departamento de ingeniería e investigaciones tecnológicas UNLAM

2017

Prueba de Caja Negra

TRABAJO PRÁCTICO N 1

Lucas Secchi, Pablo López, Fabián Norberto Escobar, Micaela Ramos

Programación Avanzada

INTRODUCCIÓN

Las pruebas de caja negra son pruebas funcionales dedicadas a mirar en el exterior de lo que se prueba, en este caso, software. También son conocidas como pruebas de caja opaca, pruebas de entrada/salida, pruebas inducidas por datos. Dichas pruebas se centran principalmente en lo que se quiere para determinar si se atiende a las especificaciones, para solventar problemas a nivel de datos externos.

Estas actividades se limitan a que el tester pruebe con datos de entrada el comportamiento del software y estudie las salidas producidas, sin preocuparse de lo que ocurre en su interior, ya que de eso se tratan las cajas negras: elementos que son estudiados desde el punto de vista de las entradas que recibe y las salidas que produce, desconociendo su funcionamiento interno. Lo que nos interesa de las cajas negras es la forma de interactuar con el medio que las rodea, entendiendo qué es lo que hacen pero sin darle importancia a cómo lo hacen. Por lo tanto, deben estar muy bien definidas sus entradas y salidas, lo que conforma la interfaz.



Suelen centrarse en módulos de interfaz de usuario, como ser pantallas, ficheros, canales de comunicación, pero suelen ser útiles en cualquier módulo ya que todos o la mayoría de los módulos tiene datos de entrada y de salida que pueden ser erróneos, y por lo tanto, es necesario comprobar y verificar. Como cualquier prueba, se apoyan y basan en la especificación de los requisitos y documentación funcional, requisitos que suelen ser más complejos que los requisitos internos, por lo que se realiza una cobertura de especificación para conseguir probar el mayor campo que sea posible.

Las pruebas funcionales se dedican a comprobar que los sistemas informáticos funcionan acorde a las especificaciones y requisitos del cliente, y es de gran ayuda para detectar los posibles defectos derivados de errores en la fase de programación. Los defectos encontrados y solventados durante la fase de pruebas supondrán un ahorro en tiempo y dinero y una reducción en los riesgos. En muchos casos, se hace necesario llevar a cabo pruebas de integración, ya que siempre pueden existir defectos que no se están teniendo en cuenta, y ellos acarrearán problemas ya sea en corto o largo plazo.

Si el sistema se encuentra formado por módulos que cumplan con las características de caja negra, será más sencillo de entender ya que permitirá dar una visión más clara del conjunto. El sistema también será más robusto y fácil de mantener, ya que en caso de ocurrir un fallo, éste podrá ser aislado y abordado ágilmente.

EL PROBLEMA

Dado lo crítico de las tareas que realiza cualquier sistema informático que tenga relación con la atención a pacientes en un hospital y ante la inminente puesta en producción del sistema de gestión de Historias Clínicas, se deberá testear en profundidad el módulo de ordenamiento de historias clínicas de los pacientes.

**Datos de entrada**: Dicho modulo a testear recibe un archivo de texto h\_clinicas.in conteniendo:   
〉 En el primer renglón la cantidad N de historias clínicas a ordenar.   
〉 En los siguientes N renglones los números de historia clínica, Nombre y Apellido separados por un blanco, que serán procesados por el módulo.

**Restricciones**:   
〉 Tenga en cuenta que el hospital atiende a más de 40000 pacientes al año.   
〉 Los números de historia clínica pudieran no ser correlativos.

**Datos de salida**: El archivo de salida hc\_ordenado.out generado contiene:   
〉 N renglones con los datos de los pacientes (números de historia clínica, Nombre y Apellido), ordenados de manera ascendente por número de historia clínica.

EL LOTE DE PRUEBA

**01\_CasoTrivial**

**Descripción:** Todas las historias clínicas están desordenadas.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  10 Nicanor Gonzalez  5 Pedro Picapiedras  7 Pablo Marmol  23 Toribio Tevez  2 Angel Croce  40 Eduardo Sport  55 Alberto Lopez  6 Nadia Nicomedes  9 Abel Avila  15 Victor Tazo  16 Zoilo Zenon | 2 Angel Croce  5 Pedro Picapiedras  6 Nadia Nicomedes  7 Pablo Marmol  9 Abel Avila  10 Nicanor Gonzalez  15 Victor Tazo  16 Zoilo Zenon  23 Toribio Tevez  40 Eduardo Sport  55 Alberto Lopez |

**02\_OrdenadosDescendentemente**

**Descripción:** Todas las historias clínicas están ordenadas de forma descendente.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  55 Alberto Lopez  40 Eduardo Sport  23 Toribio Tevez  16 Zoilo Zenon  15 Victor Tazo  10 Nicanor Gonzalez  9 Abel Avila  7 Pablo Marmol  6 Nadia Nicomedes  5 Pedro Picapiedras  2 Angel Croce | 2 Angel Croce  5 Pedro Picapiedras  6 Nadia Nicomedes  7 Pablo Marmol  9 Abel Avila  10 Nicanor Gonzalez  15 Victor Tazo  16 Zoilo Zenon  23 Toribio Tevez  40 Eduardo Sport  55 Alberto Lopez |

**03\_OrdenadosAscendentemente**

**Descripción:** Todas las historias clínicas están ordenadas de forma ascendente.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  2 Angel Croce  5 Pedro Picapiedras  6 Nadia Nicomedes  7 Pablo Marmol  9 Abel Avila  10 Nicanor Gonzalez  15 Victor Tazo  16 Zoilo Zenon  23 Toribio Tevez  40 Eduardo Sport  55 Alberto Lopez | 2 Angel Croce  5 Pedro Picapiedras  6 Nadia Nicomedes  7 Pablo Marmol  9 Abel Avila  10 Nicanor Gonzalez  15 Victor Tazo  16 Zoilo Zenon  23 Toribio Tevez  40 Eduardo Sport  55 Alberto Lopez |

**04\_HistoriasCorrelativas**

**Descripción:** Los números de las historias clínicas son correlativos entre sí.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 7  1 Nicanor Gonzalez  2 Pedro Picapiedras  3 Pablo Marmol  5 Toribio Tevez  4 Angel Croce  7 Eduardo Sport  6 Alberto Lopez | 1 Nicanor Gonzalez  2 Pedro Picapiedras  3 Pablo Marmol  4 Angel Croce  5 Toribio Tevez  6 Alberto Lopez  7 Eduardo Sport |

**05\_NúmerosDeHCAltos**

**Descripción:** En este caso se tienen números de historia clínica muy altos.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  1 Angel Croce  11111 Pedro Picapiedras  1111 Nadia Nicomedes  11111111 Pablo Marmol  1111111111 Abel Avila  11 Nicanor Gonzalez  111111 Victor Tazo  111 Zoilo Zenon  11111111111 Toribio Tevez  1111111 Eduardo Sport  111111111 Alberto Lopez | 1 Angel Croce  11 Nicanor Gonzalez  111 Zoilo Zenon  1111 Nadia Nicomedes  11111 Pedro Picapiedras  111111 Victor Tazo  1111111 Eduardo Sport  11111111 Pablo Marmol  111111111 Alberto Lopez  1111111111 Abel Avila  11111111111 Toribio Tevez |

**06\_UnaHistoriaClínica**

**Descripción:** Hay un único número de historia clínica.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 1  5 Angel Croce | 5 Angel Croce |

**07\_NombresRepetidos**

**Descripción:** En este caso los nombres de los pacientes están repetidos.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 7  1 Nicanor Gonzalez  22 Nicanor Picapiedras  3 Nicanor Marmol  54 Nicanor Tevez  4 Nicanor Croce  71 Nicanor Sport  6 Nicanor Lopez | 1 Nicanor Gonzalez  3 Nicanor Marmol  4 Nicanor Croce  6 Nicanor Lopez  22 Nicanor Picapiedras  54 Nicanor Tevez  71 Nicanor Sport |

**08\_ApellidosRepetidos**

**Descripción:** En este caso los apellidos de los pacientes están repetidos.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  10 Nicanor Picapiedras  5 Pedro Picapiedras  7 Pablo Picapiedras  23 Toribio Picapiedras  2 Angel Picapiedras  40 Eduardo Picapiedras  55 Alberto Picapiedras  6 Nadia Picapiedras  9 Abel Picapiedras  15 Victor Picapiedras  16 Zoilo Picapiedras | 2 Angel Croce  5 Pedro Picapiedras  6 Nadia Picapiedras  7 Pablo Picapiedras  9 Abel Picapiedras  10 Nicanor Picapiedras  15 Victor Picapiedras  16 Zoilo Picapiedras  23 Toribio Picapiedras  40 Eduardo Picapiedras  55 Alberto Picapiedras |

**09\_Nombres&ApellidosRepetidos**

**Descripción:** En este caso los nombres y apellidos de los pacientes están repetidos.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| 11  10 Nicanor Picapiedras  5 Nicanor Picapiedras  7 Nicanor Picapiedras  23 Nicanor Picapiedras  2 Nicanor Picapiedras  40 Nicanor Picapiedras  55 Nicanor Picapiedras  6 Nicanor Picapiedras  9 Nicanor Picapiedras  15 Nicanor Picapiedras  16 Nicanor Picapiedras | 2 Nicanor Picapiedras  5 Nicanor Picapiedras  6 Nicanor Picapiedras  7 Nicanor Picapiedras  9 Nicanor Picapiedras  10 Nicanor Picapiedras  15 Nicanor Picapiedras  16 Nicanor Picapiedras  23 Nicanor Picapiedras  40 Nicanor Picapiedras  55 Nicanor Picapiedras |

**10\_FatigaRandom**

**Descripción:** Se prueba que el módulo ordene 4000 registros con números al azar

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| Hecho con generador |  |

**11\_FatigaOrdenDescendente**

**Descripción:** Se prueba que el módulo ordene de forma ascendente 4000 registros ordenados de forma descendente.

|  |  |
| --- | --- |
| **h\_clinicas.in** | **hc\_ordenado.out** |
| Hecho con generador (ejemplo)  4000 a a  3999 b b  3998 c c  .  .  . | 1 a a  2 b b  3 c c  4 d d  .  .  . |

DESARROLLO

Se cuenta con ocho archivos ejecutables (hClinicas\_1.exe, hClinicas\_2.exe, hClinicas\_3.exe, hClinicas\_4.exe, hClinicas\_5.exe, hClinicas\_6.exe, hClinicas\_7.exe, hClinicas\_8.exe) que automatizan la resolución del problema, cada uno con una implementación diferente, la cual nos resulta desconocida y no nos interesa para realizar las pruebas, y con los casos de prueba comentados en la sección anterior.

Se cuenta también con los archivos de entrada y de salida, separados en carpetas por el nombre de caso, y con la documentación del lote de prueba.

Se procede a analizarlos de a uno por vez, ejecutándolo con el archivo de entrada de cada uno de los casos de prueba, donde se produce una salida que es comparada con el archivo de salida esperado. Si coinciden las salidas, entonces el ejecutable funciona correctamente con ese caso de prueba.

Una vez finalizadas todas las pruebas con un ejecutable, se pasa al siguiente. Se anotará cada falla en el informe de fallas por cada ejecutable y luego se redactarán las conclusiones de la prueba de esta prueba de caja negra.

Si durante el análisis de los ejecutables se considera que es necesario realizar nuevos casos de prueba para el lote, así se hará y se comentará de su existencia en el informe de fallas.

INFORME DE FALLAS

**.hClinicas\_1.exe:**

Caso01: Ordena correctamente los números de historia, sin embargo pisa algunos nombres con otros.

Caso02: Ordena los números pero todos los nombres los pisa con el primer ordenado

Caso03: FUNCIONA

Caso04: Ordena los números, pero pisa los nombres ANTERIORES que tengan un número mayor de HC.

Caso05: Ordena los números, sigue con los mismos errores y por alguna razón toma números de 10 dígitos y no de 7 y 9, pero sí de 8. Int max: 2,147,483,647

Caso06: FUNCIONA

Caso07: Misma observación que en el caso 4

Caso08: Misma observación que en el caso 4

Caso09: Parecería que anda bien pero tiene el mismo error encontrado antes.

Observaciones hasta el momento: Depende de que tan bien ordenado este el archivo de entrada, si esta ordenado al revés cambia el nombre de todos con el nombre del primero

**.hClinicas\_2.exe:**

Caso01: Ordena todos bien EXCEPTO el último registro.

Caso02: Ordena todos bien EXCEPTO el penúltimo registro.

Caso03: FUNCIONA

Caso04: FUNCIONA

Caso05: Ordena los números, sigue con los mismos errores y por alguna razón toma números de 10 dígitos y no de 7 y 9, pero sí de 8. Int max: 2,147,483,647

Caso06: FUNCIONA

Caso07: FUNCIONA (pero sigue con los mismos errores si lo cambiamos)

Caso08: IDEM caso01

Caso09: IDEM

**.hClinicas\_3.exe:**

Caso01: FUNCIONA

Caso02: FUNCIONA

Caso03: FUNCIONA

Caso04: FUNCIONA

Caso05: Está hecho con short evidentemente.

Caso06: FUNCIONA

Caso07: FUNCIONA

Caso08: FUNCIONA

Caso09: FUNCIONA

**.hClinicas\_4.exe:**

Caso01: FUNCIONA

Caso02: FUNCIONA

Caso03: FUNCIONA

Caso04: FUNCIONA

Caso05: Ordena los números, sigue con los mismos errores y por alguna razón toma números de 10 dígitos y no de 7 y 9, pero sí de 8. Int max: 2,147,483,647

Caso06: FUNCIONA

Caso07: FUNCIONA

Caso08: FUNCIONA

Caso09: FUNCIONA

**.hClinicas\_5.exe:**

Caso01: ROMPE

Caso02: ROMPE

Caso03: FUNCIONA (ya que ya está ordenado, se sospecha que rompe al intentar hacer intercambios)

Caso04: FUNCIONA

Caso05: Ordena los números, sigue con los mismos errores y por alguna razón toma números de 10 dígitos y no de 7 y 9, pero sí de 8. Int max: 2,147,483,647

Caso06: FUNCIONA

Caso07: FUNCIONA

Caso08: ROMPE

Caso09: ROMPE

**.hClinicas\_6.exe:**

Caso01: FUNCIONA

Caso02: ROMPE

Caso03: FUNCIONA (ya que ya está ordenado, se sospecha que rompe al intentar hacer intercambios)

Caso04: FUNCIONA

Caso05: Ordena los números, sigue con los mismos errores y por alguna razón toma números de 10 dígitos y no de 7 y 9, pero sí de 8. Int max: 2,147,483,647

Caso06: FUNCIONA

Caso07: FUNCIONA

Caso08: ROMPE

Caso09: ROMPE

**.hClinicas\_7.exe:**

**.hClinicas\_8.exe:**

CONCLUSIONES

**.Conclusión del hClinicas\_1.exe:**

El ejecutable ordena bien los números, sin embargo al tomar un registro, pisa los nombres y apellidos de los registros anteriores que tengan un número de historia clínica MENOR a ese registro, y lo hace en el primer intercambio que se aplique a este registro, si llegara a hacerse otro intercambio conserva el nombre y apellido que tenga en ese momento.

**.Conclusión del hClinicas\_2.exe:**

El programa empieza ordenando los dos últimos registros (es decir, verifica si tiene que hacer o no un intercambio entre esos dos), lo cual provoca que algunas veces el último registro quede mal ordenado ya que, independientemente de si realiza el intercambio o no, marca al último registro como ordenado y no lo vuelve a mirar.

**.Conclusión del hClinicas\_3.exe:**

Funciona correctamente. Utiliza el tipo de dato short int para almacenar los números de historia clínica, por lo que solo llega a almacenar números hasta el 32,767

**.Conclusión del hClinicas\_4.exe:**

Funciona correctamente. Utiliza el tipo de dato int para almacenar los números de historia clínica, por lo que solo llega a almacenar números hasta el 2,147,483,647. Si un número supera el rango de int, todos los registros de abajo se ven afectados.

**.Conclusión del hClinicas\_5.exe:**

**NO SE, ROMPE CON TODO ESTÁ RE MAL HECHO SEGURO LO HIZO SECCHI**

**LO IMPLEMENTÓ CON BOGOSORT**

**.Conclusión del hClinicas\_6.exe:**

Rompe cuando el mínimo se encuentra en la última posición. Se podría decir que

**.Conclusión del hClinicas\_7.exe:**

El programa realiza la lectura de los campos de cada registro como char – string – string, por lo que solo toma la primera cifra de los números. En otras palabras, no es capaz de generar una salida correcta si los números de historia clínica tienen más de una cifra.

**.Conclusión del hClinicas\_8.exe:**

El ejecutable no realiza intercambios con el primer registro. Todos los demás quedan bien ordenados.

**.Casos de fatiga:**

Ningún ejecutable llego al resultado esperado con ningún caso de fatiga (números de historia aleatorios y números de historia ordenados de forma descendente).