


- **Nombre:** Salvador Moreno Sánchez
- **DNI:** 
- **Centro asociado:** Motril (Granada)

Índice

- [Codificación: Gestión de Alquiler de Maquinaria Agrícola \(GesAMA\)](#)
 - [Observaciones previas sobre el diseño del que se parte](#)
 - [Inconsistencias en el diseño dado y soluciones](#)
 - [Instalación de herramientas necesarias](#)
 - [Capturas de la ejecución de la aplicación](#)

Codificación: Gestión de Alquiler de Maquinaria Agrícola (GesAMA)

Observaciones previas sobre el diseño del que se parte

Se ha decidido iniciar el desarrollo de la aplicación basándose únicamente en los diagramas UML proporcionados, ya que en el diseño que se me ha proporcionado no se han tenido en cuenta los documentos de Especificación de Requisitos de Software (SRD) y Diseño Arquitectónico (ADD) que tradicionalmente se desarrollan en fases anteriores del ciclo de vida del desarrollo de software y cuyo formato se establece en el libro de texto de la asignatura.

Los diagramas UML proporcionan una representación visual de los diferentes aspectos de la aplicación, incluyendo casos de uso, clases, secuencias, colaboración, estado y actividad. Estos diagramas son valiosos para comprender la estructura y el comportamiento del sistema, pero por sí solos no proporcionan la profundidad y claridad necesarias para guiar todo el proceso de desarrollo de software de manera efectiva.

Partiendo únicamente de los diagramas UML, nos enfrentamos a diversos desafíos y riesgos al comenzar la codificación de la aplicación:

- **Interpretación ambigua de requisitos:** los diagramas UML pueden no capturar todos los detalles y matices de los requisitos del cliente. Sin un SRD detallado, se podría interpretar de manera incorrecta o incompleta las necesidades del usuario, lo que lleva a la implementación de funcionalidades erróneas o incompletas.
- **Desajuste entre diseño y requisitos:** sin un ADD que guíe la arquitectura del software, existe el riesgo de que la implementación no esté alineada con los requisitos del sistema. Esto puede resultar en una arquitectura inadecuada que dificulte la expansión, mantenimiento y escalabilidad del software.
- **Cambio constante de requisitos:** la falta de un SRD sólido puede llevar a cambios frecuentes en los requisitos a medida que avanza el desarrollo, lo que resulta en retrasos, aumento de costos y falta de coherencia en el producto final.
- **Falta de modularidad y escalabilidad:** sin un ADD que defina una arquitectura modular y escalable, la aplicación puede volverse difícil de mantener y actualizar a medida que crece en tamaño y complejidad.

En conclusión, iniciar el desarrollo de la aplicación sin los documentos SRD y ADD desarrollados puede llevar a una serie de dificultades, incluida una comprensión ambigua de los requisitos, una arquitectura inadecuada y cambios frecuentes en los requisitos. En un entorno profesional real, sería esencial completar estos documentos antes de proceder con la codificación para garantizar un proceso de desarrollo más eficiente y satisfacer las necesidades del cliente de manera efectiva.

Inconsistencias en el diseño dado y soluciones

A pesar de no poseer los documentos de Especificación de Requisitos de Software (SRD) y Diseño Arquitectónico (ADD), el diseño recibido posee unos diagramas UML suficientemente detallados y completos para realizar la codificación de manera óptima. Aún así, a continuación, dejo pequeños ajustes que he realizado durante la elaboración del código y que cabría señalar:

- En la clase `GesAMA`, se ha incluido una variable llamada `meses_del_ano`, que alberga un array con los nombres de los meses del año. Esta variable se utilizó en el método `planMensualMaquina` para mostrar los nombres de los meses en el plan mensual de la máquina.
- Las función `editar()`, la cual aparece tanto en la clase `Maquina` como en la clase `Finca`, es omitible ya que se puede hacer una edición de las máquinas y fincas directamente desde la clase `GesAMA`, tal y como se plantea en el código elaborado.
- Se han incluido importaciones de librerías propias de Python para gestionar las fechas (`import datetime`) y la generación de rangos aleatorios (`import random`) para simular las distancias geográficas de las fincas.

Instalación de herramientas necesarias

Para ejecutar la aplicación Python que hemos desarrollado, necesitarás tener instalado Python en tu sistema. Además, como estamos utilizando las bibliotecas estándar de Python `datetime` y `random`, no necesitas instalar ninguna biblioteca adicional. Aquí tienes los pasos para instalar Python y ejecutar la aplicación:

Instalación de Python

Windows:

1. Ve al sitio web oficial de Python: python.org
2. Descarga el instalador de Python para Windows.
3. Ejecuta el instalador y sigue las instrucciones en pantalla. Asegúrate de marcar la casilla "Agregar Python X.X a la variable de entorno PATH" durante la instalación.
4. Una vez instalado, abre CMD o PowerShell y prueba ejecutando `python --version` para verificar que Python se haya instalado correctamente.

macOS:

1. macOS generalmente ya viene con Python preinstalado. Sin embargo, puedes instalar una versión más reciente utilizando Homebrew u otros administradores de paquetes si lo prefieres.
2. Abre Terminal y ejecuta el siguiente comando para instalar Python 3 con Homebrew:

```
brew install python
```

3. Después de la instalación, verifica que Python se haya instalado correctamente ejecutando `python3 --version`.

Linux:

1. La mayoría de las distribuciones de Linux ya vienen con Python preinstalado. Si no es así, puedes instalarlo a través del administrador de paquetes de tu distribución. Por ejemplo, en Ubuntu, puedes ejecutar:

```
sudo apt update
sudo apt install python3
```

2. Verifica la instalación ejecutando `python3 --version`.

Ejecución de la Aplicación

Una vez que hayas instalado Python, puedes ejecutar la aplicación Python siguiendo estos pasos:

1. Abre una terminal o línea de comandos y navega hasta el directorio donde están guardados los archivos `.py` que conforman la práctica.
2. Ejecuta el archivo Python con el comando `python Main.py` (o `python3 Main.py` en macOS y Linux).

Capturas de la ejecución de la aplicación

Menú principal

```
[salva PEC2_Codificacion % python3 Main.py]

GesAMA: Gestión de Alquiler de Máquinas Agrícolas

    Editar Máquina          (Pulsar M)
    Editar Finca            (Pulsar F)
    Listar Máquinas         (Pulsar L)
    Estado Fincas           (Pulsar E)
    Alquiler Máquina        (Pulsar A)
    Plan Mensual Máquina    (Pulsar P)
    Salir                   (Pulsar S)

Teclear una opción válida (M|F|L|E|A|P|S):
```

Editar Máquina

```
Editar Máquina:

Identificador (número entre 1 y 10)? 3
Nombre (entre 1 y 20 caracteres)? Apolo
Tipo (G-Grano, U-Uva, A-Aceituna, B-Borrar)? G
Capacidad (hectáreas/día)? 7
Ubicación inicial (Latitud)? 37.755
Ubicación inicial (Longitud)? -4.467

IMPORTANTE: Esta opción borra los datos anteriores. ¿Son correctos los nuevos datos (S/N)? S
```

Editar Finca

```
Editar Finca:

Identificador (número entre 1 y 20)? 5
Nombre (entre 1 y 20 caracteres)? Espartal
Tipo (G-Grano, U-Uva, A-Aceituna, B-Borrar)? G
Tamaño (Hectáreas)? 75
Ubicación (Latitud)? 42.454
Ubicación (Longitud)? -4.059

IMPORTANTE: Esta opción borra los datos anteriores. ¿Son correctos los nuevos datos (S/N)? S
```

Listar Máquinas

```
Tipo de máquinas a listar (G-Grano, U-Uva, A-Aceituna, T-Todas): G
Id      Nombre  Tipo   Capacidad
3       Apolo    G      7 hectáreas/día
```

Estado Fincas

```
Estado de fincas a listar (G-Grano, U-Uva, A- Aceituna, T- Todas): T
Id      Nombre  Tipo   Tamaño  Latitud Longitud
5       Espartal  G      75.0 hectáreas  42.454  -4.059
```

Alquiler Máquina

```
Alquiler Máquina:


Fecha comienzo cosecha: Día? 9
Fecha comienzo cosecha: Mes? 7
Fecha comienzo cosecha: Año? 2024
Identificador de finca (número entre 1 y 20)? 5
Identificador de máquina (número entre 1 y 10)? 3

      Resumen alquiler:

Máquina alquilada: Apolo (Id = 3)
Finca a cosechar: Espartal (Id = 5)
Traslado desde: finca Bercial (Id = 2)
Distancia entre fincas: 1000 km en línea recta
Tiempo de traslado: 09/07/2024 (1 día)
Fecha comienzo: 09/07/2024 Duración cosecha: 11 días
Fecha finalización: 20/07/2024

Es correcta la operación (S/N)? S
```

Plan Mensual Máquina

 Como se observa en la captura, este parte del programa queda pendiente de ser optimizada. Sobre todo, la visualización del calendario.

Plan Mensual Máquina:

Identificador máquina? 3
Selección Mes? 7
Selección Año? 2024

Plan mensual Máquina:

Plan Máquina: Apolo

		Julio				2024			
		L	M	X	J	V	S	D	
Tr	C1		Tr	C1		Tr	C1		C1
		C2				C2		Tr	
Tr	C2		Tr		Tr		Tr		
				C3				C3	Tr
Tr	C3		Tr				C3		
		C4		Tr		Tr		Tr	
	C4		Tr				C4		

Tiempo de traslados (Tr): 1 días
Tiempo de esperas: 10 días
Tiempo total de cosechas (C#): 22 días