

Pruebas unitarias

Testing en GeneXus

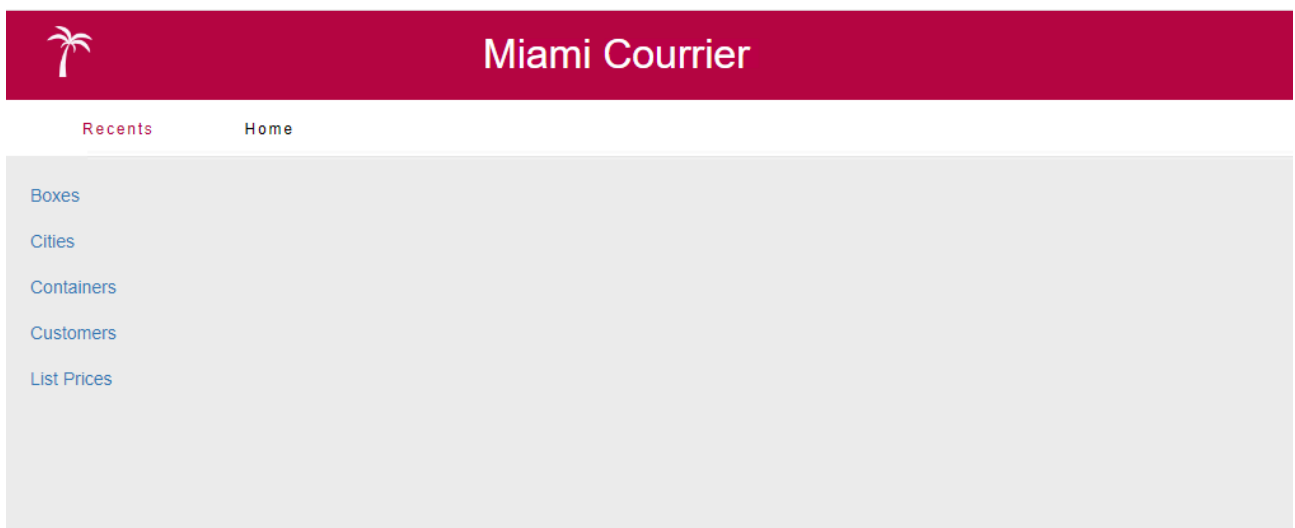
Copyright ♥ Abstracta

All rights reserved. This document may not be reproduced by any means without the express permission of Abstracta. The information contained herein is intended for personal use only

Introducción

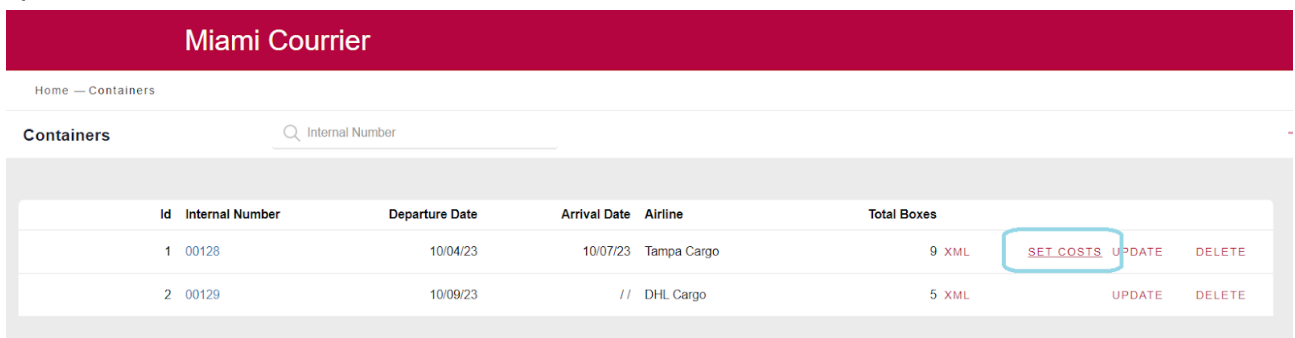
Durante el taller hands-on de GXtest vas a introducirte a la creación y ejecución de pruebas unitarias, pruebas de interfaz y suite de pruebas en GeneXus.

Durante el taller vamos a trabajar sobre la aplicación Miami Courier. Esta KB fue creada con fines de capacitación. La misma contiene un panel lateral para gestionar las Cajas, Contenedores, Clientes y Listas de precios:



Vamos a automatizar la funcionalidad “**Establecer el costo de las cajas de un contenedor**” a nivel unitario y a nivel de interfaz.

Esta funcionalidad se puede visualizar implementada en el panel de **Containers** de la aplicación.



Al hacer clic en el *link* SET COSTS se setea el costo de las cajas asociadas al contenedor. **No haga clic en SET COSTS**, ya que alteraría el estado de la base de datos de la aplicación.

1 - Introducción a pruebas unitarias

Como comentamos anteriormente, al hacer clic en el *link* SET COSTS se setea el costo de las cajas asociadas al contenedor. Esto se hace invocando a un procedimiento llamado “*SetCostToBoxesInContainer*”. Automatizamos este procedimiento para verificar su funcionamiento siempre que liberemos una nueva versión de la aplicación.

Ejecutaremos tres escenarios diferentes sobre de *Unit Test*:

Caso 1- Unit test pasa: ejecutaremos un unit tests con valores correctos para lograr la ejecución del test de forma exitosa.

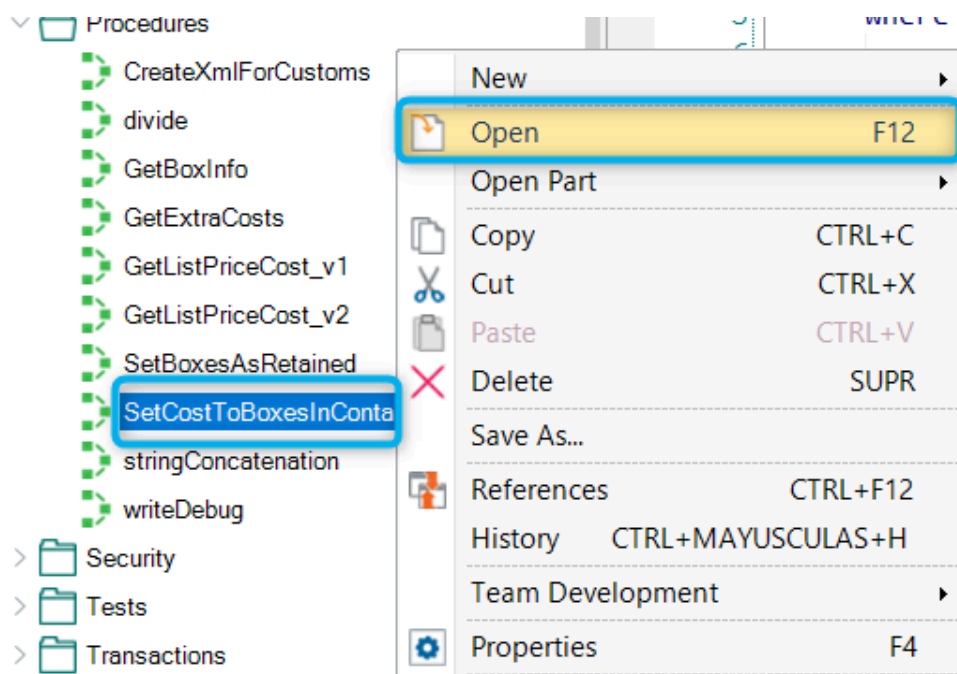
Caso 2 - Unit test falla: con datos de prueba incorrectos para descubrir qué sucede cuando falla un test.

Caso 3 - Validación de base de datos: agregaremos a la prueba unitaria una validación del estado de la base de datos.

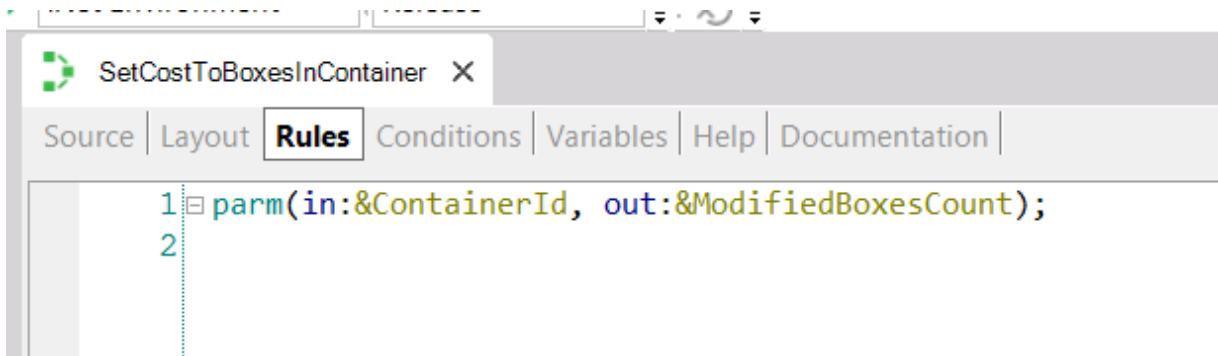
Implementación de Unit Tests

En esta sección le guiaremos para que pueda ejecutar distintos escenarios de prueba unitaria. La funcionalidad **Establecer el costo de las cajas de un contenedor**, está implementada por el procedimiento “*SetCostToBoxesInContainer*” de nuestra KB. ¡Veamos el procedimiento!

1) Abrir el procedimiento haciendo clic derecho sobre él y seleccionando “Open” o simplemente haciendo doble clic sobre el mismo.



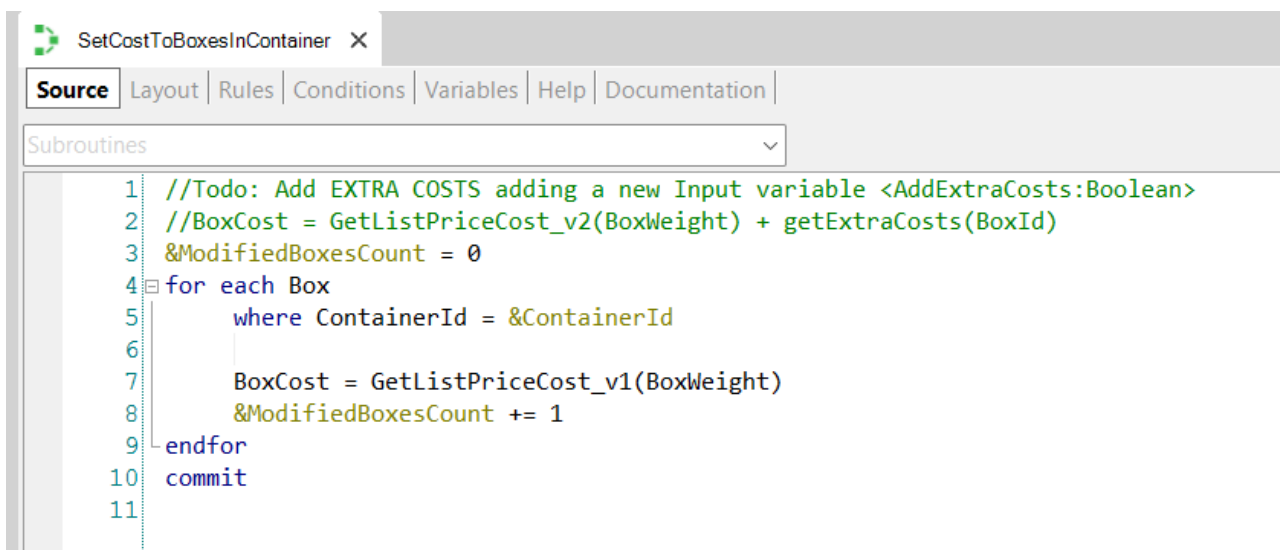
2) Haga clic en la pestaña Rules. Podrá ver que el procedimiento recibe como parámetro de entrada *&ContainerId* (id del contenedor al cual le va a establecer el costo de las cajas) y devuelve como parámetro de salida *&ModifiedBoxesCount* (cantidad de cajas modificadas):



The screenshot shows a window titled 'SetCostToBoxesInContainer' with a tab labeled 'Rules'. The code in the editor is as follows:

```
1 parm(in:&ContainerId, out:&ModifiedBoxesCount);  
2
```

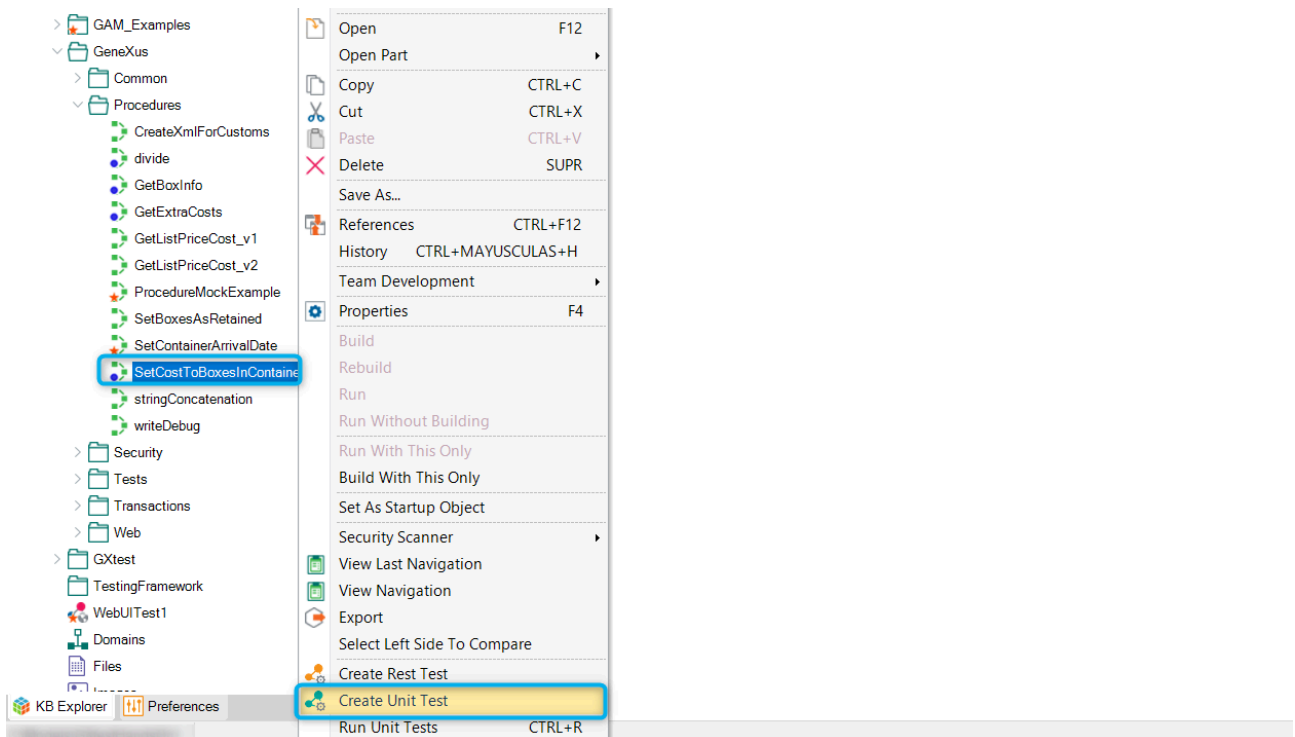
3) Hacer clic en la pestaña Source. Podrá observar, en la lógica del procedimiento, que el proc asigna el costo de cada caja del contenedor cuyo id coincide con el parámetro de entrada *&ContainerId*:



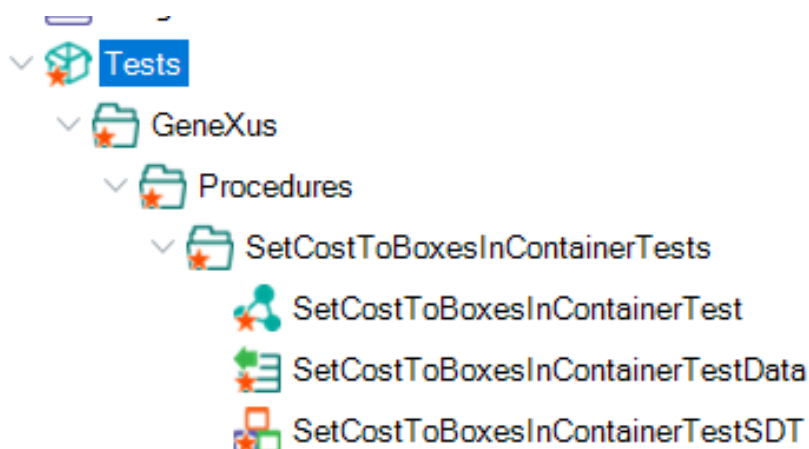
The screenshot shows the same window with the 'Source' tab selected. The code in the editor is as follows:

```
1 //Todo: Add EXTRA COSTS adding a new Input variable <AddExtraCosts:Boolean>  
2 //BoxCost = GetListPriceCost_v2(BoxWeight) + getExtraCosts(BoxId)  
3 &ModifiedBoxesCount = 0  
4 for each Box  
5     where ContainerId = &ContainerId  
6  
7     BoxCost = GetListPriceCost_v1(BoxWeight)  
8     &ModifiedBoxesCount += 1  
9 endfor  
10 commit  
11
```

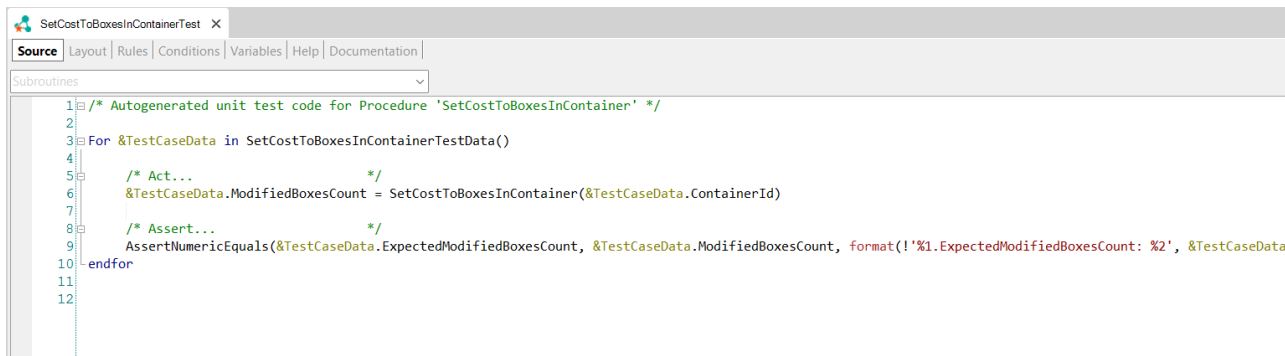
4) Seleccionar el procedimiento “SetCostToBoxesInContainer” en el panel *KB Explorer*, hacer clic derecho y elegir la opción “Create Unit Test”.



5) Espere hasta que se generen los objetos del test “SetCostToBoxesInContainerTest”, “SetCostToBoxesInContainerTestData” y “SetCostToBoxesInContainerTestSDT” que aparecerán en el módulo “Tests”, dentro de la carpeta con el nombre del Proc “SetCostToBoxesInContainerTests”



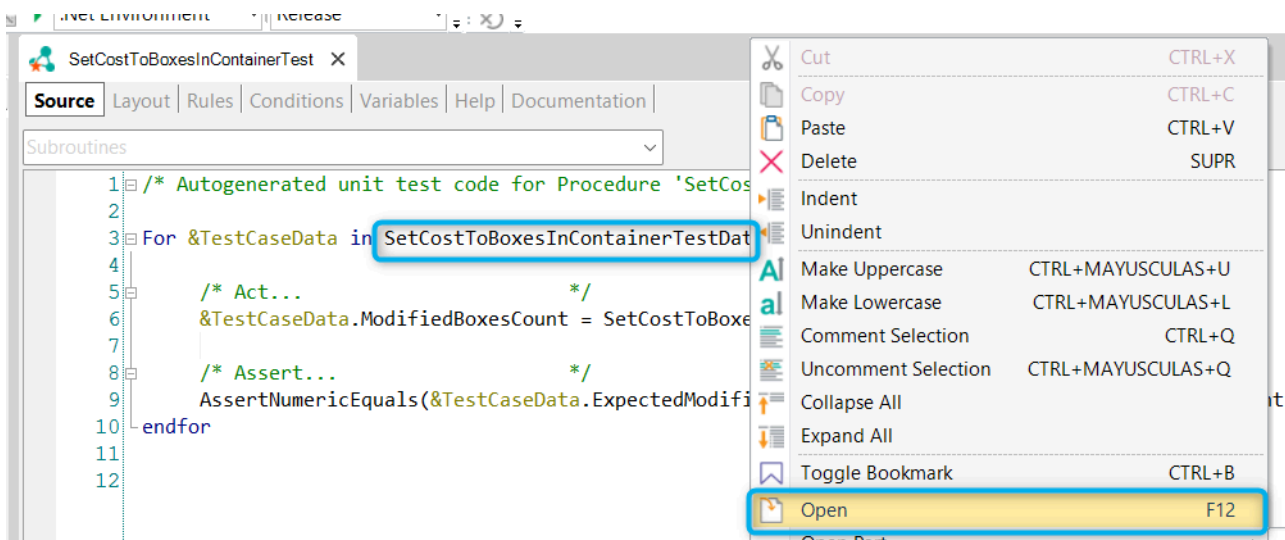
- 6) Como puede observar, GXtest genera automáticamente un template de código GeneXus iterando sobre el Data Provider “SetCostToBoxesInContainerTestData”, el cual contiene los juegos de datos que el test ejecutará.



```
1 /* Autogenerated unit test code for Procedure 'SetCostToBoxesInContainer' */
2
3 For &TestCaseData in SetCostToBoxesInContainerTestData()
4
5     /* Act... */
6     &TestCaseData.ModifiedBoxesCount = SetCostToBoxesInContainer(&TestCaseData.ContainerId)
7
8     /* Assert... */
9     AssertNumericEquals(&TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount, format('!%1.ExpectedModifiedBoxesCount: %2', &TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount))
10 endfor
```

Caso 1: Unit Test pasa

- 7) Hacer clic derecho sobre el objeto “SetCostToBoxesInContainerTestData” y seleccionar “Open”.

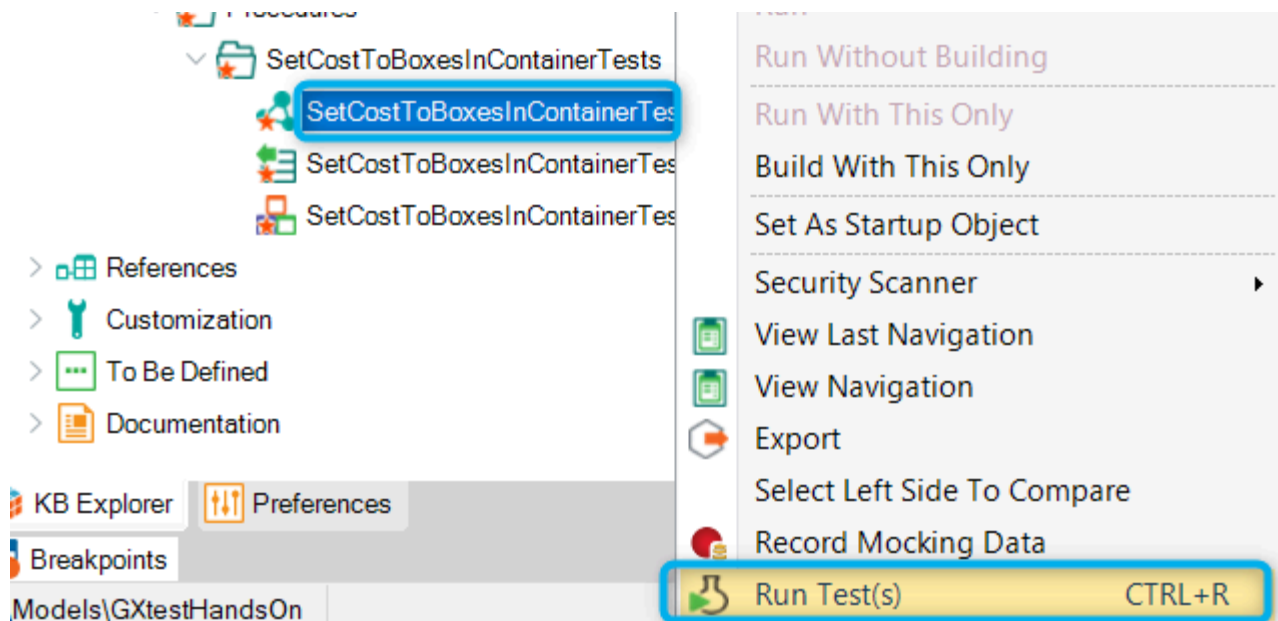


- 8) Ingresar en el objeto SetCostToBoxesInContainerTestData los valores **ContainerId = 1**, **ExpectedModifiedBoxesCount = 9** y **MsgModifiedBoxesCount = '9'**. Con estos datos probaremos si el costo de las 9 cajas del contenedor 1 se setea exitosamente.

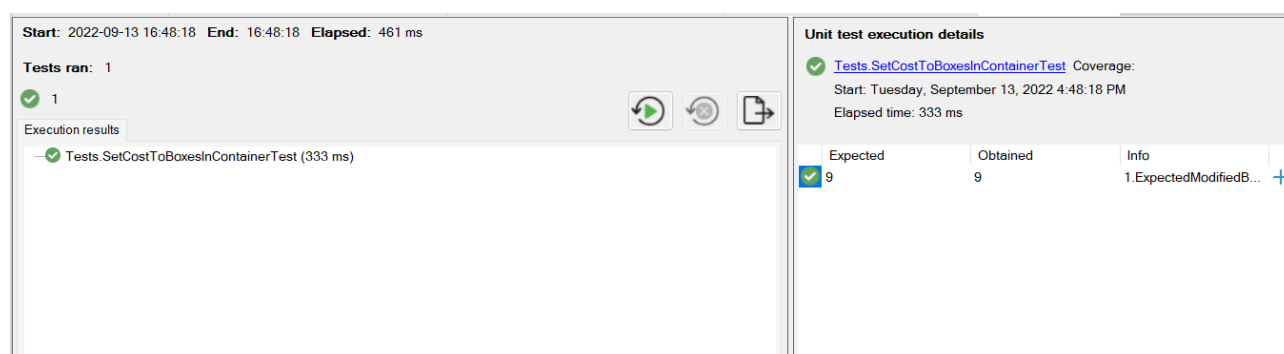
El código del Data Provider es:

```
SetCostToBoxesInContainerTestSDT
{
    TestCaseld = '1'
    ContainerId = 1
    ExpectedModifiedBoxesCount = 9
    MsgModifiedBoxesCount = '9'
}
```

- 9) Guardar los cambios, hacer clic derecho sobre el unit test y seleccionar la opción “Run Test(s)”.



- 10) Al terminar la ejecución del *test*, se mostrarán los resultados en el panel *Tests Results*. Vemos que el test pasa ya que el número esperado de cajas modificadas es igual al obtenido.



Caso 2: Unit Test falla

Para ejecutar un test que se espera que falle utilizaremos como caso de prueba el contenedor 2. Sabemos que el contenedor 2 tiene 5 cajas, pero ejecutaremos el caso indicando que esperamos que el procedimiento modifique el costo de 4 cajas para hacerlo fallar.

Por favor, intente agregar el caso de prueba en el dataprovider 'SetCostToBoxesInContainerTestData', y ejecute el test seleccionando la opción "Run Test".

El test deberá fallar, indicando las diferencias entre el resultado obtenido y el esperado.

SOLUCIÓN:

- 11) Abrir el objeto 'SetCostToBoxesInContainerTestData'
- 12) Modificar los valores actuales seteando: **ExpectedModifiedBoxesCount = 4** y **MsgModifiedBoxesCount = 'Contenedor 2 con 4 cajas'**.

El código del Data Provider debería ser:

```
SetCostToBoxesInContainerTestSDT
{
    TestCaseId = '1'
    ContainerId = 2
    ExpectedModifiedBoxesCount = 4
    MsgModifiedBoxesCount = 'Contenedor 2 con 4 cajas'
}
```

- 13) Guardar los cambios y ejecutar el test seleccionando la opción "Run Test(s)".
- 14) Observar que al terminar la ejecución del test, se mostrarán los resultados en el panel Tests Results, indicando el fallo y la diferencia entre el resultado obtenido y el resultado esperado.

The screenshot displays the 'Tests Results' window. On the left, the 'Tests ran' section shows 1 test. Below it, the 'Execution results' section shows a failed test: 'Tests.SetCostToBoxesInContainerTest (796 ms)' with a red 'X' icon. On the right, the 'Unit test execution details' panel provides more information. It shows the test name 'Tests.SetCostToBoxesInContainerTest' with a red 'X' icon, coverage status, start time (Wednesday, October 4, 2023 10:52:13 AM), and elapsed time (796 ms). Below this, a table compares 'Expected' and 'Obtained' values:

	Expected	Obtained	Info
	4	5	1.ExpectedModifiedBoxesCount: Contenedor 2 con 4 cajas

A 'Set as expected' button is located at the bottom right of the details panel.

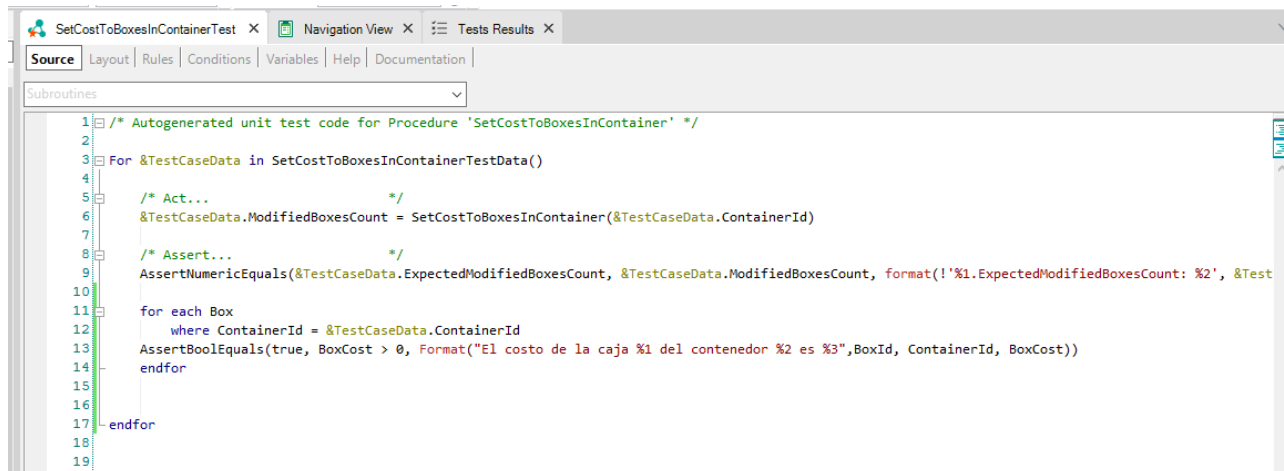
Caso 3: Validación del estado de la base de datos

Validemos ahora que, cuando el procedimiento es ejecutado, la base de datos efectivamente cambia de estado. Para esto, agregaremos una validación (assertion) sobre el estado de la base de datos. En este caso, la validación chequeará si el costo de cada caja del contenedor tiene un valor mayor a 0.

- 1) Abrir el objeto 'SetCostToBoxesInContainerTest'.
- 2) Agregar la siguiente validación debajo del assertion ya existente:

```
/* Assert del estado de la base de datos. */  
  
for each Box  
  
    where ContainerId = &TestCaseData.ContainerId  
  
    AssertBoolEquals(true, BoxCost > 0, Format("El costo de la caja %1 del contenedor %2 es %3",BoxId,  
    ContainerId, BoxCost))  
  
endfor
```

El unit test le quedará con dos aserciones, una sobre el parametro de salida y otra sobre los valores de los atributos en la base de datos:



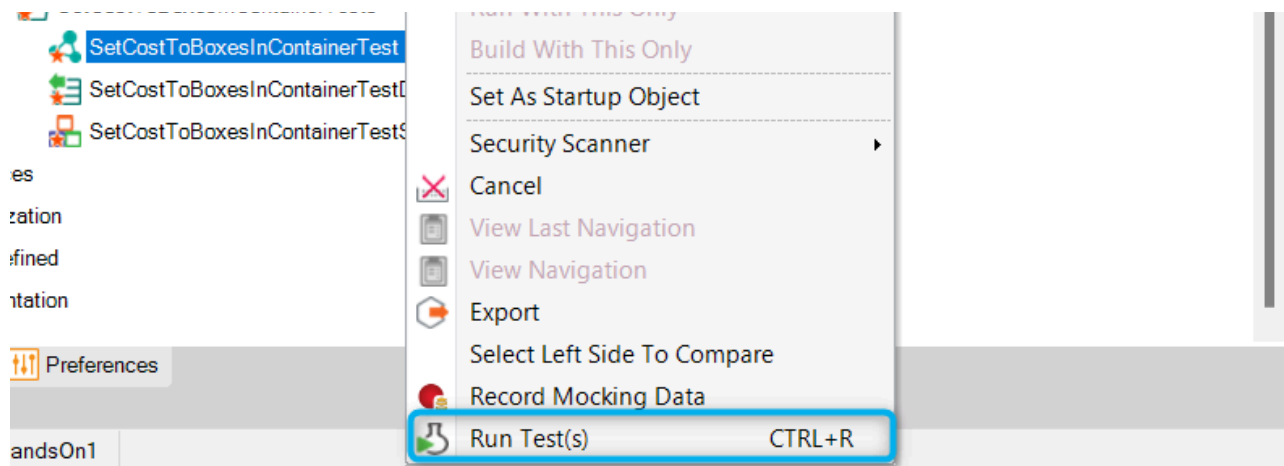
```
1  /* Autogenerated unit test code for Procedure 'SetCostToBoxesInContainer' */  
2  
3  For &TestCaseData in SetCostToBoxesInContainerTestData()  
4  
5      /* Act... */  
6      &TestCaseData.ModifiedBoxesCount = SetCostToBoxesInContainer(&TestCaseData.ContainerId)  
7  
8      /* Assert... */  
9      AssertNumericEquals(&TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount, format('!%1.ExpectedModifiedBoxesCount: %2', &Test  
10  
11      for each Box  
12          where ContainerId = &TestCaseData.ContainerId  
13          AssertBoolEquals(true, BoxCost > 0, Format("El costo de la caja %1 del contenedor %2 es %3",BoxId, ContainerId, BoxCost))  
14      endfor  
15  
16  endfor  
17  
18  
19
```

- 3) Escriba los datos de prueba para el contenedor 1 en el data provider y guarde los cambios:

El código del Data Provider es:

```
SetCostToBoxesInContainerTestSDT  
{  
    TestCaseId = '1'  
    ContainerId = 1  
    ExpectedModifiedBoxesCount = 9  
    MsgModifiedBoxesCount = '9'  
}
```

- 4) Correr el test haciendo botón derecho sobre el objeto Unit Test y seleccionando la opción "Run Test".



En el panel de resultados podrá ver, en la primera línea la validación del parámetro de salida del proc y en el resto las validaciones del estado de la base de datos del costo de cada caja.

Unit test execution details				
<div> Tests.SetCostToBoxesInContainerTest Coverage: </div> <div> Start: Thursday, September 15, 2022 12:25:32 PM </div> <div> Elapsed time: 458 ms </div>				
Expected	Obtained	Info		
9	9	1.ExpectedModifiedBoxesCount: 9		+
✓ true	true	El costo de la caja 1 del contenedor 1 es 48.30		+
✓ true	true	El costo de la caja 2 del contenedor 1 es 18.40		+
✓ true	true	El costo de la caja 3 del contenedor 1 es 37.40		+
✓ true	true	El costo de la caja 4 del contenedor 1 es 72.00		+
✓ true	true	El costo de la caja 5 del contenedor 1 es 39.60		+
✓ true	true	El costo de la caja 6 del contenedor 1 es 95.00		+
✓ true	true	El costo de la caja 7 del contenedor 1 es 60.00		+
✓ true	true	El costo de la caja 8 del contenedor 1 es 22.00		+
✓ true	true	El costo de la caja 9 del contenedor 1 es 42.00		+



MONTEVIDEO - URUGUAY
SAN FRANCISCO - USA
LONDRES - UK

Sarmiento 2465, 11300
100 Pine St., Ste. 1250, CA 94111
71-75 Shelton Street, WC2H 9JQ

+598 2711 0561
+1 415 745 3678
+44 203 696 6682