

P07- Diseño de pruebas del sistema

Pruebas de sistema vs pruebas de aceptación

El objetivo de esta práctica es trabajar con el diseño de pruebas del sistema, en concreto con el método de Transición de Estados.

En clase hemos contrastado ambos tipos de pruebas para que veáis con mayor claridad las diferencias entre ellas. Es importante tenerlo puesto que en ambos casos la SUT es la misma.

La principal diferencia entre los dos tipos de pruebas es que sus objetivos son muy diferentes. En una prueba del sistema estamos VERIFICANDO (es decir, queremos evidenciar defectos), mientras que en una prueba de aceptación queremos VALIDAR las expectativas del cliente. Como nuestro objetivo es diferente, y como ha ocurrido en ocasiones anteriores, los conjuntos de pruebas en cada caso también serán diferentes.

Cuando realizamos pruebas del sistema, lo hacemos desde el punto de vista del desarrollador y por lo tanto, tenemos en cuenta qué componentes participan en cada caso de prueba, de forma que encontraremos errores en el flujo de información que se intercambian entre ellos (los valores deben ser correctos y en la secuencia adecuada para proporcionar la funcionalidad probada). Necesitamos que los requisitos (funcionalidades del sistema) estén claros.

Cuando realizamos pruebas de aceptación, lo hacemos desde el punto de vista del usuario final, y por lo tanto, consideramos los usos concretos que dicho usuario final previsiblemente hará sobre la aplicación. Se trata de ver si se satisfacen los criterios de aceptación acordados con el cliente al inicio de proyecto.

Algo que tienen en común ambos tipos de pruebas es que los tests pueden contener múltiples verificaciones, puesto que los comportamientos a probar tendrán diferentes conjuntos de entradas, que provocarán resultados esperados "intermedios". El resultado esperado de los tests será el resultado de ejecutar la secuencia de entradas y acciones correspondientes.

En esta práctica vamos a usar el método de diseño de transición de estados, aplicable a pruebas del sistema. En clase también hemos ilustrado métodos de diseño para pruebas de aceptación (que deberás tener en cuenta de cara al examen).

Con respecto al método de diseño de transición de estados, obviamente será aplicable a sistemas con estado. Usaremos una representación particular de grafo, en concreto será una máquina de estados, la cual modelará los cambios de estado sobre la entidad observable de dicho sistema que queremos probar.

Es fundamental que tengas claros los diferentes elementos del diagrama: estados, transiciones, eventos, acciones y guardas. Y que explícites cada uno de ellos, documentando así todos los elementos utilizados.

Representar el modelo de estados es sólo una parte del proceso, recuerda que el objetivo es diseñar un conjunto de casos de prueba que ejerciten todas las transiciones. Aunque obviamente siempre buscamos eficiencia (además de efectividad), en esta ocasión no vamos a centrarnos en obtener un conjunto mínimo y sólo consideraremos el hecho de los tests ser efectivos.

GitHub

El trabajo de esta sesión debes subirlo a *GitHub*. Los ejercicios de esta práctica deberán estar en el directorio **P07-STD** dentro de tu espacio de trabajo. Puedes usar la herramienta draw.io de sesiones anteriores. El resultado debes subirlo a GitHub en formato de imagen (png o jpg).

⇒ Ejercicio 1: Especificación terminal de supermercado

Los ficheros de este ejercicio estarán en la carpeta **supermercado**, que deberás crear en el directorio **P07-STD**

Recuerda que el diagrama debe quedar perfectamente documentado. En el directorio **Plantillas_P07** tenéis el fichero **supermercado.txt** que deberéis rellenar con la información sobre los diferentes elementos del cada diagrama de transición de estados.

Se trata de probar las acciones realizadas en un **terminal de un supermercado**. Para poder utilizar el sistema, el encargado del mismo deberá **autenticarse** en el sistema utilizando un código de acceso de 6 dígitos (se permite cualquier número de intentos, y no es posible introducir ningún carácter que no sea un dígito). Si el **código es incorrecto**, se mostrará el mensaje "Error de acceso". Una vez autenticado, se **activará el uso de la caja** y se mostrará el mensaje "Caja activada". El sistema **esperará** a que el cliente coloque los productos sobre terminal para poder comenzar a pasar por el escáner los productos (de uno en uno) que el cliente desee comprar. Cada vez que se escanee un producto, **se mostrara el subtotal en la pantalla** (inicialmente estará a cero para cada nuevo cliente). Si el **escáner no detecta bien el código de un producto**, mostrará el mensaje "Vuelva a pasar el producto" y será necesario volver a pasar dicho producto por el escáner. Cuando se hayan escaneado todos los productos que el usuario ha dejado sobre el terminal, se **se procederá al pago de la compra**. Se admitirá cualquier tarjeta de pago de cualquier banco. Si se produce un **error al procesar la tarjeta**, se mostrará el mensaje "Error en pago" y se deberá volver a intentar el procesamiento del pago. Una vez **procesado el pago correctamente** se procederá a **entregar el ticket** de compra generado al cliente, mostrando los códigos de los productos adquiridos, su precio, el número de unidades y el total de la compra. El cliente puede **cancelar la compra en el momento del pago**. Una vez atendido un cliente, el encargado del terminal seguirá a la espera del siguiente cliente, repitiendo el proceso hasta que termine su turno, en cuyo caso, el encargado del terminal procederá a **salir del sistema**. En ningún caso la salida del sistema se producirá a menos que se se termine de atender a un cliente.

Aplica el método de transición de estados y diseña la tabla de casos de prueba correspondiente.

⇒ Ejercicio 2: Especificación actas de asignatura

Los ficheros de este ejercicio estarán en la carpeta **actas**, que deberás crear en el directorio **P07-STD**

Recuerda que el diagrama debe quedar perfectamente documentado. En el directorio **Plantillas_P07** tenéis el fichero **supermercado.txt** que deberéis rellenar con la información sobre los diferentes elementos del diagrama de transición de estados.

Queremos diseñar pruebas para el **proceso de gestión de actas de alumnos**. Dicho proceso, comprende las acciones de **crear, modificar y cerrar un acta** para cada asignatura, en ese orden. El **profesor debe entrar en el sistema** y se autenticará mediante un código de acceso alfanumérico (se permite cualquier número de intentos, mostrándose el **mensaje de error "Error de acceso"**, en cada intento fallido, y el **mensaje "Hola, bienvenido"** si el código introducido es correcto). Una vez **autenticado**, se **creará automáticamente el acta de la asignatura**, a menos que **ya esté creada de una ejecución previa**, en cuyo caso, **se podrá introducir notas, siempre y cuando el acta no esté cerrada**.

Una vez **creada el acta**, el **profesor podrá introducir las notas** (a todos los efectos la **introducción de una nota o la modificación de las mismas son procesos equivalentes**). Para introducir las notas, el profesor **podrá proceder de forma manual** (introduciendo la **nota numérica de cada alumno individualmente**) o **podrá introducir todas las notas a partir de un fichero** (**sólo se podrá introducir las notas a partir de un fichero inmediatamente después de crear el acta**). Cuando el profesor termine de poner las notas, éste **guardará las notas**, y a continuación se **mostrará por pantalla un mensaje con las notas incorrectas**, o **"Notas guardadas correctamente"** si **no hay errores**. A pesar de que haya notas no correctas (con formatos o valores no válidos), se podrán seguir introduciendo notas.

El profesor sólo podrá **cerrar un acta** después de haber introducido y guardado las notas (no es necesario introducirlas todas), y siempre y cuando no haya ninguna nota incorrecta. Si el profesor intenta cerrar un acta con notas erróneas **se mostrará el mensaje "No se puede cerrar el acta"**. Una vez que el **acta esté cerrada**, ya no podrá ser modificada, y se mostrará en pantalla el **mensaje "Acta cerrada"**. El profesor puede abandonar el sistema en cualquier momento, pulsando la opción "Salir", y retomar el proceso más tarde. Si el acta ya está cerrada, el profesor solo podrá salir del sistema.

Resumen



¿Qué **conceptos** y **cuestiones** me deben quedar **CLAROS** después de hacer la práctica?



PRUEBAS DEL SISTEMA Y ACEPTACIÓN

- Las pruebas del sistema son pruebas de VERIFICACIÓN, mientras que las pruebas de aceptación son pruebas de VALIDACIÓN. En ambos casos nuestro SUT estará formado por todo el código de nuestra aplicación.
- Las pruebas del sistema se diseñan desde el punto de vista del desarrollador mientras que las pruebas de aceptación se diseñan desde el punto de vista del usuario final. En ambos casos se deben tener en cuenta varias entradas y resultados esperados "intermedios".
- Las pruebas del sistema se basan en los requerimientos de la aplicación, y las pruebas de aceptación se basan en las propiedades emergentes. El objetivo de las pruebas del sistema es encontrar defectos en las funcionalidades de dicho sistema, mientras que el objetivo de las pruebas de aceptación es comprobar que se satisfacen los criterios de aceptación.

DISEÑO DE PRUEBAS DE SISTEMA Y ACEPTACIÓN

- En ambos casos se usan técnicas de caja negra.
- En una prueba del sistema la selección de comportamientos a probar se hace desde un punto de vista técnico (se tiene en cuenta cómo se han implementado los componentes implicados en las funcionalidades probadas).
- Dos ejemplos de métodos de diseño para pruebas del sistema son: diseño basado en casos de uso, y transición de estados.
- En una prueba de aceptación el diseño se realiza pensando siempre en el uso "real" de nuestra aplicación por el usuario o usuarios finales. (no tenemos en cuenta qué componentes están implicados).
- Dos ejemplos de métodos de diseño para pruebas de aceptación son: diseño basado requerimientos, y diseño basado en escenarios. En ambos casos podemos determinar los casos de prueba particionando las entradas en particiones equivalentes.
- En cualquier caso es necesario fijar estrategias de pruebas para poder ser efectivos y eficientes. Dichas estrategias nos permitirán definir un objetivo claro (efectividad) para poder ser lo más eficientes posible.

MÉTODO DE DISEÑO DE PRUEBAS DE TRANSICIÓN DE ESTADOS

- Es un método de caja negra. Y sólo es aplicable si se trata de un sistema con estado, permitiéndonos modelar los diferentes cambios de estado durante la ejecución de la aplicación.
- Es importante que queden claramente documentados todos los elementos del diagrama. Recuerda que los eventos son entradas al sistema que pueden provocar cambios de estado. Las acciones son funciones del sistema desencadenadas por los eventos. Las guardas son condiciones que nos permiten discriminar entre diferentes transiciones desde un estado. Tanto las acciones como las guardas son opcionales.
- Una vez que hemos creado el diagrama definiremos un criterio de selección. Dos criterios habituales son: recorrer todos los estados, y recorrer todas las transiciones, siendo este último el más usado.
- Los casos de prueba estarán formados por la secuencia de eventos y acciones desde el estado inicial que nos conducirán al estado final del mismo. Es posible que no haya un estado final, pero siempre habrá un estado inicial.
- Tendremos que indicar todas las entradas, y proporcionar resultados esperados intermedios correspondientes a los cambios de estado.