



- 1. Si la clase A está vinculada con la clase B mediante una relación de asociación.  
¿La estructura de la clase B se ve impactada? Explique con detalle.**

No necesariamente. En una asociación de una sola dirección, solo la clase A necesita conocer a la clase B. La clase B, en cambio, no tiene por qué saber que existe la clase A, por lo que su estructura interna no cambia. Solo se ve afectada la clase que apunta a la otra.

- 2. Si la clase P está vinculada con la clase Q mediante una relación de agregación,  
¿la estructura de la clase B se ve impactada? Explique con detalle.**

Sí, porque debe contener una colección o referencia a objetos de la clase Q. En cambio, la clase Q no se ve afectada, ya que puede existir de forma independiente.

- 3. Si la clase A está vinculada con la clase B mediante una relación de asociación  
y la clase B está vinculada con la clase A mediante una relación de asociación,  
¿la estructura de ambas clases se ve impactada? Explique con detalle.**

Sí, en este caso ambas clases se ven impactadas. Cada una tendrá un atributo que hace referencia a la otra. Esto crea una dependencia mutua, lo que significa que ambos deben conocerse en su diseño, afectando su estructura y aumentando el vínculo entre ellas}

- 4. ¿Un objeto de tipo Z podría enviar mensajes a otro objeto de tipo W, aun cuando no existe un vínculo (de asociación o agregación) entre la clase Z y la clase W? Explique con detalle.**

No, no podría. Para que un objeto Z envíe un mensaje a un objeto W, Z debe tener una referencia a W (ya sea como atributo, parámetro de un método o variable local). Sin un vínculo en el diseño (como asociación o agregación), no hay forma de que Z conozca a W ni acceda a él. El envío de mensajes requiere esa conexión previa.

5. En un diagrama de clase con detalles de implementación, suponga que existe una relación de asociación entre la clase P y la clase Q. Suponga también que esa relación tiene los cinco elementos respectivos en el diagrama. ¿Eso es suficiente para establecer de forma completa el vínculo de asociación entre P y Q? Explique con detalle.

Sí, es suficiente. Si el diagrama incluye los cinco elementos, representación UML, navegabilidad, rol, modificador de acceso y cardinalidad, entonces se tiene toda la información necesaria para implementar la relación en código. Por ejemplo, se sabe qué clase tiene el atributo, qué visibilidad tiene etc. Cómo se llama (rol), y cuántos objetos puede referenciar (cardinalidad). Con eso, un programador puede traducir el diseño a Java sin ambigüedades.

## **Reflexion**

Esta actividad me ayudó a entender que los objetos no viven aislados: su poder está en cómo se relacionan e interactúan. Aprendí a distinguir claramente entre una asociación (dos objetos independientes que se conocen) y una agregación (un TODO que contiene PARTES, pero que no las controla por completo). Además, valoré mucho el uso del Javadoc, no solo como requisito académico, sino como una herramienta real para documentar código y facilitar el trabajo en equipo. Por último, el diagrama UML dejó de ser solo dibujos:

ahora veo en él instrucciones precisas para programar. Fue una experiencia que conectó teoría, diseño y código de forma muy concreta.