



TALLER 6

1. DATOS INFORMATIVOS

Carrera: Ingeniería de Software

Asignatura: Analisis y Diseño de Software

Tema del taller: Relación entre el Análisis de Requisitos Funcionales y la Generación de Casos de Uso con IAGen (PlantUML)

Docente: Ing. Jenny Ruiz

Integrantes: Stefany Díaz, Moisés Benalcázar, Mateo Medranda

Fecha: 29 de oct. de 25

NRC : 27837

2. DESARROLLO

Parte 1. Lectura y análisis (10 min)

Revisa los RF de tu ERS al menos 1 RF por cada estudiante de grupo lo siguiente es solo un ejemplo:

- RF01 – Gestión de Medicamentos
- RF02 – Recordatorios de Medicación
- RF03 – Informes del Historial de Medicamentos

Identifica:

1. El actor principal en cada requisito.
2. Las acciones principales que el sistema debe ejecutar.
3. Las relaciones de inclusión (<<include>>) o extensión (<<extend>>) que aparecen en los diagramas UML.

Parte 2. Pensamiento crítico (20 min)

Responde con tus propias palabras:

1. ¿Qué diferencias observas entre los Casos de Uso derivados de entrevistas o descripciones textuales y los Casos de Uso generados automáticamente en PlantUML?

(Piensa en la precisión, completitud y claridad visual del modelo.)

2. ¿De qué manera el uso de PlantUML facilita (o limita) el trabajo del analista al modelar los requisitos funcionales?

(Considera aspectos como automatización, trazabilidad y comunicación entre analista y desarrollador.)

RF-02 Solicitud de prestamo

Tabla 1.

Elementos del requisito para un caso de uso (indicaciones parte 1)

Elemento	Detalles
Actor principal	Usuario/Solicitante
Acciones principales del sistema	<ul style="list-style-type: none"> • Validar sanciones del solicitante. • Solicitud de un proyector por su disponibilidad. • Mostrar disponibilidad de franja horaria. • Notificar solicitud al coordinador.
Relaciones de inclusión (<<include>>)	<p>Otra relación de inclusión es la de la disponibilidad de la franja horaria ya que esto tiene que hacerse cuando se vaya a realizar la solicitud y el usuario pueda consultar la disponibilidad en la franja horaria que tiene.</p> <p>Además un <<include>> se da entre Solicitud de un proyector con la comprobación de sanciones ya que si el solicitante tiene más de 3 sanciones no se le es posible realizar el préstamo de un proyector.</p>
Relaciones de extensión (<<extend>>)	Se tiene una relación <<extend>> con la notificación al coordinador, puesto que se realiza luego de realizar la solicitud del préstamo.

Figura 1.

Diagrama de casos de uso del requisito funcional RF-02

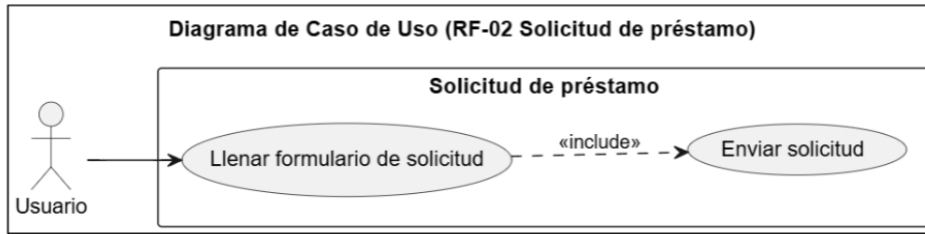
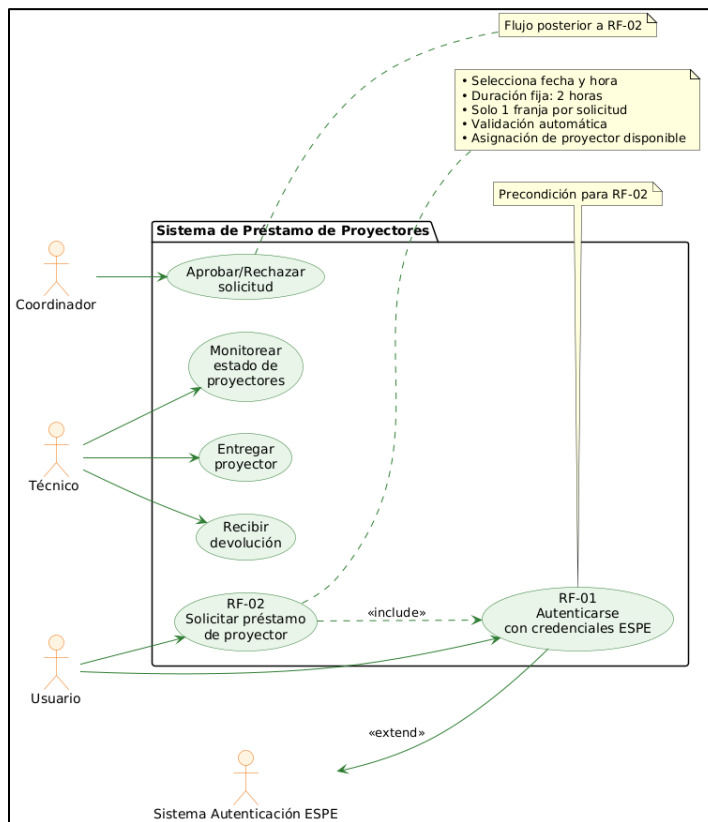


Figura 2.

Diagrama de casos de uso del requisito funcional RF-02 generado automáticamente



RF-03 Asignación automática de un proyector

Tabla 2.

Elementos del requisito para un caso de uso (indicaciones parte 1)

Elemento	Detalles
Actor principal	Usuario
Acciones principales del sistema	<ul style="list-style-type: none"> • Registrar solicitud de préstamo • Asignación automática de proyector • Mostrar entregas pendientes • Registrar entrega
Relaciones de inclusión (<<include>>)	<p>Relación entre solicitar préstamo por parte del usuario, y la asignación automática del proyector por parte del sistema, corresponde a “Include” porque una vez que se realice la solicitud, el sistema inmediatamente deberá asignar un dispositivo de forma automática.</p> <p>También posee una relación de inclusión desde la asignación automática con el bloqueo de disponibilidad de dicho dispositivo.</p>
Relaciones de extensión (<<extend>>)	Posee una relación de extensión definiendo una acción opcional en caso de necesitar aprobar una solicitud de forma manual, pero solo para casos excepcionales.

Figura 3.

Diagrama de casos de uso del requisito funcional RF-03

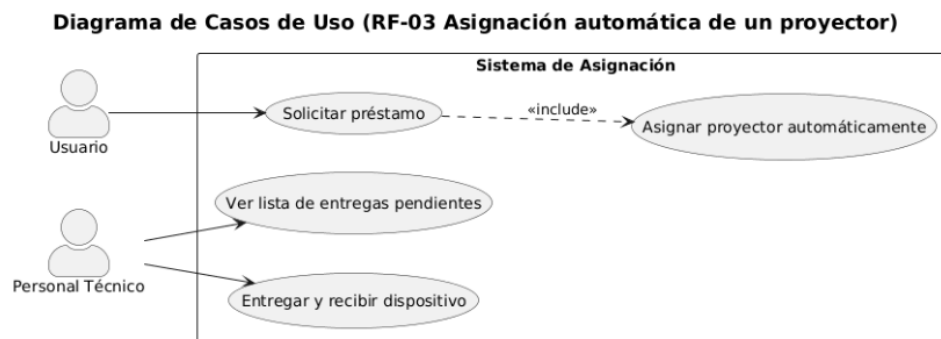
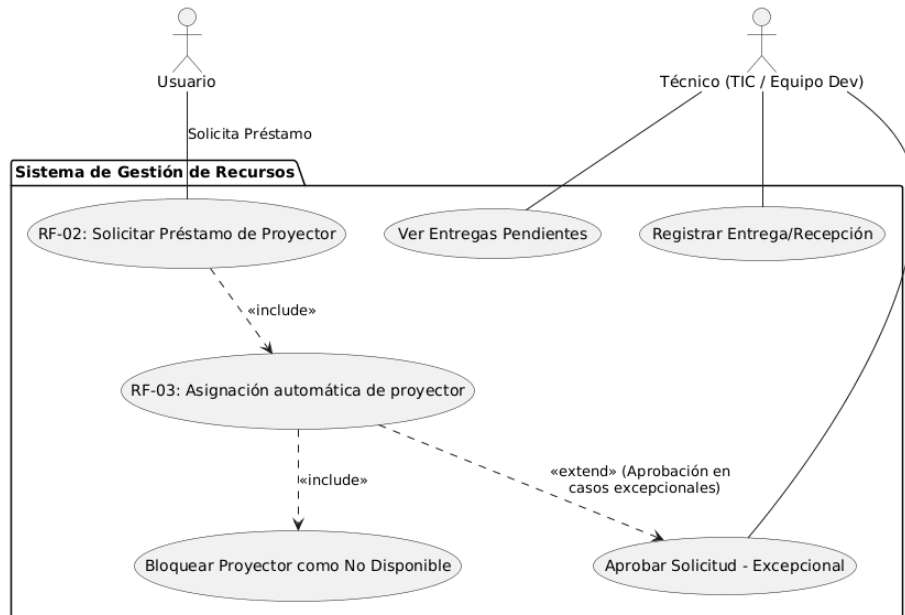


Figura 4.

Diagrama de casos de uso del requisito funcional RF-03 generado automáticamente



RF-07 Alertas por Retraso y Gestión de Sanciones

Tabla 3

Elementos del requisito para un caso de uso (indicaciones parte 1)

Elemento	Detalles
Actor principal	Sistema (Automático)El sistema actúa como actor principal porque es quien detecta automáticamente los retrasos en la devolución de proyectores y ejecuta acciones sin intervención humana (envío de alertas y registro de sanciones). Se identifica así porque el comportamiento depende de condiciones temporales que el sistema monitorea en segundo plano.
Actores principales del sistema	Usuario y Administrador <ul style="list-style-type: none"> - Usuario: Recibe las alertas cuando tiene una devolución pendiente o retrasada, y puede visualizar sus sanciones en el sistema. - Administrador: Supervisa los registros de sanciones, puede revisar alertas generadas y gestionar excepciones (por ejemplo, eliminar sanciones injustificadas). Estos actores interactúan con las funciones del sistema, pero no las inician directamente.
Relaciones de inclusión (<>)	<<include>> <i>Enviar alerta al usuario</i> <<exclude>> <i>Registrar sanción automática</i> Ambas se incluyen dentro del caso principal “Detectar retrasos en devoluciones”, porque son acciones obligatorias que siempre se ejecutan

	cuando el sistema detecta una devolución fuera del tiempo límite. Estas relaciones representan subprocesos indispensables dentro del flujo principal.
Relaciones de extensión (<>)	<<extend>> <i>Revisar y gestionar sanciones (Administrador)</i> Esta relación se considera de extensión porque la revisión de sanciones solo ocurre bajo ciertas condiciones, por ejemplo, cuando el usuario apela una sanción o se requiere verificación manual. No siempre se ejecuta, por eso se marca como una extensión opcional del flujo principal.

Figura 5.

Diagrama de casos de uso del requisito funcional RF-07

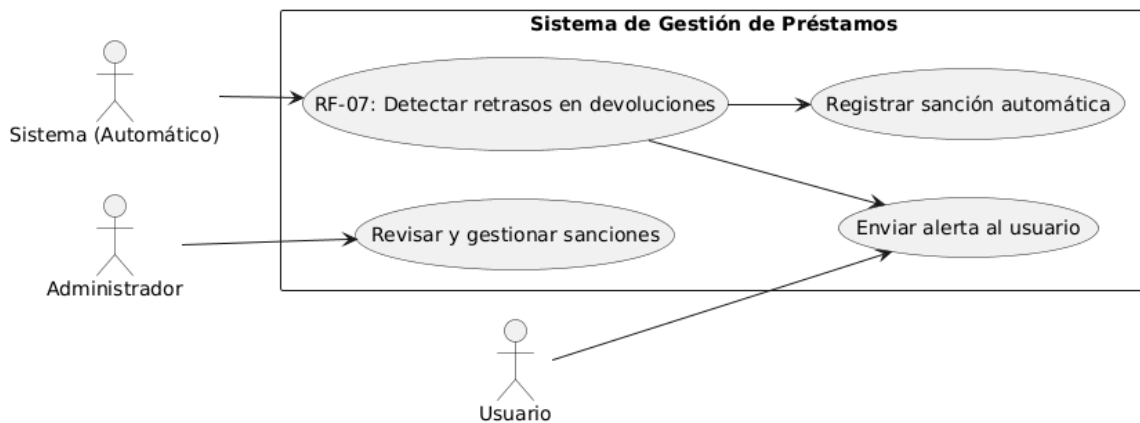
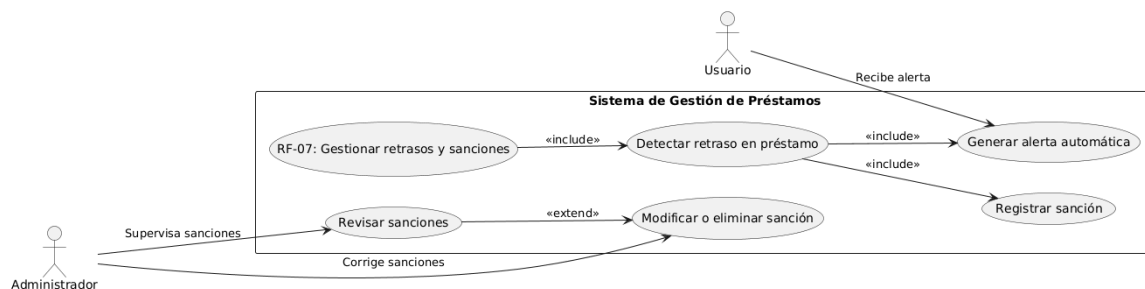


Figura 6.

Diagrama de casos de uso del requisito funcional RF-03 generado automáticamente



1. ¿Qué diferencias observas entre los Casos de Uso derivados de entrevistas o descripciones textuales y los Casos de Uso generados automáticamente en PlantUML?

Los casos de uso definidos mediante una automatización son mucho más sencillos de realizar con herramientas de generación, y en este caso la diferencia es clara, en primer lugar, el

caso de uso generado manualmente es más general y enfocado en el requisito en sí de la asignación automática del recurso tecnológico, mientras que, generado con herramientas de forma automatizada, se enfoca en la profundidad y relación de dependencia con otros requisitos, lo cual permite analizar que en ciertos casos la profundidad y nivel de detalle puede ser óptimo con el fin de tener una definición adecuada del requisito, pero también de forma manual, se evita caer en situaciones como integrar otras funcionalidades en un diagrama específico.

Al momento de generarlo de forma manual se puede omitir algunos casos de usos que en nuestro contexto pueden ser obvios, pero para el sistema y entendimiento de este son realmente importantes a diferencia de generarlo con una herramienta de forma automática que es capaz de tomar puntos que posiblemente se está pasando por alto, en este contexto la diferencia principal se ve en el aumento significativo de casos de uso y actores.

En el caso del RF-07, los Casos de Uso derivados de entrevistas o descripciones textuales ofrecen una visión más detallada y humana del requisito, ya que permiten comprender las motivaciones, excepciones y contexto de los actores, como la importancia de evitar sanciones injustas o de recibir notificaciones claras. Estos casos suelen incluir condiciones previas, flujos alternativos y justificaciones que explican el “por qué” detrás de cada acción. En cambio, los Casos de Uso generados en PlantUML son más estructurados, técnicos y orientados al sistema, pues se centran en representar la lógica funcional y las relaciones entre actores y acciones de manera formal. En el caso del RF-07, el diagrama en PlantUML muestra los pasos automáticos, detectar retraso, enviar alerta, registrar sanción y revisar casos, con sus relaciones de inclusión y extensión bien definidas, pero sin detallar las razones o situaciones excepcionales que surgen desde la perspectiva del usuario.

2. ¿De qué manera el uso de PlantUML facilita (o limita) el trabajo del analista al modelar los requisitos funcionales?

El uso de PlantUML facilita el desarrollo de diagramas de forma automatizada con una especificación de requerimientos definida, es útil porque como se pudo apreciar, permite observar la dependencia en cada caso de uso del diagrama, con otros requisitos existentes, además, al poder generar de forma automatizada, la trazabilidad del requisito permite reconocer diferentes etapas del mismo en diferentes modelos de casos de uso, siendo en esencia una herramienta muy útil, pero que es necesario revisar antes de dar por hecho un modelo final, ya que los errores como por ejemplo el nombre del sistema, o algunos actores, pueden no ser los adecuados, por eso es necesario revisar y corregir de ser el caso.

El uso de PlantUML puede generar limitaciones al momento de poder organizar los elementos del caso de uso cuando se modela los requisitos de un sistema muy grande ya que, la herramienta si le va a permitir generar el diagrama pero se pueden cruzar las relaciones, <<extends>> he <<includes>> pasando encima de otros caso de uso, o sobre las mismas relaciones generando complicaciones al momento de interpretar la funcionalidad el sistema, haciendo lo

contrario de el motivo por el que se realizan los casos de uso siendo demostrar gráficamente las funcionalidades del sistema para una mejor comprensión.

Además, es importante destacar que el verdadero valor de PlantUML no radica únicamente en la generación automática de diagramas, sino en su capacidad para mantener la coherencia y trazabilidad entre los diferentes modelos del sistema a lo largo del ciclo de vida del proyecto. Al poder versionar y editar los diagramas mediante código, el analista puede documentar los cambios de forma precisa y reproducible, algo que resulta más difícil con herramientas visuales convencionales. Sin embargo, su uso requiere una buena comprensión de los requisitos y del lenguaje UML, ya que la herramienta no interpreta la intención del analista, sino que representa exactamente lo que se codifica. Por ello, PlantUML debe verse como un apoyo técnico complementario al análisis humano, que combina automatización con juicio analítico para asegurar que el modelo refleje fielmente la funcionalidad esperada del sistema.

3. CONCLUSIONES

PlantUML es una herramienta muy útil para la construcción de diagramas de casos de uso, permitiendo llevar un requisito descrito en un estándar, a un modelo gráfico de forma automatizada y rápida, lo cual da un boceto sobre su interacción con los usuarios y con otras funcionalidades dentro del sistema.

La generación de UML de diagramas de caso de uso son mas entendibles y permite una mejor comprensión al promotor el cual va a desear conocer una estructura general para ver si su idea esta bien planteada ante el levantamiento de requisitos, pero puede demostrar dificultad visual debido a que se genera automáticamente y no se tiene la posibilidad de acomodar por donde se cruzan o la separación de las relaciones dentro de los casos de uso.

Además, el uso de PlantUML contribuye significativamente a mantener la coherencia y trazabilidad entre los modelos generados y los requisitos definidos, lo que facilita la documentación técnica y la revisión de versiones del sistema. Su integración dentro del proceso de análisis permite visualizar dependencias, relaciones y posibles inconsistencias desde etapas tempranas, reduciendo errores en fases posteriores del desarrollo. Sin embargo, su automatización no reemplaza la interpretación analítica ni el criterio profesional del analista, siendo fundamental combinar la precisión técnica del modelado con la comprensión funcional del sistema para obtener resultados completos y confiables.

4. RECOMENDACIONES

Pese a que se ha demostrado la utilidad de PlantUML, no debe verse como una sustitución del trabajo manual del analista de sistemas, ya que se debe analizar el resultado generado y comprender su interacción, para evaluar si cumple o no cumple con lo que realmente se necesita desarrollar, o si realmente refleja las especificaciones del requisito, es por eso que se recomienda una revisión manual una vez generado el modelo.



Se recomienda tener en cuenta los casos de uso generados automáticamente para añadir casos de uso o autores los cuales pueden ser parte fundamental para el requisito funcional, para presentarlos en una prueba de validación con el promotor y comprobar su aceptación.

Asimismo, se recomienda utilizar PlantUML como herramienta complementaria dentro de un proceso iterativo de análisis, diseño y validación, asegurando que los diagramas generados sean revisados de forma colaborativa con el equipo y los interesados del proyecto. Para proyectos de mayor complejidad, es aconsejable dividir los diagramas en módulos o subsistemas para mejorar la claridad visual y facilitar la comprensión de las interacciones. Finalmente, se sugiere mantener un registro versionado de los modelos UML, alineado con las actualizaciones de los requisitos, de manera que se preserve la trazabilidad y la evolución del sistema a lo largo de su ciclo de vida.

5. REFERENCIAS