PRUEBA DEL SOFTWARE

Dividiremos las pruebas en tres etapas:

* Pruebas de unidad.
* Pruebas de integración.
* Pruebas del sistema

Pruebas de Unidad.

Las pruebas de unidad se enfocan en probar el software con diferentes pruebas unitarias. Una **prueba unitaria** es una forma de comprobar el correcto funcionamiento de una unidad de código. Se pueden enfocar de diferentes formas:

1. Caja Negra
2. Caja Blanca
3. Conjetura de errores
4. Enfoque aleatorio

**Caja Negra:**

Se centra en los requisitos funcionales del software, se usaran datos de entrada y se esperará una salida conforme a los requisitos expuestos para el software.

Usaremos la técnica de Partición de clases de equivalencia.

Veremos diferentes casos de prueba a la hora de añadir contactos:

|  |  |  |  |
| --- | --- | --- | --- |
| Casos de Prueba | Entrada | Dato | Salida |
| CP1 | Nombre |  | Correcto |
| CP2 | Nombre | David | Correcto |
| CP3 | Nombre | David | Incorrecto |
| CP4 | Teléfono | 913456879 | Correcto |
| CP5 | Teléfono | 65897f797789 | Incorrecto |
| CP6 | Teléfono |  | Incorrecto |
| CP7 | Teléfono | 913456879 | Correcto |

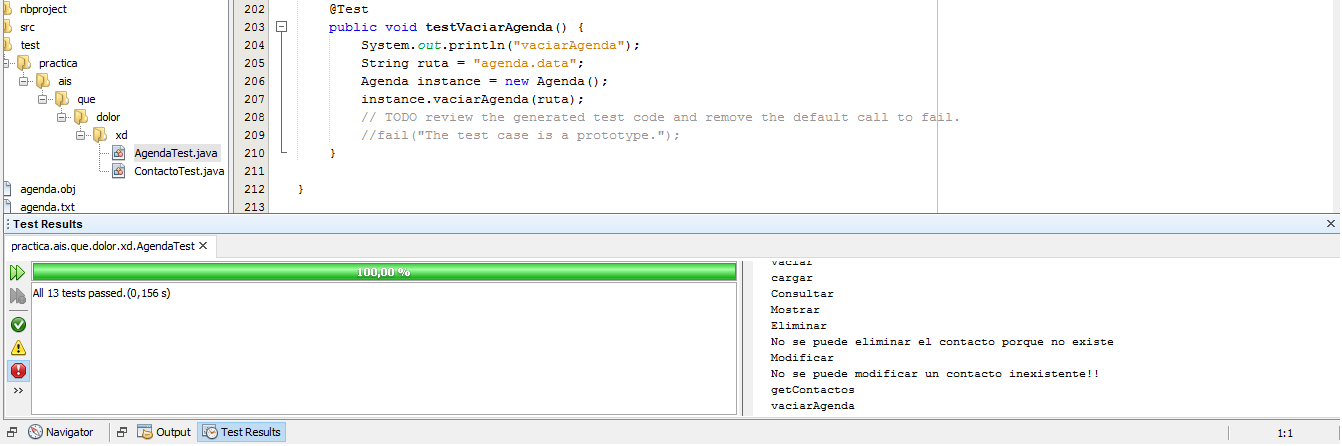
No se aceptan números de teléfono menores de tres cifras:

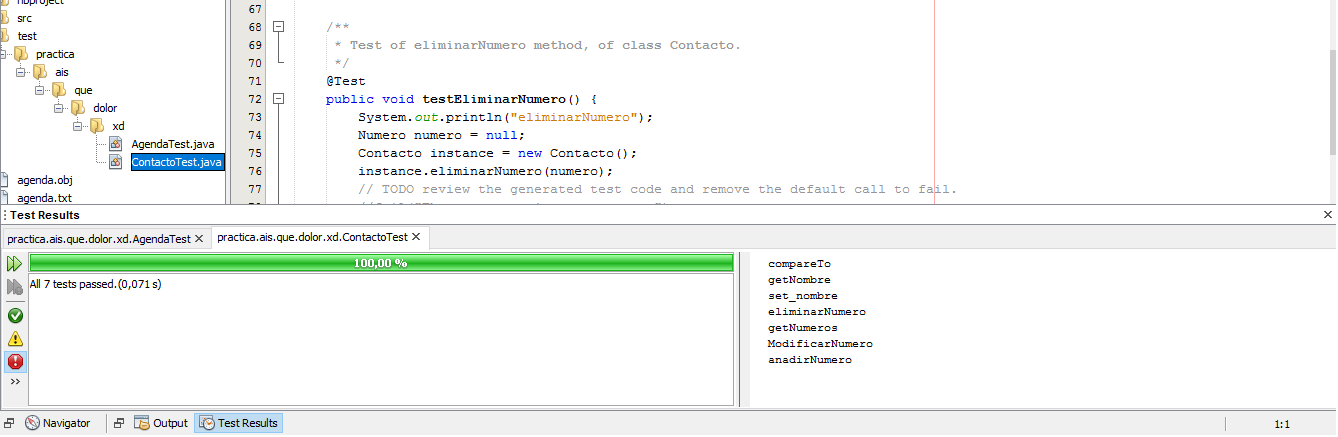
|  |  |  |  |
| --- | --- | --- | --- |
| Casos de Prueba | Entrada | Dato | Salida |
| CP1 | Teléfono | 658978897 | Correcto |
| CP2 | Teléfono | 12 | Incorrecto |
| CP3 | Teléfono | 112 | Correcto |

Casos de Prueba de añadir contacto completos:

|  |  |  |  |
| --- | --- | --- | --- |
| Casos de Prueba | Nombre | Teléfono | Salida |
| CP1 | David | 913456879 | Correcto |
| CP2 |  | 653456879 | Correcto |
| CP3 | Jaime | 913456879 | Incorrecto |
| CP4 |  |  | Incorrecto |
| CP5 | Rubén | 12 | Incorrecto |
| CP6 | Emergencias | 112 | Correcto |
| CP7 | Yoel |  | Incorrecto |

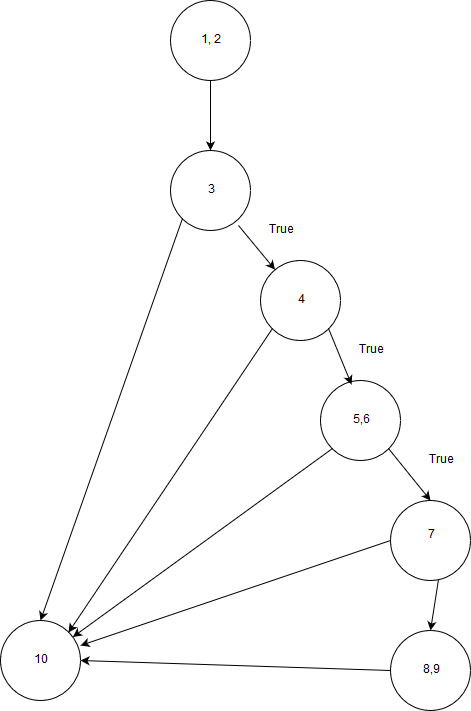
Otra de las formas para asegura el buen funcionamiento de los métodos/funciones de nuestro software es usar pruebas automáticas. En este caso hemos usado los Junit para probar todos los métodos. Dejamos las capturas para mostrar cómo pasan dichos test, esto no quiere decir que no tengan fallos, solo que para nuestras pruebas automatizadas los métodos funcionan correctamente.





**Caja Blanca:**

Usaremos el camino básico representado por un grafo de flujo:



Puntos del Grafo:

1. Código del botón añadir contacto.
2. Ventana de la interfaz de añadir contacto.
3. Comprobar si los parámetros no son vacíos.
4. Comprobar que el número tiene más de tres letras.
5. Crear y añadir número.
6. Código de la agenda.
7. Verificar que no existe otro contacto igual.
8. Añadir el contacto.
9. Guardar el fichero.
10. Fin de añadir

**PRUEBAS DE INTEGRACIÓN**

En este caso se probarán los módulos del programa poco a poco, detectando fallos en cada uno, esto nos permite localizar donde se encuentras estos fallos.

Por ejemplo si nos falla a la hora de mostrar los contactos que han sido agregados, empezaremos por probar a agregar un contacto en la lista y verificar que se ha creado correctamente, iremos a probar el modulo que se ocupa en mostrar por pantalla que se ejecuta correctamente. Si en alguno de los módulos encontramos algún error se subsana dicho error, de otra forma si se intenta buscar en la totalidad del código, se hace muy laborioso y difícil encontrar donde puede estar dando el error. Este modelo de pruebas tiene semejanza con la integración ascendente que se basa en probar primero los módulos atómicos e ir subiendo a los módulos más complejos.

**PRUEBAS DEL SISTEMA**

Usaremos este tipo de pruebas para el manejo de errores y los posibles fallos en la interfaz.

Las pruebas de recuperación son muy importantes, ya que si se introducen datos que no son los correctos, nuestra aplicación puedas recuperarse de dicho fallo y seguir su funcionamiento. Para dichas pruebas hemos listado los posibles fallos que se pueden introducir en el software y probar que la aplicación se recupera de ellos:

* Añadir dos nombres iguales
* Números de teléfono con menos de 3 cifras
* Eliminar contactos cuando no hay ninguno en la agenda
* Añadir letras en el número (Esto creo que falla en nuestra aplicación)
* Si se te ocurre alguna más.

Otra de las partes importantes de las pruebas del Sistema es comprobar los posibles errores en la interfaz y el correcto funcionamiento de ella:

CUANDO ESTE LA INTERFAZ SACAR UNAS CAPTURAS.