Trabajo Práctico N° 1 – Modelos y Sistemas

Alumna: Sofia Atkins

1. ¿Qué es el Modelado de Sistemas?

El modelado de sistemas es una técnica usada para representar, mediante gráficos y diagramas, los distintos elementos que forman parte de un sistema y cómo se relacionan entre sí. Su propósito principal en ingeniería de software es facilitar la comprensión, el análisis y el diseño del sistema antes de construirlo, permitiendo detectar errores o mejoras desde una etapa temprana.

2. ¿Cuándo se usan modelos en el desarrollo de software?

Los modelos se aplican a lo largo de todo el ciclo de vida del desarrollo de software. En la fase de análisis, sirven para entender los requerimientos. Durante el diseño, permiten estructurar cómo va a funcionar el sistema internamente. En la implementación, guían a los desarrolladores, y más adelante son útiles en las pruebas y el mantenimiento, ya que ayudan a entender qué se hizo y por qué.

3. Cuatro perspectivas fundamentales del modelado

- Estructural: describe qué cosas existen en el sistema (como clases, objetos y sus relaciones).
- Comportamiento: muestra cómo actúan los elementos del sistema frente a diferentes eventos o situaciones.
- Funcional: se enfoca en qué hace el sistema, cuáles son sus funciones principales.
- Arquitectónica: muestra cómo están organizados los componentes internamente y cómo se comunican.

4. ¿Qué es UML y qué tipos de diagramas tiene?

UML significa *Unified Modeling Language* (Lenguaje Unificado de Modelado). Es un lenguaje gráfico estándar utilizado para visualizar, especificar, construir y documentar sistemas de software. Sus diagramas se dividen en:

- Estructurales: clase, objetos, componentes, etc.
- De comportamiento: casos de uso, actividades, máquinas de estados.
- De interacción: secuencia, comunicación, etc.

5. **Cinco tipos de diagramas UML y su propósito**

- Caso de Uso: muestra qué funciones ofrece el sistema a los usuarios o actores externos.
- Clase: representa la estructura del sistema mediante clases, atributos y métodos.
- Actividad: describe flujos de trabajo, decisiones y ramificaciones del sistema.
- Secuencia: ilustra el intercambio de mensajes entre objetos con respecto al tiempo.
- Estado: refleja los distintos estados que puede tener un objeto y cómo cambia.

6. **Función de los Modelos de Contexto**

Permiten representar el sistema como una "caja negra", destacando sus interacciones con actores o sistemas externos, pero sin entrar en detalles internos. Sirve para delimitar qué pertenece al sistema y qué no, ayudando a definir sus requisitos principales y el alcance del proyecto.

7. **Relaciones que no se muestran en los Modelos de Contexto**

Las relaciones internas del sistema o entre componentes internos no se representan, ya que el foco está en lo externo. Esto es importante porque evita confundir la función general del sistema con sus detalles técnicos, priorizando su interacción con el entorno.

8. **Símbolos de inicio y fin en el Diagrama de Actividad**

- Inicio: círculo negro lleno.
- Fin: círculo con borde grueso y un punto negro adentro.
  Estos elementos permiten identificar claramente dónde comienza y termina un proceso.

9. **Significado de las barras en un Diagrama de Actividad**

- Barras de sincronización: se usan para indicar cuándo varias actividades deben empezar o terminar al mismo tiempo.
- Si el flujo llega a la barra, significa que todas las actividades anteriores deben completarse antes de continuar.
- Si el flujo parte de la barra, indica que se ejecutan múltiples actividades en paralelo.

10. Importancia del modelado de interacción

Es esencial para entender cómo los distintos elementos del sistema (usuarios, objetos, componentes) se comunican entre sí. Permite identificar posibles errores en la lógica de comunicación y asegurar que los procesos funcionen correctamente.

11. Dos enfoques del modelado de interacción

- Diagrama de Secuencia: se centra en el orden de los mensajes a lo largo del tiempo.
- Diagrama de Comunicación: se enfoca en las conexiones entre los objetos y cómo se intercambian mensajes, sin detallar el tiempo.

12. ¿Qué es un Caso de Uso y cómo se representa?

Es una representación gráfica de una funcionalidad que el sistema proporciona a un actor (usuario o sistema externo). Se dibuja como una elipse con el nombre de la función, conectada a actores (dibujados como muñequitos) mediante líneas.

13. Uso de flechas en diagramas de Casos de Uso

En algunos casos no es necesario usar flechas porque la relación entre actor y caso de uso es obvia. Sin embargo, cuando se quiere mostrar una dependencia (como que un caso extiende o incluye otro), se utilizan flechas con etiquetas como «include» o «extend».

14. ¿Qué es un Diagrama de Secuencia y para qué sirve?

Es un diagrama que muestra cómo interactúan los objetos a lo largo del tiempo. Representa los mensajes que se envían para cumplir con una funcionalidad, permitiendo identificar el orden exacto en que se dan las acciones.

15. Rectángulo de activación

También llamado barra de activación, aparece en la línea de vida de un objeto e indica el período durante el cual ese objeto está realizando una acción. Es útil para entender la duración de procesos y qué objetos están activos.

16. Modelos Estructurales y tipos principales

Son representaciones que muestran la estructura del sistema, es decir, sus partes y cómo se conectan. Los dos tipos más comunes son:

- Diagrama de Clase: define clases, atributos y relaciones.
- Diagrama de Componentes: muestra los módulos de software que componen el sistema.

17. Cuándo crear modelos estructurales

Se recomienda crearlos en las fases iniciales del análisis y diseño, ya que ayudan a entender bien cómo está compuesto el sistema antes de empezar a programar.

18. Modelos de Comportamiento y estímulos

Representan cómo se comporta el sistema frente a distintos eventos. Dos tipos de estímulos que provocan respuestas son:

- Eventos del entorno (clics, mensajes, acciones del usuario).
- Cambios internos (valores, condiciones que se cumplen).

19. Diferencia entre sistemas impulsados por datos y por eventos

- Impulsados por datos: responden a cambios en la información, por ejemplo, actualizar una factura si se cambia un precio.
- Impulsados por eventos: reaccionan a acciones específicas, como un clic del usuario.
  Ambos se usan según el tipo de aplicación que se está desarrollando.

20. Desafíos del modelado en sistemas grandes

En sistemas muy complejos, con muchos estados y transiciones, el modelo puede volverse difícil de entender y mantener. Para evitar esto, se pueden usar técnicas como:

- Dividir el sistema en módulos o subsistemas.
- Usar jerarquías para agrupar estados relacionados.
- Aplicar modelos combinados para simplificar.