CASOS DE USO

INTRO o DEFINICION BASICA DE CASO DE USO:

Son tecnicas para analizar y especificar el comportamiento de un sistema o una parte de un sistema describiendo que hace, pero NO como lo hace. Son parte del standard UML.

Otra definicion es que un caso de uso es una forma de expresar como un ACTOR externo al sistema interactúa con el mismo, para obtener un beneficio.

ACTOR:

Es quien usa el sistema. Puede ser “algo o alguien”, es decir, puede ser una persona, u otro sistema, o maquina, etc que interactua con nuestro sistema. Este actor es siempre externo al sistema, por lo tanto, identificar estos actores, hace que podamos identificar a su vez, los limites de nuestro sistema.

ACTOR VS USUARIO:

Un actor es quien representa un rol, mientras que un usuario es quien cuando usa un sistema, asume un rol, y ese usuario puede asumir distintos roles en distintos momentos.

Por ejemplo, una persona puede trabajar en un banco como ejecutivo de cuentas y asumira ese rol de usuario para interactuar con el sistema en ese momento. Perso si tambien tiene cuenta en ese banco y lo quiere usar de esa forma, asumira el rol de Cliente en ese momento.

TIPOS DE ACTORES:

Pueden ser personas, sistemas de software o hardware.

Ejemplo anterior Banco: El actor es una persona. (Persona como actor)

Ejemplo Robot: Si estoy construyendo un sistema que controla los movimientos de un robot, el robot, seria el actor. (Hardware como actor)

Ejemplo Contabilidad: Si estoy construyendo un sistema que genere asientos contables para que sean procesados por un Sistema de Contabilidad, el Sistema de Contabilidad, seria el actor. (Software como actor)

REPRESENTACION DE ACTOR:

Se representa con un stick man con el nombre del rol que esta cumpliendo debajo, o el nombre del sistema, depende que tipo de actor sea. EJ:

Diagrama

Descripción generada automáticamente

GENERALIZACION DE ACTORES:

Se puede interpretar como la herencia en los lenguajes de programación. Sucede cuando hay mas de un actor, y uno de esos actores, puede iniciar todos los casos de uso del primer actor, y algunos mas aparte de ellos.

EJ: Un actor como supervisor de ventas, puede iniciar todos los casos de uso de un actor empleado de ventas, pero a su vez, puede, por ejemplo, autorizar ventas, que el empleado no puede.

REPRESENTACION DE GENERALIZCION O HERENCIA:

Diagrama

Descripción generada automáticamenteSe representan los actores como en la representacion de un actor, pero tiene una flecha que indica la herencia entre uno y otro: EJ:

INICIO DE CASO DE USO:

Un caso de uso se puede iniciar por un único actor a la vez y esto debe representar una funcionalidad completa del sistema. La ejecución de este caso de uso debe dejar resultados observables y de interés para el actor que lo incio, es decir, un beneficio.

DEFINICION AVANZADA DE CASO DE USO:

Se utilizan para especificar los requisitos funcionales de un sistema. Cada caso de uso, en el que el actor interactua con el sistema, se debe involucrar todos los pasos para llevar a cabo la funcionalidad dada describiendo que hace el sistema, pero no como lo hace. Es decir, se puede ver la entrada y la salida del sistema, pero nunca el proceso. Al describirlo, hay que dar ESCENARIOS principales y alternativos. Los alternativos, son escenarios de errores o eventos excepcionales.

ESCENARIOS:

Es una secuencia especifica de acciones que describe un comportamiento. Un caso de uso, esta definido por uno o mas escenarios. Existen escenarios principales, que describen el flujo normal del caso de uso, y escenarios secundarios que describen alternativas.

REPRESENTACION DE CASO DE USO:

Es un óvalo con el nombre del caso de uso en su interior. Siempre es un verbo en infinitivo seguido del objeto o entidad afectada por el caso de uso. EJ:

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

CARACTERISTICAS DE UN CASO DE USO:

1. Expresados desde el punto de vista del actor externo y no del sistema. EJ: Ingresar Pedido, que es algo que puede hacer un actor.
2. Se usa texto informal, ya que está dirigido al cliente, entonces hay que usar lenguaje sencillo que se entienda.
3. Describen lo que hace el sistema (NO como lo hace), y lo que hace el actor cuando interactua con él.
4. Son iniciados por un único actor a la vez. No puede haber casos de uso iniciados sin actor.
5. Están acotados solo a una funcionalidad completa del sistema. EJ: Registrar venta tiene en su interior varias funcionalidades, como ingresar articulos vendidos, que no es necesario indicarsela al sistema, simplemente será una funcionalidad que estará incluida cuando iniciemos el caso de uso de Registrar Venta.
6. Los casos de uso deben dejar siempre un beneficio o resultado de interés del actor. EJ: Si un empleado de ventas inicia el caso de uso Registrar Venta, su beneficio es que la venta quede registrada.

MODELO DE CASOS DE USO:

Comunmente denominado MCU, representa al sistema desde el punto de vista del actor y su objetivo facilitar la comunicación entre el cliente y el equipo de desarrollo para llegar a un acuerdo sobre los requisitos, condiciones y posibildades. Luego de que el cliente lo revisa, este se irá refiniando.

Está compuesto por:

* DIAGRAMA DE CASOS DE USO.
* DESCRIPCION DE CASOS DE USO.

REPRESENTACION DEL DIAGRAMA DE CASOS DE USO:

Está formado por un rectángulo que es el que delimita el alcance del sistema, el cual incluye el nombre de este. En su intererior se representarán los casos de uso y su relación. A su vez, se representan los actores en su exterior, porque son externos al sistema, y se relaciona a los actores con los casos de uso. Se debe poder ver claramente que casos de uso puede inciar cada actor, y si existiera una generalizacion entre actores, también se representa en este diagrama. EJ:

Diagrama

Descripción generada automáticamente

DESCRIPCION DE LOS CASOS DE USO:

Se incluye luego del diagrama de casos de uso y como lo dice su nombre, hace una descripción completa de lo que hace cada caso de uso (NO como lo hace) incluidos dentro del diagrama. Por lo general, se sigue una plantilla del siguiente estilo:

* **Nombre del caso de uso**: Verbo en infinitivo y expresado desde el punto de vista del actor. Se explica en representacion de un caso de uso.
* **Actor Principal**: Es el que inicia el caso de uso e interactua con el sistema.
* **Actor Secundario**: Participa una vez iniciado el caso de uso, puede ser uno, varios o ninguno, no van en el diagrama.
* **Descripcion general a trazo grueso**: Breve descripcion de lo que hace el caso de uso, especifica al actor, la funcionalidad a llevar a cabo y el beneficio para el actor. Puede incluir interfal tentativa.
* **Precondicion**: Se indican los casos de uso anteriores que deben haberse iniciado para ejecutar este caso de uso.
* **PostCondicion**: Resultado de interes o beneficio para el actor.
* **Descripcion detallada o trazo fino**: Indica paso a paso que hace el actor y que responde el sistema (lo que muestra), por ejemplo, el sistema hace calculos, o el sistema realiza busqueda. También es común acompañar la descripción con alguna ilustración o interfaz, donde se puede visualizar lo que se describe.

La descripción detallada se divide en el **flujo normal** y el **flujo alternativo**

* + **Flujo Normal**: Descripción del caso feliz o que todo sale bien. Puede ocurrir que se puedan repetir pasos, por ejemplo, seleccionar varios productos para una venta. Esto se suele representarse como “Para cada X seleccionado, se repiten los pasos Y a Z” o “Se repite X hasta que ocrurra Y”.
  + **Flujo Alternativo**: Descipción de las alternativas al flujo normal. Las alternativas son a la respuesta del sistema, es decir, ante una determinada accion del actor, que se espera que el sistema responda de una determinada manera, lo hace de otra manera. Hay que especificar a que punto del caso de uso vuelve luego, o bien si luego de esto se finaliza el caso de uso. Siempre:
    - **Representan un error o excepción**
    - **No tienen sentido por si mismas fuera del contexto de un caso de uso**

EJ: En Registrar Venta, si el actor ingresa un código de un producto inexistente, el sistema deberá informar dicha situación y regresar al punto de flujo normal donde se permita que el actor ingrese nuevamente el código del producto.

Si bien es común documentarlas al final de un flujo normal, también es común y útil documentarlas en tablas, mostrando el flujo normal en la primera columna y el alternativo en la segunda.

RELACIONES ENTRE CASOS DE USO:

Existen 2 tipos de relaciones entre casos de uso, **relaciones de extención** y **relaciones de inclusión**.

* **Relaciones de extension**: Son casos de usos que extienden a otros casos de uso, agregandoles alguna funcionalidad. Son una serie de pasos opcionales dentro de una funcionalidad que se ejecutará solo en algunas ocasiones. Sus características son:
  + Representan una funcionalidad que no siempre ocurre.
  + La funcionalidad que existende al caso de uso, tambien es un caso de uso en si mismo.
  + Si bien es lo más común, no siempre vienen de un error o una excepción.

EJ: En Registrar Venta, al vender a un cliente nuevo, puede aparecer una excepción, que consista en interrumpir la ejecución y ejecutar el caso de uso Registrar Cliente. Es decir, Registrar Cliente extiende o agrega una funcionalidad a Registrar Venta, pero a su vez, por si solo también es un caso de uso.

* **Relaciones de inclusión**: Surgen cuando una serie de pasos se repite en dos o mas casos de uso y estos pasos generan una funcionalidad completa. En estos casos, a partir de estos pasos, se puede crear un nuevo caso de uso, que estará incluido en cada caso de uso que utilize esta funcionalidad y se ejecutará siempre que estos se ejecuten.

Sus características son:

* Aparecen como una funcionalidad común luego de haber especificado varios casos de uso.
* Las funcionalidades incluidas son un caso de uso en sí mismos.
* Se ejecutan siempre que el caso de uso que las incluya se ejecute.

EJ: La funcionalidad Buscar Producto puede ser accedida por más de un caso de uso, como Registrar Venta o Registrar Pedido, y a su vez, es un caso de uso en si mismo. Este se ejecutará siempre que se ejecute uno de estos casos de uso.

PRINCIPAL DIFERENCIA ENTRE LAS RELACIONES DE CASO DE USO:

La diferencia principal entre una relacion de extensión y una de inclusión, es que la de extensión se ejecutará de manera opcional, mientras que la de inclusión se ejecutará siempre que se ejecute el caso de uso que contenga esta funcionalidad.

REPRESENTACION DE RELACIONES DE EXCLUSIÓN:

Se representan ambos casos de uso y luego se dibuja una flecha con sentido desde el caso de uso “que extiende a” hacia el caso de uso que es “extendido”. EJ:

Diagrama

Descripción generada automáticamente

REPRESENTACION DE RELACIONES DE INCLUSIÓN:

Se representan los casos de uso y luego se dibujan flechas con sentido hacia el caso de uso que se encuentra incluido en el caso de uso en ejecución.Diagrama

Descripción generada automáticamente

DIFERENCIAS ENTRE UNA ALTERNATIVA Y UNA RELACION DE EXTENSION:

* En una relacion de extension, la funcionalidad que extiende al caso de uso es tambien un caso de uso en si mismo mientras que la alternativa no lo es.
* Una alternativa es un error o una excepción, mientrsa que una extensión puede no serlo.

**ACÁ FINALIZA TODO LO SE NECESITA PARA REALIZAR UN EJERCICIO DE CASOS DE USO, QUEDAN POR DEFINIR ALGUNOS CONCEPTOS TEÓRICOS QUE SIRVEN PARA ENTENDER COMO IDENTIFICAR ACTORES Y CASOS DE USO CORRECTAMENTE.**

METODOLOGIA DE TRABAJO PARA LOS CASOS DE USO (ES TODO TEORICO):

Realizar correctamente un MCU consiste en:

1. Encontrar los actores que interectuaran con el sistema.
2. Encontrar los casos de uso para cada actor.
3. Describir los casos de uso brevemente y luego de forma detallada.

Encontrar esto es muy importante y suele ser tarea del analista del sistema, pero requiere la participación del cliente. Es un proceso iterativo, donde se crea un proceso preliminar para validar con el ciente y se le va dando forma hasta llegar a la versión final con cada interación.

Aplicar la metodología consiste en:

1. **Identificar Errores**: Consiste en que antes de avanzar con los casos de uso, se tiene que identificar a los actores candidatos para interactuar con el sistema, agruparlos y evaluarlos según su función en detalle, para ver que caso de uso aplicará a cada uno, ya que, en un principio, erróneamente se pueden incluir en grupos más generales.
2. **Identificar los casos de uso para cada actor**: En primera instancia se definen los casos de uso para cada actor, a los que se denomina, casos de uso candidatos. Pero luego se debe detallar minuciosamente para ir depurando cuales se terminarán usando. Se deberá dar nombre a cada uno siguiendo las reglas de nombres ya vistas. Como resultado, se obtiene el primer diagrama de casos de uso, sujeto a evaluaciones y restructuraciones.
3. **Identificar nuevos casos de uso a partir de los existentes**: En este paso se verifica si el sistema está completo o completarlo buscando nuevas funcionalidades a partir de:
   1. **Variaciones significativas de los casos de uso existentes**: EJ: Si hay mucha diferencia entre registrar una venta minorista y una mayorista. El caso de uso Registrar Venta, puede separarse en estos dos mencionados.
   2. **Casos de uso opuestos**: EJ: Si existe Registrar Venta, debería existir el Anular Venta.
   3. **Casos de uso que preceden a otros existentes**: Preguntarse que pasa antes del caso de uso en ejecución. EJ: Si se está ejecutando Registrar Venta, el cliente previamente deberia estar registrado en el sistema, con lo cual, debería existir, Registrar Cliente.
   4. **Casos de uso que suceden a los casos de uso existentes**: Preguntarse que pasa luego de la ejecución de un caso de uso. EJ: Que pasa despues de Registrar Venta, por ejemplo, si se emite una factura y eso cubre una funcionalidad completa, podría necesitarse el caso de uso Emitir factura.

Luego de esto, se obtiene como resultado un diagrama de casos de uso más completo, con los nuevos casos de uso encontrados.

1. **Describir los casos de uso con trazo grueso**: Una vez identificados los casos de uso, se documentan con este tipo de forma, es decir, una breve descripción desde el punto de vista del actor, indicando el actor involucrado, funcionalidad que se va a llevar a cabo, y el resultado o beneficio que deja para el actor.

Luego de esto se obtiene como resultado el MCU compuesto por el diagrama y la descripcion con trazo grueso de cada caso de uso.

1. **Priorizar los casos de uso**: Una vez hecho el trazo grueso, se definen prioridades para ver cuales serán tenidos en cuenta en la primera iteración del proceso de desarrollo y cuales se pueden postergar. Para priorizarlos se estila la siguiente categorización:
   1. **Requerimientos imprescindibles - prioridad alta**: Si no se implementan, el sistema no funciona y/o no tendría sentido.
   2. **Requerimientos importantes – prioridad media:** Son los requerimientos muy deseables por el cliente, pero no imprescindibles para la primera etapa. Se pueden negociar con el cliente.
   3. **Requerimientos deseables – prioridad baja:** Son requerimientos posibles de implementar pero que podrían eliminarse. Se implementan solo si hay tiempo disponible.

Una vez categorizados, por lo general, se desarrollan los de prioridad alta, se discuten los de categoría media y se descartan los de categoría baja.

Luego de esto se obtiene los casos de uso priorizados a desarrollar en cada iteración del proceso de desarrollo.

1. **Describir los casos de uso con trazo fino**: Una vez seleccionados los casos de uso a implementar, se procede a describirlos con trazo fino, es decir, desde el punto de vista del actor, se crea el flujo normal y el flujo alternativo. En este paso se crea un prototipo que ilustre lo descripto al cliente y brinde una idea de como se vería.

Luego de esto se obtiene la descripción detallada de los casos de uso junto a sus prototipos de interfaz.

1. **Identificar relaciones entre casos de uso**: Revisando el trazo fino, se pueden detectar secuencias que se repiten siempre en distintos casos de uso. Esto puede generar un nuevo caso de uso de inclusión, o de relación *include*. También puede darse el caso de una secuencia que se repita, pero de manera opcional dentro de distintos casos de uso. Esto daría pie a un nuevo caso de uso del tipo extensión, o de relación *extend*.

Luego de esto se obtiene el diagrama de casos de uso completo incluytendo las relaciones entre los casos de uso.