**Casos de Prueba**

|  |  |
| --- | --- |
| **Proyecto No. 7** | **Caso No. CPI-01** |
| **Nombre del Proyecto:** ChangeCounter | **Nombre del Caso de Uso:** Integración módulos del sistema |
| **Escrito por:** Gladys Mireya Castro, Néstor Cruz | **Fecha:** 25/03/2013 |
| **Responsable Ejecución:** | |

## Caso 9:

Descripción del caso de prueba:

El sistema debe funcionar adecuadamente después de realizarse la integración de todos sus módulos.

pre-requisitos de prueba:

* Entrega, y correcto funcionamiento de cada uno de los módulos que componen el sistema

*ERRORES POSIBLES:*

* *Error al generar la compilación tras haber integrado los módulos del sistema*
* *Error en alguna de las funcionalidades causada por la misma integración*
* *Error en la secuencia lógica de pasos*
* *Error al ingresar los archivos*
* *Error al realizar las comparaciones*
* *Error al generar las etiquetas*
* *Error al identificar las líneas de código agregadas y borradas*

|  |  |  |  |
| --- | --- | --- | --- |
| Paso | Acción | Resultados Esperados | Criterios de Aceptación |
| 1 | Ingresar la versión anterior del programa | El programa permite ingresar adecuadamente la versión anterior del programa |  |
| 2 | Ingresar la versión posterior del programa | El programa permite ingresar adecuadamente la versión posterior del programa |  |
| 3 | Solicitud de información por parte del sistema | El sistema solicita al usuario que se ingrese la información correspondiente |  |
| 4 | Ingresar información en el aplicativo | El sistema valida correctamente la información ingresada |  |
| 5 | Analizar archivos de ambas versiones | El sistema analiza sin excepciones todos los archivos de ambas versiones |  |
| 6 | Generar registros de cambios de versión | El sistema genera un registro de los cambios realizados |  |
| 7 | Visualización adecuada de mensajes en caso de error | En caso de que no existan clases .java o de que ocurra un erro, se debe visualizar un mensaje definido |  |
| 8 | Recorrer las líneas de código | El sistema recorre todas las líneas de código de ambos archivos |  |
| 9 | Identificar líneas agregadas y borradas | El sistema reconoce adecuadamente las líneas borradas y las líneas adicionas entra las versiones del programa ingresado |  |
| 10 | Llevar registro | El sistema lleva el registro completo de todas las LOC adicionadas y borradas |  |
| 11 | Recorrer las líneas de código | El sistema recorre todas las líneas de código de ambos archivos |  |
| 12 | Contar LOC agregados | El sistema cuenta la cantidad de LOC´S agregados por cada archivo |  |
| 13 | Contar LOC borrados | El sistema cuenta la cantidad de LOC´S borrados por cada archivo |  |
| 14 | Contar LOC totales | El sistema cuenta la cantidad de LOC´S totales por cada archivo |  |
| 15 | Indicar LOC´S | El sistema indica al usuario la cantidad total de LOC´S agregados y borrados |  |
| 16 | Generar número | El sistema genera un numero de cambio del programa |  |
| 17 | Agregar etiqueta | El sistema agrega etiqueta de la línea borrada con el número de cambio generado por el sistema |  |
| 18 | Adiciona el numero del cambio | El número de cambio fue adicionado en el comentario de la cabecera del programa | El número debe estar adicionado en el comentario de la cabecera del programa. |
| 19 | Adiciona la fecha del cambio | La fecha de cambio fue adicionado en el comentario de la cabecera del programa | La fecha debe estar adicionada en el comentario de la cabecera del programa. |
| 20 | Adiciona la persona que realizo el cambio | La persona que realizó el cambio fue adicionada en el comentario de la cabecera del programa | La persona que realizó el cambio debe estar adicionada en el comentario de la cabecera del programa. |
| 21 | Adiciona el por qué se realizó el cambio | El por qué se realizó el cambio fue adicionado en el comentario de la cabecera del programa | El por qué se realizó el cambio debe estar adicionado en el comentario de la cabecera del programa. |
| 22 | Verifica si existen | Valoración de la verificación de existencia de comentarios previos | El LOC adicionadas debe estar adicionado en el comentario de la cabecera del programa. |
| 23 | Verifica y almacena los cambios encontrados | Los cambios deben haber sido verificados y almacenados | El LOC cambiadas debe estar adicionado en el comentario de la cabecera del programa. |
| 24 | La información está en orden | La información está en orden en el comentario del encabezado del programa | El LOC modificadas debe estar adicionado en el comentario de la cabecera del programa. |
| 25 | Toma toda la información previamente almacena | Haber recuperado la información del comentario del encabezado |  |
| 26 | Toma la información almacenada en las líneas con cambios del programa | El toma toda la información previamente almacenada en las líneas con cambios del programa |  |
| 27 | Guarda contenido del programa | El sistema guarda todo el contenido del programa junto con la información con los cambios previamente analizados | Programa guardado con la información histórica de cambios. |
| 28 | Guarda un nuevo archivo | El sistema guarda en un nuevo archivo toda la información | Archivo generado con todo la información. Si se produce fallo en la salida del archivo se debe notificar al usuario. |