


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

# Documento de Diseño del Proyecto: Gestión de Alquiler de Maquinaria Agrícola (GesAMA)

## Documento de Especificación de Requisitos de Software (SRD)

---

### 1. Introducción

#### 1.1. Objetivo

El objetivo de este documento es definir los requisitos funcionales y no funcionales del sistema GesAMA, así como establecer las expectativas de rendimiento, usabilidad, fiabilidad y seguridad.

#### 1.2. Ámbito

El sistema GesAMA es un programa desarrollado en Java para la gestión del alquiler de maquinaria agrícola utilizado en la recolección de distintos productos en fincas agrícolas.

#### 1.3. Definiciones, siglas y abreviaturas

- GesAMA: Gestión de Alquiler de Maquinaria Agrícola

### 2. Requisitos Funcionales

#### 2.1. Editar Máquina

- **Descripción:** Permite introducir los datos de una máquina cosechadora.
- **Entradas:**

- Identificador de la máquina (número entre 1 y 10)
  - Nombre de la máquina (entre 1 y 20 caracteres)
  - Tipo de máquina (G-Grano, U-Uva, A-Aceituna, B-Borrar)
  - Capacidad de la máquina (hectáreas/día)
  - Ubicación inicial (Latitud y Longitud)
- **Salida:** Confirmación de la edición de la máquina.

## 2.2. Editar Finca

- **Descripción:** Permite introducir los datos de una finca agrícola.
- **Entradas:**
  - Identificador de la finca (número entre 1 y 20)
  - Nombre de la finca (entre 1 y 20 caracteres)
  - Tipo de cultivo de la finca (G-Grano, U-Uva, A-Aceituna, B-Borrar)
  - Tamaño de la finca (Hectáreas)
  - Ubicación (Latitud y Longitud)
- **Salida:** Confirmación de la edición de la finca.

## 2.3. Listar Máquinas

- **Descripción:** Permite listar todas las máquinas disponibles o solo las del tipo seleccionado.
- **Entradas:** Tipo de máquinas a listar (G-Grano, U-Uva, A-Aceituna, T-Todas)
- **Salida:** Listado de máquinas con su identificador, nombre, tipo y capacidad.

## 2.4. Estado Fincas

- **Descripción:** Permite conocer el estado de todas las fincas o solo las del tipo seleccionado.
- **Entradas:** Tipo de fincas a listar (G-Grano, U-Uva, A-Aceituna, T-Todas)
- **Salida:** Estado de las fincas listadas con su identificador, nombre, tipo, tamaño, latitud y longitud.

## 2.5. Alquilar Máquina

- **Descripción:** Permite planificar el alquiler de una máquina para realizar la cosecha de una finca.
- **Entradas:**
  - Fecha de inicio de la cosecha
  - Identificador de la finca

- Identificador de la máquina
- **Salida:** Confirmación del alquiler y resumen del mismo.

## 2.6. Plan Mensual de Máquina

- **Descripción:** Permite generar el plan mensual de una máquina para un mes y año específicos.
- **Entradas:**
  - Identificador de la máquina
  - Mes y año
- **Salida:** Plan mensual de la máquina especificada para el mes y año seleccionados.

## 3. Requisitos No Funcionales

### 3.1. Usabilidad

- La interfaz de usuario debe ser intuitiva y fácil de usar, especialmente para los usuarios no técnicos.
- Se debe proporcionar retroalimentación clara al usuario sobre el resultado de sus acciones.

### 3.2. Rendimiento

- El sistema debe ser capaz de manejar la gestión de hasta 10 máquinas y 20 fincas de manera eficiente, incluso con grandes volúmenes de datos.

### 3.3. Confiabilidad

- El sistema debe ser robusto y resistente a fallos, minimizando la posibilidad de errores y garantizando la integridad de los datos.

### 3.4. Seguridad

- Se deben implementar medidas de seguridad para proteger los datos del sistema, incluyendo la autenticación de usuarios y el control de acceso.

## 4. Requisitos de Interfaz de Usuario

- El sistema GesAMA utilizará una interfaz de línea de comandos (CLI) para interactuar con los usuarios.

## 5. Requisitos de Hardware y Software

- El sistema GesAMA requerirá un entorno de ejecución Java (JRE) para su funcionamiento.
- Se recomienda una máquina con al menos 2 GB de RAM y un procesador de doble núcleo para un rendimiento óptimo.

# Documento de Diseño Arquitectónico (ADD)

---

## 1. Introducción

### 1.1. Objetivo

El objetivo de este documento es describir en detalle la arquitectura del sistema GesAMA, destacando los componentes principales y su interacción.

### 1.2. Ámbito

El ADD se centra en la estructura y organización del sistema GesAMA, así como en los principios arquitectónicos que guiarán su desarrollo.

### 1.3. Definiciones, siglas y abreviaturas

Se utilizan las definiciones establecidas en el SRD.

## 2. Panorámica del sistema

GesAMA estará compuesto por los siguientes módulos principales:

- **GesAMAController:** Coordinará las operaciones del sistema y gestionará la interacción con el usuario.
- **Máquina:** Representará las máquinas cosechadoras, con atributos como id, nombre, tipo, capacidad y ubicación.
- **Finca:** Representará las fincas agrícolas, con atributos como id, nombre, tipo, tamaño y ubicación.
- **Alquiler:** Gestionará los alquileres de máquinas para la cosecha de fincas, con atributos como id de máquina, id de finca, fecha de inicio y duración.

## 3. Contexto del sistema

GesAMA interactuará con los usuarios a través de una interfaz de línea de comandos (CLI) y gestionará datos sobre máquinas, fincas y alquileres en un entorno de memoria.

## 4. Diseño del sistema

### 4.1. Arquitectura General

La arquitectura del sistema GesAMA seguirá un enfoque de tres capas:

- **Capa de Presentación:** Interacción con el usuario a través de la interfaz de línea de comandos (CLI).
- **Capa de Lógica de Negocio:** Contiene la lógica de negocio del sistema, incluyendo la gestión de máquinas, fincas y alquileres.
- **Capa de Acceso a Datos:** Gestiona el acceso a los datos almacenados en memoria.

### 4.2. Componentes del Sistema

#### GesAMAController

- **Funcionalidad:**
  - Coordinar las operaciones del sistema.
  - Gestionar la interacción con el usuario.
- **Interfaces:**
  - Interactúa con las capas de Lógica de Negocio y Presentación.

#### Máquina

- **Funcionalidad:**
  - Representar las máquinas cosechadoras.
  - Almacenar información sobre cada máquina, como id, nombre, tipo, capacidad y ubicación.
- **Interfaces:**
  - Interactúa con la capa de Acceso a Datos para almacenar y recuperar datos.

#### Finca

- **Funcionalidad:**
  - Representar las fincas agrícolas.
  - Almacenar información sobre cada finca, como id, nombre, tipo, tamaño y ubicación.
- **Interfaces:**

- Interactúa con la capa de Acceso a Datos para almacenar y recuperar datos.

## **Alquiler**

- **Funcionalidad:**

- Gestionar los alquileres de máquinas para la cosecha de fincas.
- Almacenar información sobre cada alquiler, como id de máquina, id de finca, fecha de inicio y duración.

- **Interfaces:**

- Interactúa con la capa de Acceso a Datos para almacenar y recuperar datos.

## **5. Diseño de los Componentes**

### **5.1. Componente: GesAMAController**

#### **5.1.1. Identificador**

Nombre del componente: GesAMAController.

#### **5.1.2. Tipo**

Tipo de componente: Controlador.

#### **5.1.3. Objetivo**

El objetivo del controlador GesAMAController es coordinar las operaciones del sistema GesAMA y gestionar la interacción con el usuario.

#### **5.1.4. Función**

El controlador GesAMAController realiza las siguientes funciones:

- Coordinar la ejecución de las operaciones del sistema.
- Gestionar la entrada y salida de datos con el usuario.
- Interpretar los comandos del usuario y dirigirlos a los componentes correspondientes para su ejecución.

#### **5.1.5. Subordinados**

El controlador GesAMAController utiliza los siguientes componentes:

- Máquina

- Finca
- Alquiler

#### **5.1.6. Dependencias**

El controlador GesAMAController es utilizado por los otros componentes del sistema para solicitar operaciones y obtener resultados.

#### **5.1.7. Interfaces**

El controlador GesAMAController interactúa con el usuario a través de la interfaz de línea de comandos (CLI). Utiliza métodos para procesar los comandos del usuario y mostrar resultados.

#### **5.1.8. Recursos**

El controlador GesAMAController utiliza recursos del sistema Java, como la entrada y salida estándar, para interactuar con el usuario.

#### **5.1.9. Referencias**

- Documento de Especificación de Requisitos de Software (SRD).

#### **5.1.10. Proceso**

El controlador GesAMAController implementa algoritmos para procesar los comandos del usuario y dirigir las operaciones correspondientes a los componentes adecuados.

#### **5.1.11. Datos**

El controlador GesAMAController no almacena datos internos, pero utiliza los datos de los otros componentes del sistema para realizar sus funciones.

### **5.2. Componente: Máquina**

#### **5.2.1. Identificador**

Nombre del componente: Máquina.

#### **5.2.2. Tipo**

Tipo de componente: Clase.

#### **5.2.3. Objetivo**

El objetivo de la clase Máquina es representar las máquinas cosechadoras en el sistema GesAMA y almacenar información relevante sobre cada máquina.

#### **5.2.4. Función**

La clase Máquina realiza las siguientes funciones:

- Almacenar información sobre las máquinas cosechadoras, como su identificador, nombre, tipo, capacidad y ubicación.
- Proporcionar métodos para acceder y modificar los datos de las máquinas.

#### **5.2.5. Subordinados**

La clase Máquina no tiene componentes subordinados.

#### **5.2.6. Dependencias**

La clase Máquina es utilizada por el controlador GesAMAController para realizar operaciones relacionadas con la gestión de máquinas.

#### **5.2.7. Interfaces**

La clase Máquina proporciona métodos públicos para acceder y modificar los datos de las máquinas.

#### **5.2.8. Recursos**

La clase Máquina utiliza recursos de memoria para almacenar los datos de las máquinas.

#### **5.2.9. Referencias**

- Documento de Especificación de Requisitos de Software (SRD).

#### **5.2.10. Proceso**

La clase Máquina implementa métodos para realizar operaciones relacionadas con la gestión de máquinas, como la edición de datos y la generación de informes.

#### **5.2.11. Datos**

La clase Máquina almacena datos sobre las máquinas cosechadoras, como su identificador, nombre, tipo, capacidad y ubicación.

### **5.3. Componente: Finca**



### **5.3.1. Identificador**

Nombre del componente: Finca.

### **5.3.2. Tipo**

Tipo de componente: Clase.

### **5.3.3. Objetivo**

El objetivo de la clase Finca es representar las fincas agrícolas en el sistema GesAMA y almacenar información relevante sobre cada finca.

### **5.3.4. Función**

La clase Finca realiza las siguientes funciones:

- Almacenar información sobre las fincas agrícolas, como su identificador, nombre, tipo, tamaño y ubicación.
- Proporcionar métodos para acceder y modificar los datos de las fincas.

### **5.3.5. Subordinados**

La clase Finca no tiene componentes subordinados.

### **5.3.6. Dependencias**

La clase Finca es utilizada por el controlador GesAMAController para realizar operaciones relacionadas con la gestión de fincas.

### **5.3.7. Interfaces**

La clase Finca proporciona métodos públicos para acceder y modificar los datos de las fincas.

### **5.3.8. Recursos**

La clase Finca utiliza recursos de memoria para almacenar los datos de las fincas.

### **5.3.9. Referencias**

- Documento de Especificación de Requisitos de Software (SRD).

### **5.3.10. Proceso**

La clase Finca implementa métodos para realizar operaciones relacionadas con la gestión

de fincas, como la edición de datos y la generación de informes.

#### **5.3.11. Datos**

La clase Finca almacena datos sobre las fincas agrícolas, como su identificador, nombre, tipo, tamaño y ubicación.

### **5.4. Componente: Alquiler**

#### **5.4.1. Identificador**

Nombre del componente: Alquiler.

#### **5.4.2. Tipo**

Tipo de componente: Clase.

#### **5.4.3. Objetivo**

El objetivo de la clase Alquiler es gestionar los alquileres de máquinas para la cosecha de fincas en el sistema GesAMA y almacenar información relevante sobre cada alquiler.

#### **5.4.4. Función**

La clase Alquiler realiza las siguientes funciones:

- Gestionar los alquileres de máquinas para la cosecha de fincas, almacenando información como el identificador de la máquina, el identificador de la finca, la fecha de inicio y la duración del alquiler.
- Proporcionar métodos para acceder y modificar los datos de los alquileres.

#### **5.4.5. Subordinados**

La clase Alquiler no tiene componentes subordinados.

#### **5.4.6. Dependencias**

La clase Alquiler es utilizada por el controlador GesAMAController para realizar operaciones relacionadas con la gestión de alquileres.

#### **5.4.7. Interfaces**

La clase Alquiler proporciona métodos públicos para acceder y modificar los datos de los alquileres.

#### **5.4.8. Recursos**

La clase Alquiler utiliza recursos de memoria para almacenar los datos de los alquileres.

#### **5.4.9. Referencias**

- Documento de Especificación de Requisitos de Software (SRD).

#### **5.4.10. Proceso**

La clase Alquiler implementa métodos para realizar operaciones relacionadas con la gestión de alquileres, como la creación de nuevos alquileres y la generación de informes.

#### **5.4.11. Datos**

La clase Alquiler almacena datos sobre los alquileres de máquinas, como el identificador de la máquina, el identificador de la finca, la fecha de inicio y la duración del alquiler.

### **6. Viabilidad de recursos estimados**

#### **6.1. Recursos Necesarios**

##### **6.1.1. Recursos de Hardware**

Para ejecutar el sistema GesAMA de manera eficiente, se recomienda el siguiente hardware:

- Procesador: Se recomienda un procesador de doble núcleo o superior para manejar las operaciones del sistema de manera eficiente.
- Memoria RAM: Se sugiere un mínimo de 2 GB de RAM para garantizar un rendimiento óptimo del sistema.
- Almacenamiento: El sistema GesAMA no requiere una cantidad significativa de almacenamiento en disco, ya que los datos se gestionan en memoria durante la ejecución.

##### **6.1.2. Recursos de Software**

El sistema GesAMA se desarrollará en el lenguaje de programación Java, por lo que se necesitará:

- Entorno de ejecución Java (JRE): Se requiere una versión compatible de JRE para ejecutar el sistema en cualquier plataforma compatible con Java.
- Entorno de desarrollo integrado (IDE): Se recomienda el uso de un IDE como Eclipse o IntelliJ IDEA para el desarrollo y depuración del código.

- Dependencias de terceros: Es posible que se requieran bibliotecas o frameworks adicionales para la implementación de ciertas funcionalidades del sistema, como el manejo de fechas o la interacción con la interfaz de línea de comandos.

## **6.2. Evaluación de Viabilidad**

### **6.2.1. Viabilidad Técnica**

El sistema GesAMA se basa en tecnologías estándar y no requiere componentes o plataformas especializadas. La arquitectura propuesta es viable desde el punto de vista técnico y puede implementarse utilizando herramientas y recursos ampliamente disponibles.

### **6.2.2. Viabilidad Económica**

El desarrollo del sistema GesAMA no requiere inversiones significativas en términos de hardware o software. Las herramientas necesarias para el desarrollo están disponibles de forma gratuita o a un costo mínimo. Además, el sistema se puede implementar en cualquier plataforma compatible con Java, lo que reduce los costos de infraestructura.

### **6.2.3. Viabilidad Operativa**

El sistema GesAMA se diseñará para ser fácil de usar y mantener. Se proporcionará documentación detallada para facilitar la instalación, configuración y uso del sistema. Además, se seguirán prácticas de desarrollo de software como la modularidad y la reutilización de código para facilitar futuras actualizaciones y modificaciones.

## **6.3. Conclusiones**

Basado en la evaluación de viabilidad técnica, económica y operativa, se concluye que el desarrollo e implementación del sistema GesAMA es factible y puede llevarse a cabo con recursos razonables. El sistema proporcionará una solución eficiente y fácil de usar para la gestión del alquiler de maquinaria agrícola.

## **7. Matriz de requisitos componentes**

En esta sección, se establecerá una matriz que relaciona los requisitos del sistema con los componentes arquitectónicos correspondientes. Esto proporcionará una visión clara de cómo cada componente contribuye a cumplir los requisitos del sistema.

| Requisito               | GesAMAController | Máquina | Finca | Alquiler |
|-------------------------|------------------|---------|-------|----------|
| Editar Máquina          | X                | X       |       |          |
| Editar Finca            | X                |         | X     |          |
| Listar Máquinas         | X                | X       |       |          |
| Estado Fincas           | X                |         | X     |          |
| Alquilar Máquina        | X                | X       | X     | X        |
| Plan Mensual de Máquina | X                | X       |       |          |

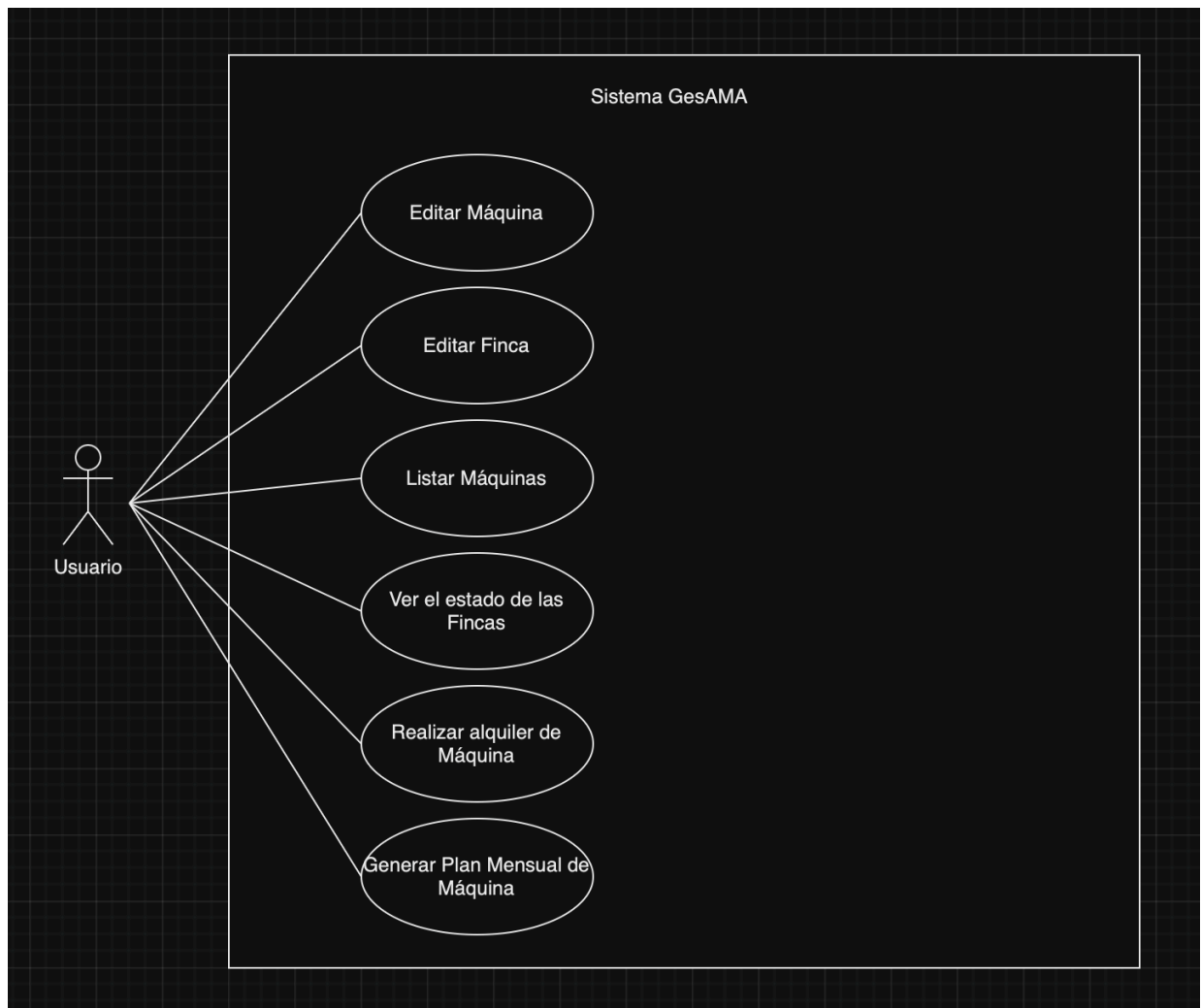
Esta matriz muestra cómo cada requisito del sistema se relaciona con los diferentes componentes arquitectónicos del sistema GesAMA. Por ejemplo, el requisito "Editar Máquina" está asociado al controlador GesAMAController y a la clase Máquina, ya que implica la interacción del usuario con la interfaz para editar los datos de una máquina. De manera similar, el requisito "Alquilar Máquina" está asociado al controlador GesAMAController, la clase Máquina, la clase Finca y la clase Alquiler, ya que implica la interacción del usuario para alquilar una máquina para una finca específica.

Esta matriz proporciona una guía clara para el diseño e implementación de los diferentes componentes del sistema GesAMA, asegurando que cada requisito del sistema esté correctamente asignado y cubierto por los componentes correspondientes.

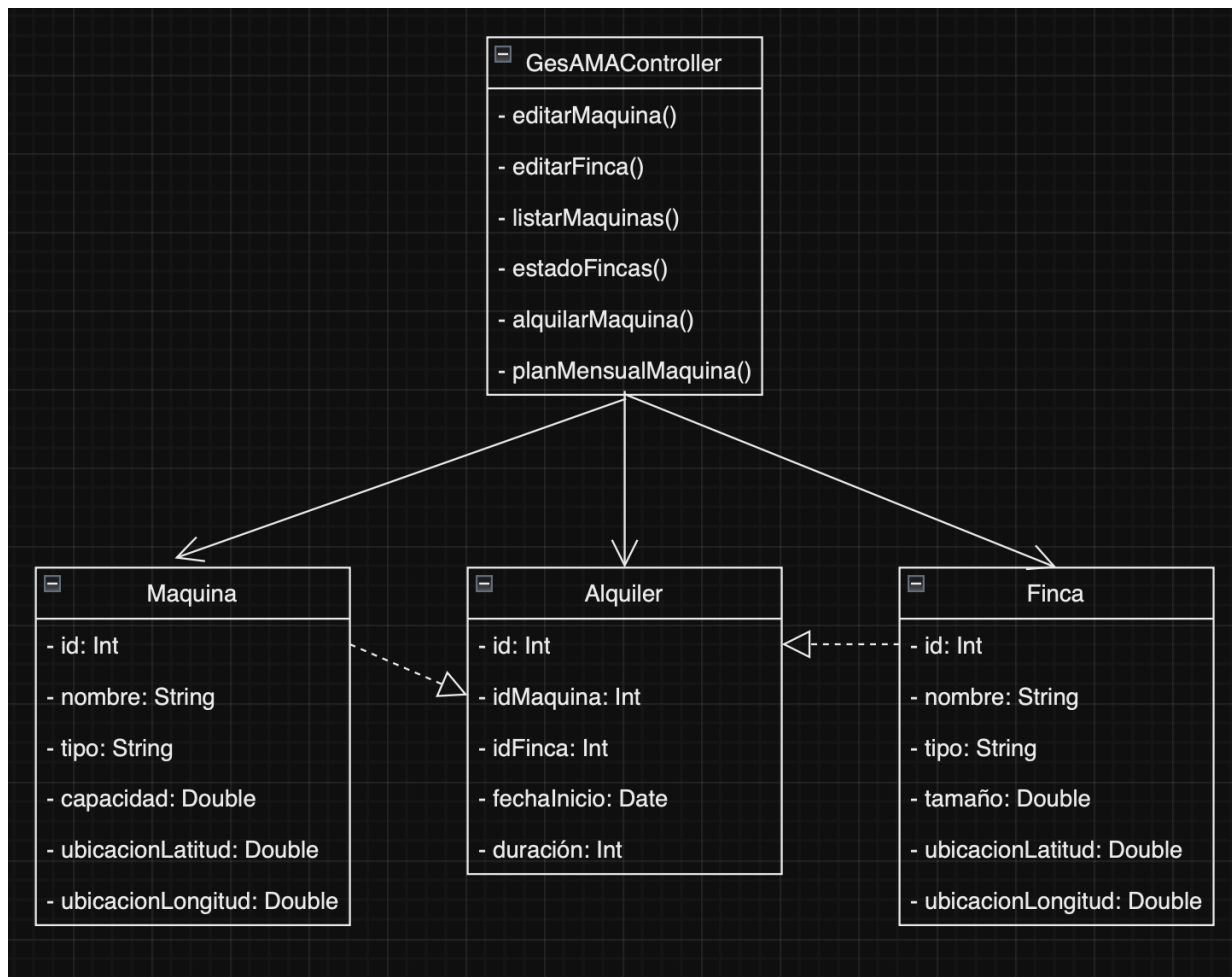
## Diagramas UML

---

### Diagrama de casos de uso



## Diagrama de clases



La descripción de las relaciones que hay entre las clases en el diagrama se dá a continuación:

#### 1. GesAMAController -> Máquina

- Relación: Asociación
- Descripción: El controlador GesAMAController interactúa con la clase Máquina para realizar operaciones como editar máquinas, listar máquinas disponibles, etc.

#### 2. GesAMAController -> Finca

- Relación: Asociación
- Descripción: El controlador GesAMAController interactúa con la clase Finca para realizar operaciones como editar fincas, listar el estado de las fincas, etc.

#### 3. GesAMAController -> Alquiler

- Relación: Asociación
- Descripción: El controlador GesAMAController interactúa con la clase Alquiler para realizar operaciones como alquilar una máquina para una finca específica.

#### 4. Alquiler -> Máquina

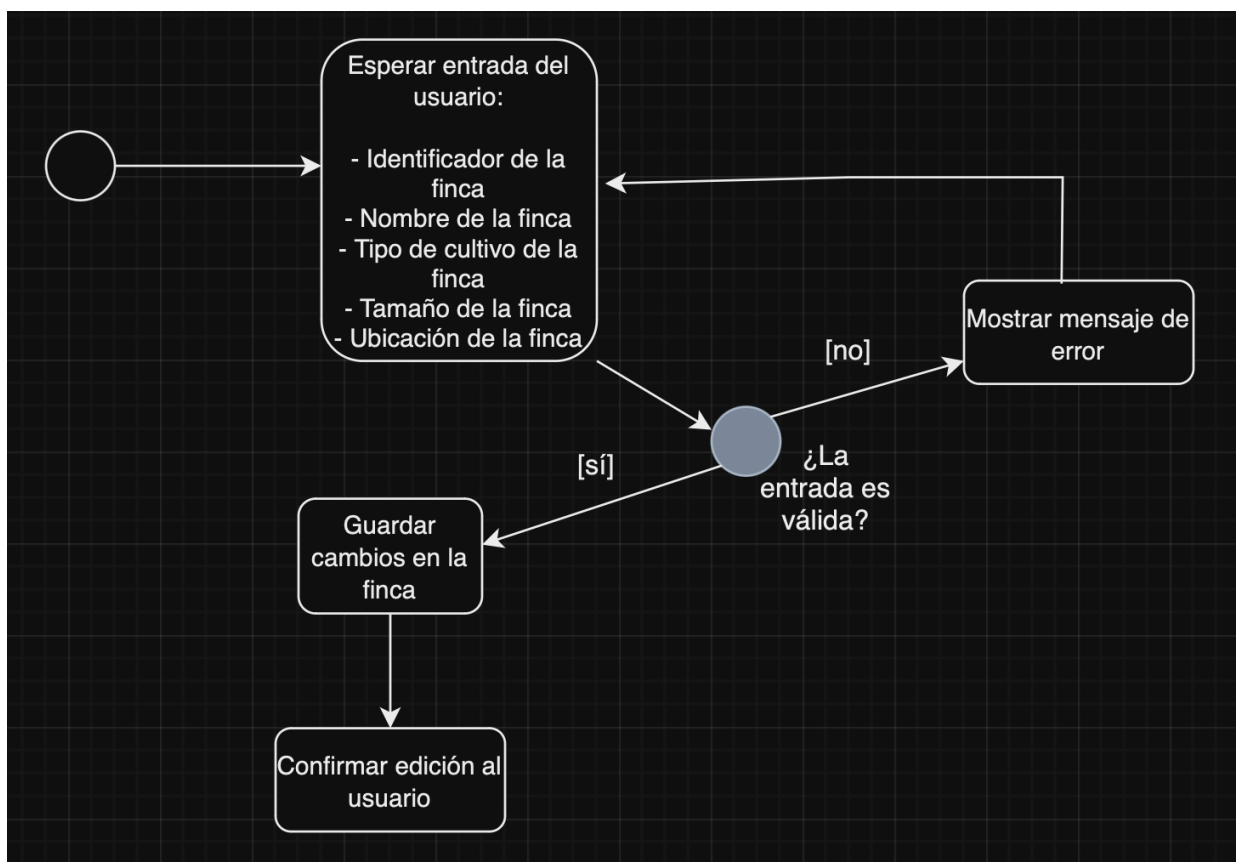
- Relación: Dependencia
- Descripción: La clase Alquiler depende de la clase Máquina para almacenar el identificador de la máquina que se alquila.

## 5. Alquiler -> Finca

- Relación: Dependencia
- Descripción: La clase Alquiler depende de la clase Finca para almacenar el identificador de la finca para la que se alquila la máquina.

## Diagrama de estado

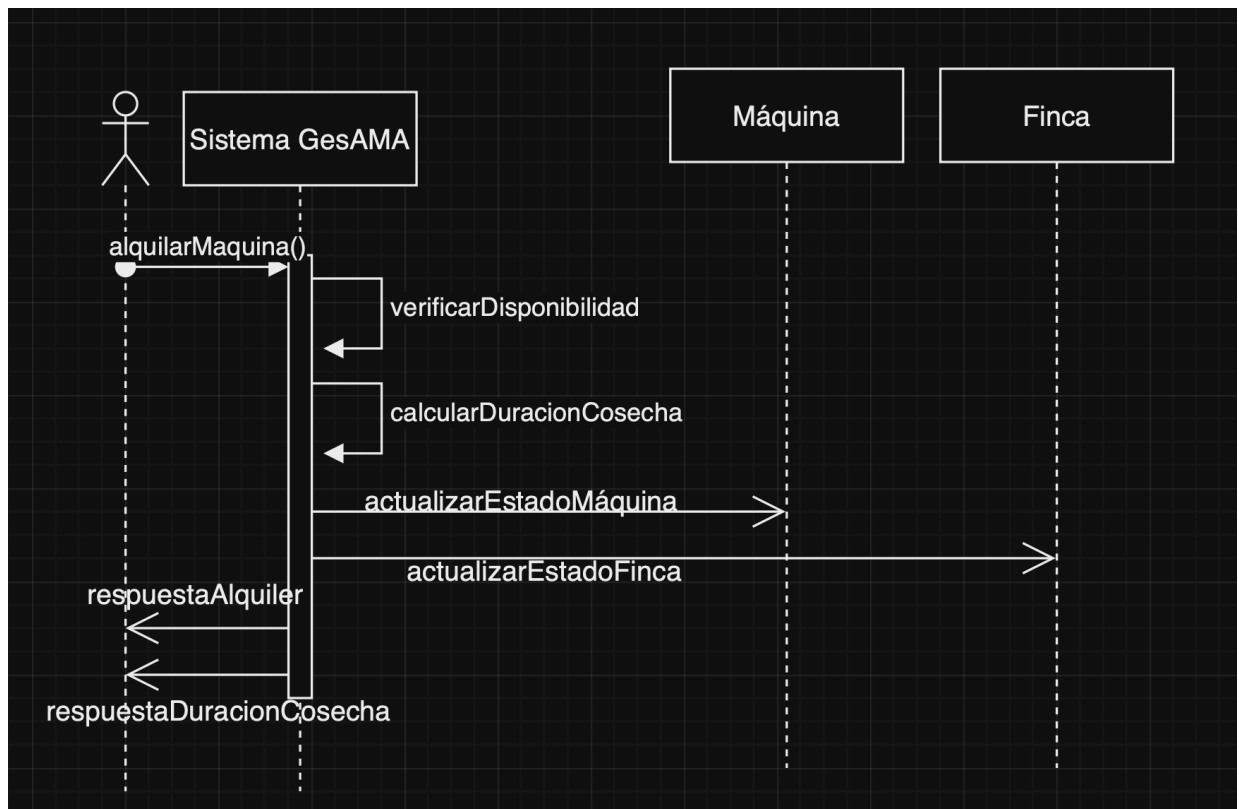
### Ejemplo: Editar Finca



## Diagrama de secuencias

### Ejemplo: Alquilar Máquina





## Diagrama de actividad

### Ejemplo: Editar Máquina

