

TP N°2

ALUMNO: Dylan Alvarez

Parte 1: Modelos de Casos de Uso

1. Definición Profunda de Caso de Uso

Un caso de uso describe, desde la perspectiva externa, la secuencia de interacciones entre un usuario (actor) y el sistema para alcanzar un objetivo específico. No se centra en el diseño interno o en cómo está implementado el sistema, sino en lo que es visible y útil para el usuario. Esta visión externa es crucial en las primeras etapas de modelado porque permite entender claramente lo que el usuario necesita, sin distraerse con detalles técnicos que pueden cambiar más adelante.

2. Propósito Esencial en el Contexto de Interacción

El principal propósito del modelado de casos de uso es capturar y documentar los requisitos funcionales del sistema desde el punto de vista del usuario. Permite identificar cómo interactúan los diferentes actores con el sistema y qué funciones esperan obtener, asegurando que el desarrollo esté alineado con las necesidades reales y no con suposiciones del equipo técnico.

3. Representación Gráfica y Simbolismo

En UML, los casos de uso se representan como elipses que contienen el nombre de la funcionalidad. Los actores se dibujan como figuras de persona (stickman). La relación entre actor y caso de uso se indica con una línea recta sin flechas. En las etapas iniciales, el diagrama debe ser simple y de alto nivel, mostrando únicamente las funcionalidades principales y su relación con los actores para no generar complejidad innecesaria.

4. Tipos y Roles de Actores

- **Actor humano:** Personas que interactúan directamente con el sistema (ej. recepcionista médica).
- **Sistema externo:** Otro software que intercambia información (ej.

un sistema de gestión hospitalaria conectado).

- **Dispositivo físico:** Equipos que envían o reciben datos (ej. una impresora de recetas médicas).
Cada uno representa una entidad externa con la que el sistema debe comunicarse.

5. Notación de Flechas y su Justificación

En casos de uso, las líneas entre actor y caso de uso no llevan flechas porque la relación es bidireccional: ambos participan en la interacción. Si se usaran flechas, se sugeriría erróneamente un flujo de control o dependencia unidireccional, lo que no es el objetivo del diagrama.

6. Ampliación y Refinamiento de Detalles

Tres métodos para profundizar en un caso de uso son:

- **Descripción textual estructurada:** detallando pasos, condiciones y excepciones.
- **Diagramas de secuencia:** mostrando el orden y lógica de los mensajes.
- **Diagramas de actividad:** representando visualmente el flujo de procesos.
Este nivel adicional permite a los desarrolladores entender claramente qué debe hacer el sistema y cómo manejar situaciones especiales.

7. Estructura Detallada de la Descripción Textual (MHC-PMS)

Ejemplos de secciones:

- **Identificador:** un código único para el caso de uso.
- **Descripción breve:** resumen del objetivo.
- **Actores involucrados:** quién participa.

- **Flujo básico:** secuencia normal de pasos.
- **Flujos alternativos:** qué pasa si ocurre una variación.
Cada sección garantiza que el caso de uso esté completamente documentado y sea fácil de rastrear.

8. Análisis de Caso de Uso Específico

En "Ver Información del Paciente", el actor principal es la recepcionista médica. El flujo básico sería: la recepcionista solicita los datos → el sistema valida permisos → el sistema busca la información → se muestra en pantalla. El objetivo es consultar la ficha completa del paciente.

9. Vínculo Crucial: Casos de Uso y Requisitos del Usuario Los casos de uso actúan como puente entre la comunicación con el usuario y la especificación técnica. Ayudan a validar que el sistema cubrirá los requisitos funcionales esperados, evitando malentendidos y asegurando que cada funcionalidad tiene un propósito claro.

10. Ventajas Estratégicas del Modelado de Casos de Uso

- Clarifica los requisitos funcionales.
- Facilita la comunicación entre equipo técnico y usuarios.
- Sirve como base para el diseño y las pruebas del sistema.

Parte 2: Modelos de Interacción (Diagramas de Secuencia)

1. Objetivo y Enfoque

Los diagramas de secuencia muestran cómo interactúan los objetos o componentes de un sistema en un orden temporal específico. A diferencia de los casos de uso, que son generales, estos se enfocan en el detalle de la comunicación paso a paso.

2. Elementos Estructurales y Líneas de Vida

En la parte superior se encuentran:

- **Actores u objetos:** entidades que participan.
- **Nombre del rol o clase:** especifica quién es cada uno. Las líneas punteadas (líneas de vida) representan la existencia del objeto en el tiempo y cuándo está activo.

3. Interacciones, Mensajes y Activaciones

Los mensajes se representan con flechas horizontales. El rectángulo sobre la línea de vida indica un **periodo de activación**, es decir, cuándo el objeto está ejecutando una operación. La longitud indica cuánto dura esa ejecución.

4. Principio de Lectura y Flujo Temporal

Se leen de arriba hacia abajo, siguiendo la secuencia cronológica. Esta convención evita confusión y asegura que todos interpreten el orden de los eventos de la misma manera.

5. Información Adicional en Mensajes

Puede incluir parámetros, condiciones o resultados. Ejemplo: `buscarPaciente(id)` aclara qué información se envía y por qué.

6. Modelado de Alternativas y Condicionales

Se usa el **bloque alt** (combined fragment) en UML. Las condiciones se escriben entre corchetes, como `[si paciente existe]` y `[si paciente no existe]`, para indicar qué flujo se sigue.

7. Análisis de Escenario Específico – Inicio Detallado

En "Ver Información del Paciente", el actor **Recepcionista** envía el mensaje `solicitarDatosPaciente(id)` al objeto **Sistema de Gestión**. El parámetro relevante es el ID del paciente.

8. Análisis – Proceso de Autorización

El sistema envía un mensaje al módulo de seguridad para validar permisos. Si la autorización es exitosa, continúa con la búsqueda; si falla, devuelve un mensaje de error y termina la interacción.

9. Análisis – Transferencia de Datos (Alternativas)

Dos posibles caminos:

- Transferencia exitosa al PRS.
- Error en la transferencia (datos incompletos o fallo de conexión).
Las condiciones dependen de la validación de datos y disponibilidad del sistema.

10. Impacto Estratégico

Permiten validar la lógica de interacción antes de programar, encontrar errores de diseño temprano y asegurar que las funcionalidades cumplan con lo esperado por el usuario.