 Los 13 diagramas UML y sus componentes

1. 1. Tipos de diagramas UML
2. 2. Diagrama de Casos de Uso Captura los requisitos de un sistema. Los casos de uso son un medio de comunicación con los usuarios y otros interesados acerca de lo que se piensa hacer del sistema. • Nombre y Descripción: Se nombra como una frase verbal y se le da una descripción textual informal. • Requisitos: Los requisitos definen los requisitos funcionales formales que un caso de uso debe proveer al usuario final. • Restricciones: Los requisitos definen los requisitos funcionales formales que un caso de uso debe proveer al usuario final. • Escenarios: Una restricción es una condición o restricción bajo la cual opera un caso de uso y que incluye pre, y post condiciones y condiciones invariantes. • Diagramas de escenarios: Un escenario es una descripción formal del flujo de eventos que ocurren durante la ejecución de una instancia de casos de uso. • Información adicional Actores: Representan los roles que pueden incluir usuarios humanos, un hardware externo u otros sistemas. Actor Generalizado Caso de Uso: Un caso de uso es una sola unidad de trabajo significativo. Este provee una vista de alto nivel de comportamiento observable para alguien o algo fuera del sistema. Conector “use”: Puede tener valores múltiples en cada final, en el ejemplo se muestra un cliente que puede tener usa sesión de “retiro” a la vez pero que el servicio puede atender a múltiples usuarios
3. 2. Diagrama de Casos de Uso (Cont.) Incluir caso de uso: Los casos de uso pueden contener la funcionalidad de otro caso de uso como parte de su proceso normal. En general se asume que cualquier caso de uso incluido se llamará cada vez que se ejecute una ruta básica. Extender: Se pude usar para extender el comportamiento de otro, en el ejemplo la autorización puede o no implicar una modificación de la orden y puede ser antes o después de la autorización. Punto de Extensión: El punto al cual un caso de uso extendido se agrega se puede definir por medio de un punto de extensión. Límite del Sistema: Usualmente se usa para mostrar casos de uso dentro del sistema y actores fuera del sistema.
4. 3.Diagrama de Clases Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento. Elementos • Clase: Atributos, métodos y visibilidad (public+, private-, protected#) • Relaciones: Herencia, composición, agregación, asociación y Uso Herencia Agregación Asociación Dependencia/Instanciación/USO
5. [4.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-3-638.jpg?cb=1406283424)Diagrama de objeto Se puede considerar un caso especial de un diagrama de clase. Los diagramas de objetos usan un sub conjunto de elementos de un diagrama de clase para enfatizar la relación entre las instancias de las clases en algún punto en el tiempo. Estos son útiles para entender los diagramas de clases. Estos no muestran nada diferente en su arquitectura a los diagramas de secuencia, pero reflejan multiplicidad y roles. Clase Objeto – Pertenece a Estado en tiempo de ejecución -Atributos -Operaciones Clases Objetos
6. [5.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-17-638.jpg?cb=1406283424)Diagrama de Secuencia Es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida.
7. 6. Diagrama de Secuencia (Cont.) Inicio/Final de Línea de Vida: Una línea de vida se puede crear o destruir durante la escala de tiempo representada por un diagrama de secuencia. En el último caso, la línea de vida se termina por un símbolo de detención, representado como una cruz. Restricciones de Tiempo/Duración Puerta: Punto de conexión para conectar un mensaje dentro de un fragmento con un mensaje fuera del fragmento. Descomposición en Parte: Un objeto puede tener más de una línea de vida que viene de ésta. Esto permite mensajes de entre e intra objetos para que se muestren en el mismo diagrama. Continuaciones/Invariantes de Estado: Una invariante de estado es una restricción ubicada en una línea de vida que debe ser verdadera en el tiempo de ejecución.
8. 7. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos. Línea de Vida: representa un participante individual en un diagrama, algunas veces un diagrama de secuencia tendrá una línea de vida con un símbolo del elemento actor en la parte superior. Este usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso. Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida. Mensajes: Se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales. Síncrono Asíncrono Retorno asíncrono Ocurrencia de Ejecución Mensaje “self”: Puede representar una llamada recursiva de una operación, o un método llamando a otro método perteneciente al mismo objeto. Mensajes Perdidos/Encontrados: Son aquellos que han sido enviados pero que no han llegado al destino esperado, o que han llegado a un destino que no se muestra en el diagrama actual. Los mensajes encontrados son aquellos que llegan de un remitente no conocido, o de un remitente no conocido en el diagrama actual.
9. 8. Diagrama de Actividades se usa para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades. Los Diagramas de Actividades son útiles para el Modelado de Negocios donde se usan para detallar el proceso involucrado en las actividades de negocio. Actividad: Una actividad es la especificación de una secuencia parametrizada de comportamiento. Acciones: Una acción representa un solo paso dentro de una actividad. Restricciones de Acción: Las restricciones se pueden adjuntar a una acción. Flujo de Control: Muestra el flujo de control de una acción a otra. Nodo Inicial Nodo Final
10. 9. Diagrama de Actividades (Cont.) Nodo Final de Flujo: Representa el final de un flujo de control más no el final de todos los flujos. Flujos de Objeto y Objeto: es la ruta a lo largo de la cual pueden pasar objetos o datos. Ejemplo Objeto de almacenaje “Datastore” Flujo de objeto Notación de acceso rápida Objetos Nodos de Decisión y Combinación Nodos de Bifurcación y Unión
11. [9.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-11-638.jpg?cb=1406283424) Diagrama de Actividades (Cont.) Región de Expansión: Una región de expansión es una región de actividad estructurada que se ejecuta muchas veces. Los nodos de expansión de salida y entrada se dibujan como un grupo de tres casillas representando una selección múltiple de ítems. La clave reiterativa, paralelo, o flujo se muestra en la esquina izquierda arriba de la región. Gestores de excepciónRegión de Actividad Interrumpible: Iinterrumpible rodea un grupo de acciones que se pueden interrumpir; en este caso el proceso de la orden será interrumpido si sucede una interrupción del tipo “Cancelar Orden” Partición: Una partición de una actividad se muestra como calles horizontales o verticales; en el ejemplo se describen dos actividades que realizan los departamentos
12. 10. Diagrama de Colaboración: define un conjunto de roles co- operativos usados colectivamente para ilustrar una funcionalidad especifica. Una colaboración debería solo mostrar los roles y los atributos requeridos para lograr sus tareas o funciones definidas. Enlace de Roles: Se dibuja desde una colaboración a un clasificador que completa el rol. Representa: Se dibuja desde una colaboración a un clasificador para mostrar que una colaboración se usa en el clasificador. Ocurrencia: Se puede dibujar desde una colaboración a un clasificador para mostrar que la colaboración representa el clasificador.
13. [11.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-8-638.jpg?cb=1406283424)Diagrama de Paquetes Se usan para reflejar la organización de paquetes y sus elementos. Cuando se usan para representaciones, los diagramas de paquete de los elementos de clase se usan para proveer una visualización de los espacios de nombres. Los usos más comunes para los diagramas de paquete son para organizar diagramas de casos de uso y diagramas de clase, a pesar de que el uso de los diagramas de paquete no es limitado a estos elementos UML. -Combinación de Paquetes: Cuando un conector «merge» se usa en un paquete, la fuente de la combinación importa los contenidos importados y anidados del destino. Si existe un elemento dentro del origen y el destino, las definiciones del elemento origen se expandirán para incluir las definiciones del elemento contenidas en el destino. Todos los elementos agregados o actualizados por una combinación se notan por una relación de generalización desde el origen hasta el destino. -Importación de paquetes: El conector «import» indica que los elementos dentro del paquete destino, que en este ejemplo es una sola clase, se importarán al paquete origen. El espacio de nombre del paquete origen ganará acceso a la Clase/s de Destino; el espacio de nombre del destino no está afectado. -Conectores Anidados: El conector anidado entre el paquete destino y los paquetes de origen reflejan lo que muestran los contenidos del paquete.
14. 12. Diagramas de Componentes Ilustran las piezas del software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase – usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema. Elementos • Representación de componentes • Interfaces requeridas (Conector de Ensamble) • Componentes con puertos Componentes Conector de Ensamble ProporcionaConsume Puerto Proporciona Consume
15. 13. Diagrama de Despliegue
16. 14. Diagrama de Estructura Compuesta es un diagrama que muestra la estructura interna de un clasificador, incluyendo sus puntos de interacción a otras partes del sistema. Esto muestra la configuración y relación de las partes que juntas realizan el comportamiento de clasificador contenido. Se describe la forma en que las clases se pueden mostrar como elementos compuestos exponiendo interfaces y conteniendo puertos y partes. Parte: Representa un conjunto de una o más instancias que pertenecen a una instancia del clasificador contenida. Una parte se puede quitar de sus padres antes de que el padre se elimine, para que la parte no se elimine al mismo tiempo. Puerto: Es un elemento escrito que representa una parte visible externa de una instancia del clasificador contenido. Un Puerto puede especificar los servicios que un clasificador provee así como también los servicios que este requiere de su entorno. Interface: es similar a una clase pero con un número de restricciones. Todas las operaciones de la interfaz son públicas y abstractas, y no proveen ninguna implementación predeterminada. Todos los atributos de la interfaz deben ser constantes. Sin embargo, mientras que una clase puede solo heredar de una sola super- clase, puede implementar interfaces múltiples. Interface Provista: Una interfaz provista se muestra como una “pelota en un palo” adjuntada al borde de un elemento clasificador. Una interfaz requerida se muestra como una “copa en un palo” adjuntada al borde de un elemento clasificador.
17. [14.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-6-638.jpg?cb=1406283424) Diagrama de Estructura Compuesta (Cont.) Delegar: se usa para definir los trabajos internos de los puertos e interfaces externas del componente. Un conector delegar se muestra como una flecha con un estereotipo «delegar». Esto conecta un contrato externo de un componente como se muestra por sus puertos a la realización interna del comportamiento de la parte del componente. [7.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-7-638.jpg?cb=1406283424)Diagrama de despliegue Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Nodo: Elemento de HW o SW Instancia de Nodo: Su nombre esta subrayado y tiene dos puntos antes del tipo de nodo base. Artefacto: Es un producto del proceso de desarrollo de software, que puede incluir los modelos del proceso, archivos fuente, ejecutables, documentos de diseño, reportes de prueba, prototipos, manuales de usuario y más. Ejemplos Asociación: una asociación representa una ruta de comunicación entre los nodos. Nodo como Contenedor: Un nodo puede contener otros elementos, como componentes o artefactos.
18. 15. Diagrama de Maquina de Estados Modela el comportamiento de un solo objeto, especificando la secuencia de eventos que un objeto atraviesa durante su tiempo de vida en respuesta a los eventos. Estado Estado Inicial/Final Transiciones: Las transiciones desde un estado al siguiente se denotan por líneas con flechas. Una transición puede tener un disparador, una guarda y un efecto Acciones de Estado: Si el estado de destino tenía muchas transiciones llegando al mismo, y cada transición tenía el mismo efecto asociado con este, sería mejor asociar el efecto con el estado de destino en lugar de con las transiciones. Esto se puede realizar para definir una acción de entrada para el estado. Transiciones Recursivas: Un estado puede tener una transición que retorna a sí misma, siendo útil cuando un efecto se asocia con la transición. Estados Compuestos
19. [15.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-15-638.jpg?cb=1406283424)Diagrama de Maquina de Estados (Cont.) Punto de Entrada: Es posible utilizar cuando el estado inicial normal no requiere una sub maquina. Punto de Salida Pseudo Estado “choice” Pseudo Estado “Junction” (unión) Pseudo Estado “terminate” (terminar): Indica que la línea de vida de la maquina de estado ha terminado.
20. [15.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-16-638.jpg?cb=1406283424)Diagrama de Maquina de Estados (Cont.) Estado “history” (historial): se usa para recordar el estado anterior de una maquina de estado cuando fue interrumpida. Regiones Recientes: se usa para recordar el estado anterior de una maquina de estado cuando fue interrumpida. Los pseudo estados en lugar de los pseudo estados elección y combinación. Estos símbolos se usan para sincronizar los hilos concurrentes.
21. [19.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-19-638.jpg?cb=1406283424)Un fragmento combinado es una o más secuencias de procesos incluidas en un marco y ejecutadas bajo circunstancias nombradas específicas. Los fragmentos disponibles son: El fragmento Alternative (denotedo “alt”) modela estructuras if…then…else. El fragmento Option (denotado “opt”) modela estructuras switch. El fragmento Break modela una secuencia alternativa de eventos que se procesa en lugar de todo del resto del diagrama. El fragmento Parallel (denotado “par”) modela procesos concurrentes. El fragmento de secuenciado Weak (denotado “seq”) incluye un número de secuencias para las cuales todos los mensajes se deben procesar en un segmento anterior, antes de que el siguiente segmento pueda comenzar, pero que no impone ningún secuenciado en los mensajes que no comparten una línea de vida. El fragmento de secuenciado Strict (denotado “strict”) incluye una serie de mensajes que se deben procesar en el orden proporcionado. El fragmento Negative (denotado “neg”) incluye una serie de mensajes inválidos. El fragmento Critical incluye una sección crítica. El fragmento Ignore declara un mensaje o mensajes que no son de ningún interés si este aparece en el contexto actual. El fragmento Consider es el opuesto del fragmento Ignore: cualquier mensaje que no se incluya en el fragmento Consider se debería ignorar. El fragmento Assertion (denotado “assert”) designa que cualquier secuencia que no se muestra como un operando de la aserción es inválida. El fragmento Loop incluye una serie de mensajes que están repetidos. Diagrama de Secuencia (Cont.)
22. [20.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-20-638.jpg?cb=1406283424)Diagrama de Comunicaciones Inicialmente llamado un diagrama de colaboración, es un diagrama de interacción que muestra información similar a los diagramas de secuencia pero su foco principal es en la relación de objetos. En los diagramas de comunicaciones, los objetos como se muestran con conectores de asociación entre ellos. Los mensajes se agregan a las asociaciones y se muestran como flechas cortas apuntando en la dirección del flujo del mensaje. La secuencia de los mensajes se muestra a través de un esquena enumerado.
23. [21.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-21-638.jpg?cb=1406283424)Diagrama de Tiempo Se usan para mostrar el cambio en el estado o valor de uno o más elementos en el tiempo. Este también puede mostrar la interacción entre los eventos de tiempos, las restricciones de tiempos y la duración que los gobiernan. Línea de Vida del Estado: Muestra el cambio de estado de ítem en el tiempo. El eje-X muestra el tiempo trascurrido en cualquier unidad que se elija mientras que el eje-Y se nombra con una lista de estados proporcionados. Línea de Vida del Valor: Muestra el cambio del valor de un ítem en el tiempo. El eje-X muestra el tiempo transcurrido en cualquier unidad que se elija, lo mismo que para la línea de vida del estado. El valor se muestra entre el par de líneas horizontales que se cruzan en cada cambio del valor. Mostrar Ambos Juntos: Las líneas de vida y del estado se pueden ubicar una arriba de otro en cualquier combinación. Estas deben tener el mismo eje-X. Los mensajes se pueden pasar de una línea de vida a otra. Cada transición del estado o valor puede tener un evento definido, una restricción de tiempo que indica cuándo debe ocurrir un evento, y una restricción de duración que indica cuánto tiempo debe estar en efecto un valor o estado.
24. [22.](https://image.slidesharecdn.com/uml-140725101452-phpapp01/95/los-13-diagramas-uml-y-sus-componentes-22-638.jpg?cb=1406283424)Diagrama de Descripción de la Interacción es una forma de diagrama de actividad en el cual los nodos representan diagramas de interacción. Los diagramas de interacción pueden incluir diagramas de secuencia, comunicación, de descripción de la interacción y de tiempos. La mayoría de la notación para los diagramas de descripción de la interacción es la misma que para los diagramas de actividad, por ejemplo los nodos inicial, final, decisión, combinación, bifurcación y unión son todos lo mismo. Sin embargo, los diagramas de descripción de la interacción introducen dos elementos nuevos, ocurrencias de interacción y elementos de interacción. Ocurrencia de la Interacción: Son referencias a los diagramas de interacción existentes. Una ocurrencia de interacción se muestra como un marco de referencia, es decir un marco con “ref” en la esquina superior izquierda. Elementos de Interacción: Son similares a los de ocurrencias de interacción en el hecho de que estos muestran una representación de diagramas de interacción existentes dentro de un marco rectangular. Estos difieren en el hecho de que muestran los contenidos de los diagramas de referencia en línea.