

Taller 5

1. DATOS INFORMATIVOS

Carrera: Ingeniería de Software

Asignatura: Análisis y Diseño de Software

Tema del taller: Casos de Uso

Docente: Ing. Jenny Ruiz

Integrantes: Stefany Díaz, Moisés Benalcázar, Mateo Medranda.

Fecha: 14 de oct. de 25 **NRC:** 23837

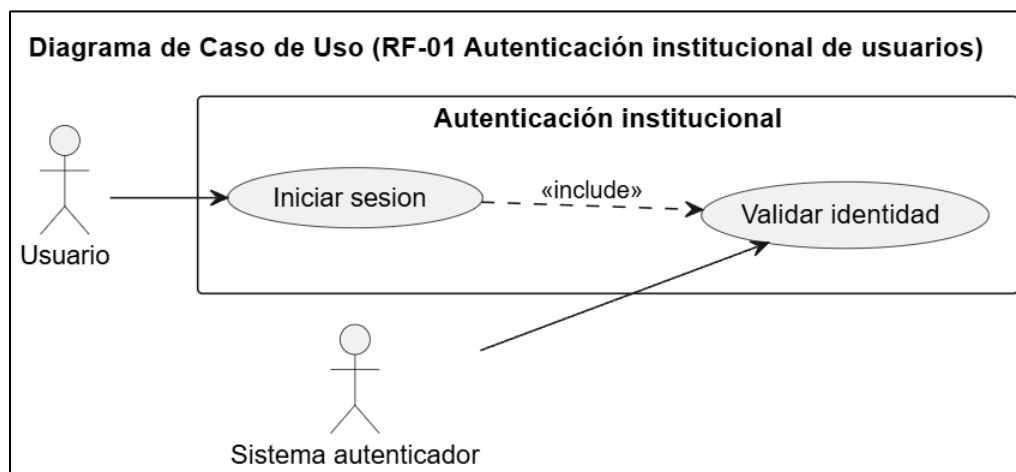
2. DESARROLLO

Para la elaboración de los diagramas de casos de uso correspondientes a los siete requisitos funcionales del sistema, se utilizó la herramienta PlantUML (2025), la cual permite generar diagramas a partir de código en lenguaje de modelado.

Primero, se identificaron los actores principales y las interacciones del sistema con base en los requisitos funcionales definidos previamente. Luego, se redactó el código de cada caso de uso en formato PlantUML, cuidando la correcta relación entre actores y funcionalidades.

Finalmente, se generaron los diagramas automáticos desde el código para su validación visual.

RF-01 *Diagrama de Casos de Uso*



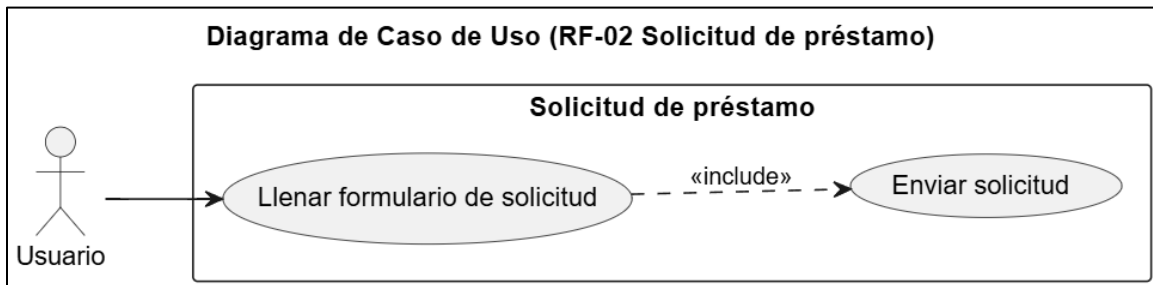
RF-01 *Código*

```
@startuml
title Diagrama de Caso de Uso (RF-01 Autenticación institucional de usuarios)
left to right direction

actor Usuario
actor "Sistema autenticador"

rectangle "Autenticación institucional" {
    usecase "Iniciar sesion" as CU1
    usecase "Validar identidad" as CU2
}
Usuario --> CU1
CU1 ..> CU2 : <<include>>
"Sistema autenticador" --> CU2
@enduml
```

RF-02 Diagrama de Casos de Uso



RF-02 Código:

```
@startuml
left to right direction
title Diagrama de Caso de Uso (RF-02 Solicitud de préstamo)

actor Usuario

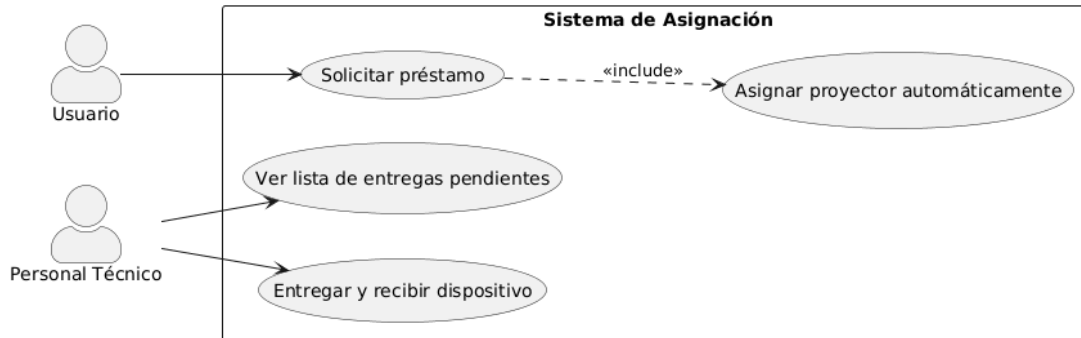
rectangle "Solicitud de préstamo" {
    usecase "Llenar formulario de solicitud" as CU1
    usecase "Enviar solicitud" as CU2
}

Usuario --> CU1
CU1 ..> CU2 : <<include>>
@enduml
```

RF-03

Diagrama de Casos de Uso

Diagrama de Casos de Uso (RF-03 Asignación automática de un proyector)



Código:

```

@startuml
left to right direction
skinparam packageStyle rectangle
skinparam actorStyle awesome
skinparam titleBorderRoundCorner 15
skinparam titleFontSize 18
title Diagrama de Casos de Uso (RF-03 Asignación automática de un proyector)

```

```

actor Usuario
actor "Personal Técnico" as Tecnico

```

```

rectangle "Sistema de Asignación" {
    (Solicitar préstamo) as UC_Solicitud
    (Asignar proyector automáticamente) as UC_Asignacion
    (Ver lista de entregas pendientes) as UC_VerLista
    (Entregar y recibir dispositivo) as UC_Entregar
}

```

```

Usuario --> UC_Solicitud
UC_Solicitud ..> UC_Asignacion : <<include>>
Tecnico --> UC_VerLista
Tecnico --> UC_Entregar

```

```

@enduml

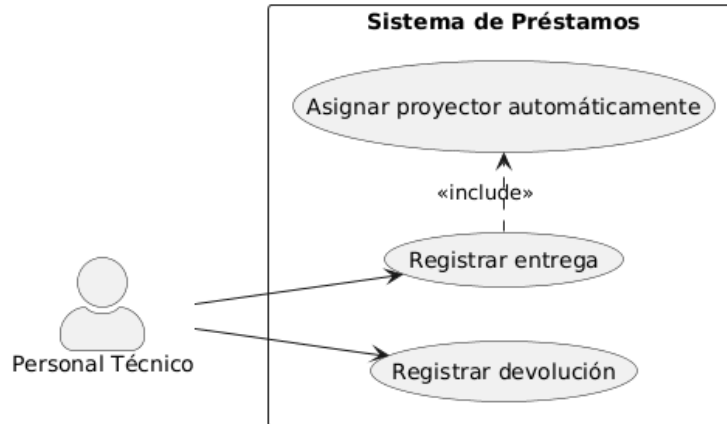
```

RF-04

Diagrama de Casos de Uso



Diagrama de Casos de Uso (RF-04 Registro de Entrega y Devolución)



Código

```
@startuml
left to right direction
skinparam packageStyle rectangle
skinparam actorStyle awesome
skinparam titleBorderRoundCorner 15
skinparam titleFontSize 18
title Diagrama de Casos de Uso (RF-04 Registro de Entrega y Devolución)

actor "Personal Técnico" as Tecnico

rectangle "Sistema de Préstamos" {
    (Registrar entrega) as UC_Registro
    (Registrar devolución) as UC_Devolucion

    (Asignar proyector automáticamente) as UC_Asignacion
}

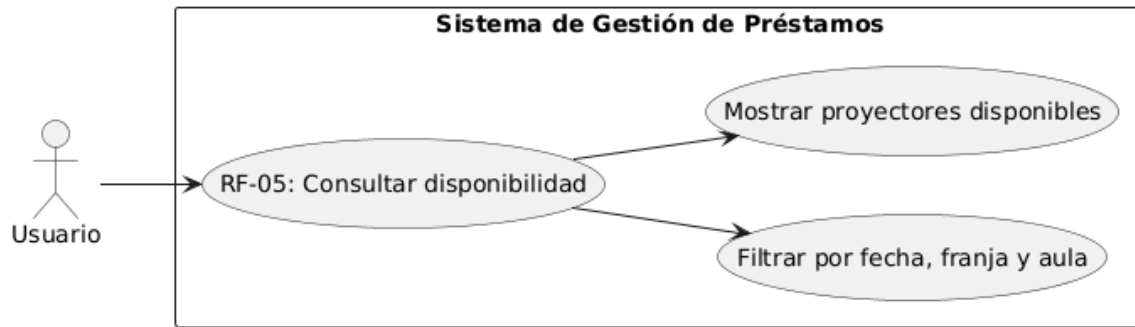
Tecnico --> UC_Registro
Tecnico --> UC_Devolucion

UC_Registro .> UC_Asignacion : <<include>>

@enduml
```

RF-05

Diagrama de Casos de Uso



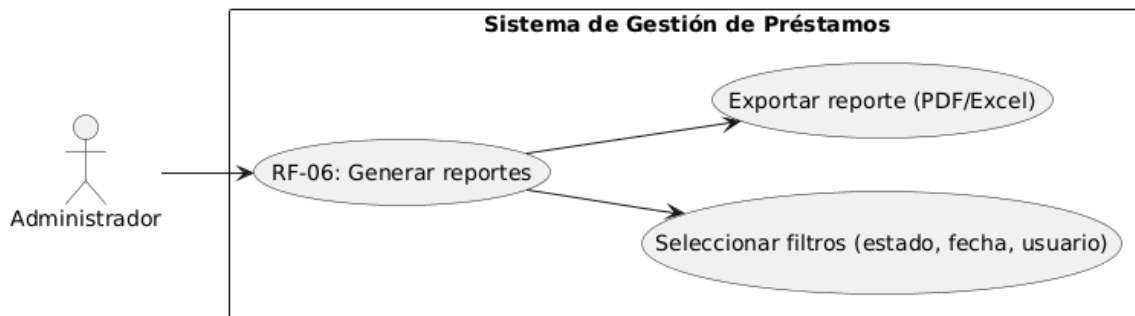
Código

```
@startuml
left to right direction
actor "Usuario" as U
rectangle "Sistema de Gestión de Préstamos" {
    usecase "RF-05: Consultar disponibilidad" as UC05
    usecase "Filtrar por fecha, franja y aula" as UC05A
    usecase "Mostrar proyectores disponibles" as UC05B
}
```

```
U --> UC05
UC05 --> UC05A
UC05 --> UC05B
@enduml
```

RF-06

Diagrama de Casos de Uso



Código

```
@startuml
left to right direction
actor "Administrador" as A
rectangle "Sistema de Gestión de Préstamos" {
    usecase "RF-06: Generar reportes" as UC06
}
```

```

usecase "Seleccionar filtros (estado, fecha, usuario)" as UC06A
usecase "Exportar reporte (PDF/Excel)" as UC06B
}

```

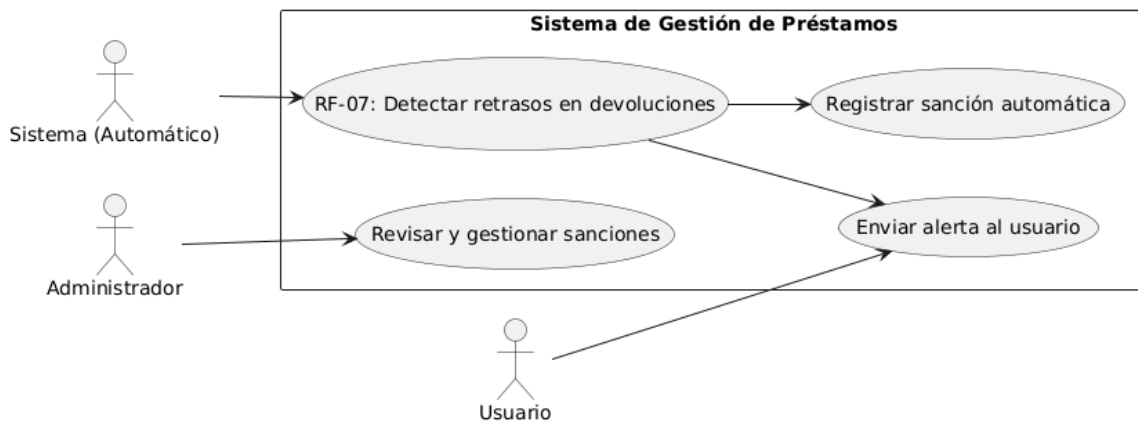
```

A --> UC06
UC06 --> UC06A
UC06 --> UC06B
@enduml

```

RF-07

Diagrama de Casos de Uso



Código

```

@startuml
left to right direction
actor "Sistema (Automático)" as S
actor "Usuario" as U
actor "Administrador" as A

rectangle "Sistema de Gestión de Préstamos" {
    usecase "RF-07: Detectar retrasos en devoluciones" as UC07A
    usecase "Enviar alerta al usuario" as UC07B
    usecase "Registrar sanción automática" as UC07C
    usecase "Revisar y gestionar sanciones" as UC07D
}

S --> UC07A
UC07A --> UC07B
UC07A --> UC07C
A --> UC07D
U --> UC07B
@enduml

```

3. CONCLUSIONES

El análisis de los Requisitos Funcionales (RF) es crucial para asegurar que el sistema desarrollado cumpla con las expectativas del usuario y ofrezca las funcionalidades necesarias para resolver los problemas planteados. El uso de PlantUML para generar los diagramas de casos de uso facilita una representación visual clara y precisa de las interacciones entre los actores y el sistema, lo cual es clave para la comprensión y validación de los requisitos. Con estos diagramas, se obtiene una visión completa de los procesos y cómo cada uno contribuye al funcionamiento del sistema. Además, al haber cubierto todos los RF con casos de uso, se ha logrado garantizar que todas las funcionalidades necesarias estén debidamente modeladas y documentadas, lo que minimiza el riesgo de omisiones durante las etapas posteriores del desarrollo.

4. RECOMENDACIONES

Es recomendable continuar utilizando herramientas como **PlantUML** para documentar casos de uso a lo largo del proyecto, ya que permite una visualización fácil y ágil para los diferentes stakeholders (desarrolladores, clientes, testers). También es fundamental revisar periódicamente los diagramas de casos de uso con los usuarios o clientes para asegurarse de que se mantengan alineados con sus necesidades y expectativas, evitando malentendidos o modificaciones costosas más adelante. Además, se debe seguir asegurando la trazabilidad de cada caso de uso, de modo que se conecte claramente con los requisitos y los resultados de las pruebas en las fases posteriores del desarrollo. La implementación de una metodología ágil o estructurada puede contribuir a realizar estos ajustes de manera eficiente, garantizando la calidad y el cumplimiento de los objetivos.

5. REFERENCIAS

PlantUML. (2025). *PlantUML [software]*. <https://plantuml.com>