

## TALLER PENSAMIENTO CRÍTICO

### 1. DATOS INFORMATIVOS

Carrera: Ingeniería de Software

Asignatura: Análisis y Diseño de Software

Tema de la tarea: Diagramas de Caso de Uso en PlantUML

Docente: Mgt. Jenny Alexandra Ruiz Robalino

Integrantes:

- Amaguaña Casa Kevin Fernando
- Bonilla Hidalgo Jairo Smith
- Guamán Pulupa Alexander Daniel
- Tipán Ávila Reishel Dayelin

Fecha: 28/10/2025

Paralelo: 27835

### 2. OBJETIVO DE APRENDIZAJE

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

### 3. DESARROLLO

#### CU3.1 Visualizar producto

##### Parte 1. Lectura y análisis

Identifica:

1. **El actor principal:** Cliente (visitante)
2. **Las acciones principales que el sistema debe ejecutar:**
  - Cargar y mostrar el listado público de productos
  - Exponer controles de filtrado (categoría, etc.) y actualizar resultados
  - Acceder a la información específica de un artículo seleccionado
  - Ver imágenes y descripciones, Nostrar los elementos multimedia y la información precio en el detalle
3. **Las relaciones de inclusión (<<include>>) o extensión (<<extend>>) que aparecen en el diagrama UML**
  - Visualizar catálogo <<extend>> Filtrar productos: Acción opcional, se usa si el cliente desea aplicar filtros o búsquedas
  - Visualizar catálogo <<extend>> Ver detalles de producto: El cliente puede o no entrar al detalle de un producto
  - Ver detalles de producto <<include>> Ver imágenes y descripciones: Siempre que se consulta el detalle, se incluyen imágenes y descripciones del producto

## Parte 2. Pensamiento crítico

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?

Los casos de uso escritos a partir de entrevistas o descripciones textuales son más precisos y completos, ya que explican cómo el cliente interactúa con el sistema, los posibles errores y las condiciones que deben cumplirse, en cambio, el modelo en PlantUML ofrece una claridad visual superior al resumir toda esa información en un diagrama fácil de entender. Sin embargo, esa misma simplicidad hace que se pierdan detalles importantes del comportamiento del sistema y de los escenarios alternos que sí están descritos en el texto.

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

PlantUML facilita el trabajo del analista gracias a su automatización para generar diagramas de forma rápida, a su trazabilidad porque se puede mantener actualizado junto al código o documentación, y a la mejora de la comunicación

con los desarrolladores al representar de forma visual el flujo del proceso. Su limitación está en que no captura toda la lógica ni las validaciones internas, por lo que el analista debe complementar el diagrama con explicaciones textuales más detalladas.

### CU1.3 Consultar producto

#### Parte 1. Lectura y análisis

Identifica:

1. **El actor principal:** Administrador (Admin)
2. **Las acciones principales que el sistema debe ejecutar:**
  - Permitir la búsqueda de productos mediante filtros (nombre, categoría, material, color y rango de precio)
  - Validar que existan coincidencias en el catálogo
  - Mostrar los resultados en una tabla con la información básica del producto (nombre, categoría, precio, material, color)
  - Permitir al administrador seleccionar un producto para ver sus detalles y, si es necesario, modificarlo o eliminarlo
4. **Las relaciones de inclusión (<<include>>) o extensión (<<extend>>) que aparecen en el diagrama UML**
  - Consultar producto <<include>> Buscar/filtrar productos: el sistema siempre realiza la búsqueda como parte del proceso
  - Buscar productos <<include>> Validar existencia de resultados: la verificación de coincidencias forma parte obligatoria del flujo
  - Consultar producto <<include>> Mostrar resultados en tabla: el sistema presenta la información obtenida
  - Mostrar resultados en tabla <<extend>> Mostrar “Sin resultados”: si no se hallan coincidencias, se activa este flujo alternativo
  - Mostrar resultados en tabla <<extend>> Mostrar detalles del producto: el administrador puede decidir visualizar la ficha completa del artículo

#### Parte 2. Pensamiento crítico

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?

Los casos de uso creados mediante entrevistas logran mayor precisión y completitud, ya que incluyen las reglas de negocio, los filtros disponibles, los mensajes de error y los flujos alternos que pueden surgir. Por su parte, el modelo en PlantUML destaca por su claridad visual, mostrando las relaciones entre el administrador y las funciones del sistema de manera ordenada; no obstante, al simplificar tanto el contenido, el diagrama puede dejar fuera aspectos importantes que sí aparecen en la versión textual.

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

El uso de PlantUML ayuda al analista a mejorar la automatización del modelado, porque los diagramas se crean mediante código y se pueden actualizar fácilmente. Además, aporta trazabilidad, ya que el flujo visual se relaciona directamente con los requisitos del sistema, y mejora la comunicación entre analista y desarrollador al brindar una representación comprensible del proceso. La limitación está en que no muestra detalles específicos ni decisiones condicionales, por lo que debe ir acompañado de la documentación escrita para ser totalmente útil.

## CU2.1 Generar Proforma

### Parte 1. Lectura y análisis

Identifica:

1. **El actor principal:** Administrador (Admin)
2. **Las acciones principales que el sistema debe ejecutar**
  - Seleccionar productos del catálogo para componer la proforma
  - Validar (stock / precios, según tu modelado)
  - Calcular totales (subtotal, impuestos, total a pagar)
  - Guardar la proforma en el sistema
3. **Las relaciones de inclusión (<<include>>) o extensión (<<extend>>) que aparecen en el diagrama UML**
  - Generar proforma <<include>> Ingresar datos del cliente e ID de proforma
  - Generar proforma <<include>> Seleccionar productos del catálogo
  - Generar proforma <<include>> Registrar ofertas/promociones (si aplica)
  - Generar proforma <<include>> Calcular subtotal, impuestos y total
  - Generar proforma <<include>> Mostrar resumen de proforma
  - Generar proforma <<include>> Guardar proforma en el sistema

## Parte 2. Pensamiento crítico

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?

Los casos de uso redactados desde entrevistas o descripciones son más precisos, ya que explican paso a paso la captura de datos, las validaciones y los resultados esperados, logrando así mayor completitud. El diagrama en PlantUML, en cambio, ofrece claridad visual, pues condensa el proceso en una secuencia lógica y fácil de interpretar; sin embargo, al hacerlo más gráfico, se pierden algunos matices importantes como los mensajes de error o las condiciones previas que forman parte del texto.

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

PlantUML facilita el modelado al permitir automatizar la creación de diagramas coherentes con los requisitos, contribuye a la trazabilidad porque los casos de uso se pueden mantener sincronizados con el documento funcional, y mejora la comunicación con el equipo técnico al mostrar visualmente los pasos principales. Aun así, limita la representación de detalles internos del proceso, por lo que siempre es necesario acompañarlo de la descripción completa para mantener la fidelidad funcional.

## CU2.3 Consultar Proforma

### Parte 1. Lectura y análisis

Identifica:

1. **El actor principal:** Administrador (Admin)
2. **Las acciones principales que el sistema debe ejecutar**
  - Permitir búsqueda/filtrado de proformas por código, cliente o fecha
  - Validar la existencia de resultados para los criterios ingresados
  - Listar coincidencias y permitir seleccionar una proforma
  - Mostrar los detalles de la proforma (productos, totales, estado)
  - Visualizar el historial/estado asociado a la proforma
3. **Las relaciones de inclusión (<<include>>) o extensión (<<extend>>) que aparecen en el diagrama UML**
  - Consultar proforma <<include>> Buscar proforma por código, cliente o fecha

- Buscar proforma por código, cliente o fecha<<include>> Validar existencia de proforma
- CU 2.3 Consultar proforma <<include>> Mostrar detalles de proforma
- Mostrar detalles de proforma <<include>> Ver historial / estado de proforma

## Parte 2. Pensamiento crítico

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?

Los casos de uso basados en entrevistas son más ricos en precisión y completitud, ya que describen de forma detallada los filtros disponibles, las validaciones y los posibles resultados, mientras que el diagrama generado en PlantUML prioriza la claridad visual para mostrar la estructura general del flujo. El texto profundiza en la lógica del negocio; el diagrama, en cambio, facilita la comprensión rápida del proceso, aunque deja fuera algunos detalles técnicos.

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

PlantUML como herramienta ayuda al analista a automatizar la documentación visual, garantizando una trazabilidad clara entre los requisitos y los diagramas, y fomentando una mejor comunicación con los desarrolladores gracias a la representación visual del flujo de acciones. Sin embargo, limita la descripción de condiciones específicas o escenarios excepcionales, lo que obliga a complementar el modelo con explicaciones textuales para lograr una visión completa del comportamiento del sistema.

## 4. CONCLUSIONES

- Los casos de uso textuales permiten comprender mejor las necesidades del usuario y el contexto funcional, mientras que los generados con PlantUML aportan mayor precisión y claridad visual en la documentación técnica.
- PlantUML facilita la automatización y trazabilidad de los diagramas, mejorando la organización y coherencia de los modelos UML.
- Sin embargo, su carácter técnico puede limitar la comunicación con usuarios no especializados, al perder parte del contexto narrativo que ofrecen las descripciones textuales.

## 5. RECOMENDACIONES

- Utilizar PlantUML para documentar los diagramas de manera formal y mantener la trazabilidad de los cambios en los modelos.
- Complementar los diagramas generados con descripciones textuales que expliquen el contexto, los actores y las reglas de negocio.
- Promover la comunicación continua entre analistas y desarrolladores para asegurar que los modelos UML reflejen fielmente los requisitos funcionales.
- Capacitar a los analistas en el uso de herramientas como PlantUML para aprovechar sus ventajas sin descuidar la claridad en la interpretación del modelo.

## 6. REFERENCIAS

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley.

Object Management Group (OMG). (2017). *Unified Modeling Language (UML) Specification, Version 2.5.1*.

PlantUML. (s.f.). *PlantUML Documentation*. Recuperado de <https://plantuml.com>

Sommerville, I. (2016). *Ingeniería del Software* (10ª ed.). Pearson Educación.