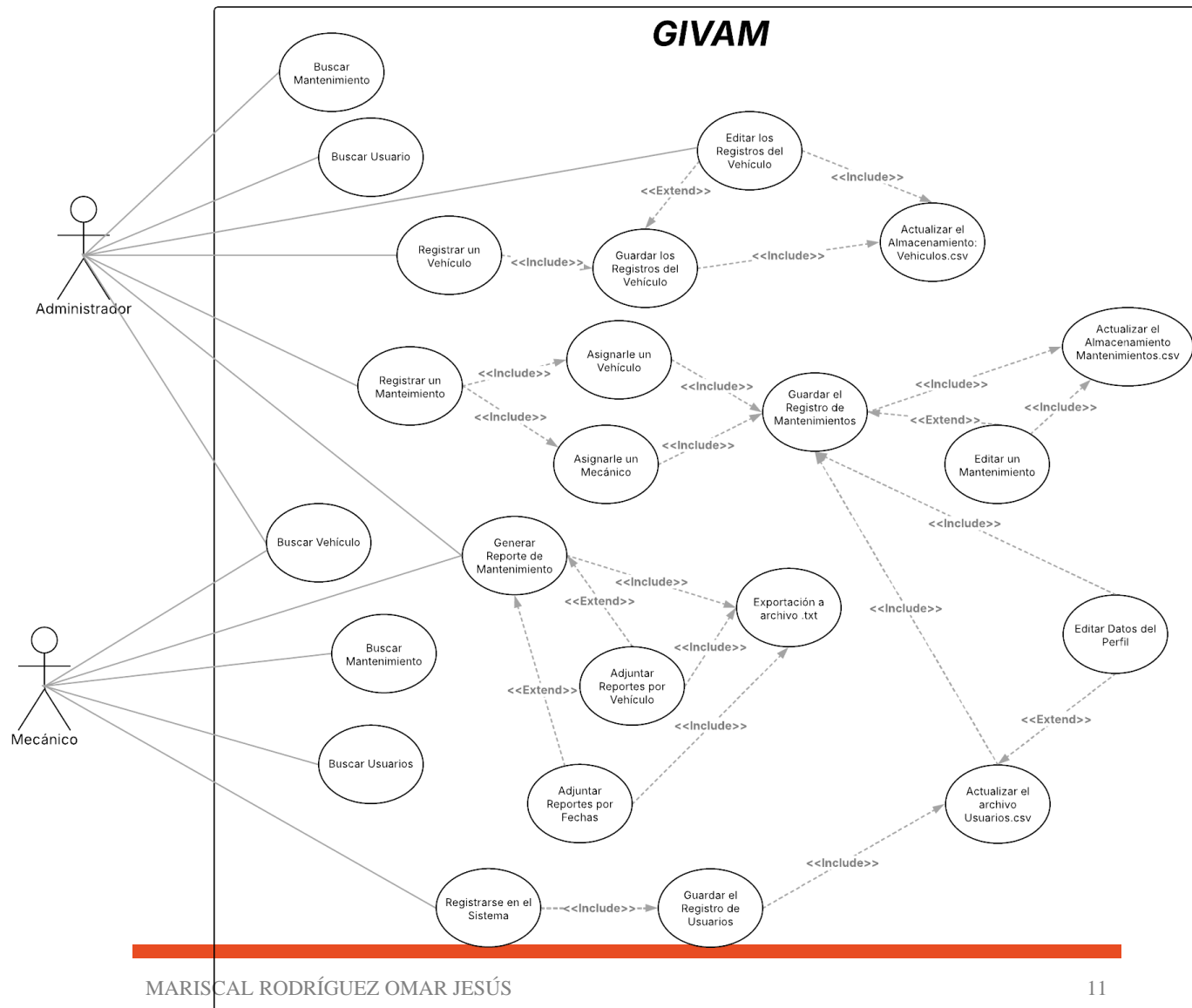
Mecánico
+getTipoUsuario(): string sobreescritura

Menu
-sistema: GIVAM& -usuario: Usuario*
-InformacionGIVAM(): void -opcionIncorrecta(): void -faltaRegistro(string): void -accesoDenegado(): void -cantidadFueraDeRango(): void -regresandoMenuAnterior(): void
-limiteInt(string, int, int): int -limiteFloat(string, float, float): float
-pedirPlacas(string): Vehiculo* -pedirMecanico(string): Mecanico* -pedirUsuario(string): Usuario* -pedirMantenimiento(string): Mantenimiento*
-menuPrincipal(): void -gestionVehiculos(): void -gestionUsuarios(): void -gestionMantenimientos(): void -gestionReportes(): void
-registrarNuevoVehiculo(): void -registrarNuevoUsuario(): void -registrarNuevoMantenimiento(): void
-eliminarVehiculo(): void -eliminarMantenimiento(): void -eliminarUsuario(): void
-editarUsuario(): void -editarVehiculo(): void -editarMantenimiento(): void
-buscarUsuario(): void -buscarVehiculo(): void -buscarMantenimientoPorId(): void
-mostrarTodoRegistroVehiculos(): void -mostrarTodoRegistroUsuarios(): void -consultarHistorialPorVehiculo(): void
-generarReporteUnico(): void -generarReportesEntreFechas(): void -generarReporteDeVehiculo(): void
+login(): void

Moto
-tipoMoto: string
+getTipoVehiculo(): string Sobreescritura +mostrarInformacion(): void Sobreescritura
+getTipoMoto(): string
+setTipoMoto(string): void



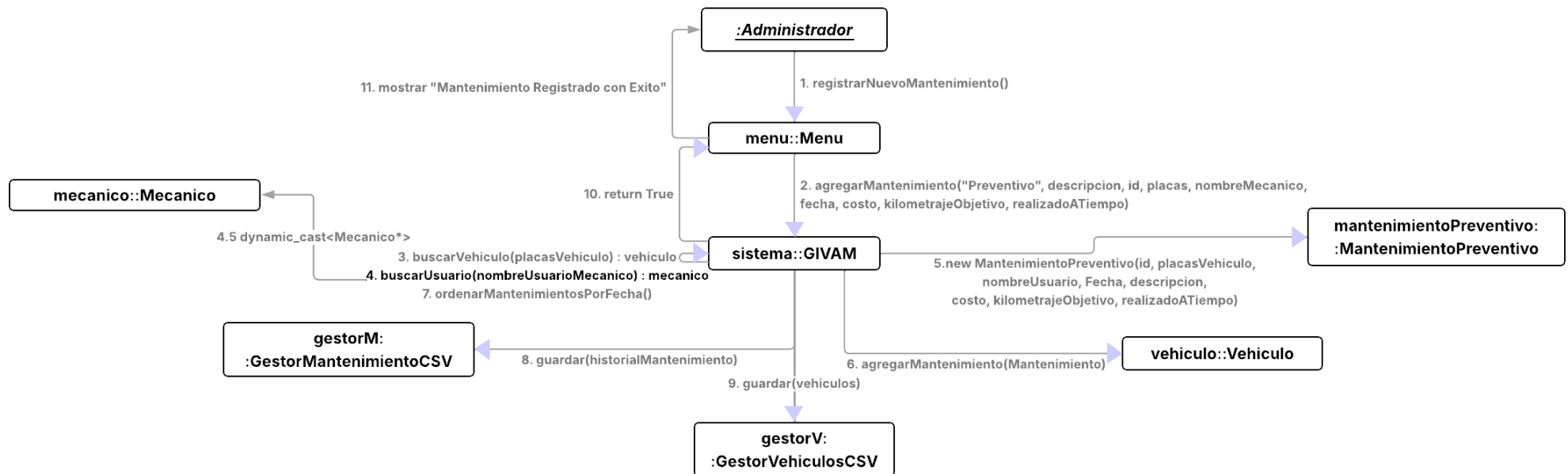
## Diagrama de Casos de Uso



## Diagrama de Comunicación

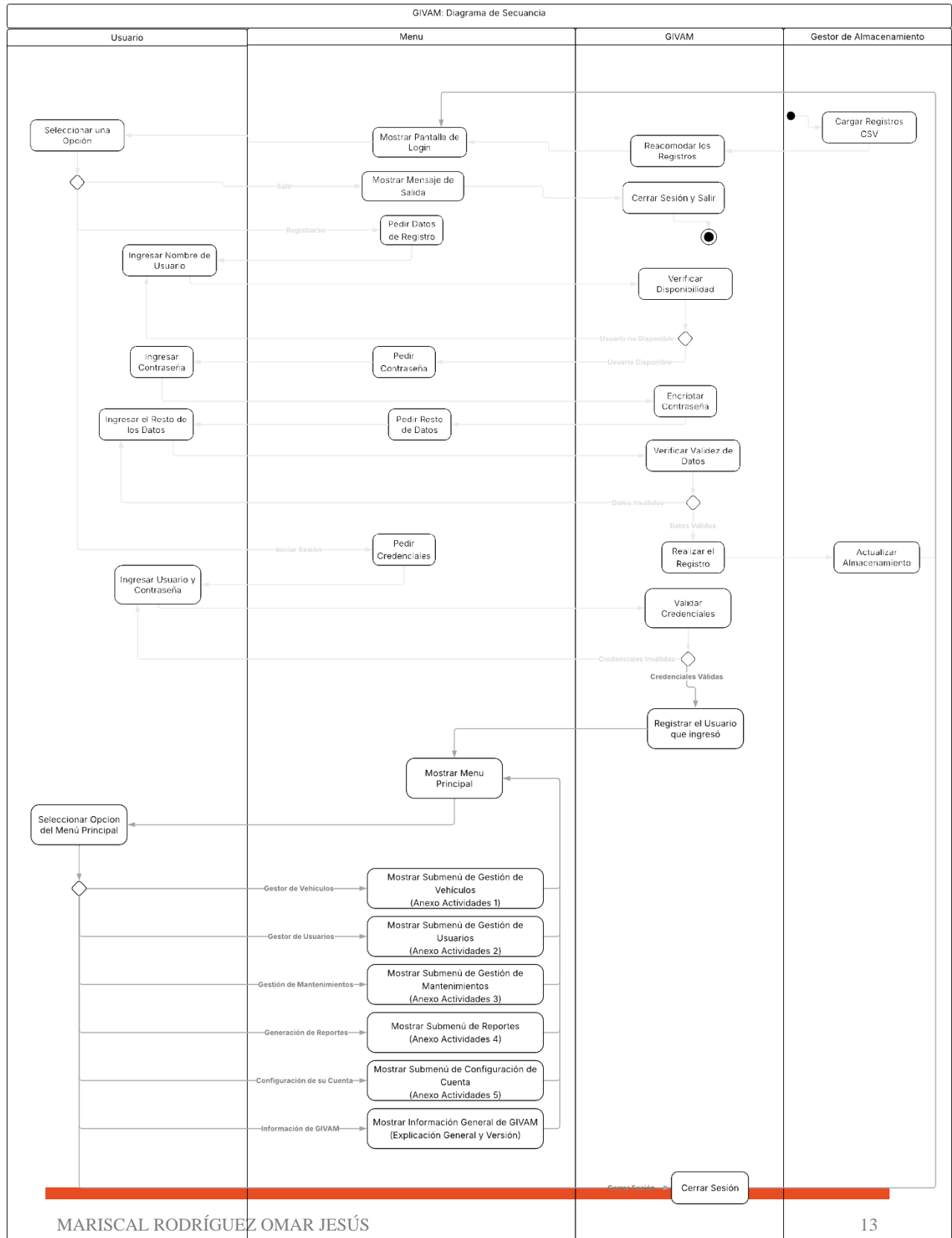
El diagrama organiza como es que los distintos objetos se comunican entre sí y qué mensajes se mandan para llevar a cabo la acción de registrar un nuevo mantenimiento e intervienen varias veces para que esta acción se lleve correctamente a cabo.

**Diagrama de Comunicación:**  
Registrar un Nuevo Mantenimiento

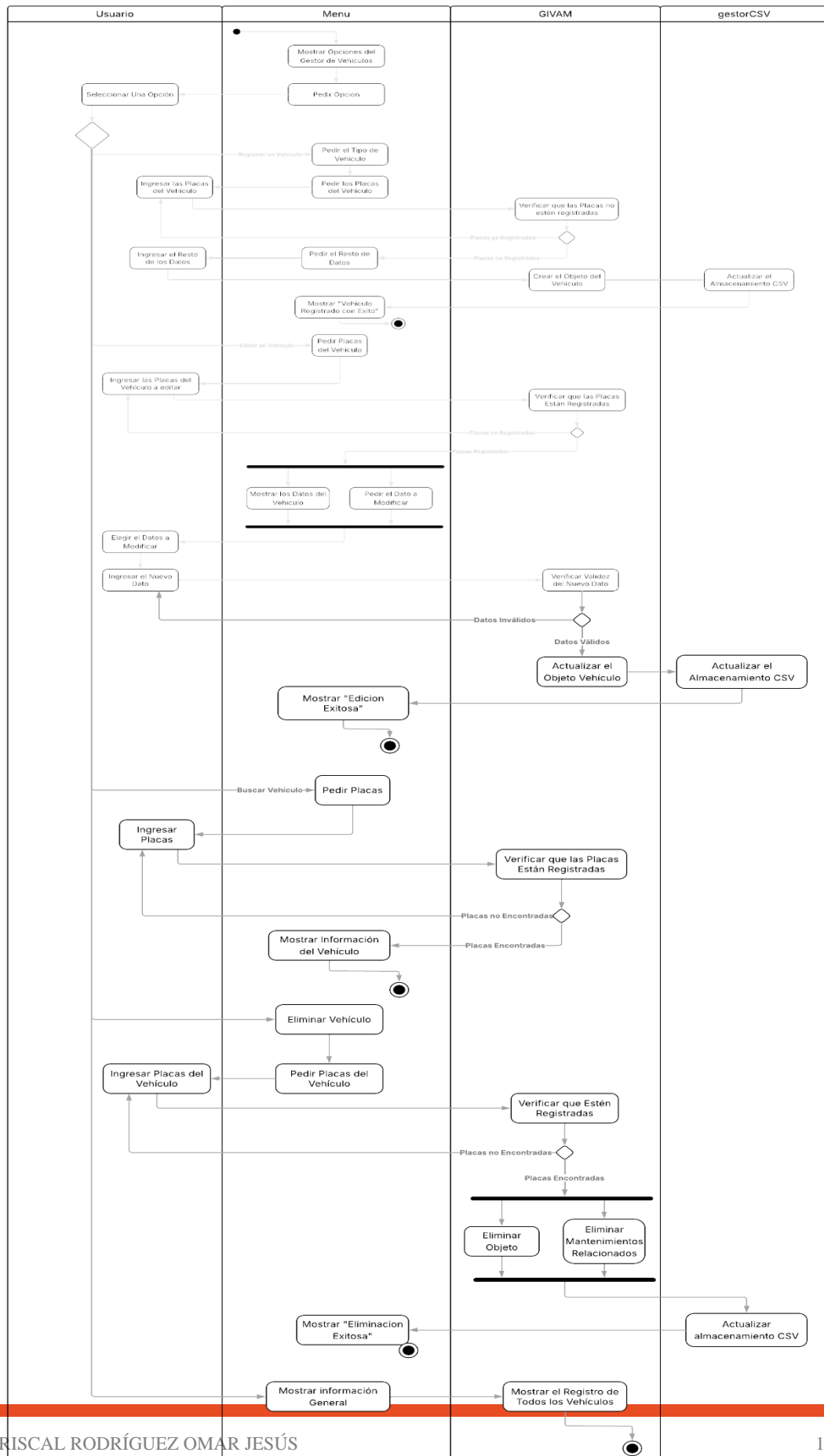


## Diagrama de Actividades

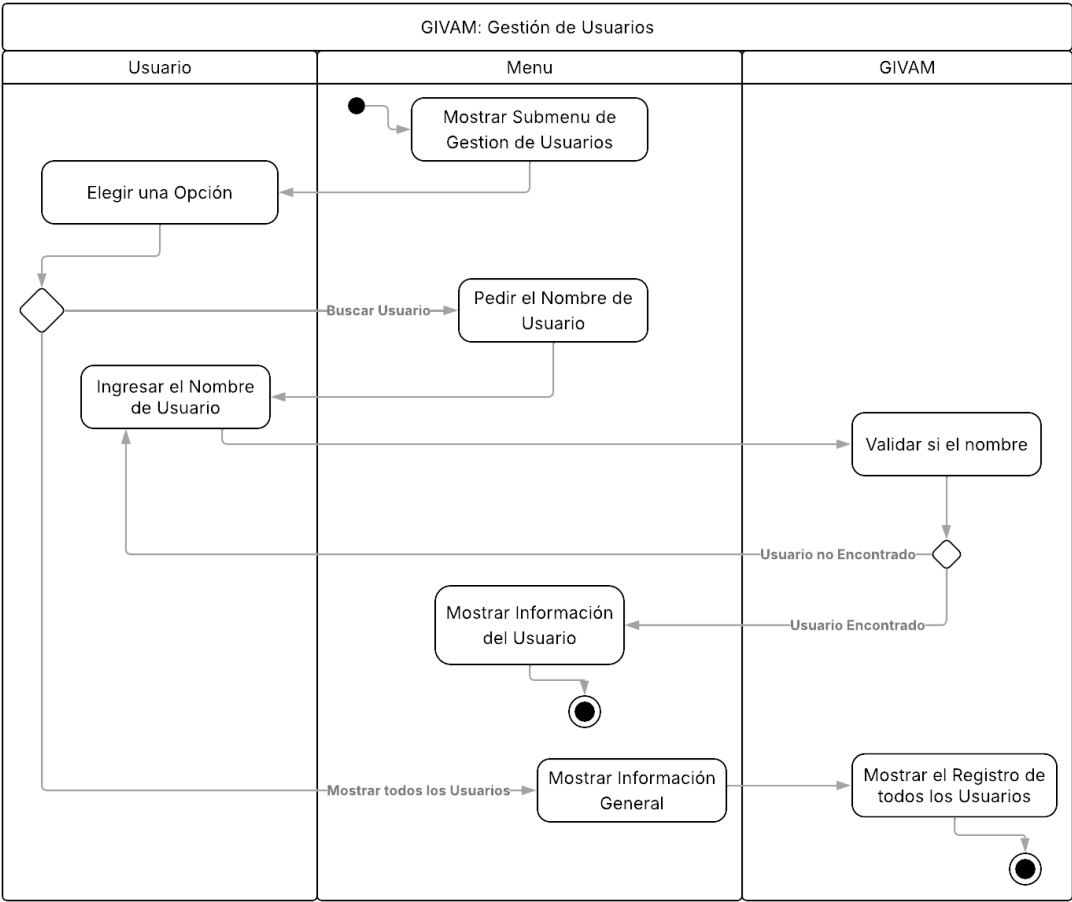
El diagrama de Actividades que a continuación se presenta, representa el flujo sintetizado del funcionamiento de todo el programa de GIVAM, por la extensión del mismo, el diagrama de actividades se ha segmentado en el diagrama principal que abarca desde la validación de credenciales, la elección del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



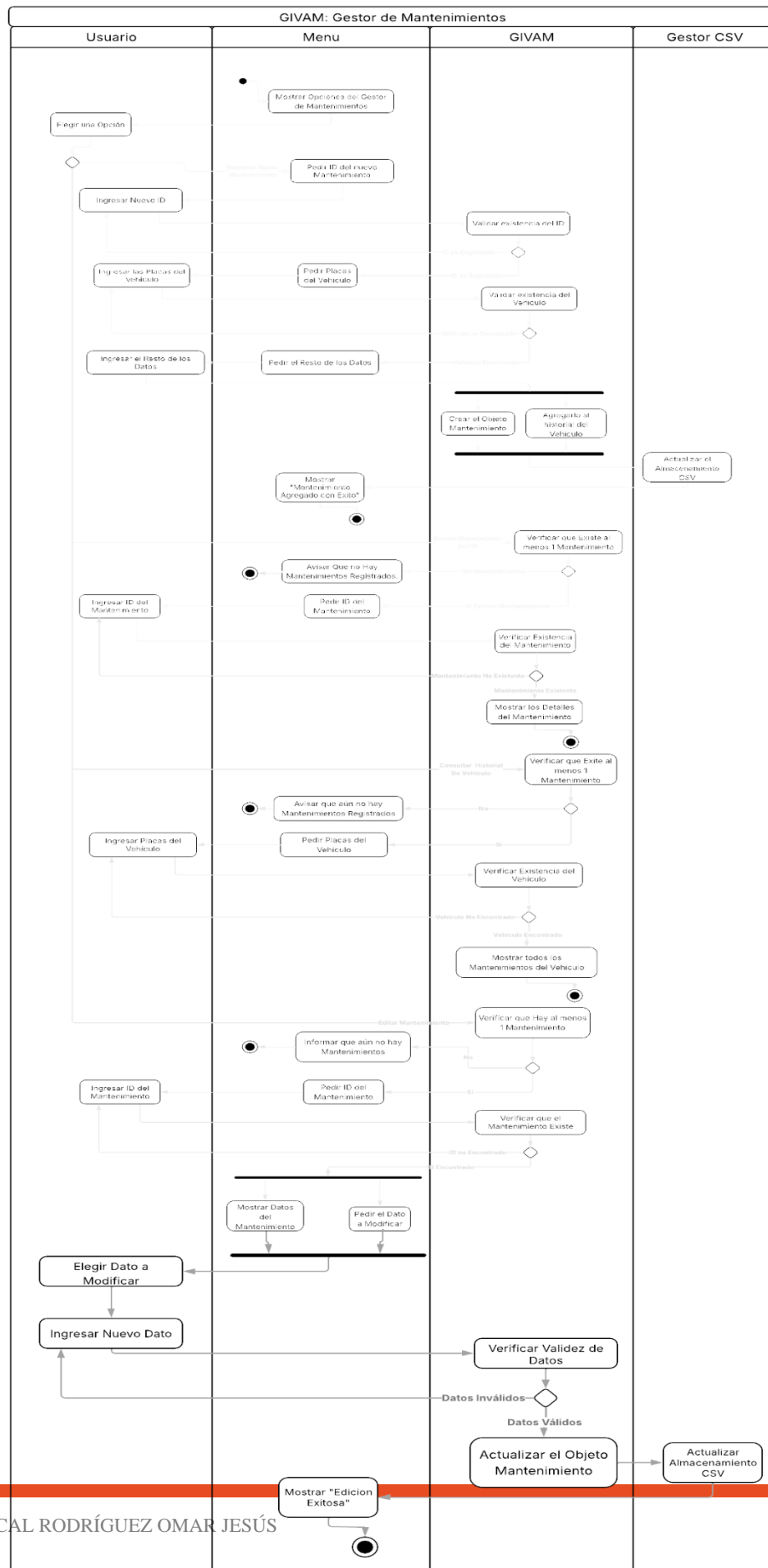
## Anexo Actividades 1: “Gestor de Vehículos”



Anexo Actividades 2: “Gestor de Usuarios”

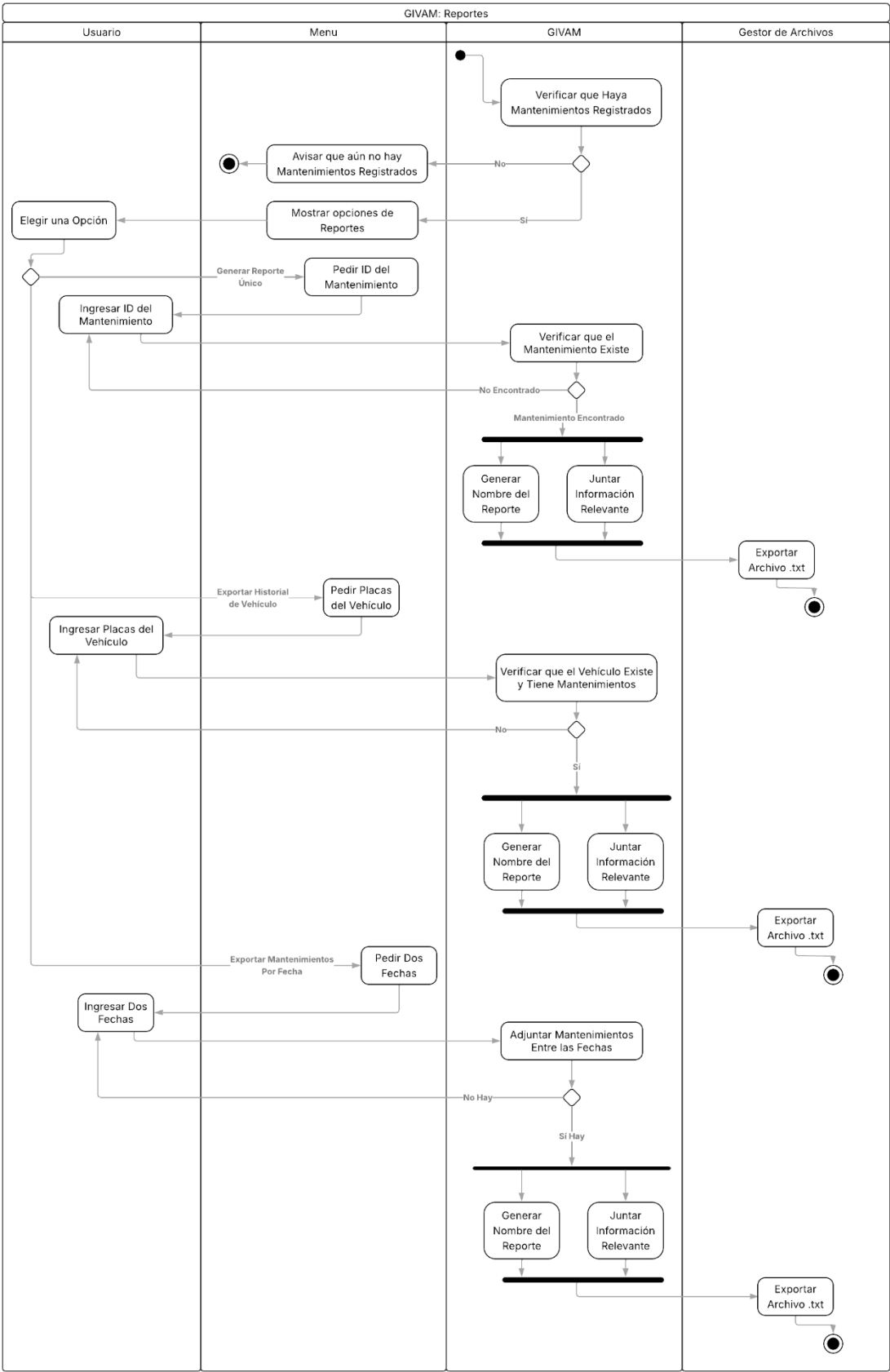


### **Anexo Actividades 3: “Gestor de Mantenimientos”**

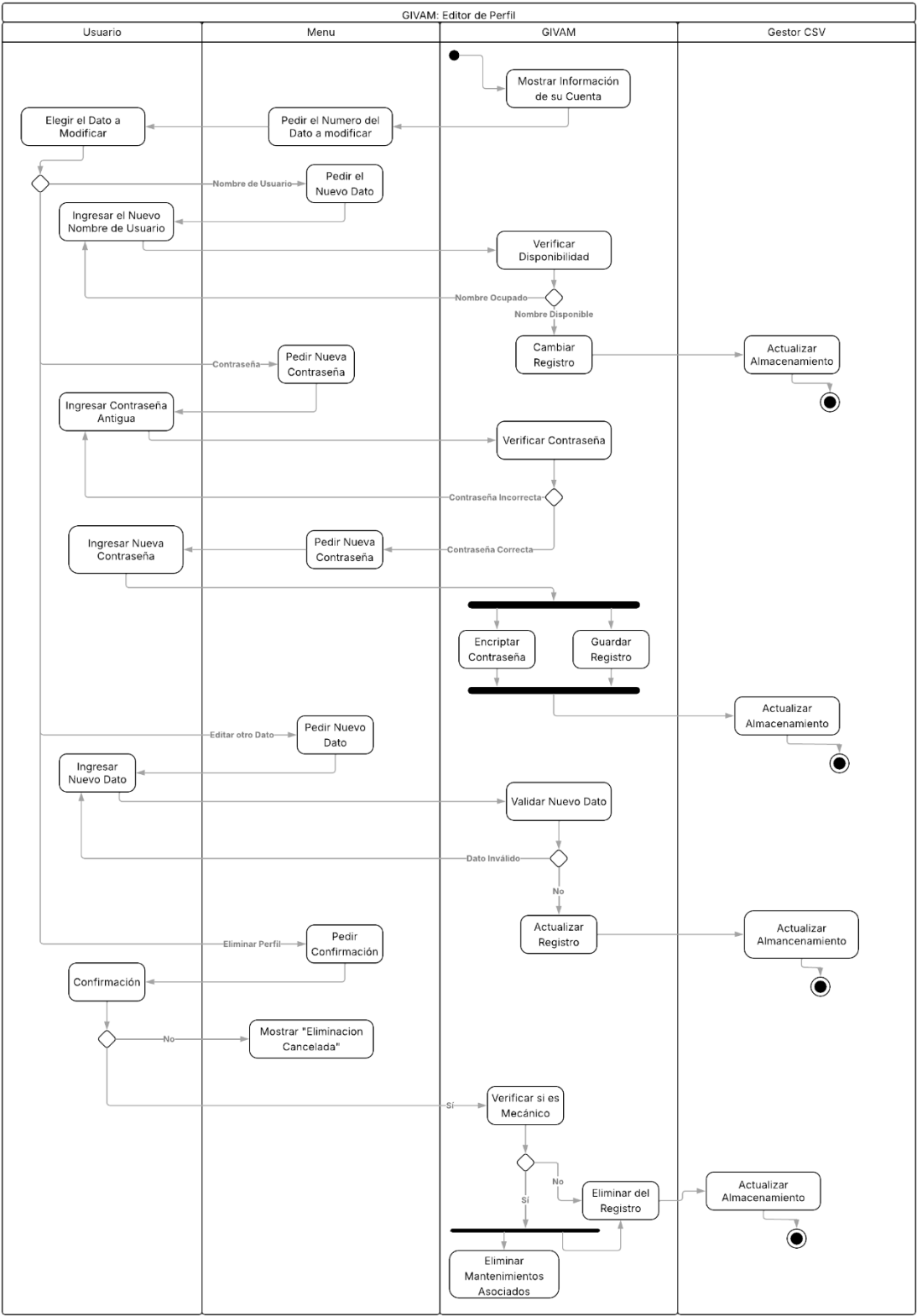




Anexo Actividades 4: “Actividades de Reportes”



Anexo Actividades 5: “Configuración de la Cuenta”



# Implementación de la Solución

Para este programa y aplicar los principios y conocimientos recabados de POO para GIVAM se ha utilizado el lenguaje C++, y a continuación se muestran capturas del programa corriendo:

## Login y Registro de GIVAM

Iniciando el programa, GIVAM pone esta como primer menú:

```
Bienvenido/a al GIVAM
Gestor Inteligente de Vehiculos y Administracion de Mantenimiento
Desea Iniciar Sesion o Registrarse en GIVAM?
1. Iniciar Sesion.
2. Registrarse
3. Cerrar GIVAM
```

Dándote la bienvenida y presentándose distintas opciones ante usted; cabe destacar que el programa viene preparado para si en cualquier menú pone una opción inválida (alguna que no se presente como por ejemplo podría ser 4 en este caso), se muestra el siguiente mensaje de error:

```
-----
La cantidad ingresada esta fuera de rango.
Por favor, vuelva a intentarlo.
-----

Presione una tecla para continuar . . .
```

Y se regresa al menú anterior, así, evitando errores o casos inesperados de números fuera de rango.

Al ser el primer puesta en marcha del programa y no tener registros aún, si intentamos iniciar sesión aparecerá lo siguiente:

```
No hay Usuarios, Registrados.
Accediendo al Registro...
Presione una tecla para continuar . . . |
```

Como no hay usuarios para acceder, en automático nos pasa al registro de perfil, que es el mismo menú si hubiéramos presionado la segunda opción para registrarse.

```
GIVAM: Registro de Nuevo Usuario-----
Ingrese el tipo de Usuario que Desea Registrar
Tenga en Cuenta que Cuentan con Distintos Privilegios:
1. Administrador.
2. Mecanico.
3. Salir.
```

Aquí podemos seleccionar el tipo de usuario con el que nos registraremos, tenga en cuenta que los administradores son los que tienen un control total del programa como para editar registros por ejemplo, mientras que un mecánico no tiene tales privilegios; sin embargo y bien como se muestra en el diagrama de clases, GIVAM se compone de Usuarios, así que para que el programa funcione bien, además de administradores necesitamos mecánicos, ya que a ellos se les hará responsables de los mantenimientos registrados, registremos primero un administrador con la opción 1:

```
Ingrese su Nombre de Usuario: JMV
Ingrese su contrasena: NA17
Ingrese su Nombre: Jose Madero
Ingrese su edad: 44
```

```
Usuario Agregado con Exito.
Presione una tecla para continuar . . .
```

Al ingresar los datos que piden (con restricciones en la edad para un mínimo, por ejemplo), nos avisa que el usuario ha sido agregado con éxito y con ello, nos mandará al apartado de Login donde podremos acceder al sistema:

```
Para Regresar al Menu Anterior Ingrese -1
Ingrese su nombre de Usuario:
```

Sin embargo, antes de entrar a GIVAM, registraré un par de usuarios más para demostrar bien su funcionamiento:

Primero a la Administradora Laura:

```
Ingrese su Nombre de Usuario: LP78
Ingrese su contrasena: QueS10
Ingrese su Nombre: Laura Pergolizzi
Ingrese su edad: 44
```

```
Usuario Agregado con Exito.
Presione una tecla para continuar . . . |
```

Y a los mecánicos Alan Fernandez y Pedri Gonzalez:

```
Ingrese su Nombre de Usuario: Yas25
Ingrese su contrasena: GeRY
Ingrese su Nombre: Alan Fernandez
Ingrese su edad: 25
```

```
Usuario Agregado con Exito.
Presione una tecla para continuar . . .
```

```
Ingrese su Nombre de Usuario: PolRy
Ingrese su contrasena: LabRot
Ingrese su Nombre: Pedri Gonzalez
Ingrese su edad: 22
```

```
Usuario Agregado con Exito.
Presione una tecla para continuar . . .
```

Y ahora sí, paremos al apartado de Login, se nos preguntarán las credenciales, que son el nombre de usuario y la contraseña, y en caso de ser ingresadas erróneamente se mostrará el siguiente mensaje:

```
Para Regresar al Menu Anterior Ingrese -1
Ingrese su nombre de Usuario: JMV
Ingrese su contraseña: Contraseña
```

```
-----
Usuario y/o contraseña incorrecta.
Por favor, intente nuevamente
-----
```

```
Presione una tecla para continuar . . .
```

Pidiéndonos nuevamente las credenciales, si las ingresamos correctamente, se nos da un mensaje de bienvenida con el nombre registrado.

```
Para Regresar al Menu Anterior Ingrese -1
Ingrese su nombre de Usuario: JMV
Ingrese su contraseña: NA17
```

```
Sesion Iniciada Correctamente!
Bienvenido Administrador Jose Madero
Ingresando al Sistema
Presione una tecla para continuar . . .
```

Entrando así a GIVAM y mostrándonos el Menú Principal:

```
GIVAM
Gestor Inteligente de Vehiculos y Administracion de Mantenimiento.
-----
Menu Principal
Seleccione una de las Siguietes Opciones:
1. Gestion de Vehiculos
2. Gestion de Usuarios
3. Gestion de Mantenimientos.
4. Reportes.
5. Configuracion de su Cuenta.
6. Informacion de GIVAM.
7. Cerrar Sesion
```

Desde aquí tenemos el control al sistema, dentro de algunos submenús, habrá opciones de las cuales, solo los usuarios del tipo administrados pueden disponer; veremos aquello cuando toque alguna de esas opciones.

## Gestión de Vehículos

Seleccionando el primer menú tenemos el despliegue del submenú: Gestión de Vehículos:

```
GIVAM: Gestion de Vehiculo
-----
1. Registrar un Nuevo Vehiculo.
2. Editar un Vehiculo Existente.
3. Buscar un Vehiculo.
4. Eliminar un Vehiculo.
5. Mostrar todo el Registro de Vehiculos.
6. Salir al Menu Principal.
```

### Registrar un Vehículo

Intentado acceder a la primera opción, para registrar un nuevo vehículo, es una funcionalidad reservada para los administradores, si intentamos acceder teniendo sesión iniciada con un mecánico aparecerá el siguiente mensaje:

```
-----
El tipo de usuario Mecanico no tiene privilegios para esta funcion
Por favor, seleccione otra funcionalidad o inicie sesion con otros privilegios.
-----
Presione una tecla para continuar . . . |
```

Y nos regresa al menú principal; ahora, accediendo con un administrador se nos despliega el siguiente menú:

```
GIVAM: Registrar un Nuevo Vehiculo: .....
Indique que tipo de Vehiculo desea registrar.
1. Carro.
2. Camioneta
3. Moto
4. Regresar
```

donde podemos elegir el tipo de vehículo a registrar, según el que seleccionemos, se nos pedirán ciertos datos, por ejemplo, elijamos un Carro:

```
Registrando un Vehiculo
Ingrese las placas del Nuevo Vehiculo: JSD5717
Ingrese la Marca del Vehiculo: KIA
Ingrese el modelo el Vehiculo: K3
Ingrese el Anio del Vehiculo: 2025
Ingrese el Kilomtraje del Vehiculo: 12500
Ingrese la Cantidad de Puertas del Carro: 4
La Traccion es Automatica (1.Si/2.No): 1

Vehiculo Agregado Correctamente.
Presione una tecla para continuar . . .
```

Una vez validados los datos, se nos avisa que el vehículo ha sido agregado correctamente, por lo que la siguiente ventana que nos sale es la repetición:

```
Desea Agregar un Nuevo Vehiculo?
1. Si
2. No
```

Aquí podemos elegir si regresar al menú anterior o seguir registrando vehículos, intentemos ingresar ahora, una camioneta, pero, le pondremos las mismas placas que el vehículo que acabamos de registrar, como las placas es el identificar de cada vehículo, es decir, lo aquello con lo que GIVAM los diferencia, no pueden existir dos placas iguales para el correcto funcionamiento, así que si intentamos ingresar placas ya registradas, GIVAM nos avisa de aquello de la siguiente manera:

```
El Vehiculo ya esta registrado.
Se ve de la siguiente manera:
Registro del Carro:
Tipo De Vehiculo: Carro
Placas: JSD5717
Marca: KIA
Modelo: K3
Anio: 2025
Kilometraje: 12500
No se tiene registrado el ultimo mantenimiento.
cantidad de Puertas: 4
Tipo de Manejo: Automatico

Por favor, ingrese placas distintas:

Presione una tecla para continuar . . .
```

Nos muestra la información del vehículo que posee dichas placas y nos pide ingresar unas distintas (Nótese que como todavía, dicho vehículo no le hemos registrado mantenimientos, expresa esto mismo en la información), así que ingresando datos válidos, ingresaremos una camioneta y una moto:

```
Registrando un Vehiculo
Ingrese las placas del Nuevo Vehiculo: JSR2717
Ingrese la Marca del Vehiculo: GMC
Ingrese el modelo el Vehiculo: Sierra
Ingrese el Anio del Vehiculo: 2020
Ingrese el Kilometraje del Vehiculo: 10000
Ingrese el Tipo de Traccion de la Camioneta: 4x4
Ingrese si la Camioneta tiene Caja (1.Si/2.No): 1
```

```
Vehiculo Agregado Correctamente.
Presione una tecla para continuar . . .
```

```
Registrando un Vehiculo
Ingrese las placas del Nuevo Vehiculo: JKY2513
Ingrese la Marca del Vehiculo: BMW
Ingrese el modelo el Vehiculo: S 1000 RR
Ingrese el Anio del Vehiculo: 2023
Ingrese el Kilometraje del Vehiculo: 5200
Ingrese el Tipo de Moto: Deportiva
```

```
Vehiculo Agregado Correctamente.
Presione una tecla para continuar . . . |
```

Así podemos registrar cuantos vehículos sean necesarios.

## Editar un Vehículo

Ahora, siguiendo con la exploración del gestor de Vehículos, la segunda opción es la de editar un vehículo ya existente

```
GIVAM: Editar Informacion de Vehiculos-----  
Ingrese las Placas del Vehiculo: |
```

En este menú podemos editar cualquiera de las especificaciones con la que lo hemos registrado, accedemos a ellos mediante las placas (Existe un mensaje de error y retorno si aún no hay vehículos registrados), placas no existentes se avisa de ello:

```
No existe un Vehiculo en el Sistema con las placas JSR5717  
Por favor, vuelva a intentarlo
```

Ingresando las placas de la camioneta (un vehículo existente) se muestra lo siguiente:

```
Ingrese las Placas del Vehiculo: JSR2717  
A continuacion se muestra la informacion del Vehiculo Seleccionado  
Ingrese el numero de lo que desea modificar.  
1. Placas: JSR2717  
2. Marca: GMC  
3. Modelo: Sierra  
4. Anio: Sierra  
5. Kilometraje: 10000.000000  
6. Tipo de Traccion: 4x4  
7. Tiene Caja
```

Podemos ingresar el número de la característica a modificar (Cabe destacar que las placas deben ser distintas, sino pasará lo mismo que si intentamos registrar un vehículo con placas registradas: nos mostrará la información del vehículo registrado y regresa), modifiquemos el Kilometraje:

```
Ingrese el Kilometraje Actualizado del Vehiculo  
Nota: Solo se permite un Kilometraje Superior  
200000  
Cambio Realizado con Exito.Presione una tecla para continuar . . .
```

Nos avisa que el cambio se realizó exitosamente y regresamos al menú, luego nos pregunta si queremos realizar otra edición:

```
Desea Realizar Otra Edicion? (1.Si/2.No):
```

Y así cuanto necesite el administrados.



## Buscar un Vehículo

En esta opción simplemente nos pide las placas de un vehículo registrado y muestra su información (Una opción disponible para todo tipo de usuarios)

```
Ingresa las Placas del Vehiculo:
JSR2717
Registro de la Camioneta:
Tipo De Vehiculo: Camioneta
Placas: JSR2717
Marca: GMC
Modelo: Sierra
Año: 2020
Kilometraje: 200000
No se tiene registrado el ultimo mantenimiento.
Tipo de Traccion: 4x4
Con Cajon

Presione una tecla para continuar . . . |
```

Nótese que el kilometraje se ha actualizado como lo hemos hecho en con anterioridad.

## Eliminar un Vehículo

Se nos piden las placas de un Vehículo, nos muestra su información y nos pide una confirmación, después de ello, se nos avisa que la eliminación fue exitosa.

```
GIVAM: Eliminar Vehiculos del Registro-----
Ingresa las placas del Vehiculo que Desea Eliminar:
Si Desea Regresar, Ingresa -1
JSR2717
A continuacion, se muestra la informacion del Vehiculo seleccionado:
Registro de la Camioneta:
Tipo De Vehiculo: Camioneta
Placas: JSR2717
Marca: GMC
Modelo: Sierra
Año: 2020
Kilometraje: 200000
No se tiene registrado el ultimo mantenimiento.
Tipo de Traccion: 4x4
Con Cajon

Esta Seguro/a que Desea Eliminar Este Vehiculo? (1.Si/2.No)
Nota: Tambien seran eliminados los mantenimientos Registrados a este Vehiculo
1
Eliminacion Exitosa.
Desea Eliminar otro Vehiculo del Registro? (1.Si/2.No): |
```

Ahora ya no aparecerá más en los registros, si intentamos buscar sus placas ya no aparecerán registradas:

```
GIVAM: Busqueda de Vehiculo
Ingresa las Placas del Vehiculo: JSR2717

No existe un Vehiculo en el Sistema con las placas JSR2717
Por favor, vuelva a intentarlo
```

## Mostrar el Registro de Todos los Vehículos

La última opción del gestor de vehículos es la de mostrar todo el registro, al seleccionarla, nos imprime la información de todos los vehículos registrados:

```
GIVAM: Registro Completo de Vehiculos.-----
A continuacion, se muestra el total de Vehiculos registrados en GIVAM
Registro del Carro:
Tipo De Vehiculo: Carro
Placas: JSD5717
Marca: KIA
Modelo: K3
Año: 2025
Kilometraje: 12500
No se tiene registrado el ultimo mantenimiento.
cantidad de Puertas: 4
Tipo de Manejo: Automatico
.....

Registro de la Moto:
Tipo De Vehiculo: Moto
Placas: JKY2513
Marca: BMW
Modelo: S 1000 RR
Año: 2023
Kilometraje: 5200
No se tiene registrado el ultimo mantenimiento.
Tipo de Moto: Deportiva
.....

Un total de: 2 vehiculos registrados.
```

y se nos da el total de registros.

## Gestión de Usuarios

La segunda opción del menú principal es el gestor de usuarios, en él, se nos muestran estas dos opciones:

```
GIVAM: Gestion de Usuarios
-----
1. Buscar un Usuario Registrado.
2. Mostrar Todos los Usuarios Registrados.
3. Salir al Menu Principal.
|
```

## Buscar un Usuario Registrado

La primera de estas, nos pide un nombre de usuario registrado y muestra la información de aquel:

```
GIVAM: Buscar Usuario Registrados por Nombre de Usuario-----
-----
Ingrese el Nombre de Usuario del Mecanico que quiera buscar: LP78
Usuario del Tipo Administrador:
Nombre del Administrador:LP78
Nombre: Laura Pergolizzi
Edad: 44

Desea Realizar Una Nueva Busqueda? (1.Si/2.No):
```

## Mostrar Registro de Usuarios Registrados

La segunda opción del gestor de usuarios, al igual que en los vehículos, imprime la información de todos los usuarios registrados y contando el total:

```
GIVAM: Registro Completo de Usuarios.-----
A continuacion, se muestra el total de Usuarios registrados en GIVAM
Usuario del Tipo Administrador:
Nombre del Administrador:JMV
Nombre: Jose Madero
Edad: 44
-----
Usuario del Tipo Administrador:
Nombre del Administrador:LP78
Nombre: Laura Pergolizzi
Edad: 44
-----
Usuario del Tipo Mecanico:
Nombre del Mecanico:Yas25
Nombre: Alan Fernandez
Edad: 25
-----
Usuario del Tipo Mecanico:
Nombre del Mecanico:PolRy
Nombre: Pedri Gonzalez
Edad: 22
-----

Total de Usuarios Registrados: 4
Presione una tecla para continuar . . . |
```

## Gestor de Mantenimientos

El gestor de mantenimientos nos presenta las siguientes opciones:

```
GIVAM: Gestion de Mantenimientos
-----
1. Registrar un Nuevo Mantenimiento.
2. Buscar Mantenimiento por ID.
3. Consultar Historial por Vehiculo.
4. Editar Mantenimiento
5. Salir al Menu Principal.
```

Cada uno es un homónimo a las del vehículo

## Registrar un Nuevo Mantenimiento

```
GIVAM: Registrar Mantenimiento-----
Ingrese el tipo de mantenimiento que desea registrar:
1. Mantenimiento Preventivo.
2. Mantenimiento Correctivo.
3. Salir
```

Primero se nos pide el tipo de mantenimiento, si es uno preventivo (como lo podría ser un cambio de aceite, chequeo rutinario, revisión de presión de llantas, etc) o uno correctivo (si el vehículo ya presenta alguna falla consigo), cada uno recabando información propia, primero, agreguemos un Mantenimiento Preventivo:

```
Ingrese el ID del Nuevo Mantenimiento: 1
Ingrese las Placas del Vehiculo: JDY1020
```

```
No existe un Vehiculo en el Sistema con las placas JDY1020
Por favor, vuelva a intentarlo
```

```
Ingrese las Placas del Vehiculo: |
```

Para ambos mantenimientos, se nos pedirá un ID numérico y las placas del vehículo al cuál se le aplican.

Estas placas deben estar registradas, caso contrario se nos avisa y nos pide unas existentes (como nota, si no hay vehículos registrados, ni siquiera podremos acceder a registrar un mantenimiento, ya que no habrá vehículos a los cuales asignárselas)

```
Ingrese el ID del Nuevo Mantenimiento: 1
Ingrese las Placas del Vehiculo: JDY1020

No existe un Vehiculo en el Sistema con las placas JDY1020
Por favor, vuelva a intentarlo

Ingrese las Placas del Vehiculo: JSD5717
Ingrese el Nombre de Usuario del Mecanico: Yas25

Para Ingresar la Fecha:
1. Ingresar la Fecha de Hoy.
2. Ingresar una Fecha Diferente
2
Ingrese el dia de realizacion del mantenimiento: 10
Ingrese el mes de realizacion del mantenimiento: 01
Ingrese el anio de realizacion del mantenimiento: 2023
Ingrese la descripcion del Mantenimiento: Cambio de Aceite Rutinario
Ingrese el costo total del Mantenimiento: 1200
Ingrese el Kilometraje Recomendado para el Siguiete Mantenimiento: 20000
Ingrese si el Mantenimiento se Realizo en el Tiempo Recomendado (1.Si/2.No): 1
Desea Ingresar Tareas Programadas para Este Mantenimiento (1.Si/2.No): 1
Ingrese el numero de tareas programadas a agregar al Mantenimiento Preventivo: 2
Tarea 1: Revisar las Condiciones del Motor
Tarea 2: Cambiar el Aceite
Se ha registrado correctamente el nuevo Mantenimiento
Presione una tecla para continuar . . .
```

Al igual que los vehículos, los mantenimientos se anexan a un mecánico, por lo que es necesario ingresar el nombre de usuario de uno existente, los demás son datos propios del mantenimiento. Una vez ingresados, se nos avisa que el mantenimiento fue registrado con éxito. Para seguir con el ejemplo, agregaré un Mantenimiento Correctivo.

Si intentamos ingresar un ID ocupado, nos aparece lo siguiente:

```
Ingrese el ID del Nuevo Mantenimiento: 1

El ID 1 ya esta registrado con un mantenimiento.
Dicho mantenimiento es el siguiente:
Detalles del Mantenimineto con id 1
Mantenimiento Preventivo
Informacion del Vehiculo-----
Carro de placas JSD5717
Responsable del Mantenimiento Mecanico: Alan Fernandez
Fecha de Realizacion: 10/1/2023
Descripcion del Proceso:
Cambio de Aceite Rutinario
Costo Total: 1200
Tareas Programadas para el Mantenimiento Preventivo:
1. Revisar las Condiciones del Motor
2. Cambiar el Aceite
El proximo mantenimiento preventivo se recomienda al proximo kilometraje: 20000
El mantenimiento fue realizado a tiempo recomendado.

Por favor, seleccione otro ID para su Nuevo Mantenimiento.
Presione una tecla para continuar . . . |
```

Así, que registrando un Mantenimiento Correctivo con un ID disponible:

```
Ingrese el ID del Nuevo Mantenimiento: 2
Ingrese las Placas del Vehiculo: JKY2513
Ingrese el Nombre de Usuario del Mecanico: PolRy

Para Ingresar la Fecha:
1. Ingresar la Fecha de Hoy.
2. Ingresar una Fecha Diferente
2
Ingrese el dia de realizacion del mantenimiento: 25
Ingrese el mes de realizacion del mantenimiento: 5
Ingrese el año de realizacion del mantenimiento: 2024
Ingrese la descripción del Mantenimiento: Reparación del Parachoques Delantero
Ingrese el costo total del Mantenimiento: 12500
Ingrese el Componente Averiado que se trato en el Mantenimiento Correctivo: Cubierta Exterior
Ingrese la Causa Detectada de la Falla: Choque
Ingrese si se Requirio Reemplazo de la Pieza Afectada (1.Si/2.No): 1
Se ha registrado correctamente el nuevo Mantenimiento
Presione una tecla para continuar . . .
```

## Buscar Mantenimiento por ID

Se pide un ID de mantenimiento registrado y se muestra la información si es que este existe:

```
GIVAM: Consulta de Mantenimiento.
Ingrese el ID del Mantenimiento Registrado: 2
Detalles del Mantenimineto con id 2
Mantenimiento Correctivo
Informacion del Vehiculo-----
Moto de placas JKY2513
Responsable del Mantenimiento Mecanico: Pedri Gonzalez
Fecha de Realizacion: 25/5/2024
Descripcion del Proceso:
Reparación del Parachoques Delantero
Costo Total: 12500
Componente Averiado: Cubierta Exterior
Causa Detectada de la Falla: Choque
El mantenimiento requirio de un reemplazo de pieza.

Presione una tecla para continuar . . . |
```

## Consultar Historial por Vehículo

Aquí podemos, mediante las placas de un vehículo, imprimir todos los mantenimientos que se le hayan realizado:

```
GIVAM: Consulta del Historial del Mantenimiento de un Vehiculo-----
Ingrese las Placas del Vehiculo: JSD5717
Registro de Mantenimientos a Carro de placas JSD5717:
Detalles del Mantenimineto con id 1
Mantenimiento Preventivo
Informacion del Vehiculo-----
Carro de placas JSD5717
Responsable del Mantenimiento Mecanico: Alan Fernandez
Fecha de Realizacion: 10/1/2023
Descripcion del Proceso:
Cambio de Aceite Rutinario
Costo Total: 1200
Tareas Programadas para el Mantenimiento Preventivo:
1. Revisar las Condiciones del Motor
2. Cambiar el Aceite
El proximo mantenimiento preventivo se recomienda al proximo kilometraje: 20000
El mantenimiento fue realizado a tiempo recomendado.

Desea realizar una nueva busqueda? (1.Si/2.No): |
```

En caso de tener múltiples, se van mostrando todos consecutivamente

## Editar un Mantenimiento

Al igual que en los vehículos, podemos editar la información ya registrada de un mantenimiento siendo administradores del sistema, ingresando un ID existente:

```
GIVAM: Editar Informacion de un Mantenimiento-----
Ingrese el ID del Mantenimiento Registrado: 2
A continuacion, se muestra la informacion del mantenimiento Seleccionado
Ingrese el Numero de lo que Desea modificar.
1. ID: 2
2. Responsable: Pedri Gonzalez
3. Fecha de Realizacion: 25/5/2024
4. Descripcion: Reparación del Parachoques Delantero
5. Costo: 12500.000000
6. Componente Averiado: Cubierta Exterior
7. Causa Falla: Choque
8. La Pieza Requirio Reemplazo: Si
```

y posteriormente, eligiendo qué cosa cambiaremos:

```
Ingrese la Nueva Causa de la Falla: Desgaste por Uso Cotidiano
Cambio realizado con Exito
Presione una tecla para continuar . . .
```

Los cambios se ven registrados si volvemos a buscar el mismo mantenimiento por ID:

```
GIVAM: Consulta de Mantenimiento.
Ingrese el ID del Mantenimiento Registrado: 2
Detalles del Mantenimineto con id 2
Mantenimiento Correctivo
Informacion del Vehiculo-----
Moto de placas JKY2513
Responsable del Mantenimiento Mecanico: Pedri Gonzalez
Fecha de Realizacion: 25/5/2024
Descripcion del Proceso:
Reparación del Parachoques Delantero
Costo Total: 12500
Componente Averiado: Cubierta Exterior
Causa Detectada de la Falla: Desgaste por Uso Cotidiano
El mantenimiento requirio de un reemplazo de pieza.

Presione una tecla para continuar . . . |
```

## Reportes

La Cuarta opción del menú principal está dedicada a la creación y exportación de reportes de mantenimientos, al seleccionarla, aparece el siguiente submenú:

```
GIVAM: Gestion de Reportes
-----
1. Exportar un Reporte de un Mantenimiento Determinado.
2. Exportar el Historial de Mantenimiento de un Vehículo. .
3. Exportar el Reporte de Todos los Mantenimientos Entre Dos Fechas
4. Salir al Menu Principal.
```

Exploremos la primera opción:

### Exportar un Reporte Determinado

```
GIVAM: Exportar Reporte de Mantenimiento-----
Ingrese el ID del Mantenimiento Registrado: 1
Reporte Generado con Exito
Desea Exportar otro Mantenimiento Registrado? (1.Si/2.No):
```

Se nos pedirá un ID de reporte registrado, y al seleccionarlo, se nos dirá que el reporte fue generado con éxito, si nos fijamos en la carpeta donde nuestro programa se ejecuta tendremos un archivo como este:

```
📄 JSD5717-Carro-ID1-Mantenimiento Preventivo-10_1_2023
```

Un archivo .txt con un título característico, y esto contiene en su interior:

```
Reportes de Mantenimiento.
Fecha de Consulta: 9/5/2025

Reporte de Mantenimiento de Id 1 -----
Carro de placas: JSD5717
Tipo de Mantenimiento: Mantenimiento Preventivo
Mantenimiento Realizado por el Mecanico: Alan Fernandez
Fecha de Realizacion: 10/1/2023
Descripcion del Mantenimiento:
Cambio de Aceite Rutinario
Costo Total: 1200
Tareas Programadas:
Tarea 1: Revisar las Condiciones del Motor
Tarea 2: Cambiar el Aceite
kilometraje Objetivo: 20000
El Mantenimiento Preventivo se Realizo a Tiempo Recomendado

Total de Reportes: 1
```

Escrito en texto plano, están las características de nuestro reporte con la información relevante de aquello.

### Exportar Reportes de un Vehículo

Esta función es similar a la anterior, pero en lugar de exportar un mantenimiento único por ID, exporta todo el historial de mantenimientos de un vehículo ingresando sus placas:

```
GIVAM: Exportar Reportes de los Mantenimientos de un Vehiculo: -----
Ingrese las Placas del Vehiculo: JKY2513
Reporte Exportado con Exito!
```

Generando un archivo .txt como el siguiente:

#### JKY2513-Moto-ReportesMantenimiento

```
Reportes de Mantenimiento.
Fecha de Consulta: 9/5/2025

Reporte de Mantenimiento de Id 1 -----
Carro de placas: JSD5717
Tipo de Mantenimiento: Mantenimiento Preventivo
Mantenimiento Realizado por el Mecanico: Alan Fernandez
Fecha de Realizacion: 10/1/2023
Descripcion del Mantenimiento:
Cambio de Aceite Rutinario
Costo Total: 1200
Tareas Programadas:
Tarea 1: Revisar las Condiciones del Motor
Tarea 2: Cambiar el Aceite
kilometraje Objetivo: 20000
El Mantenimiento Preventivo se Realizo a Tiempo Recomendado

Total de Reportes: 1
```

## Exportar Mantenimientos Entre Fechas

Aquí, el filtro para seleccionar qué mantenimientos registrará serán dos fechas dadas, ahí se incluirán todos los mantenimientos entre dichas Fechas:

```
GIVAM: Exportar Mantenimientos Entre Fechas: -----
Ingrese la Primera Fecha de la Cual Quiera Exportar los Mantenimientos:
Dia: 12
Mes: 12
Anio: 2023

Ingrese la Segunda Fecha de la Cual Quiera Exportar los Mantenimientos:
Dia: 07
Mes: 04
Anio: 2025
Reporte Exportado con Exito!
```

Generando un archivo .txt como el siguiente:

#### ReporteEntreFechas\_12-12-2023\_7-4-2025

```
Reportes de Mantenimiento.
Fecha de Consulta: 9/5/2025

Reporte de Mantenimiento de Id 2 -----
Moto de placas: JKY2513
Tipo de Mantenimiento: Mantenimiento Correctivo
Mantenimiento Realizado por el Mecanico: Pedri Gonzalez
Fecha de Realizacion: 25/5/2024
Descripcion del Mantenimiento:
Reparación del Parachoques Delantero
Costo Total: 12500
Componente Averiado: Cubierta Exterior
Causa Detectada de la Falla: Desgaste por Uso Cotidiano
La pieza requirio de ser cambiada

Total de Reportes: 1
```



## Configuración de su Cuenta

Esta opción del menú principal nos dirige a la siguiente ventana:

```
GIVAM: Editor de Perfil-----
Actualmente, su perfil de Administrador tiene la siguiente forma:
Elija el Numero de la opcion que desee modificar:
1. Nombre de Usuario: JMV
2. Nombre: Jose Madero
3. Edad: 44
4. Cambiar Contraseña.
5. Eliminar Perfil
6. Regresar.
```

nos muestra los datos sobre los cuales está iniciada nuestra sesión y aquí podemos editar cualquiera de ellos o eliminar el perfil, para demostrar más funcionalidades y ver qué pasa, seleccionemos la 5ta, eliminar el perfil:

```
Esta Seguro/a que desea Elimiar Este Perfil? (1.Si/2.No):
Nota: Para los Mecanicos, tambien seran eliminados los mantenimientos que ha realizado
```

Se nos pide una confirmación, y al aceptarla nos avisa que la eliminación fue exitosa y nos redirige a la primera pantalla de GIVAM:

```
Bienvenido/a al GIVAM
Gestor Inteligente de Vehiculos y Administracion de Mantenimiento
Desea Iniciar Sesion o Registrarse en GIVAM?
1. Iniciar Sesion.
2. Registrarse
3. Cerrar GIVAM
```

Y si intentamos acceder con las mismas credenciales pasará esto:

```
-----
Usuario y/o contraseña incorrecta.
Por favor, intente nuevamente
-----
Presione una tecla para continuar . . .
```

Por lo que tendremos que, o registrar otro perfil, o bien, acceder con otra sesión.

Así mismo, podemos editar cualquier otro dato que se nos presente.

## Información de GIVAM

Esta penúltima opción del menú principal nos arroja información relevante sobre el funcionamiento de GIVAM y sobre su versión:

```
=====
GIVAM
Gestor Inteligente de Vehiculos y Administracion de Mantenimiento
GIVAM es un gestor para taller mecanico en el cual puedes llevar el control sobre los vehiculos y los mantenimientos que estos reciben a lo largo del tiempo
.
GIVAM guarda la informacion de los Usuarios divididos en Administradores y Mecanicos
.
Los administradores tienen control sobre la eliminacion de vehiculos y mantenimientos.
Los mecanicos tienen controles mas limitados sobre el sistema.

GIVAM guarda diversa informacion en archivos CSV, separados por comas.
Para llevar los registros estan: Mantenimientos.csv, Usuarios.csv, Vehiculos.csv
Los Archivos CSV tienen la ventaja de que pueden ser editados con facilidades en programas como Excel, lo que le da un mayor control sobre los datos
Para el registro de los Usuarios, la contraseña se guarda encriptada, por lo que el que tenga acceso a los archivos no podrá conocer la contraseña
GIVAM carga y maneja automaticamente estos registros, ademas de poder generar en archivos de texto, reportes de los mantenimientos registrados

Espero que GIVAM sea de utilidad para su empresa :)

Producto Integrados de Programacion Orientada a Objetos
Desarrollador: Mariscal Rodriguez Omar Jesus
Version de GIVAM: 1.0
=====
Presione una tecla para continuar . . .
```

Útil para el control de versiones si GIVAM se llega a actualizar más adelante.

## Cerrar Sesión

Nos devuelve a la primera pantalla de GIVAM, cerrando la sesión

```
Bienvenido/a al GIVAM
Gestor Inteligente de Vehiculos y Administracion de Mantenimiento
Desea Iniciar Sesion o Registrarse en GIVAM?
1. Iniciar Sesion.
2. Registrarse
3. Cerrar GIVAM
```

Y si seleccionamos Cerrar GIVAM:

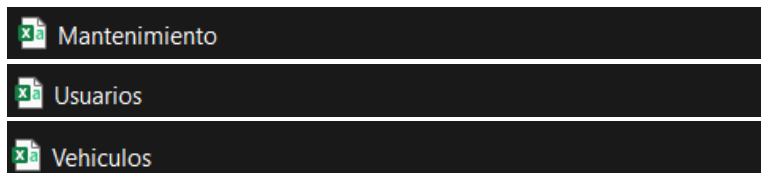
```
Saliendo de GIVAM...
Tenga un Lindo Dia :)

=====

Process exited after 224.3 seconds with return value 4294967295
Presione una tecla para continuar . . .
```

## Control de Almacenamiento

GIVAM está programado para sostener sus datos, aún si el programa cierra, para ello, todos estos registros que hemos estado haciendo se almacenan en archivos .csv dentro de la misma carpeta donde GIVAM se ejecuta



Estos archivos contienen la información que hemos registrado, al ser formato .csv, se hace sumamente fácil manejarla y visualizarla con aplicaciones externas como Excel, por lo que editando esto GIVAM en automático carga los datos aquí registrados, así que sirve como una manera de importación si una empresa adquiere GIVAM y quiere ingresar todo su catálogo, no haciéndolo uno por uno a través de la consola, sino más fácilmente con Excel

El almacenamiento tiene un look como el siguiente:

	A	B	C	D	E
1	TipoDeUsuario	NombreUsuario	ContraseñaEncriptada	Nombre	Edad
2	Administrador	LP78	2.49662E+18	Laura Pergolizzi	44
3	Mecanico	Yas25	1.43755E+19	Alan Fernandez	25
4	Mecanico	PolRy	3.81326E+18	Pedri Gonzalez	22

En este caso particular de los usuarios, la contraseña se almacena encriptada mediante un Hash, por lo que aunque alguien acceda al almacenamiento .csv, la contraseña no válida, GIVAM no desencripta la contraseña, sino lo que hace es encriptar también el intento de contraseña y compararla con este hash, por lo que la contraseña real no se almacena en ningún momento.

Para Vehículos el archivo .csv se ve como:

	A	B	C	D	E	F	G	H	I	J	K
1	TipoVehiculo	Placas	Marca	Modelo	Anio	Kilometraje	DiaUltMan	MesUltMan	AnioUltMan	Extra1	Extra2
2	Carro	JSD5717	KIA	K3	2025	12500	10	1	2023	4	1
3	Moto	JKY2513	BMW	S 1000 RR	2023	5200	25	5	2024	Deportiva	-

Y para los Mantenimientos:

	A	B	C	D	E	F	G	H	I	J	K	L
1	TipoMantenimiento	Id	PlacasVehiculo	UsuarioMecanico	DiaRealizacion	MesRealizacion	AnioRealizacion	Descripcion	Costo	Extra1	Extra2	Extra3
2	Mantenimiento Preventivo	1	JSD5717	Yas25	10	1	2023	Cambio de Aceite Rutinario	1200	Revisar las Condiciones	20000	1
3	Mantenimiento Correctivo	2	JKY2513	PolRy	25	5	2024	Reparación del Parachoques Delantero	12500	Cubierta Exterior	Desgaste por	1

Su interpretación y carga es gestionada por el mismo GIVAM.

De esta manera, hacemos que el sistema tenga su propio método de almacenamiento de datos para que el programa sea consecuente y toda su información no sea eliminada en cuanto cierra.

Completando así, los puntos que GIVAM pretendía solucionar como un sistema integral para la gestión de vehículos, usuarios y mantenimientos de un taller automotriz.

## Conclusiones de la Actividad

---

La realización del sistema GIVAM representó un buen reto para integrar todos los conocimientos y el paradigma recabado en el curso, en la práctica he aprendido muchas cosas, empezando por darle el valor que merecen a los diagramas UML, en un inicio parecen solo esquemas que retrasan el empezar a programar lo que más o menos se tiene en mente, pero con un problema más grande como fue GIVAM, el tener un modelo sólido sobre lo que GIVAM debe ser y qué camino se debe tomar agiliza mucho el proceso de la programación, sabiendo qué queremos hacer es mucho más fácil hacer el código más eficiente en términos de líneas de código y en cuestiones de tiempo; hace que no tengamos que retroceder una y otra vez para añadir más clases, más métodos o ver qué si queremos añadir una función ahora tenemos que modificar muchas más cosas. Además, puedo entender que su valor más allá incluso en el trabajo colaborativo, si cada persona debe trabajar en un módulo, una clase o ciertas funcionalidades, no es la mejor idea esperar que un programador acabe su módulo para empezar a trabajar con el mío porque requiero de un resultado de su función, sino que el plantear de antemano qué debe retornar cada función permite tener valores de prueba y trabajar a la vez en funciones que, dentro del código dependen unas de otras; una herramienta de importancia fundamental que al inicio del curso no podía valorar.

Otro gran aprendizaje que tuve en la realización de esta actividad integradora es procurar, dentro de la fase de desarrollo tener funciones específicas que cumplan tareas concretas, esto es bastante útil para mantener el código legible, si ocurre un error identificarlo más fácilmente y hacer el código más expandible; por ejemplo, dentro de mi código GIVAM se encarga de coordinar las acciones de cada una de las clases que la componen, pero GIVAM recibe datos que ya deben estar filtrados, y el incluir dentro de él también los menús o el filtro de datos rompe con el principio de responsabilidad única y es una mala praxis de programación, por lo que creé otra clase llamada Menú encargada de presentar menús y submenús y de filtrar los datos que GIVAM recibe, así identificando más fácil los errores que surgieron y haciendo más mantenible, legible y con mejor presentación el código final.

Cuestión de polimorfismo, es increíble lo sumamente útil que es, también en conjunto con las clases abstractas, no veía la necesidad de incluir métodos abstractos que no definieran su código, pero el tenerlos ahí y poder aplicar polimorfismo dinámico a sus clases padres fue algo de gran provecho en todo el código; valoré también más el tema del encapsulamiento, creo que fue el pilar de POO que más cuestioné en un inicio, pensaba por qué habría que limitar el acceso a los datos ¿no era más cómodo tenerlos a la mano y dejar de utilizar getter's y setter's que son funciones simples que podrían desaparecer sin el encapsulamiento? Con GIVAM aprendí que existen funciones o datos que como programador, sólo quiero que se ejecuten desde un punto específico del código, acceder a ellos desde otras partes puede tener consecuencias fatales.

Un refuerzo de aprendizaje que tuve también fue el manejo de memoria dinámica, si bien, el curso no va centrado únicamente en C++ sino que en el paradigma de la Programación Orientada a Objetos, trabajar con C++ y su libertad para el manejo de la memoria es una responsabilidad destacada y crucial, y estar utilizando apuntadores, vectores de apuntadores y gestionar este apartado sin tener fugas fue enriquecedor.

En conclusión, el desarrollo de GIVAM fortaleció en gran manera desde mi parte práctica hasta mi parte conceptual de los pilares de este paradigma y me hizo valorar mucho más cada uno de los por qué's de cada concepto y la importancia de los modelos en esta profesión.