

## RESUMEN 2DO PARCIAL ISW PRACTICO

### Testing de Caja Negra

- Lo primero que haces es definir las Condiciones Externas
- Estas son todas las entradas y salida del sistema
- Las entradas son todo lo que el usuario puede ingresar/seleccionar, o todo sobre lo que el usuario pueda hacer una modificación o afectarlo
- Las salidas son las respuestas que da el sistema a esas entradas
- Para definir las Condiciones externas, primero diferencia si son de E o S
- Después ponle un nombre → Debe describir que es la entrada o la salida (edad del usuario, dirección, cantidad de X cosa, opción seleccionada) → No definís exactamente un valor, si no el nombre de la opción o entrada (Calle 1, nombre de usuario, tipo de producto, productos seleccionados, etc)
- Como Condición de salida, casi siempre podes poner un “Mensaje de (lo que sea que se trato de hacer)”
- Si una condición externa es una lista que va priorizada/ordenada/valorada, tomala toda como una sola entrada (la lista de X cosas ordenadas)
- OJO! → Si esta funcionalidad solo la puede hacer un cierto usuario con ciertos permisos, el “Usuario con Permisos” es una condición externa, una entrada (no una precondition)

Después definid la clase de equivalencia con su ID (un numero secuencial, que se van contando en la misma clase, y después en la siguiente)

- Las clases de equivalencia son o validas o invalidas
- Las definís poniendo el motivo de porque son validas o invalidas, o su naturaleza (nombre que existe, nombre con mas de 6 caracteres, opción existente, calle no existe, no se ingresa calle, email con formato errónea, etc)
- Para las salidas, pones en las CDE validas todas las respuestas que da o debería dar el sistema (recorrido calculado, lista de productos, factura, etc)
- Las invalidas suelen ser mensajes de error → Podes identificar el motivo, PE Mensaje informando que el recorrido no se puede calcular)
- Trata de definir una sola clase de equivalencia VALIDA por cada condición externa → Usuario → Usuario con 6 o mas caracteres y que no existe ya en el sistema (en vez de separarlo)
- De todas formas a veces no se puede, por ejemplo, al seleccionar el tipo de pedido, vas a tener dos CDE validas si o si (comida o producto), que no se pueden hacer las dos juntas
- Casi siempre va el “no se ingresa (esta condición externa)” como CDE invalida
- En la condición de salida de mensaje, en las CDE validas pones el mensaje valido (mensaje que informa que se logro hacer tal cosa exitosamente), en las invalidas, definís cada mensaje de error por cada motivo
- Normalmente hay una Condición externa de salida que es el registro/creación/carga de la operación → Tiene una CDE valida que se registra, y varias invalidas que no se registra por cada motivo

Cuando definís los CP, tienen varios apartados

ID → El numero de la CDE que abarca ese CP (trata de abarcar los mas posibles en cada CP)

- Normalmente, se abarca muchas CDE validas con pocos CP, y las invalidas pueden probarse una con cada CP (pero no es obligatorio)

Prioridad → Va Alta, Media o Baja → El CP de las CDE validas que define el camino exitoso SIEMPRE es alta, las demás fijate

- Por ejemplo algo cosmético o de muy poca importancia es Baja
- Que se meta una contraseña o usuario con un formato erróneo es baja
- Que se meta una cuenta que ya esta cargada es media

Nombre → Tiene que representar el escenario (que sucede en ese CP)

- No es obligatorio que diga exitosa o fracasa en el nombre, pero te tenes que dar cuenta → Elegir un valor que no esta permitido
- La exitosa, la del camino feliz si puede llamarse “Carga de usuario exitosa”
- Pone realmente que estas haciendo → No pongas “poner edad invalida”, pone “Ingresar al sistema poniendo una edad invalida”
- Podes poner la funcionalidad que prueba el CP + la característica que lo hace valido o invalido → Ingresar al sistema + con una edad permitida ; Seleccionar dos calles + que se cruzan

Precondiciones → Todo lo que tiene que existir previamente a la ejecución del CP para asegurarnos que nos de el resultado que se espera

- Es un listado de condiciones
- Hacelo lo mas especifico posible
- Hacelo teniendo en cuenta el CP → PE Si en el CP vas a seleccionar los productos mochila y reloj, pone "Lista de productos cargada y disponible en el sistema, con los productos mochila y reloj disponibles"
- No pongas algo general, como "Tipos de usuarios cargados", pone "Tipos de usuario admin y dba cargados", porque vas a usar esos en el CP
- También puedes especificar cada elemento → "Producto "mochila reebok verde y gris" cargada y disponible
- SIEMPRE especifica todo en detalle → No pongas "Usuario cargado con los permisos correspondientes", especifica "Usuario Lautaro logueado con los permisos de admin"
- Fijate que a menos que sea algo que puede hacer cualquier persona (como crearse una cuenta), una precondición es que el usuario este logueado con los permisos correspondientes

Pasos → Van enumerados dentro del CP

- Son MUY específicos
- Se dice en cada paso quien lo hace, aunque se repita
- También se especifican los valores exactos que se manejan
- No se pone nada general → No pones "Se ingresa una edad mayor a 18", pones "El usuario Pedro ingresa una edad de 34 años"
- No pones "El usuario Juan ingresa un país", pones "El usuario Juan ingresa el país Argentina"

Resultado esperado → Va enumerado (pero puede que no vaya de manera secuencial)

- Representa las salidas que va dando el sistema a las entradas del mismo
- Debe ser muy especifico
- No hace falta que por cada paso haya un paso de resultado esperado, pero puede ser
- De igual forma, a cada resultado esperado le va el numero en el que se genera, respecto del paso → PE si producen salidas los pasos 1 y 5, solo vas a tener los Resultados Esperados 1 y 5
- Por ejemplo → "El sistema muestra la pantalla para la selección de vehículos, con las opciones Sedan, Fiat y Coupe"
- No decís "El sistema muestra los distintos recorridos calculados", decís "El sistema muestra el recorrido mas corto, mas rápido, etc"
- No decís "el sistema muestra las opciones", decís "el sistema muestra la opción 1, la opción 2, etc"
- El resultado esperado también puede ser la respuesta a un error → PE si el paso dice que el usuario ingresa una ciudad "asdasda", el RE sera "El sistema muestra un mensaje informando que la ciudad asdasdas no existe
- A lo mejor tenes un RE que tiene un numero mayor a cualquier paso → PE después de todo lo que hizo el usuario, el sistema registra la compra/registro/pedido (ademas de mostrar el mensaje de éxito, tiene que registrar lo que paso)

Testing de Caja Blanca

- Siempre te conviene representar la funcionalidad/implementación con un diagrama de flujo
- Existen distintos tipos de cobertura

Decisiones → Estructuras de control (if)

Condiciones → Cada parámetro interno que tienen las condiciones

Cobertura de enunciados o caminos básicos

- Representas la funcionalidad con un grafo → El While son dos nodos que van y vienen, hasta que en algún momento, el primer nodo sale por otra flecha (el segundo no sale para ningún otro lado)
- Formula de complejidad →  $M = E - N + 2 * P$
- E → Lineas en el grafo (las que van y vienen cuentan como 2)
- N → Numero de nodos en el grafo
- P → Cantidad de nodos finales (que no salen a ningún lado)

Cobertura de sentencias

- Cantidad mínima de CP que cubran la ejecución de todas las sentencias al menos una vez
- Las sentencias son los rectángulos del diagrama (son una declaración de una variable, la ejecución de un método, todo lo que sea una acción del sistema que no esta en una estructura de control)
- Por mas que se repita una sentencia (que haga exactamente lo mismo) será una sentencia distinta en el grafico
- Cada sentencia entra y sale por un lado

#### Cobertura de decisión

- Cantidad mínima de CP que cubra la ejecución de todas las decisiones, de forma que en cada una se salga al menos una vez por T y al menos una vez por F (sin importar los valores d ellas condiciones)

#### Cobertura de condición

- Cantidad mínima de CP que cubra la ejecución de forma que todas las condiciones tengan alguna vez el valor de T y alguna vez el valor de F (sin importar como sale la decisión)
- Solo se le puede asignar un valor a la condición si se llega a esa decisión

#### Cobertura de decisión/condición

- Cantidad mínima de CP que cubra la ejecución de forma que todas las condiciones tengan alguna vez el valor de T y alguna vez el valor de F, y ejecutando todas las condiciones al menos una vez saliendo por T y al menos una vez saliendo por F

#### Cobertura múltiple

- Cantidad mínima de CP que aseguren que se salio por T y por F en cada decisión de todas las maneras posible (probando todas las combinaciones posibles de las condiciones)
- Puede ser que algún caso sea imposible de ejecutar

#### Tips

- Se deben definir las condiciones, los parámetros que recibirán un valor y definirán como se ejecuta ese caso
- Se arma una tabla y se les asigna valor a estos parámetros
- Si al asignarle valor al primer parámetro, ya se anula la ejecución de todos los demás, no hace falta que se les ponga valor
- Ponele valores bien concretos a los parámetros (siempre respetando las precondiciones)
- Foto = imagen.jpg
- nombre = "Pedro López"
- Si el parámetro es algo que se calcula (por ejemplo la distancia total, o la distancia entre dos puntos) pones directamente su valor
- OJO! → Si el enunciado dice que no se puede cargar una condición sin otra (no se puede cargar un numero de calle sin elegir una calle), tenelo en cuenta para las coberturas (para la de decisión, cuando la calle sea falso (no se eligió) no podes usar ese mismo CP para poner un numero que de falso, porque no lo pudiste elegir, tenes que hacer otro CP con una calle valida y ahí elegís un numero que de falso)