Teoría de la UT6.-Diagramas de comportamientos

1. ¿Para qué sirven los **diagramas de casos de uso**?

Resuelven la falta de comunicación entre el equipo de desarrollo y el equipo que necesita de una solución software.

1. ¿Qué son los **actores** en los diagramas de casos de uso?

Representan un tipo de usuario del sistema. Se entiende como usuario cualquier cosa externa que interactúa con el sistema.

1. Define los **tipos de actores** y su representación:
   1. Primarios: interaccionan con el sistema para explotar su funcionalidad. Trabajan directa y frecuentemente con el software.
   2. Secundarios: soporte del sistema para que los primarios puedan trabajar. Son precisos para alcanzar algún objetivo.
   3. Iniciadores: no interactúan con el sistema pero desencadenan el trabajo de otro actor.
   4. Representación:
2. ¿Qué especifica un **caso de uso**?

Especifican tareas que deben poder llevarse a cabo con el apoyo del sistema que se está desarrollando.

1. Define los siguientes **datos** a tener en cuenta en los **casos de uso** y su representación:
   1. Nombre: nombre del caso de uso
   2. Actores: aquellos que interactúan con el sistema a través del caso de uso.
   3. Propósito: breve descripción de los que se espera que haga.
   4. Precondiciones: aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
   5. Flujo normal: flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente, desde el punto de vista del actor que participa y del sistema.
   6. Flujo alternativo: flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuentes. No se deben incluir aquí errores como escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
   7. Postcondiciones: las que se cumplen una vex que se ha realizado el caso de uso.
   8. Representación gráfica:
2. ¿Qué representan las **relaciones** en los diagramas de casos de uso?

Representan que actores realizan las tareas descritas en los casos de uso, en concreto que actores inician en caso de uso.

1. Define las siguientes **tipos de relaciones** entre elementos y sus representaciones:
   1. Interacción o asociación: Una asociación se representa mediante un linea continua que une un actor con un caso de uso.
   2. Generalización: se utiliza para representar relaciones de herencia entre casos de uso o actores
   3. Extensión: : se utiliza para representar relaciones entre un caso de uso que requiere la ejecución de otro en determinadas circunstancias.
   4. Inclusión: se utiliza cuando queremos dividir una tarea de mayor envergadura en otras más sencillas, que son utilizadas por la primera. Representa una relación de uso, y son muy útiles cuando es necesario reutilizar tareas
2. ¿Qué es un **diagrama frontera**?

Al diagrama de casos de uso que incluye todos los casos de uso genéricos del sistema, que podrán ser desglosados después en nuevos diagramas de casos de uso que los describan si es necesario.

1. ¿Qué es un **escenario** y cómo debería documentarse?

Un escenario es una ejecución particular de un caso de uso que se describe como una secuencia de eventos. Un caso de uso es una generalización de un escenario.

1. ¿Para qué se utilizan los **diagramas de secuencia**?

Los diagramas de secuencia completan a los diagramas de casos de uso, ya que permiten al equipo de desarrollo hacerse una idea de qué objetos participan en el caso de uso y como interaccionan a lo largo del tiempo

1. ¿Cómo se representan los objetos, línea de vida y paso de mensajes en los diagramas de secuencia?
   1. Objetos: se puede usar el nombre concreto del objeto que participa en la interacción, normalmente aparece subrayado.
   2. Línea de vida: Una linea de vida puede estar encabezada por otro tipo de instancias como el sistema o un actor queaparecerán con su propio nombre. Usaremos el sistema para representar solicitudes al mismo, comopor ejemplo pulsar un botón para abrir una ventana o una llamada a una subrutina.
   3. Paso de mensajes: Un mensaje es la especificación de una comunicación entre objetos que transmite información y desencadena una acción en el objeto destinatario.
2. ¿Qué son los **diagramas de colaboración**?

Los diagramas de colaboración son un complemento para los de secuencia cuyo objetivo es mostrar las interacciones entre los objetos del diagrama mediante el paso de mensajes entre ellos. Las interacciones entre los objetos se describen en forma de grafo en el que los nodos son objetos y las aristas son enlaces entre objetos a través de los cuales se pueden enviar mensajes entre ellos. Los objetos se conectan mediante enlaces y se comunican a través de los mensajes.

1. ¿Cómo se **representan los objetos** en los diagramas de colaboración?
   1. NombreClase: directamente se puede utilizar el nombre de la clase a la que pertenece el objeto que participa en la interacción. Pero esta representación hace referencia a la clase, el resto son objetos.
   2. NombreObjeto: se puede usar el nombre concreto del objeto que participa en la interacción, normalmente aparece subrayado
   3. :nombreClase: cuando se coloca el símbolo : delante del nombre de la clase quiere decir que hace referencia a un objeto genérico de esa clase.
   4. nombreObjeto:nombreclase:  hace referencia al objeto concreto que se nombre añadiendo la clase a la que pertenece.
2. ¿Qué representa un **mensaje** en un diagrama de colaboración?

Los diagramas de colaboración son un complemento para los de secuencia cuyo objetivo es mostrar las interacciones entre los objetos del diagrama mediante el paso de mensajes entre ellos. Las interacciones entre los objetos se describen en forma de grafo en el que los nodos son objetos y las aristas son enlaces entre objetos a través de los cuales se pueden enviar mensajes entre ellos. Los objetos se conectan mediante enlaces y se comunican a través de los mensajes.

1. Escribe la **sintaxis de un mensaje** en un diagrama de colaboración y explica qué significa cada parte:
   1. Secuencia: representa el nivel de anidamiento del envío del mensaje dentro de la interacción. Los mensajes se numeran para indicar el orden en el que se envían, y si es necesario se puede indicar anidamiento incluyendo subrangos
   2. \*:indica que el mensaje es iterativo
   3. Condición de guarda: debe cumplirse para que el mensaje pueda ser enviado
   4. ValorDevuelto: : lista de valores devueltos por el mensaje. Estos valores se pueden utilizar como parámetros de otros mensajes. Los corchetes indican que es opcional.
   5. Mensaje: nombre del mensaje
   6. Argumentos: parámetros que se pasan al mensaje.
2. ¿Para qué se utilizan fundamentalmente los **diagramas de actividad**?

Se utilizan fundamentalmente para modelar el flujo de control entre actividades en el que se puede distinguir cuales ocurren secuencialmente a lo largo del tiempo y cuales se pueden llevar a cabo concurrentemente.

1. ¿Qué es un **diagrama de actividad**?

El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones, que se compone de una serie de actividades y representa cómo se pasa de unas a otras. Las actividades se enlazan por transiciones automáticas, es decir, cuando una actividad termina se desencadena el paso a la siguiente actividad.

1. Define los siguientes **elementos** de un diagrama de actividad:
   1. Estados de actividad: Elemento compuesto cuyo flujo de control se compone de otros estados de actividad y de acción
   2. Estados de acción: Estado que representa la ejecución de una acción atómica, que no se puede descomponer ni interrumpir, normalmente la invocación de una operación.
   3. Transiciones: Relación entre dos estados que indica que un objeto en el primer estado realizará ciertas acciones y pasará al segundo estado cuando ocurra un evento específico y satisfaga ciertas condiciones. Se representa mediante una línea dirigida del estado inicial al siguiente.
   4. Tipos de transiciones:
      1. Secuencial o sin disparadores: Al completar la acción del estado origen se ejecuta la acción de salida y, sin ningún retraso, el control sigue por la transición y pasa al siguiente estado.
      2. Bifurcación (Decision Node): Especifica caminos alternativos, elegidos según el valor de alguna expresión booleana. Las condiciones de salida no deben solaparse y deben cubrir todas las posibilidades (puede utilizarse la palabra clave else). Pueden utilizarse para lograr el efecto de las iteraciones.
      3. Fusión (Merge Node): : Redirigen varios flujos de entrada en un único flujo de salida. No tiene tiempo de espera ni sincronización
      4. División (Fork Node): : Permiten expresar la sincronización o ejecución paralela de actividades. Las actividades invocadas después de una división son concurrentes
      5. Unión (Join Node): : Por definición, en la unión los flujos entrantes se sincronizan, es decir, cada uno espera hasta que todos los flujos de entrada han alcanzado la unión.
   5. Objetos: Manifestación concreta de una abstracción o instancia de una clase. Cuando interviene un objeto no se utilizan los flujos de eventos habituales sino flujos de objetos (se representan con una flecha de igual manera) que permiten mostrar los objetos que participan dentro del flujo de control asociado a un diagrama de actividades. Junto a ello se puede indicar cómo cambian los valores de sus atributos, su estado o sus role
2. ¿Qué representan los **diagramas de estados**?

Expresan el comportamiento de un objeto como una progresión a través de una serie de estados, provocada por eventos y las acciones relacionadas que pueden ocurrir.

1. ¿Qué es un **estado** en un diagrama de estado?

Un estado es una situación en la vida de un objeto en la que satisface

cierta condición, realiza alguna actividad o espera algún evento.

1. Indica los **6 elementos de un estado** de un diagrama de estado:

Los diferentes estados en los que puede estar un pedido son:

-En creación: es cuando se están seleccionando los productos que formará el pedido.

- Pendiente: está en este estado desde que se confirma el pedido hasta que se selección para preparar su envío.

-En almacén: está en este estado cuando es elaborado el paquete y se ha asignado a una ruta, hasta que se envía a través de la ruta que le corresponde.

- Servido: Cuando el pedido es enviado. En este caso se envía una señal física desde el almacén cuando el transporte abandona el almacén.

-Cancelado: puede llegarse a esta situación por dos motivos, o bien se cancela mientras se está haciendo por problemas con la tarjeta de crédito, o bien porque, una vez pendiente de su gestión el usuario decide cancelarlo, la diferencia fundamental entre ambos es que en el segundo caso hay que devolver el importe pagado por el pedido al socio que lo ha comprado.

1. Define los siguientes dos **tipos de estados especiales** que existen en un diagrama de estados:
   1. Estado inicial: es un pseudoestado que indica el punto de partida por defecto para una transición cuyo destino es el límite de un estado compuesto. El estado inicial del estado de nivel más alto representa la creación de una nueva instancia de la clase.
   2. Estado final: : Estado especial dentro de un estado compuesto que, cuando está activo, indica que la ejecución del estado compuesto ha terminado y que una transición de finalización que sale del estado compuesto está activada.
2. ¿Qué es un **evento** en un diagrama de estados?

Un evento es un acontecimiento que ocupa un lugar en el tiempo y

espacio que funciona como un estímulo que dispara una transición en

una máquina de estados. Existen eventos externos y eventos internos

según el agente que los produzca.

1. Define los siguientes **tipos de eventos** de un diagrama de estados:
   1. Señales (excepciones): : la recepción de una señal, que es una entidad a la que se ha dado nombre explícitamente (clase estereotipada), prevista para la comunicación explícita ‐ y asíncrona‐ entre objetos. Es enviada por un objeto a otro objeto o conjunto de objetos. Las señales con nombre que puede recibir un objeto se modelan designándolas en un compartimento extra de la clase de ese objeto. Normalmente una señal es manejada por la máquina de estados del objeto receptor y puede disparar una transición en la máquina de estados.
   2. Llamadas: la recepción de una petición para invocar una operación. Normalmente un evento de llamada es modelado como una operación del objeto receptor, manejado por un método del receptor y se implementa como una acción o transición de la máquina de estados.
   3. Paso de tiempo: representa el paso del tiempo (ocurrencia de un tiempo absoluto respecto de un reloj real o virtual o el paso de una cantidad de tiempo dada desde que un objeto entra en un estado).
   4. Cambio de estado: : evento que representa un cambio en el estado o el cumplimiento de una condición. Palabra clave when, seguida de una expresión booleana, que puede ser de tiempo o de otra clase: when (hora = 11:30); when (altitud < 1000).
2. ¿Qué representa una **transición** en un diagrama de estado?

Una transición de un estado A a un estado B, se produce cuando se

origina el evento asociado y se satisface cierta condición especificada,

en cuyo caso se ejecuta la acción de salida de A, la acción de entrada a B

y la acción asociada a la transición.

1. Define las siguientes **elementos de una transición** en un diagrama de estado?
   1. Estados de origen y destino: a transición se disparará si, estando en el estado origen se produce el evento de disparo y se cumple la condición de guarda (si la hay), pasando a ser activo el estado final.
   2. Evento de disparo: cuando se produce un evento, afecta a todas las transiciones que lo contienen en su etiqueta. Todas las apariciones de un evento en la misma máquina de estados deben tener la misma signatura. Los tipos de evento los hemos visto en el punto anterior.
   3. Condición de guarda: expresión booleana. Si es falsa, la transición no se dispara, y si no hay otra transición etiquetada con el mismo evento que pueda dispararse, éste se pierde.
   4. Acción: computación atómica ejecutable. Puede incluir llamadas a operaciones del objeto que incluye la máquina de estados (o sobre otros visibles), creación o destrucción de objetos o el envío de una señal a otro objeto.