

Testing automatizado em **Genexus**

CONTEÚDO

CONTEÚDO	2
OBJETIVO	3
APLICATIVO	3
TESTING	5
TESTES UNITÁRIOS	5
GEREMOS UM UNIT TEST A PARTIR DO PROC!	6
CENÁRIO 1: Unit Test falha	7
CENÁRIO 2: Unit Test aprova	9
CENÁRIO 3: Validação do estado do banco de dados	10
TESTE DE INTERFACE	12
GRAVAÇÃO DE FLUXO COM GXtest Recorder	12
TEST SUITES (OPCIONAL)	22
CENÁRIO 1: Test suite aprova	22
CENÁRIO 2: Test suite falha	23

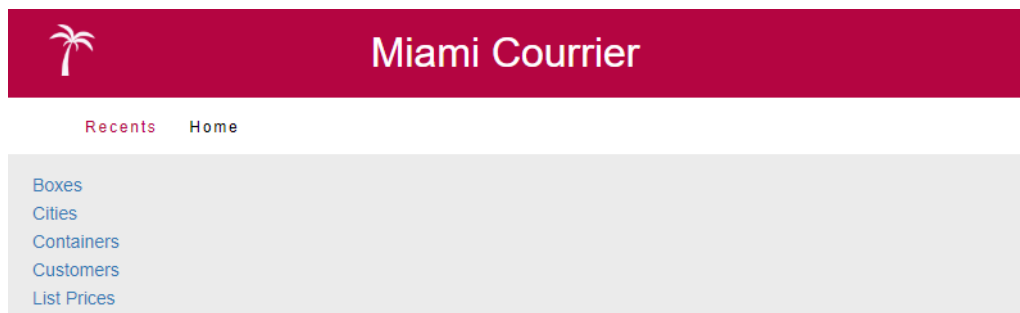
OBJETIVO

Neste workshop, você terá um guia passo a passo para construir e executar um teste unitário e um teste de interface de usuário para aplicativos web. O objetivo deste laboratório não é treinar, mas sim familiarizar-se com a simplicidade do uso da ferramenta GXtest para desenvolvedores e testadores de aplicações GeneXus.

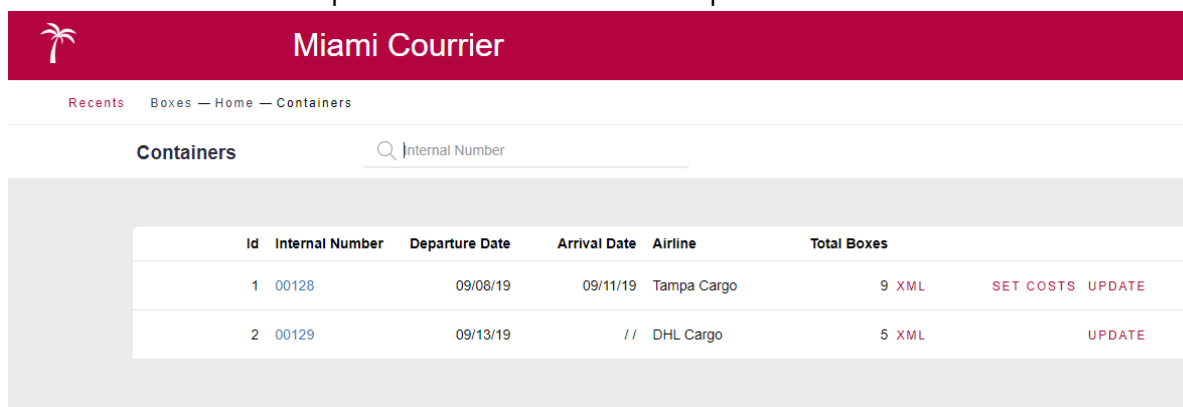
APLICATIVO

A aplicação com a qual trabalharemos simula ser um *Business Courier*, utilizado para gerenciar contêneires com as respectivas caixas, clientes e alfândegas. Vamos ver o aplicativo!


1. Abra o IDE de GeneXus
2. No caso de não abrir automaticamente, abra a KB chamada de “**GXtestHandsOn**” (File > Open > Knowledge Base , selecione de C:\Models\GXtestHandsOn o arquivo **GXtestHandsOn.gxw**)
3. Execute a partir do IDE de GeneXus (F5). A seguinte tela deverá se abrir no seu navegador Chrome:



4. Clique em “Containers”. Você pode ver que o aplicativo possui dois contêneires com suas respectivas caixas. Observe que o container 1 tem uma quantidade de 9 caixas.



5. Clique no link “00128” do container de id 1. Você pode ver mais informações do container:



Miami Courier

Recents

Home — Containers — 00128

Container Information

← CONTAINERS

Internal Number

00128

General

Box

UPDATE

DELETE

Id

1


Internal Number

00128

Departure Date

05/05/21

6. Clique na janela *Box*. Você pode ver na coluna Custo que todas as caixas têm como status inicial um custo de 0. Quando uma caixa é criada, ela é criada com custo 0.



Miami Courier

Recents

Home — Containers — 00128

Container Information

← CONTAINERS

Internal Number

00128

General

Box

Tracking	Type	Weight	Status	Purchase Amount	Cost	Is Retained	Be Delivered	Customer
1146HFGDS3	Clothes	2.30	AT DESTINATION	180.00	0.00	<input type="checkbox"/>	false	Pablo Romero
KJS7587F541	Electronics	0.80	AT DESTINATION	30.00	0.00	<input type="checkbox"/>	false	Analia Gutierrez
290DKJHXA3	Baby Accs.	1.70	AT DESTINATION	90.00	0.00	<input type="checkbox"/>	false	Joaquin Martinez
7343GFDW953	Home decor	3.60	AT DESTINATION	165.00	0.00	<input type="checkbox"/>	false	Sofia Perez
BCHS837MSM1	Clothes	1.80	AT DESTINATION	130.00	0.00	<input type="checkbox"/>	false	Pablo Romero
08KSDX73BD1	Furniture	5.00	AT DESTINATION	180.00	0.00	<input type="checkbox"/>	false	Analia Gutierrez
9BFIEN3832B	Cam Lenses	3.00	AT DESTINATION	180.00	0.00	<input type="checkbox"/>	false	Analia Gutierrez
SCYUETB98796	Stickers	1.00	AT DESTINATION	180.00	0.00	<input type="checkbox"/>	false	Sofia Perez
SCYUETB98796	Toys	2.00	AT DESTINATION	55.00	0.00	<input type="checkbox"/>	false	Joaquin Martinez

A funcionalidade mais importante do aplicativo é **Definir o custo das caixas de um container.**

TESTING

Vamos validar o correto funcionamento da funcionalidade **Definir o custo das caixas de um container**. Ele é implementado no nível da interface no painel Containers (etapa 4). Ao clicar no link *SET COSTS*, é definido o custo das caixas associadas ao container. Não clique em *SET COSTS*, pois isso alterará o estado do banco de dados do aplicativo.

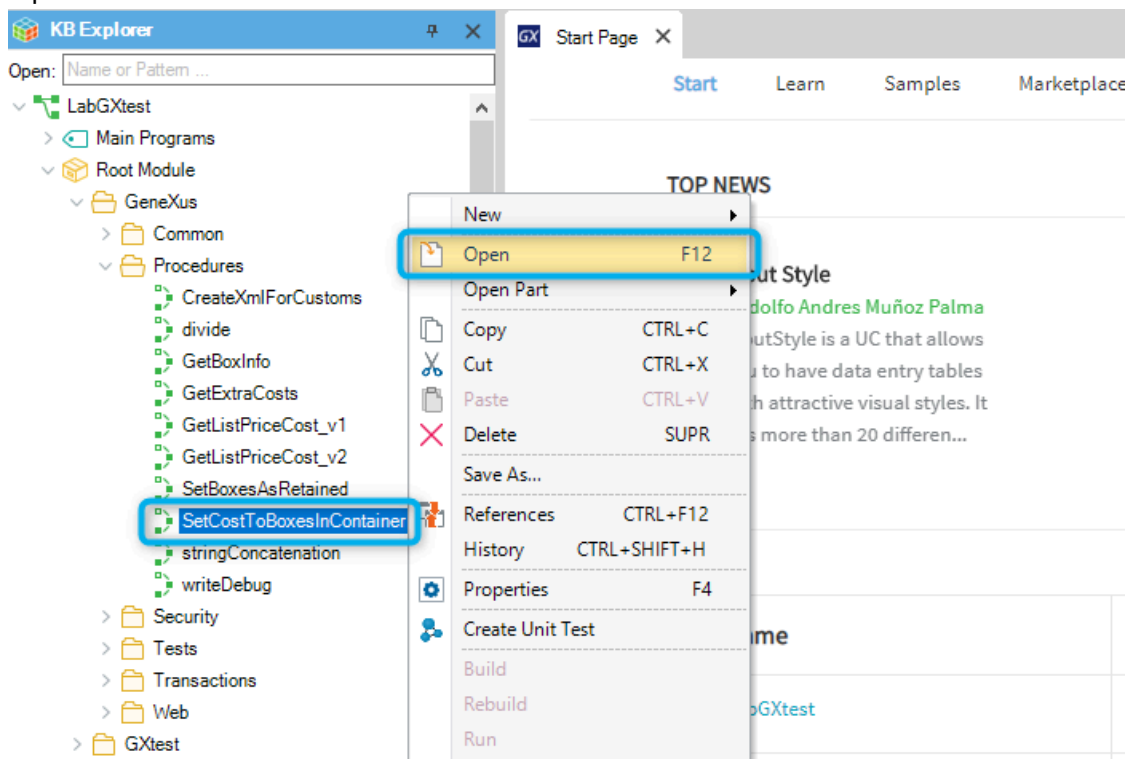
Em uma primeira instância, testaremos a funcionalidade **Definir o custo das caixas de um container** com um teste unitário, mais conhecido como *unit test*. Posteriormente, verificaremos a funcionalidade no nível da interface, executando um teste de interface de usuário, mais conhecido como *UI test (user interface test)*.

TESTES UNITÁRIOS

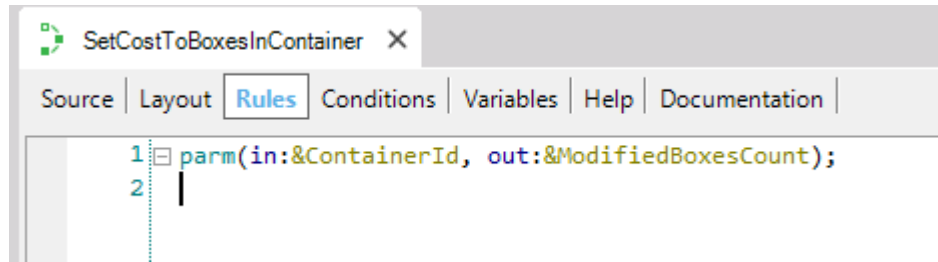
Nesta seção, o guiaremos para que você possa executar três cenários diferentes de teste unitário.

A funcionalidade **Definir o custo das caixas de um container**, é implementada pelo procedimento **"SetCostToBoxesInContainer"** do nosso KB. Vamos ver o procedimento!

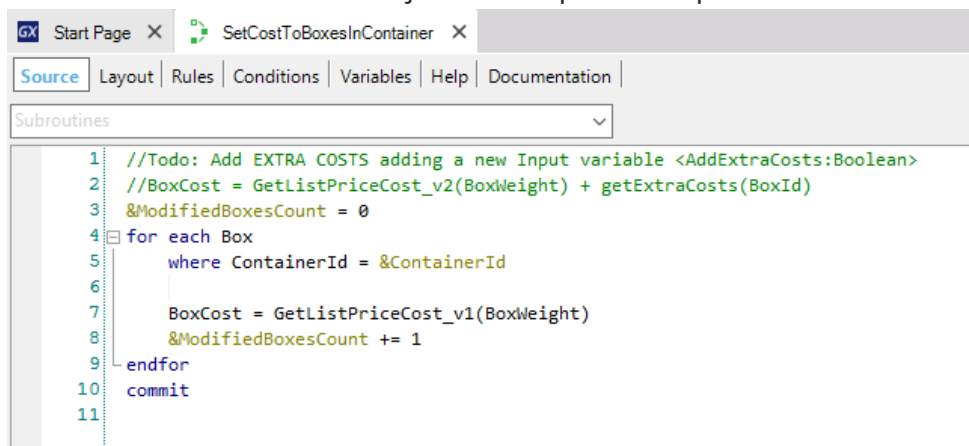
5. Abra o procedimento clicando com o botão direito sobre ele e selecionando *"Open"* ou simplesmente clicando duas vezes nele.



6. Clique na guia Rules. Você pode ver que o procedimento recebe como parâmetro de entrada `&ContainerId` (id do container para o qual vai definir o custo das caixas) e retorna o parâmetro de saída `&ModifiedBoxesCount` (número de caixas modificadas):

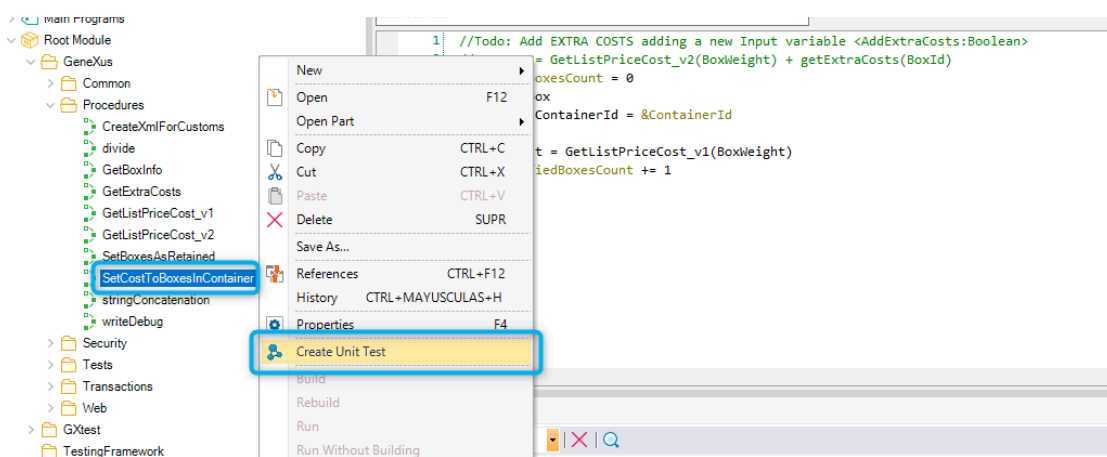


7. Clique na guia Source. Você pode ver, na lógica do procedimento, que o proc atribui o custo de cada caixa de container cujo id corresponde ao parâmetro de entrada `&ContainerId`:

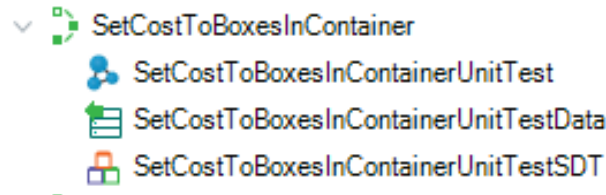


GEREMOS UM UNIT TEST A PARTIR DO PROC!

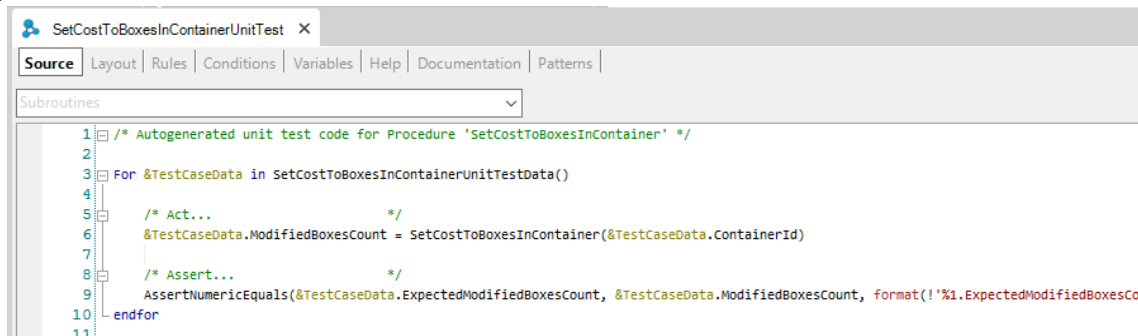
8. Selecione o procedimento *"SetCostToBoxesInContainer"* no painel KB Explorer, clique com o botão direito e escolha a opção *"Create Unit Test"*.



Aguarde até que se gerem os objetos de teste "SetCostToBoxesInContainerUnitTest", "SetCostToBoxesInContainerUnitTestData" e "SetCostToBoxesInContainerUnitTestSDT" que aparecem juntos no procedimento "SetCostToBoxesInContainer"



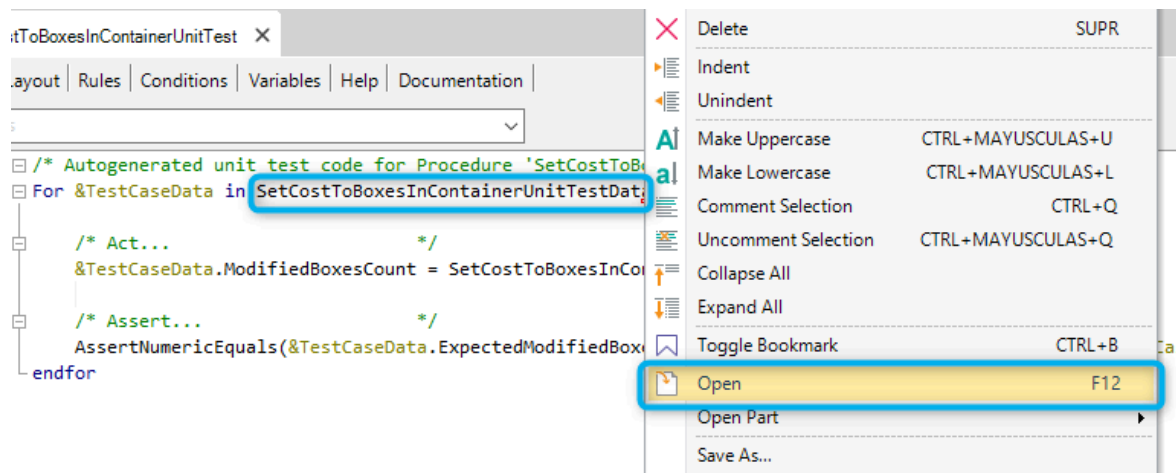
Como pode observar, GXtest gera automaticamente um *template* de código GeneXus iterando sobre o *Data Provider* "SetCostToBoxesInContainerUnitTestData", ao qual contém os conjuntos de dados que o teste executará.



Executaremos três cenários diferentes: falha no unit test, aprovação no unit test e validação do banco de dados no unit test. Primeiro, inserimos casos de teste incorretos para mostrar o que acontece quando um teste falha. No segundo, usaremos os valores corretos para obter a execução bem-sucedida do teste. Por último, no terceiro cenário, adicionaremos o teste unitário em uma validação de estado do banco de dados.

CENÁRIO 1: Unit Test falha

9. Clique com o botão direito do mouse no objeto "SetCostToBoxesInContainerUnitTestData" y seleccione "Open"

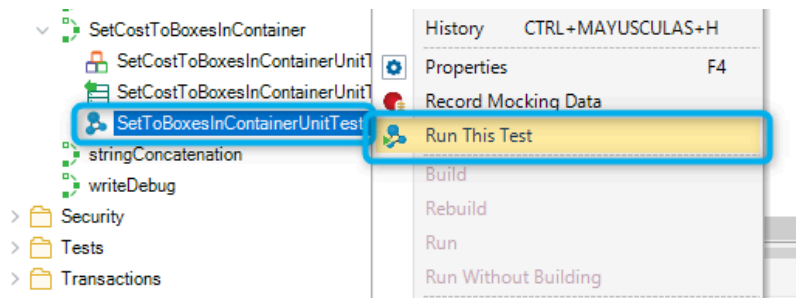


10. Insira os dados de teste para o caso de teste. Carregue o valor de *ContainerId* = 1 e *ExpectedModifiedBoxesCount* = 8 e *MsgModifiedBoxesCount* = '8'.

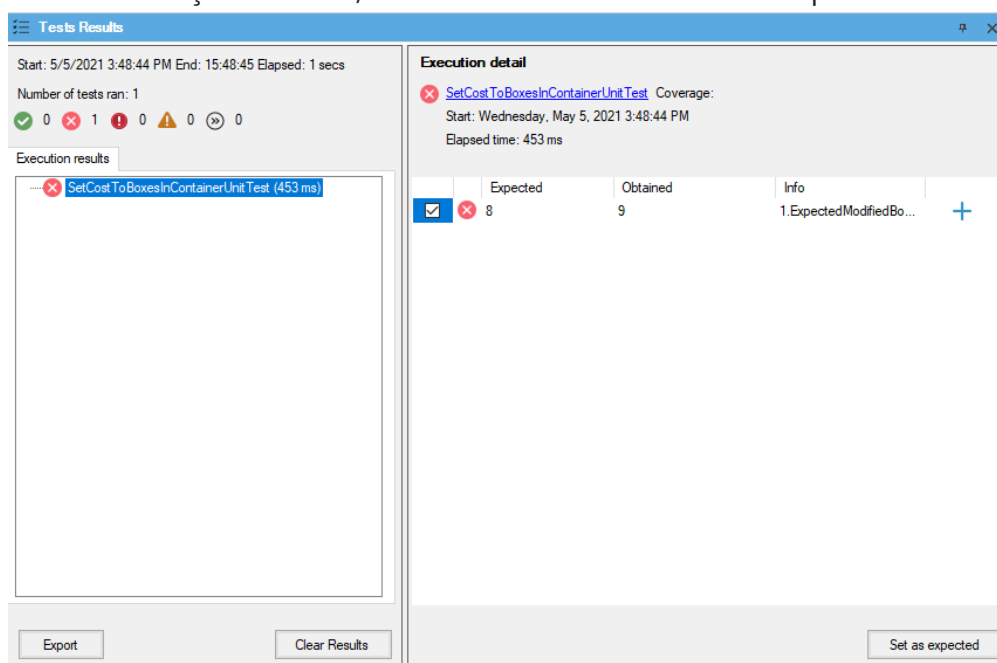
O código do *Data Provider* é:

```
SetCostToBoxesInContainerUnitTestSDT
{
    TestCaseld = '1'
    ContainerId = 1
    ExpectedModifiedBoxesCount = 8
    MsgModifiedBoxesCount = '8'
}
```

11. Salve as alterações, clique com o botão direito no *unit test* e selecione a opção "Run This Test".



Ao concluir a execução do teste, os resultados serão exibidos no painel *Tests Results*.



CENÁRIO 2: Unit Test aprova

12. Modifique os valores do *Data Provider*, modifique *ExpectedModifiedBoxesCount* = 9 e *MsgModifiedBoxesCount* = '9'.

O código do *Data Provider* é:

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

13. Salve as alterações, clique com o botão direito sobre ele e selecione a opção *"Run This Test"*.

Veremos que agora o teste não falhou, o teste passou pois o valor esperado é igual ao obtido:

The screenshot shows the 'Tests Results' window in Visual Studio. The top bar indicates the test start and end times, and the elapsed time. The 'Execution results' tab shows a single test, 'SetCostToBoxesInContainerUnit Test (303 ms)', which passed successfully, indicated by a green checkmark. The 'Execution detail' tab provides further information about the test, including the start time, elapsed time, and a table comparing expected and obtained values.

Start: 5/5/2021 5:16:46 PM End: 17:16:47 Elapsed: 1 secs

Number of tests ran: 1

✓ 1 ✗ 0 ! 0 ⚠ 0 ⏮ 0

Execution results

..... ✓ SetCostToBoxesInContainerUnit Test (303 ms)

Execution detail

✓ SetCostToBoxesInContainerUnit Test Coverage:

Start: Wednesday, May 5, 2021 5:16:46 PM

Elapsed time: 303 ms

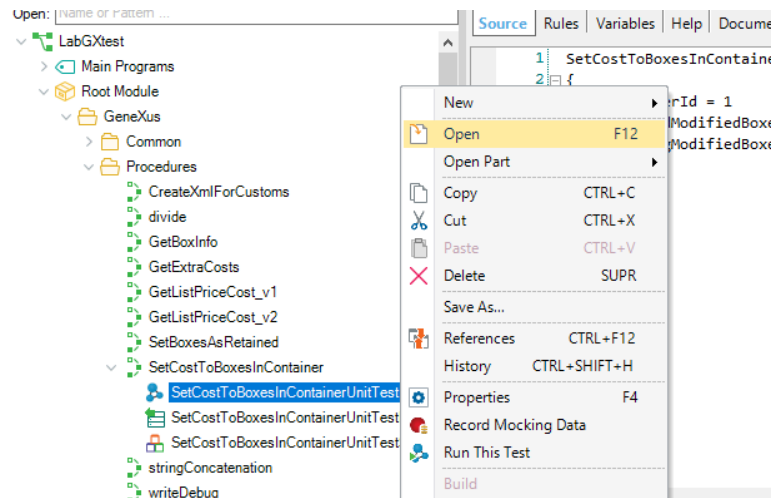
	Expected	Obtained	Info
1	9	9	1.ExpectedModifiedBox... +

Export Clear Results Set as expected

CENÁRIO 3: Validação do estado do banco de dados

Agora vamos validar que quando o proc é exercido, o banco de dados realmente muda de estado. Para isso, adicionaremos uma validação (*assertion*) sobre o estado do banco de dados. Neste caso, a validação irá verificar se o custo de cada caixa do container tem um valor maior que 0.

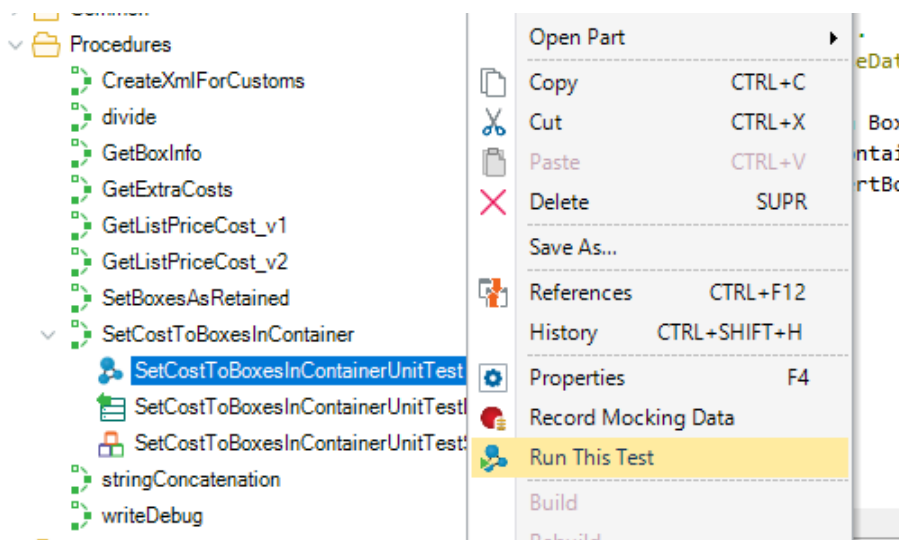
14. Abra o objeto SetCostToBoxesInContainerUnitTest.



15. Adicione a seguinte validação debaixo do *assertion* já 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
```

16. Salve as alterações e execute o *test* clicando com o botão direito sobre o objeto Unit Test e selecionando a opção “Run This Test”.



No painel de resultados você pode ver, na primeira linha, a validação do parâmetro de saída proc e nas demais as validações do estado do banco de dados do custo de cada caixa

Tests Results

Start: 5/5/2021 6:25:06 PM End: 18:25:07 Elapsed: 1 secs

Number of tests ran: 1

1 0 0 0 0 0

Execution results

SetCostToBoxesInContainerUnit Test (523 ms)

Execution detail

SetCostToBoxesInContainerUnit Test Coverage:

Start: Wednesday, May 5, 2021 6:25:06 PM

Elapsed time: 523 ms

		Expected	Obtained	Info	
<input checked="" type="checkbox"/>	✓	9	9	1.ExpectedModifiedBoxesCount: 9	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 1 del contenedor 1 es 48.30	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 2 del contenedor 1 es 18.40	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 3 del contenedor 1 es 37.40	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 4 del contenedor 1 es 72.00	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 5 del contenedor 1 es 39.60	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 6 del contenedor 1 es 95.00	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 7 del contenedor 1 es 60.00	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 8 del contenedor 1 es 22.00	+
<input type="checkbox"/>	✓	true	true	El costo de la caja 9 del contenedor 1 es 42.00	+

Export Clear Results Set as expected

O código final do unit test é:

```
/* Autogenerated unit test code for Procedure 'SetCostToBoxesInContainer' */
For &TestCaseData in SetCostToBoxesInContainerUnitTestData()

    /* Act... */
    &TestCaseData.ModifiedBoxesCount = SetCostToBoxesInContainer(&TestCaseData.ContainerId)

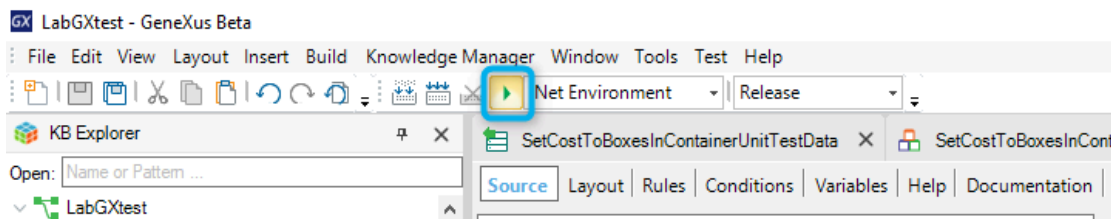
    /* Assert de la variable de salida del procedimiento */
    AssertNumericEquals(&TestCaseData.ExpectedModifiedBoxesCount, &TestCaseData.ModifiedBoxesCount, format('!%1.ExpectedModifiedBoxesCount: %2',
    &TestCaseData.TestCaseId, &TestCaseData.MsgModifiedBoxesCount))

    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
    endfor
```

TESTE DE INTERFACE

Nesta seção, iremos guiá-lo para que você possa executar um teste de interface do usuário (doravante, UI Test) no aplicativo web em que estamos trabalhando. Vamos registrar o seguinte fluxo: criar uma caixa pertencente ao container 1, definir o custo do container 1, validar que o custo foi atribuído e excluir a caixa.

Acesse o aplicativo aberto no browser ou clique no botão "Play" e aguarde até que o aplicativo seja executado no navegador Chrome novamente:

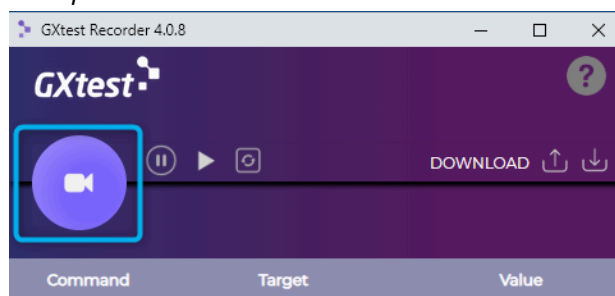


GRAVAÇÃO DE FLUXO COM GXtest Recorder

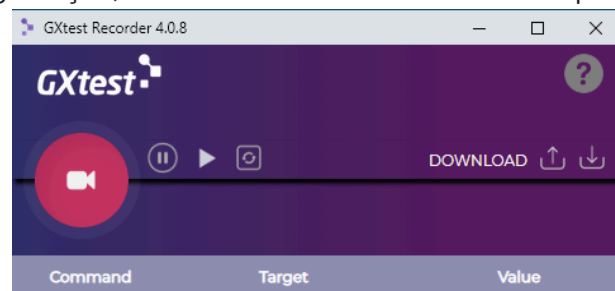
Abra a extensão GXtest Recorder clicando no ícone da imagem, na extrema direita do navegador:



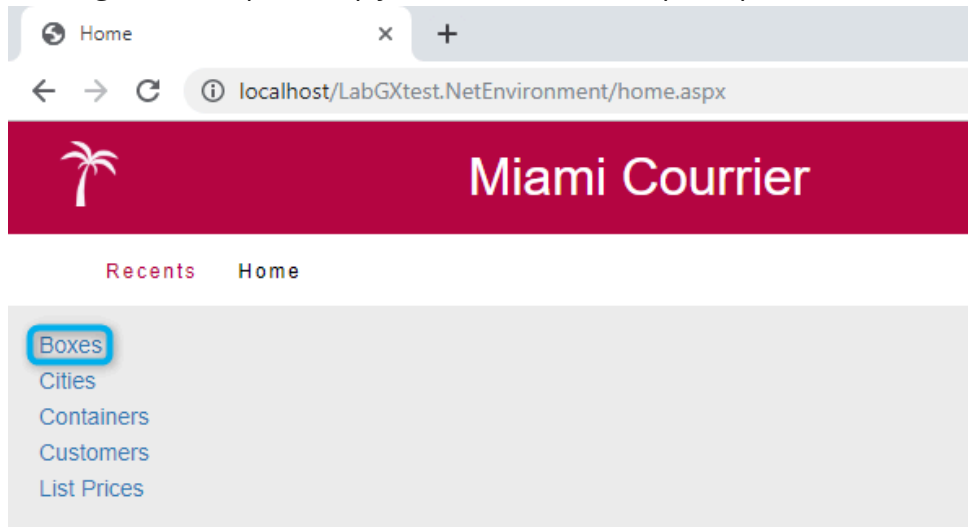
1. Clique no botão *Start/Stop record* :



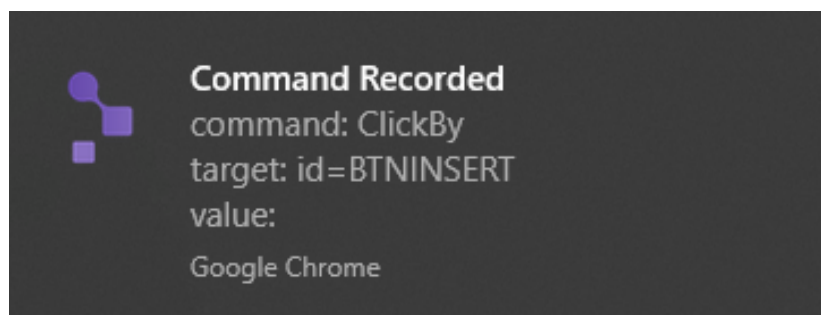
2. Quando começa a gravação, o botão muda de cor indicando que a gravação foi iniciada.



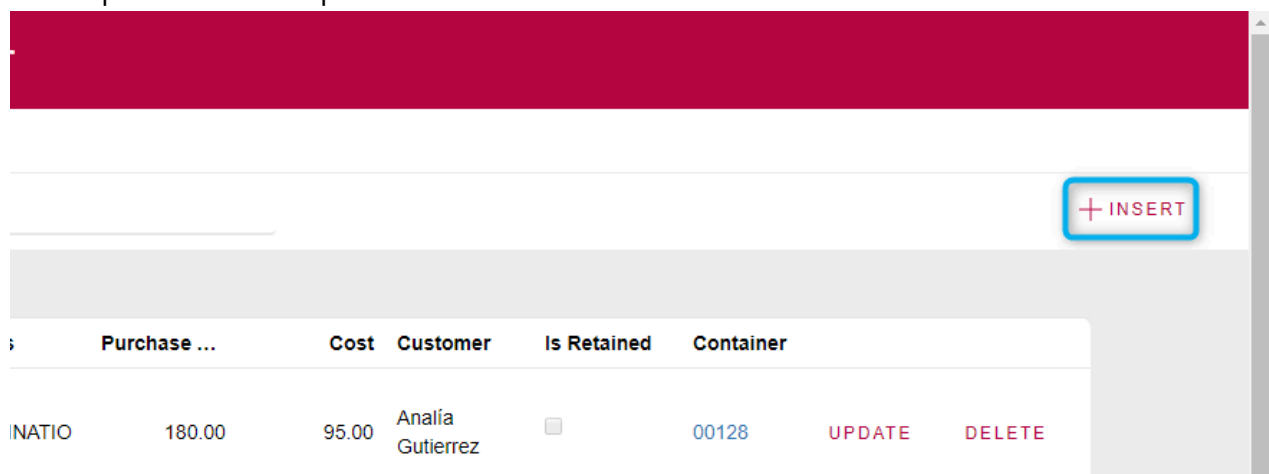
3. Volte ao navegador e clique na opção “Boxes” do menu principal



Você pode verificar se está sendo gravado o fluxo ao visualizar as notificações que aparecem na parte inferior da tela. Para cada ação gravada, a seguinte mensagem é exibida:



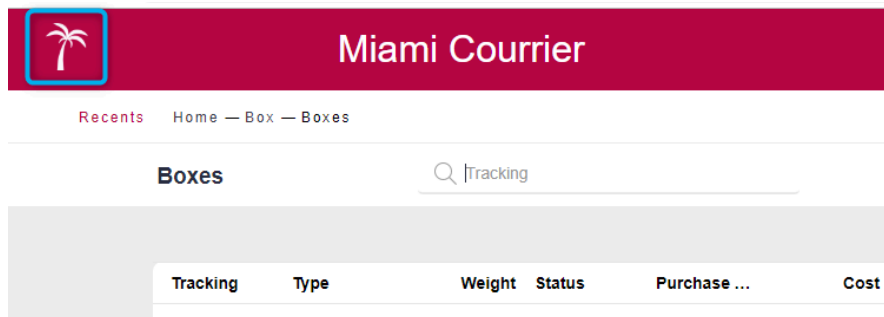
4. Clique em “INSERT” para criar uma nova caixa.



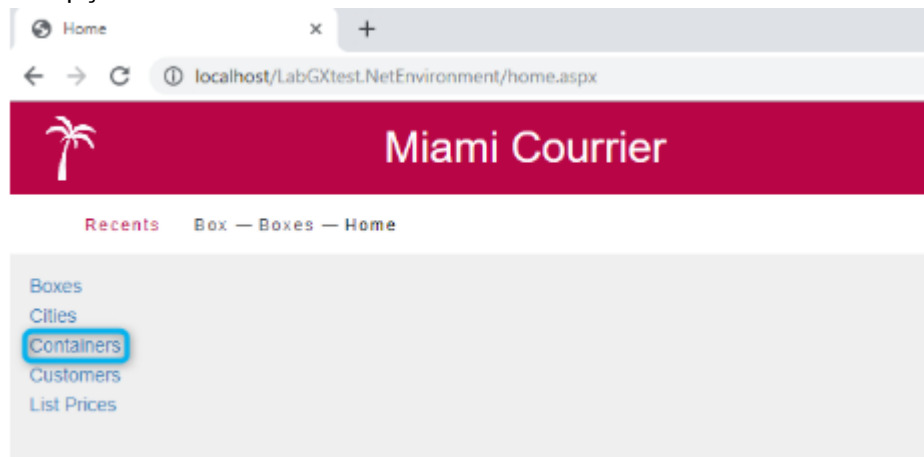
5. Complete apenas os campos **Id**= 999, **Tracking**= NEWBOX999, **Weight**= 1 e **Container Id**= 1 de caixa e clique no botão "**CONFIRM**".

Id	<input type="text" value="999"/>
Tracking	<input type="text" value="newbox999"/>
Type	<input type="text"/>
Weight	<input type="text" value="1.00"/>
Volumetric Weight	<input type="text" value="0.00"/>
Status	<input type="text" value="REGISTERED"/>
Purchase Amount	<input type="text" value="0.00"/>
Cost	<input type="text" value="0.00"/>
Arrival Date	<input type="text" value=" / / 29"/>
Arrival Date	<input type="text" value=" / / 29"/>
Delivery Date	<input type="text" value=" / / 29"/>
Is Retained	<input type="checkbox"/>
Be Delivered	<input type="text" value="false"/>
Customer Id	<input type="text" value="0"/>
Container Id	<input type="text" value="1"/>

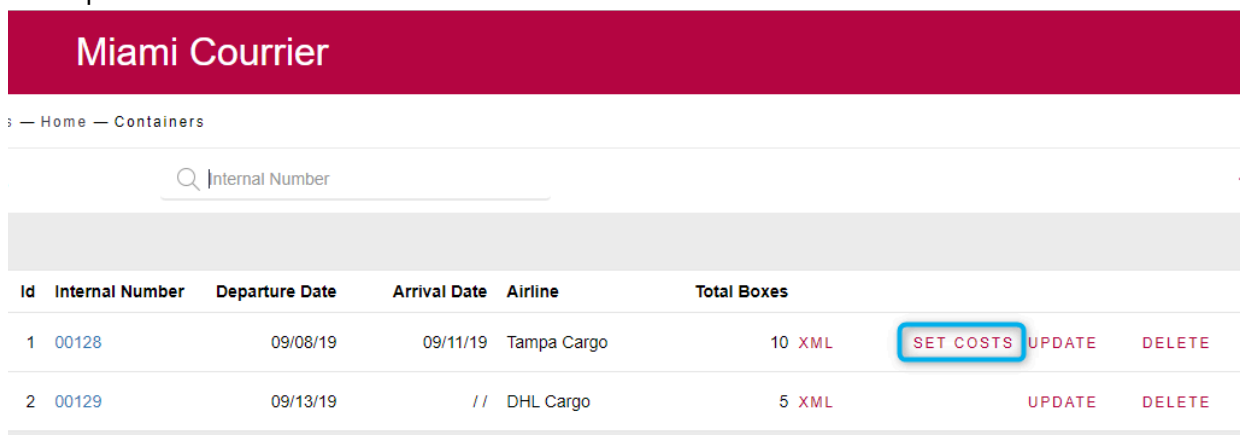
6. Clique na palmeira para voltar à Home



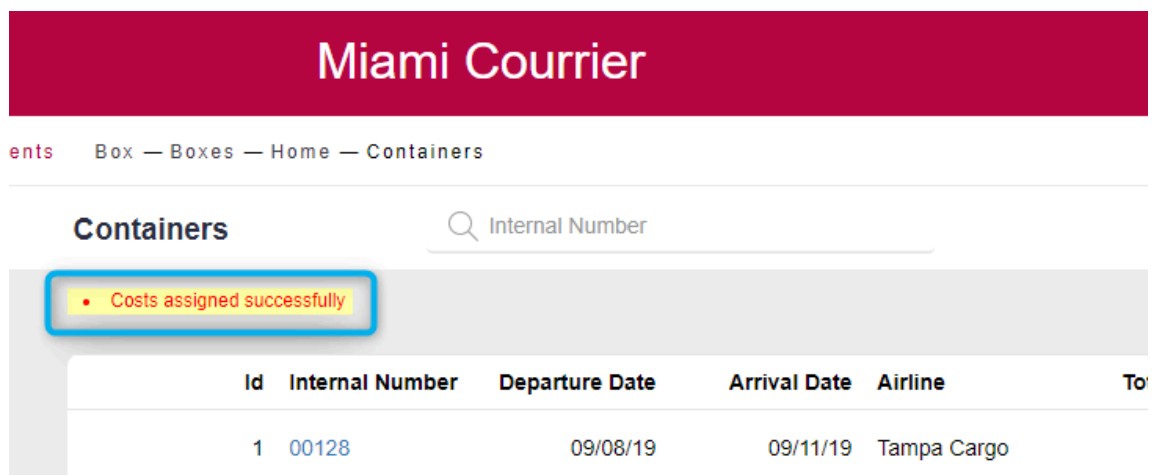
7. Selecione a opção "Containers" do menu.



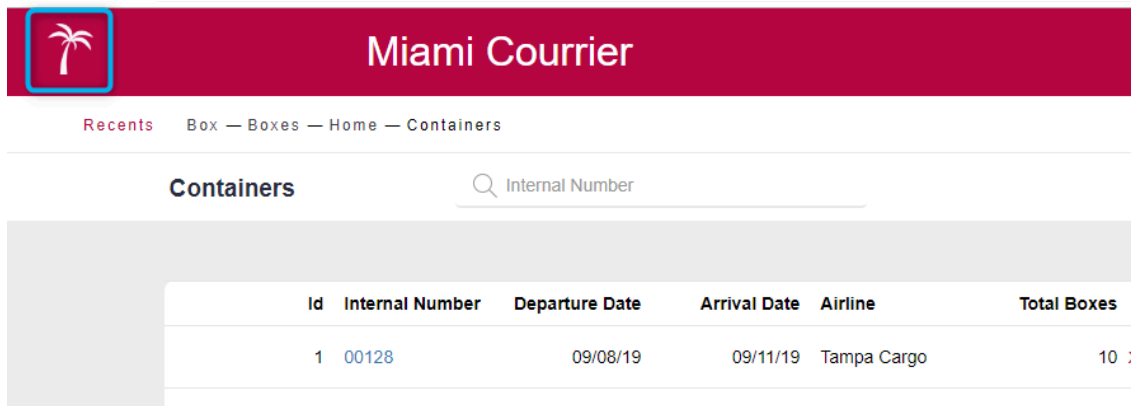
8. Clique em "SET COSTS".



A mensagem "Costs assigned successfully" será exibida indicando que os custos foram atribuídos corretamente.

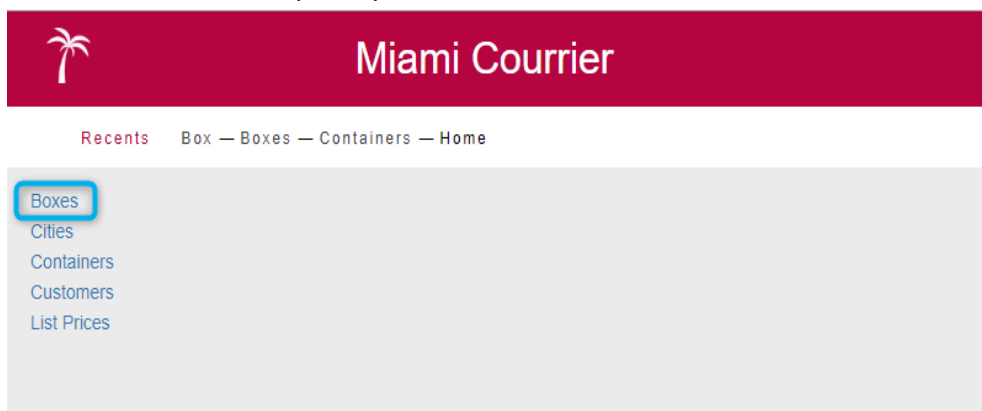


9. Volte a página principal, clique na palmeira.



Id	Internal Number	Departure Date	Arrival Date	Airline	Total Boxes
1	00128	09/08/19	09/11/19	Tampa Cargo	10 X

10. Clique em "Boxes" do menu principal.

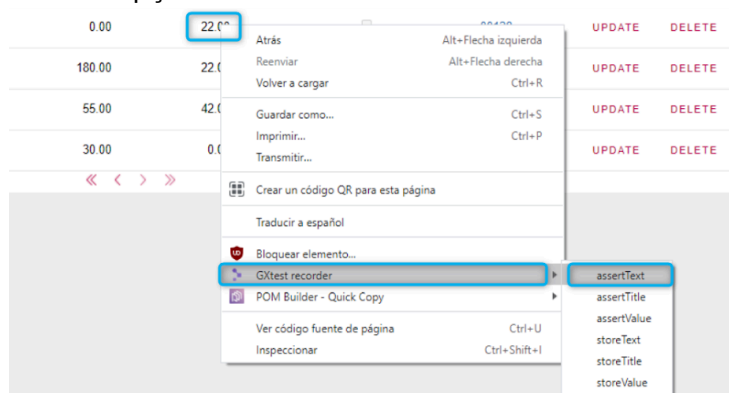


11. Clique sobre a terceira seta para avançar na lista de caixas.

3.40	IN TRANSIT	80.00	0.00	Pablo Romero	<input type="checkbox"/>
3.00	AT DESTINATION	180.00	60.00	Anal?a Gutierrez	<input type="checkbox"/>
1.80	AT DESTINATION	130.00	39.60	Pablo Romero	<input type="checkbox"/>

Navigation buttons: << < > >>

12. Clique com o botão direito sobre o valor 22,0 para validar, (neste caso o custo da caixa newbox999), dentro da opção "GXtest recorder" selecione "assertText".



13. Clique em “DELETE” e depois clique em “CONFIRM”.

Weight	Status	Purchase Am...	Cost	Customer	Is Retained	Container		
0.80	AT DESTINATION	30.00	18.40	Analia Gutierrez	<input type="checkbox"/>	00128	UPDATE	DELETE
1.00	REGISTERED	0.00	22.00		<input type="checkbox"/>	00128	UPDATE	DELETE
1.00	AT DESTINATION	180.00	22.00	Sofia Perez	<input type="checkbox"/>	00128	UPDATE	DELETE
2.00	AT DESTINATION	55.00	42.00	Joaquin Martinez	<input type="checkbox"/>	00128	UPDATE	DELETE
1.20	IN TRANSIT	30.00	0.00	Pablo Romero	<input type="checkbox"/>	00129	UPDATE	DELETE

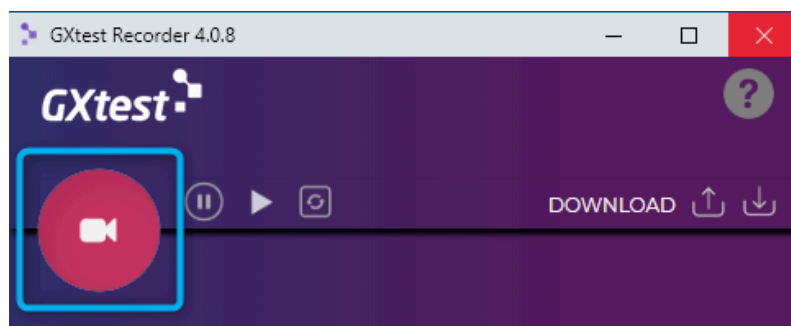
« < > »

omer Id 0

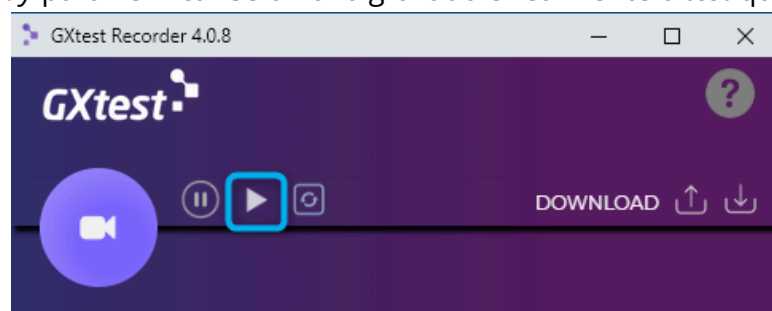
ainer Id 1

CONFIRM CANCEL

14. Pare a gravação pressionando o gravador com o botão *Start/Stop record*



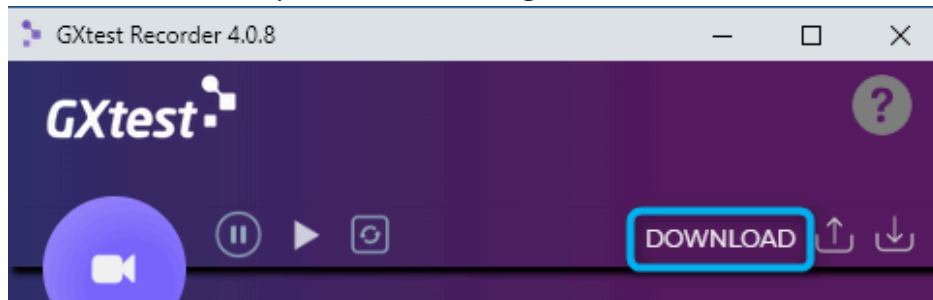
Clique no botão *Play* para verificar se o fluxo gravado é realmente o *test* que desejamos:



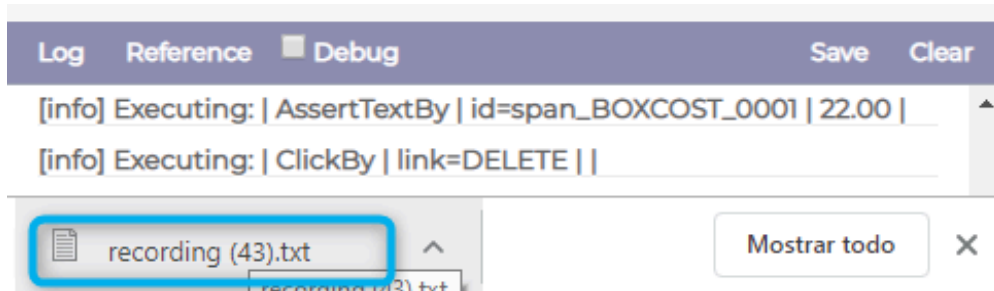
Agora que verificamos que o fluxo gravado com GXtest Recorder é de fato o teste de interface que queremos automatizar, vamos importar a gravação para a KB.

IMPORTAÇÃO DE UI TEST PARA A KB

15. Clique no botão “Download” para baixar o código do UI Test.



16. Abra o arquivo baixado a partir do recorder com o código de UI Test.



17. Copie o conteúdo do arquivo baixado:



O código gerado é:

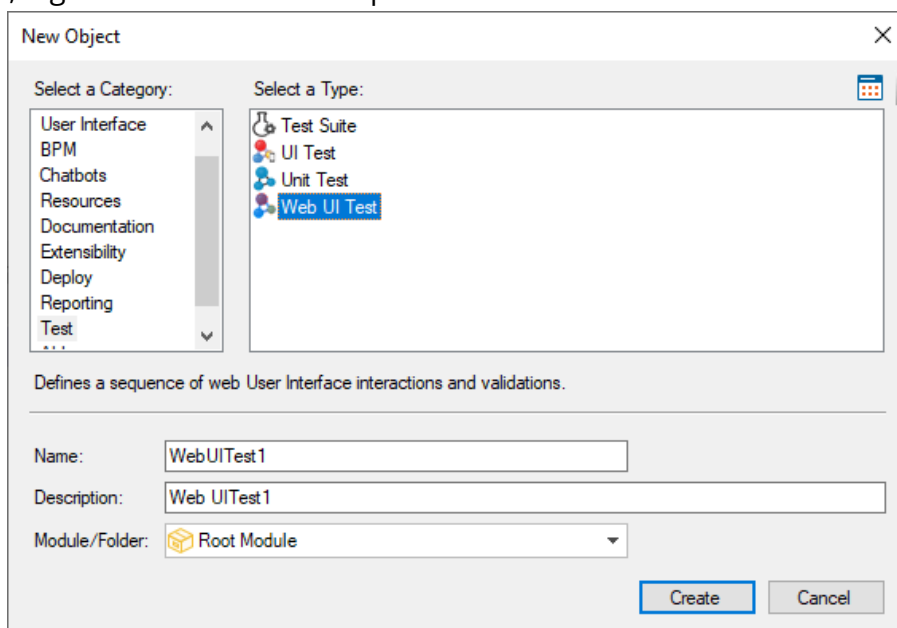
```
// Script generated using GXtest Recorder

//Start webdriver
&driver.Start()
&driver.Maximize()

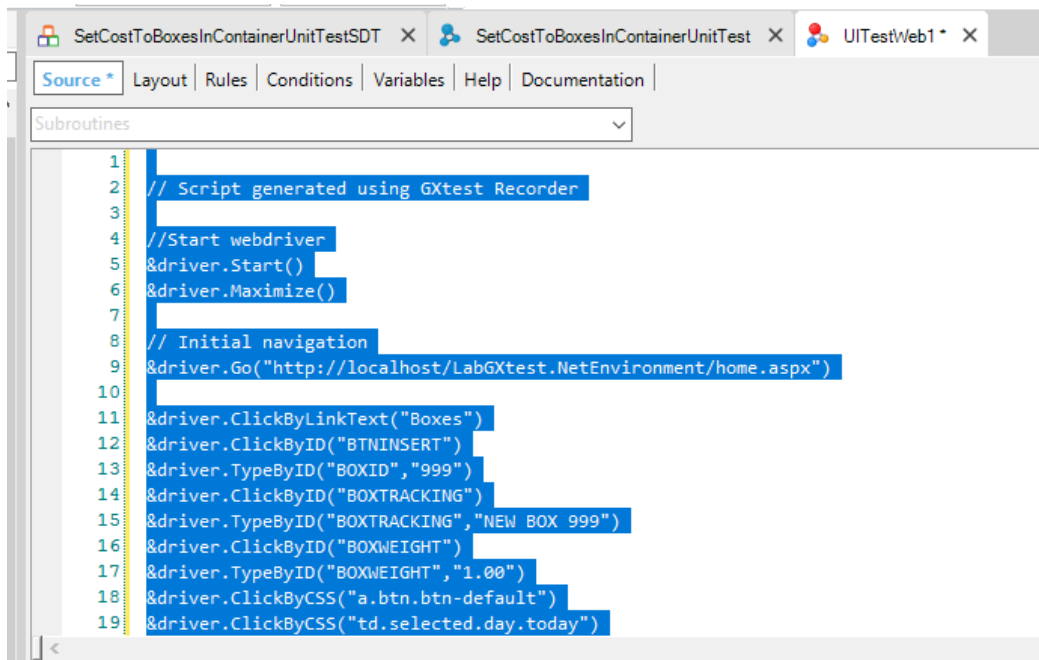
// Initial navigation
&driver.Go("http://localhost/GXtestHandsOn.NetEnvironment/wwcontainer.aspx")

&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Boxes")
&driver.ClickByID("BTNINSERT")
&driver.TypeByID("BOXID","999")
&driver.TypeByID("BOXTRACKING","newbox999")
&driver.TypeByID("BOXWEIGHT","1.00")
&driver.TypeByID("CONTAINERID","1")
&driver.ClickByID("BTN_ENTER")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Containers")
&driver.ClickByLinkText("SET COSTS")
&driver.ClickByID("IMAGE2_MPAGE")
&driver.ClickByLinkText("Boxes")
&driver.ClickByXPath("(//button[@type='button'])[3]")
AssertStringEquals("22.00",&driver.GetTextByID("span_BOXCOST_0002"),"span_BOXCOST_0002 not matching 22.00")
&driver.ClickByCSS("#span_vDELETE_0002 > a")
&driver.ClickByID("BTN_ENTER")
&driver.End()
```

18. Crie um objeto UI Test for Web em sua KB fazendo: File> New> Object, selecione a categoria “Test”, digite “UI Test for Web” e pressione “Create”.



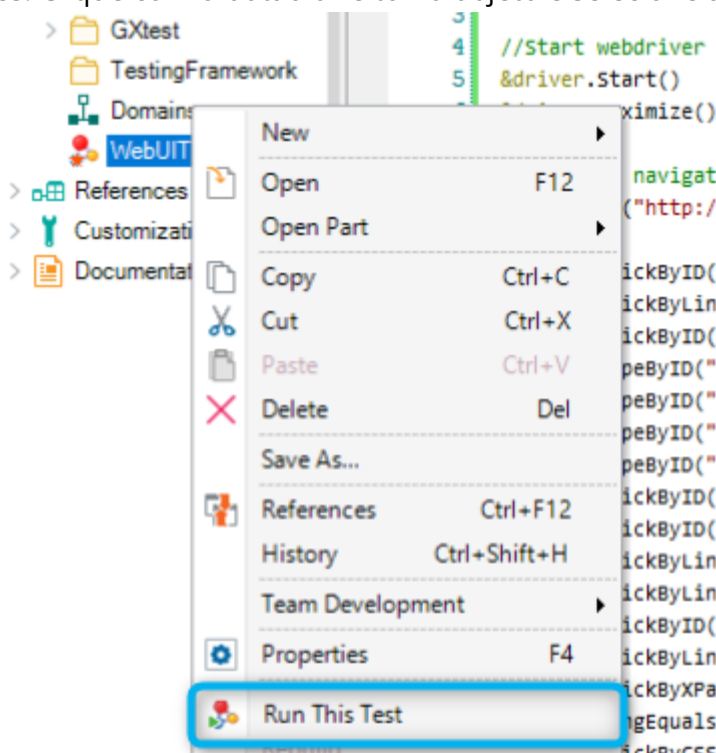
19. Cole o conteúdo do arquivo baixado de GXtest Recorder.



The screenshot shows a code editor with three tabs: 'SetCostToBoxesInContainerUnitTestSDT', 'SetCostToBoxesInContainerUnitTest', and 'UITestWeb1'. The 'Source' tab is active, displaying a test script. The script starts with a comment '// Script generated using GXtest Recorder' and includes various Selenium WebDriver actions like 'Start', 'Maximize', 'Go', 'ClickByLinkText', 'ClickByID', 'TypeByID', and 'ClickByCSS'.

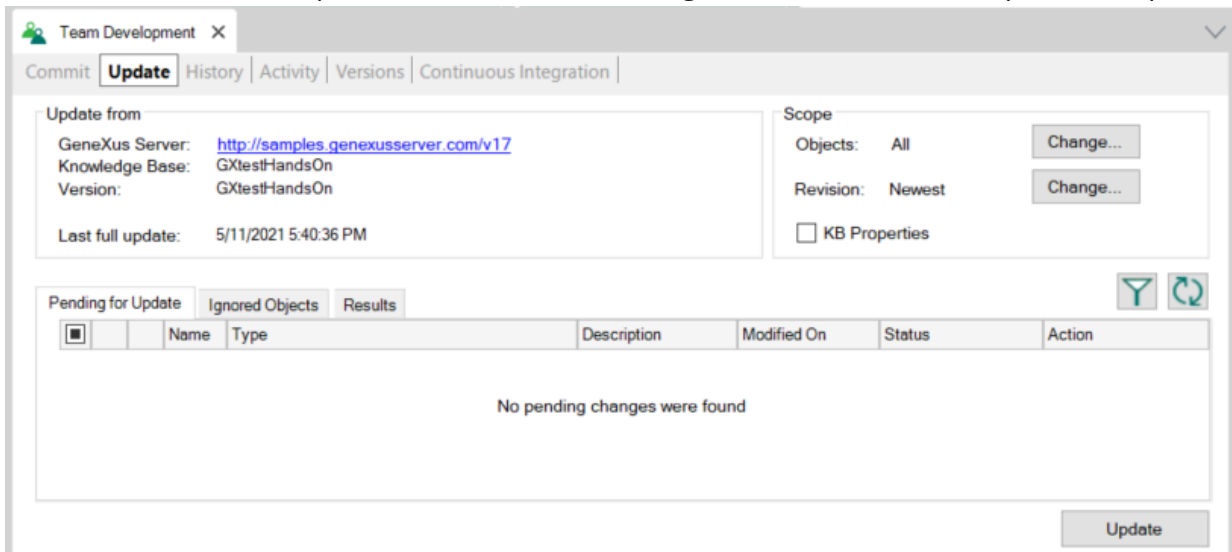
```
1 // Script generated using GXtest Recorder
2
3 //Start webdriver
4 &driver.Start()
5 &driver.Maximize()
6
7 // Initial navigation
8 &driver.Go("http://localhost/LabGXtest.NetEnvironment/home.aspx")
9
10
11 &driver.ClickByLinkText("Boxes")
12 &driver.ClickByID("BTNINSERT")
13 &driver.TypeByID("BOXID", "999")
14 &driver.ClickByID("BOXTRACKING")
15 &driver.TypeByID("BOXTRACKING", "NEW BOX 999")
16 &driver.ClickByID("BOXWEIGHT")
17 &driver.TypeByID("BOXWEIGHT", "1.00")
18 &driver.ClickByCSS("a.btn.btn-default")
19 &driver.ClickByCSS("td.selected.day.today")
```

20. Salve as alterações. Clique com o botão direito no objeto e selecione a opção "Run This Test".



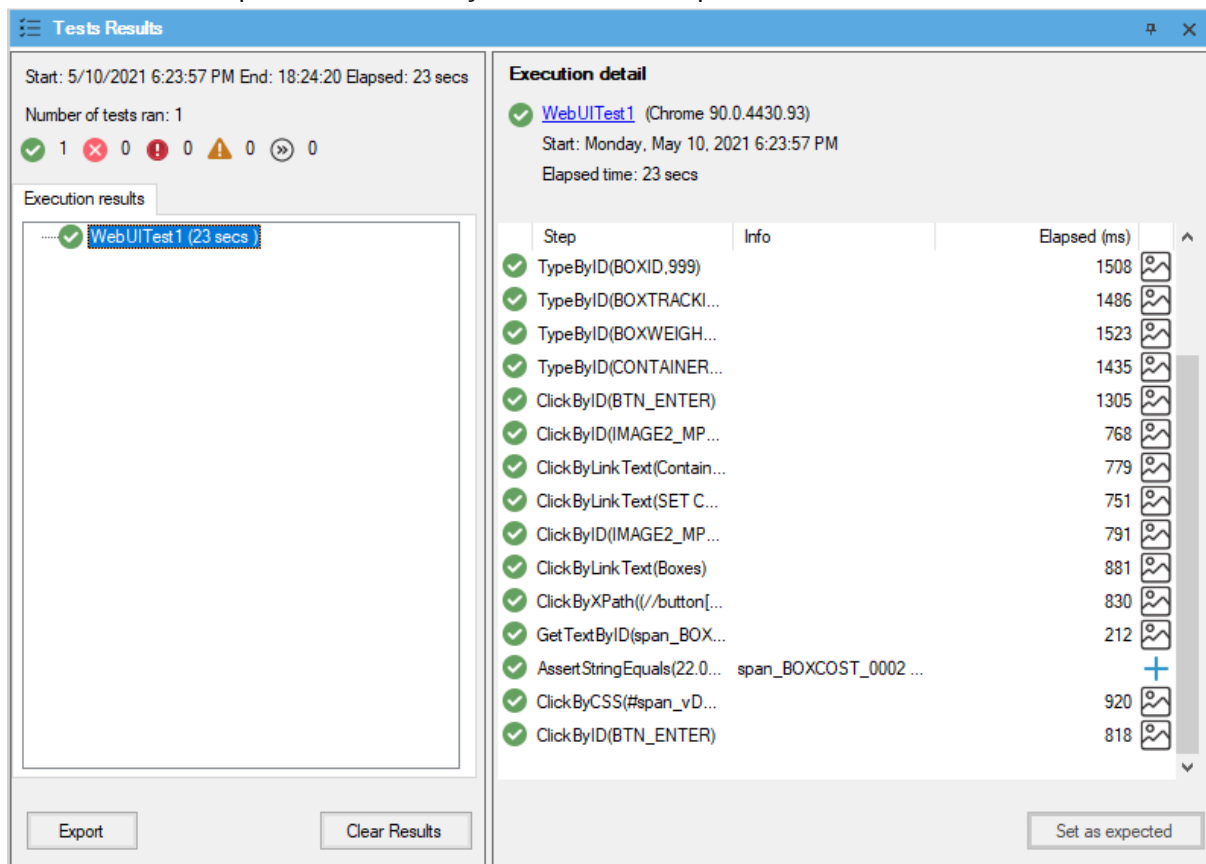
Se a execução falhar, verifique se não há diferenças entre seu código e o da etapa 3.

Em caso de erro, verifique a conexão em Knowledge Base > Team Development > Update:



21. Observe os resultados do test quando a execução terminar. Você pode ver as imagens de tela do aplicativo durante a execução de cada comando.

Explore as informações / botões no painel Resultados do teste!



TEST SUITES (OPCIONAL)

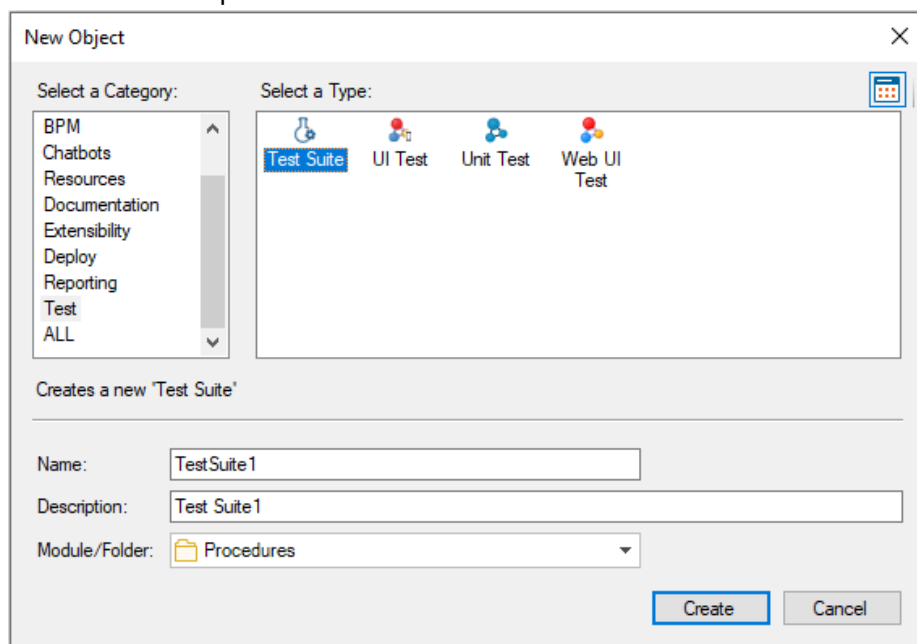
É possível agrupar testes criando um objeto de teste *Test Suite* para que possam ser executados como uma unidade na ordem especificada.

Existem várias propriedades para configurar o comportamento dos testes dentro do *Test Suite*, nesta parte opcional iremos deter no teste da propriedade para continuar ou parar a execução dos testes, em caso de falha.

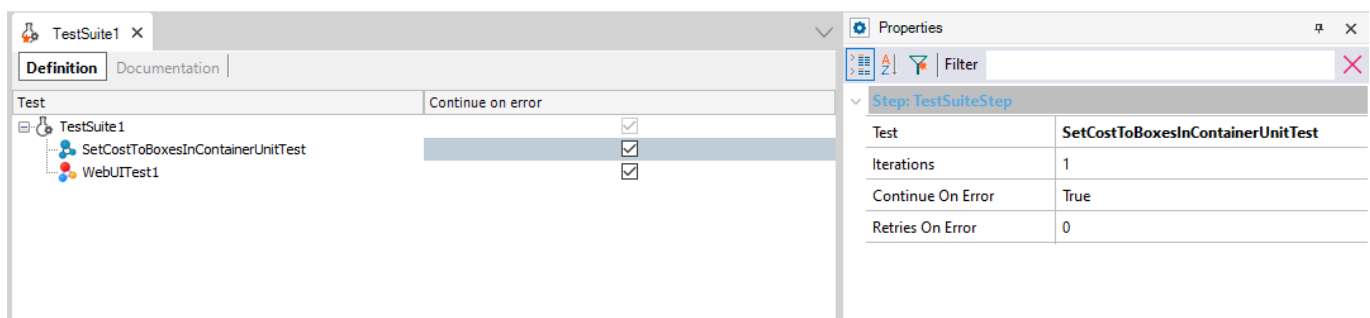
Vamos implementar um objeto Test Suite onde adicionaremos os testes criados anteriormente!

CENÁRIO 1: Test suite aprova

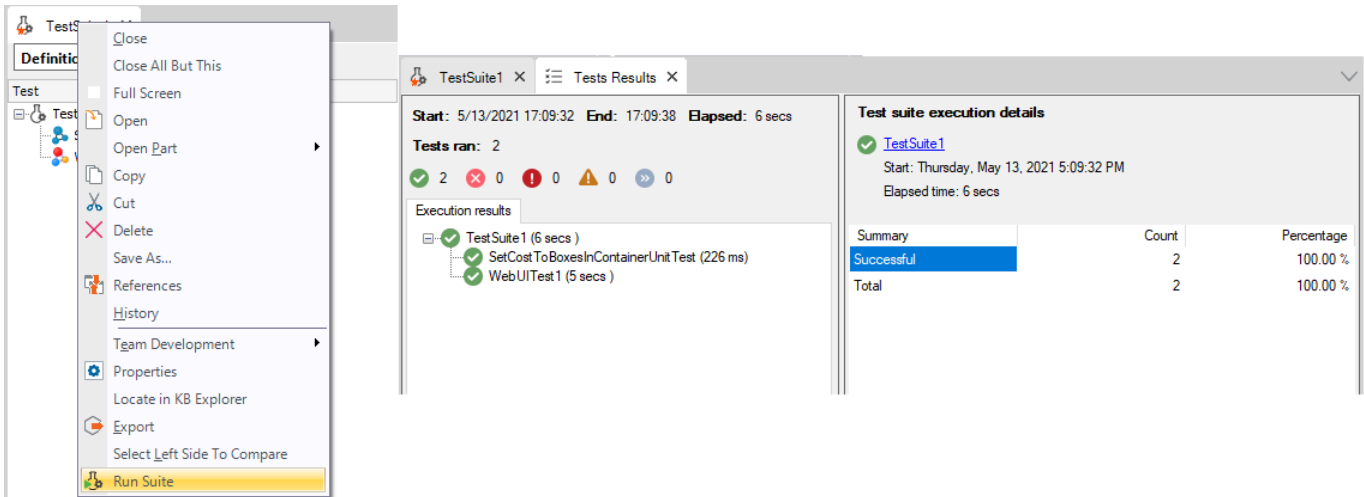
1) Crie um objeto *Test Suite* clicando em File> New> Object, selecione a categoria "*Test*", selecione o objeto "*Test Suite*" e pressione "*Create*".



2) Selecione e arraste do painel KB Explorer o *Unit Test* e o *Web UI test* criados anteriormente e solte-os no editor *Test Suite*. Deve ser assim:



3) Clique com o botão direito no novo objeto *Test Suite* e clique em “*Run Suite*”. Observe que a execução do *Test Suite* foi bem-sucedida, pois todos os testes do Suite foram aprovados.

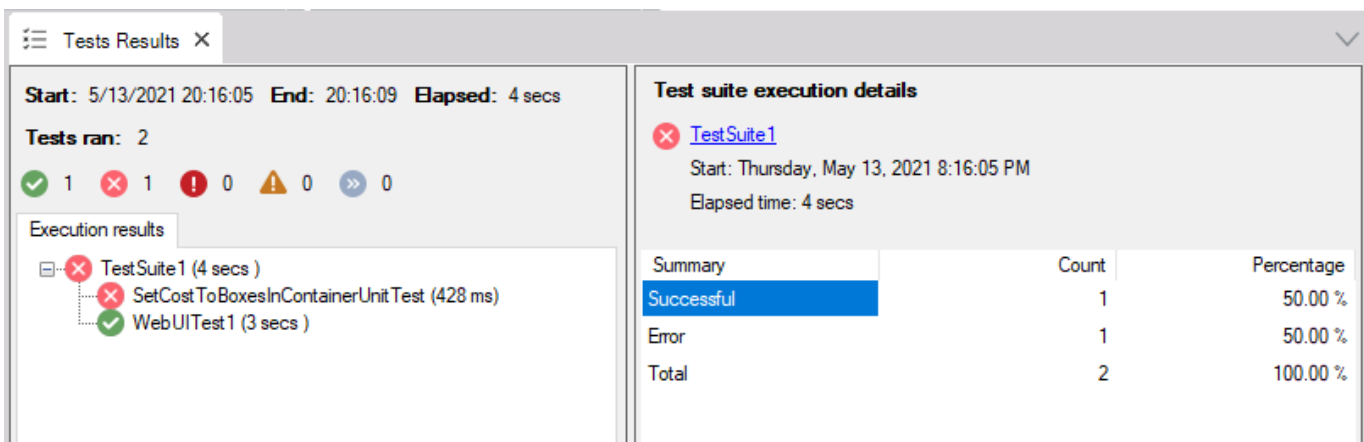


CENÁRIO 2: Test suite falha

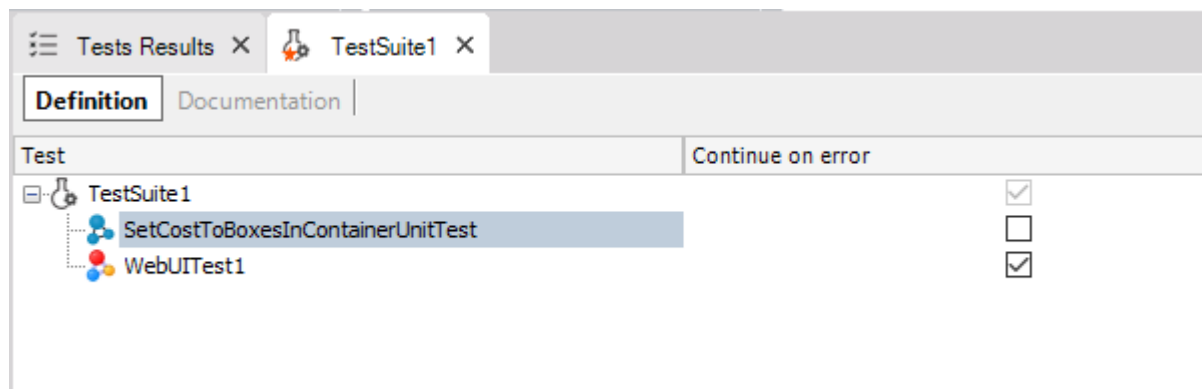
4) Modifique o objeto **SetCostToBoxesInContainerUnitTestSDT** com o seguinte código:

```
SetCostToBoxesInContainerUnitTestSDT
{
    TestCaseld = '1'
    ContainerId = 1
    ExpectedModifiedBoxesCount
= 8
    MsgModifiedBoxesCount = '8'
}
```

5) Clique com o botão direito do mouse no objeto *Test Suite* e clique em “*Run Suite*”. O *Test Suite* falha quando falha o unit test modificado.



6) Dentro do painel de edição do Suite, clique na coluna **Continue on error** para o teste **SetCostToBoxesInContainerUnitTest** para que não seja verificado.



7) Execute o *Test Suite* novamente. Clique com o botão direito do mouse no objeto *Test Suite* e clique em "Run Suite". Você deve obter o seguinte resultado:

The screenshot shows the Test Results panel. The 'Test Suite1' execution details are displayed, including the start and end times, and the elapsed time. The 'Execution results' section shows a tree view where 'Test Suite1' is marked as failed, and 'SetCostToBoxesInContainerUnitTest' is marked as failed (333 ms). 'WebUITest1' is marked as skipped (»).

Test suite execution details

Start: Thursday, May 13, 2021 8:18:52 PM
Elapsed time: < 1 sec

Summary	Count	Percentage
Error	1	50.00 %
Skipped	1	50.00 %
Total	2	100.00 %

Como pode ser visto, o teste **WebUITest1** não foi executado porque o Test Suite foi configurado para não continuar a execução se o teste **SetCostToBoxesInContainerUnitTest** falhar.



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