

 ESPE UNIVERSIDAD DE LAS FUERZAS ARMADAS INNOVACIÓN PARA LA EXCELENCIA	UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE	VERSIÓN: 1.0 FECHA ÚLTIMA REVISIÓN: 01/08/2025
--	--	---

DEPARTAMENTO:	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN	CARRERA:	SOFTWARE		
ASIGNATURA:	Análisis y Diseño de Software	PERIODO:	OCT 25 - MAR 26	NIVEL:	6to
DOCENTE:	Jenny Alexandra Ruiz Robalino	NRC:	27837	Trabajo N°:	3
Nombre Estudiantes	Carlos Ñato, Juan Granda, David Cepeda				

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

Objetivo del Taller: Desarrollar el pensamiento crítico de los estudiantes mediante la comparación entre los Casos de Uso derivados del proceso de análisis de requisitos funcionales de su ERS (SRS) y los Casos de Uso modelados gráficamente mediante PlantUML, evaluando la coherencia, trazabilidad y completitud de los modelos.

Contexto de Aprendizaje

Durante el proceso de análisis de requisitos, los analistas obtienen los requisitos funcionales (RF) a partir de entrevistas, observaciones y documentación. Cada requisito funcional puede representarse mediante un caso de uso, el cual describe cómo un actor interactúa con el sistema para alcanzar un objetivo específico. En esta práctica, los estudiantes contrastarán cómo los requisitos funcionales textuales se transforman en diagramas de casos de uso generados con PlantUML.



Actividades

Parte 1. Lectura y análisis de requisitos.

Nombre del Requerimiento	<p>RF11 MinGO - Acceder a contenido exclusivo</p> <pre>graph LR Padre((Padre)) --> ComprarMembresia(Comprar membresia) PadreConMembresia((Padre con membresia)) --> TraducirTextoBreve(Traducir texto breve) PadreConMembresia --> EnseñarConAprendizajeDinámico(Enseñar con aprendizaje dinámico) subgraph Nivel1 [Nivel 1: MinGO Acceder A Contenido Exclusivo] ComprarMembresia TraducirTextoBreve EnseñarConAprendizajeDinámico end</pre>
Actores	Padre, Padre con membresia
Descripción	<p>El sistema consulta la base de datos para verificar si el actor posee una membresía activa.</p> <ul style="list-style-type: none">• Si no tiene una membresía válida, el usuario es redirigido automáticamente a una sección informativa donde se detallan los beneficios de adquirirla y se le ofrece la opción de compra.• Si la membresía es válida, el sistema desbloquea el acceso a funcionalidades exclusivas, tales como: Traductor de palabras al lenguaje de señas y Sesión de aprendizaje dinámico a través de juegos educativos interactivos
Relaciones	<p>No existe ninguna relación include o extend según nuestro diagrama, tenemos caso de uso inmersos en este como: Comprar membresia. Traducción de texto breve Enseñar con aprendizaje dinámico el cual tiene otra explosión a requisitos más detallados.</p>

Interpretación de IAGen del caso de uso a código Plan UML (Gemini)

Prompt usado :

Actúa como un arquitecto de software senior especializado en modelado de sistemas con UML.

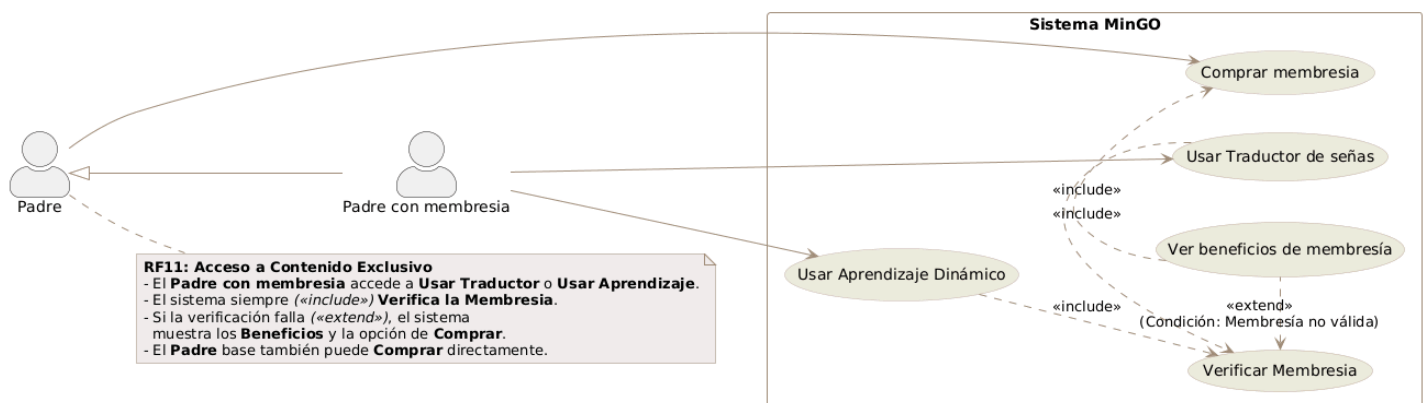
Objetivo: Tu tarea es analizar la siguiente especificación del requisito (RF11) y generar el código PlantUML para un **Diagrama de Casos de Uso** que muestre las relaciones entre los actores y los diferentes casos de uso.

Utiliza las mejores prácticas de UML para las relaciones (herencia para actores, include, extend o asociaciones donde sea apropiado).

— Especificación del Requisito —

Instrucciones de Salida: Genera código Plan UML para este caso de uso, además puedes mejorar si crees necesario el requisito.

Interpretación:



Comparación entre requisitos

Al analizar la propuesta generada, observamos que la inteligencia artificial integró un mayor nivel de detalle al sugerir nuevos casos de uso como parte funcional del sistema, específicamente “Verificar membresía” y “Ver beneficios de membresía”.

En cuanto al diseño del diagrama, consideramos que se podría mejorar el orden visual para que las conexiones entre los requisitos asociados se entiendan con mayor claridad.

Reconocemos que implementar estos dos requisitos funcionales adicionales ("extras") es una opción valiosa, ya que podrían presentarse situaciones durante la implementación donde sean requeridos.



Sin embargo, tras comparar ambas propuestas, hemos decidido no implementar estos nuevos requisitos funcionales de forma independiente en este momento, dado que implicaría una reestructuración más profunda del software. En su lugar, tomaremos esta sugerencia como una observación para integrar o unir esta lógica como parte de un requisito funcional ya existente.

ENLAZAR CLASE

Nombre del Requerimiento	<p>RF4 MinGo - Enlazar clase</p> <pre>graph LR Docente --> UC1[Visualizar progreso del padre] Docente --> UC2[Importar contenido] Usuario --> UC3[Enlazar clase] Usuario --> UC4[Registrar usuario] Usuario --> UC5[Descargar contenidos del sistema] Padre --> UC6[Guiar por funcionalidades del sistema] Padre --> UC7[Realizar prueba de conocimiento] UC1 -.-> «include» UC8[Ver historial de palabras aprendidas] UC9[Filtrar Contenido] -.-> «extend» UC10[Categorizar frases comunes] UC3 -.-> «include» UC11[Categorizar aprendizaje de señas por edad del niño] UC4 -.-> «include» UC12[Acceder a contenido exclusivo] UC6 -.-> «extend» UC13[Iniciar Sesión] UC7 -.-> «extend» UC13 UC3 -.-> UC4 UC6 -.-> UC7</pre>
Actores	Padre
Descripción	Una vez que el actor ha iniciado sesión, el sistema recibe como entrada el rol del actor. El sistema solicita el ingreso del código de clase previamente generado por un docente. El sistema valida la existencia y vigencia del código. Si es correcto, el actor es vinculado a la clase. Como salida, queda asociado a la clase, habilitando su acceso a contenidos.
Las acciones principales que el sistema debe ejecutar.	Acción
	El sistema verifica el rol del actor autenticado sea padre.
	El sistema solicita el código de la clase.
	El sistema valida el formato y existencia del código.

	EL sistema enlaza al actor a la clase si el código es válido y está disponible.
	El sistema muestra un mensaje de confirmación y habilita el acceso a la clase.
	El sistema registra la relación actor-clase en base de datos.
Relaciones	<p><<include>> con “Categorizar frases comunes” y “Categorizar aprendizaje de señas por edad del niño”, ya que ambos requisitos son necesarios para el proceso de enlace.</p> <p><<extend>> con “Guiar por funcionalidades del sistema”, que se activa después de completarse la vinculación.</p>

Interpretación de IAGen del caso de uso a código Plan UML(ChatGPT) versión gratuita

Prompt usado :

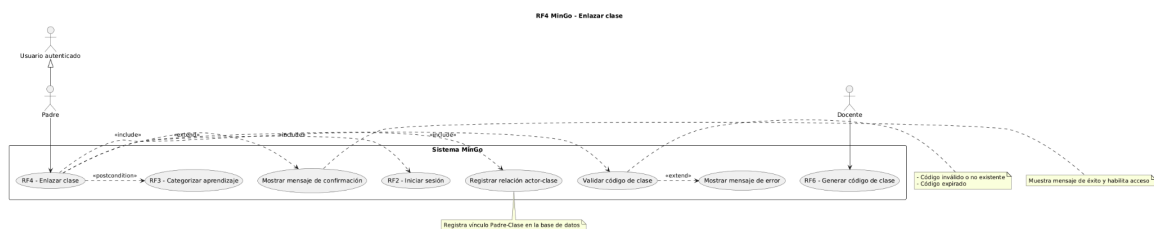
Actúa como un analista de sistemas senior especializado en modelado de sistemas con UML.

Objetivo: El objetivo es analizar la siguiente especificación del requisito y generar el código PlantUML para un Diagrama de Casos de Uso que muestre las relaciones entre los actores y los diferentes casos de uso. Utiliza las mejores prácticas de UML para las relaciones (herencia para actores, include, extend o asociaciones donde sea apropiado).

— Especificación del Requisito —

Instrucciones de Salida: Genera código Plan UML para este caso de uso, además puedes mejorar si crees necesario el requisito.

Interpretación




Comparación entre requisitos

Dado los resultados de la IAGen la propuesta generada, identificamos que la inteligencia artificial incorporó un mayor nivel de detalle al incluir nuevos casos de uso como parte del sistema en el requisito funcional.



Con el orden generado consideramos que es posible mejorar su organización visual para que las relaciones entre los requisitos sean más comprensibles y tanto los incluye como extend sea mas visibles

Comparando ambas versiones, hemos decidido no implementarlos como requisitos funcionales independientes por el momento, ya que esto implicaría una modificación estructural más amplia del sistema y un cambio en los requisitos previamente ya aceptados por el promotor. Pero tomaremos en cuenta todo dado de la IA para su posible implementación o modificación de los requisitos funcionales.

Nombre del Requerimiento	RF10 MinGO - Descargar contenido del sistema 
Requerimientos del Sistema Asociados	NF-01, NF-05
Actores	Padre
Descripción	El sistema obtiene y presenta los módulos de aprendizaje disponibles vinculados a la clase del actor. Solo se permite seleccionar módulos gratuitos activos para el usuario; los módulos de pago se visualizan con restricción e indicación de que requieren activación. Una vez que el actor selecciona un módulo gratuito y un nivel de aprendizaje (como principiante, intermedio o avanzado), el sistema le ofrece la opción de descargar el contenido educativo asociado para uso sin conexión. Si el usuario acepta y ha concedido los permisos de almacenamiento local, los recursos se descargan y almacenan en el almacenamiento interno del dispositivo. Estos datos incluyen contenido textual, audiovisual e imágenes, organizados por lección y módulo.
Relaciones	Según el Diagrama General de Casos de Uso, no se especifican relaciones de tipo <<include>> o <<extend>> para



el caso de uso "Descargar contenidos del sistema". Tampoco se proporciona un diagrama de descomposición de Nivel 1 o Nivel 2 para este requisito en el documento.

Interpretación de IAGen del caso de uso a código Plan UML(Claude)

Prompt usado:

Actúa como arquitecto de software senior especializado en UML.

CONTEXTO:

Sistema MinGO - plataforma educativa para lenguaje de señas

- Docentes: gestionan contenido
- Padres: acompañan aprendizaje
- Usuarios: estudiantes

[PEGAR DIAGRAMA]

ANALIZA el caso de uso "Descargar contenidos del sistema" y entrega:

1. PROBLEMAS IDENTIFICADOS

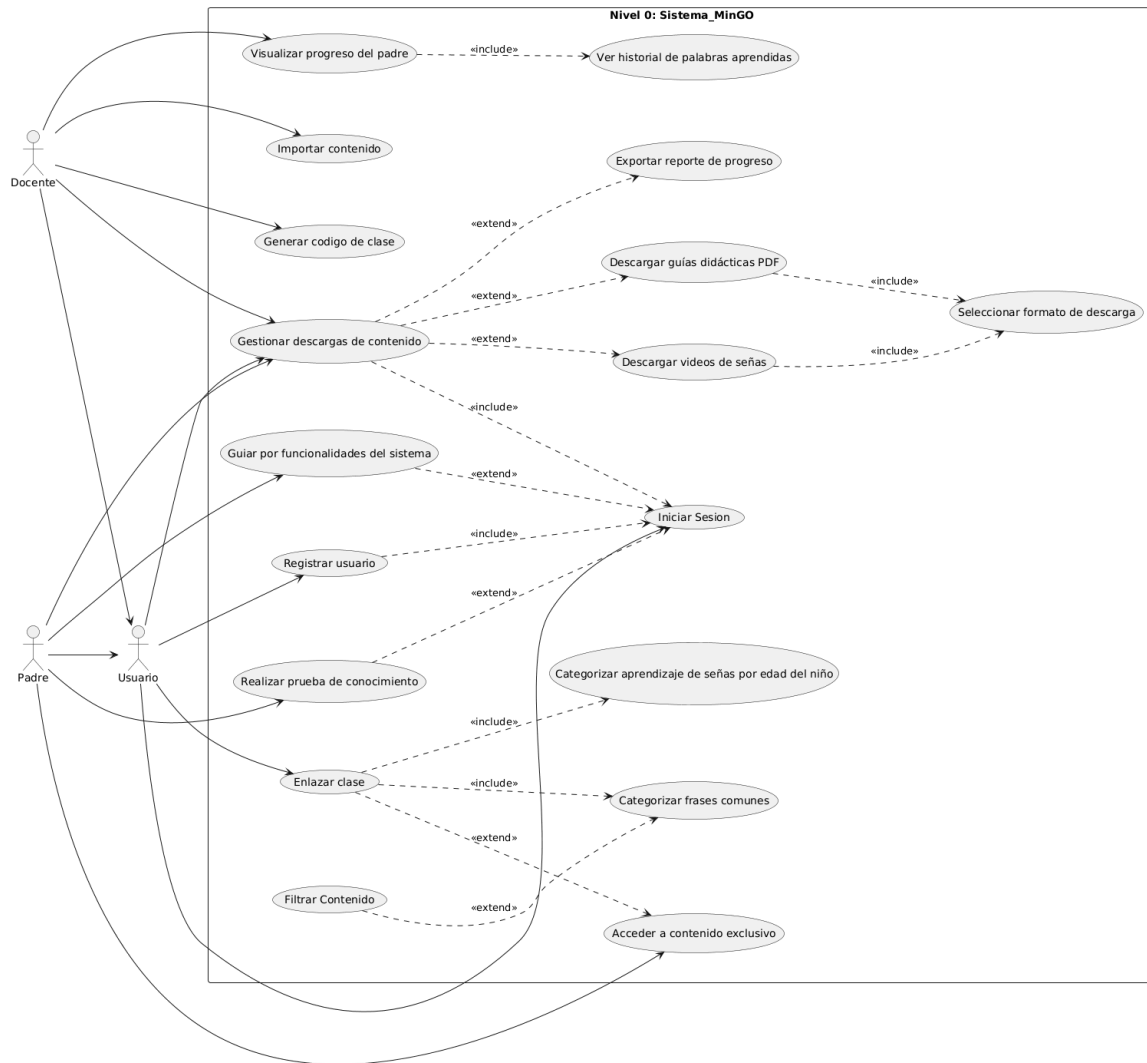
- Ambigüedades, relaciones incorrectas, actores faltantes

2. SOLUCIÓN MEJORADA

- Código PlantUML completo corregido
- Justificación técnica de cada cambio
- Casos de uso descompuestos si es necesario

Sé específico, práctico y prioriza la claridad arquitectónica.

Resultado:



Comparacion

El DCU original es ambiguo y no implementable: "Descargar contenidos" no especifica qué se descarga, limita al Padre como único actor. El DCU mejorado descompone la funcionalidad en casos concretos (videos, guías, reportes), incluye todos los actores relevantes, establece relaciones correctas con autenticación mediante <<include>>, y permite escalabilidad al agregar nuevos tipos de descarga como extensiones.

Conclusión

El DCU mejorado cumple criterios de calidad a comparación del original, siendo el único viable para implementación profesional porque transforma requisitos vagos en



especificaciones accionables para desarrollo, testing y mantenimiento del sistema MinGO.

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.)

Los Casos de Uso derivados de entrevistas o descripciones textuales suelen reflejar con mayor contexto y detalle semántico las necesidades del usuario llegando a permitir captar matices como emociones, excepciones o restricciones que no siempre están explícitas en el modelo gráfico. Sin embargo, al ser redactados manualmente, pueden carecer de uniformidad o contener ambigüedades y de lado, los casos de uso generados con IAGen y la otras herramientas (PlantUML) se caracterizan por su precisión estructural y claridad visual, ya que representan gráficamente los actores, las relaciones y dependencias de forma sistemática además que podría implementar mejoras en la construcción del software. No obstante, tienden a omitir información narrativa o contextual si el texto fuente no está completo, afectando la completitud 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.)

El PlantUML facilita significativamente el trabajo del analista porque llega a automatizar la generación de diagramas coherentes a partir del texto de los requisitos, llegando a reducir errores humanos y ahorrando tiempo en el análisis, promoviendo la trazabilidad, ya que pueden vincularse directamente con los identificadores de los RF, partiendo de otra perspectiva de la comunicación, PlantUML actúa como un puente visual entre el analista y el desarrollador asegurando que ambos comprendan las mismas relaciones funcionales. Sin embargo, su principal limitación es que no reemplaza el juicio analítico humano ya que el analista debe revisar si la IA interpretó correctamente las intenciones del usuario, verificando coherencia y completitud por lo que su integración permite transformar texto en modelos visuales automáticamente, lo que potencia la validación y comprensión de requisitos funcionales.