Mocking de Base de Datos

Testing en GeneXus

Introducción

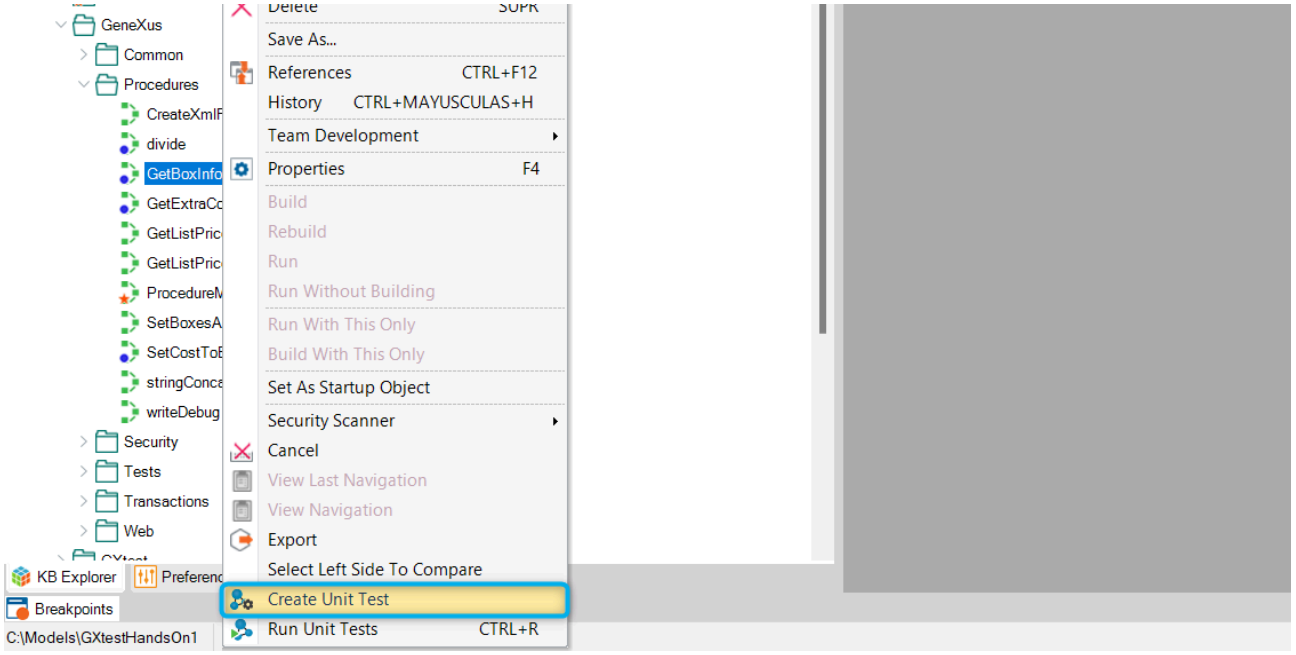
En esta capacitación práctica dispondrá de una guía para poner en práctica la funcionalidad de Mocking de Base de datos para realizar simulaciones de base de datos en la ejecución de las pruebas (Mocking).

Ejecución con mocking de Base de Datos

Supongamos que para nuestras pruebas necesitamos que el *Box* 1 del *Container 1* siempre nos devuelva el precio seteado en el ejercicio anterior, independientemente de los cambios que puedan realizarse sobre la base de datos.

Para ello grabaremos el estado actual de dicho valor utilizando la funcionalidad **Database Mocking**.

1) Crear un Unit Test para el procedimiento **"GetBoxInfo"** de nuestra KB, haciendo clic derecho sobre el mismo y seleccionando la opción *'Create Unit Test'*.

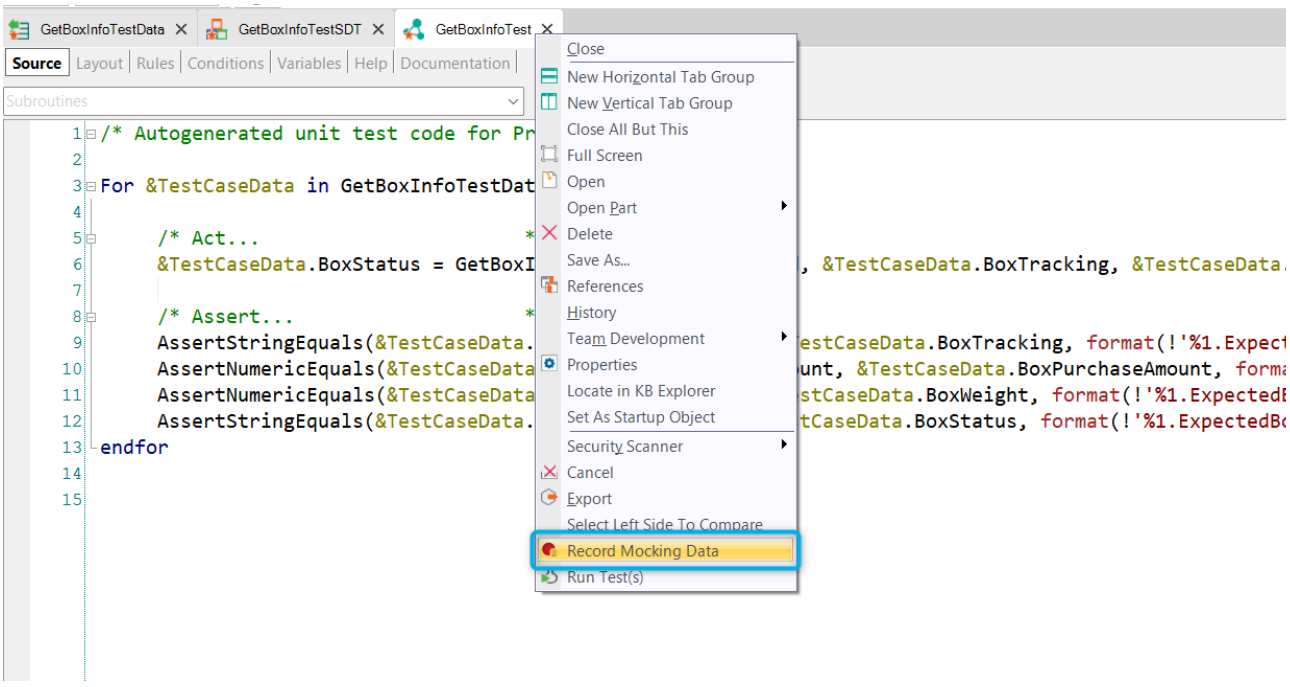


2) Abrir el objeto “GetBoxInfoTestData” y editar el valor ***ExpectedBoxCost = 48.3.*** Si queremos que todas las assertions pasen, también completamos el resto de los valores con los datos de *BoxId=1* que se pueden ver en la aplicación.

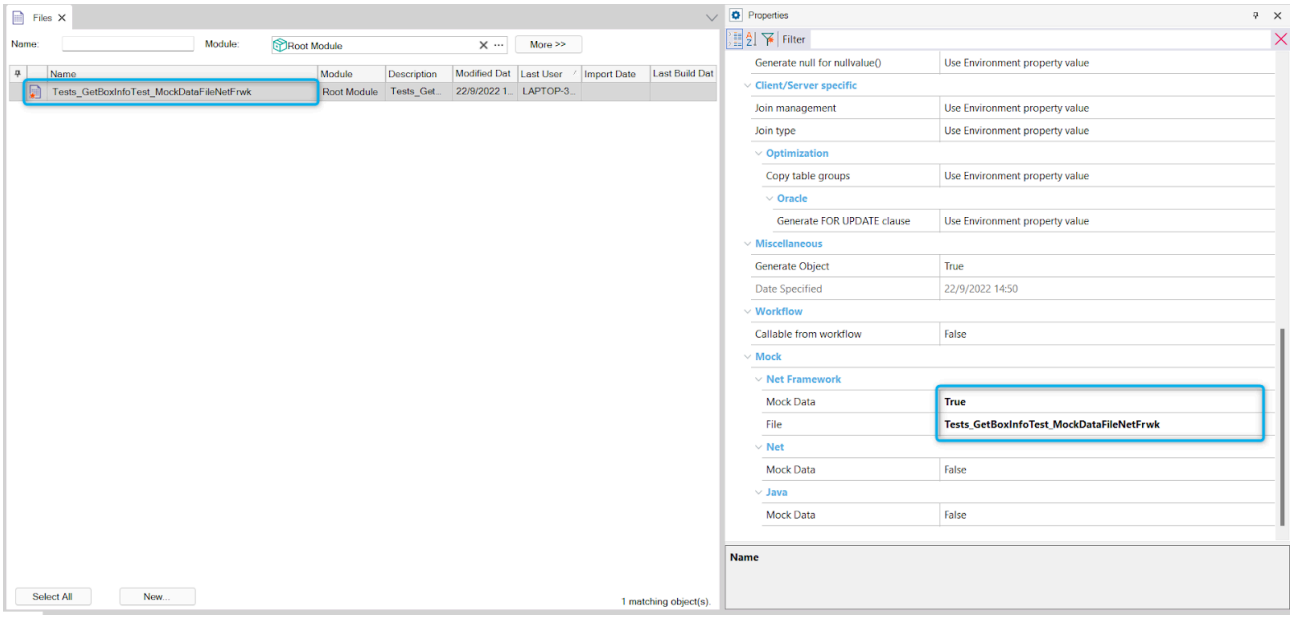
El código del Data Provider es:

```
GetBoxInfoTestSDT
{
        TestCaseId = '1'
        BoxId = 1
        ExpectedBoxTracking = '1146HFGDS3'
        MsgBoxTracking = '1146HFGDS3'
        ExpectedBoxPurchaseAmount = 180
        MsgBoxPurchaseAmount = '180'
        ExpectedBoxWeight = 2.3
        MsgBoxWeight = '2.3'
        ExpectedBoxStatus = Status.ATDESTINATION
        MsgBoxStatus = 'Status.ATDESTINATION'
        ExpectedBoxOriginArrivalDate = #2018-10-22#
        MsgBoxOriginArrivalDate = '#2018-10-22#'
        ExpectedBoxCost = 48.3
        MsgBoxCost = '48.3'
}
```

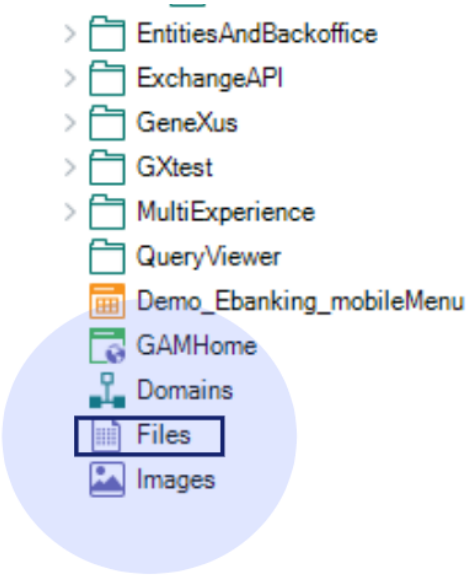
3) Guardar los cambios, hacer clic derecho sobre el unit test y seleccionar la opción *“Record Mocking Data”.*



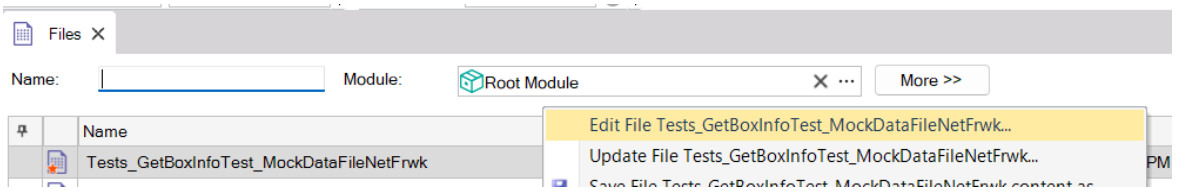
4) Vemos que se crea un archivo en *Files* con el nombre *'Tests_GetBoxInfoTest_MockDataFileNetFrwk.* La propiedad **Mock Data** del test unitario cambia a 'True' y en File se asigna automáticamente el archivo creado.



5) Hacer doble click en KB Explorer -> Files para identificar en el listado y abrir el archivo grabado



6) Seleccione “Edit file…..” haciendo click derecho en el archivo grabado



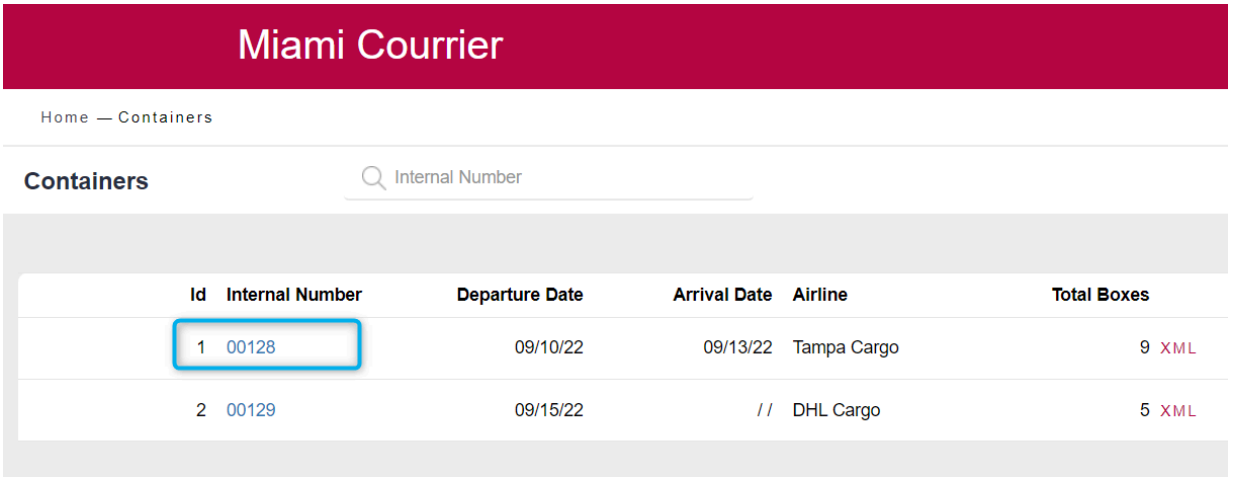

```
{
        "_array": [
                {
                        "Data": {
                                "innerArray": [
                                        [
                                                1,
                                                48.3000,
                                                "1146HFGDS3",
                                                180.0000,
                                                2.3000,
                                                "ATDESTINATION",
                                                "2018-10-22T00:00:00.000"
                                        ]
                                ]
                        },
                        "Key": "1,True,SELECT [BoxId], [BoxCost], [BoxTracking], [BoxPurchaseAmount], [BoxWeight], [BoxStatus],
[BoxOriginArrivalDate] FROM [Box] WHERE [BoxId] = @AV8BoxId ORDER BY [BoxId] ",
                        "KeyPattern": ""
                },
                null,
                null,
                null
        ],
        "_head": 0,
        "_size": 1,
        "_tail": 1,
        "_version": 2
}
```


Observe que se graba en el campo Key la sentencia SQL SELECT ejecutada por el procedure y el resultado de la misma en el campo Data, el cual fue obtenido desde la base de datos de la aplicación.

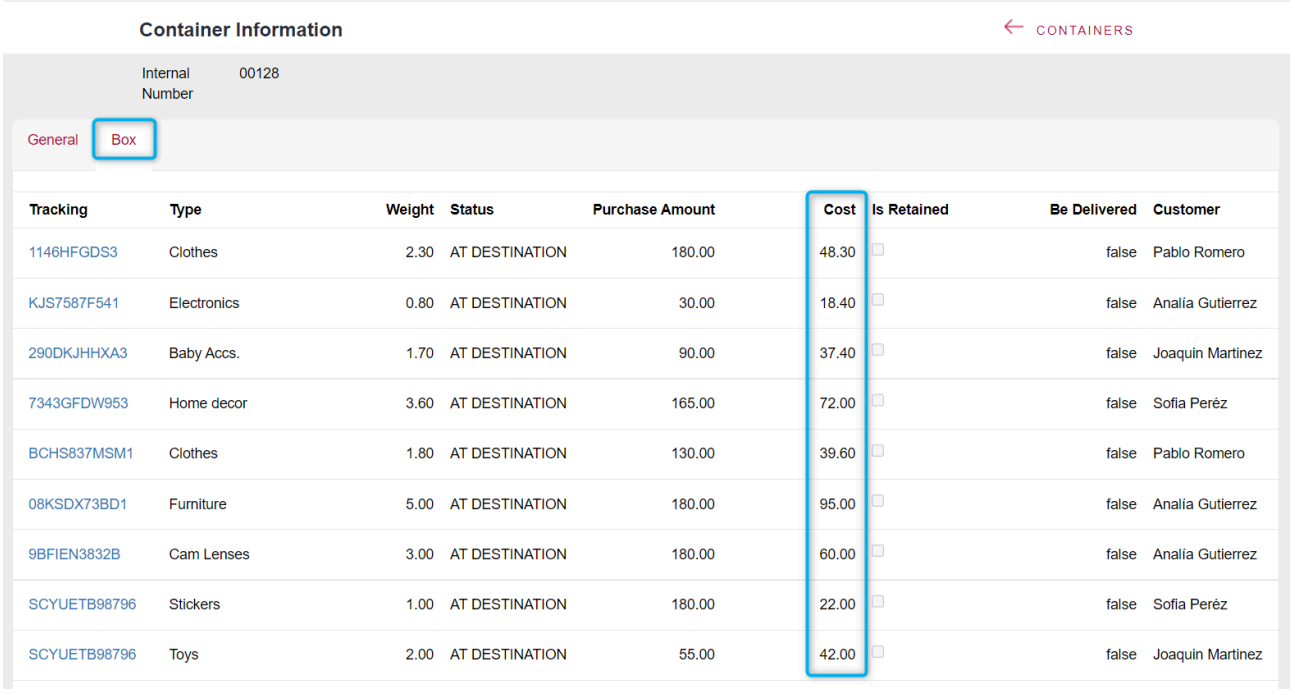
Comentario:
El JSON generado para el ambiente Java es diferente al de la captura ya que el de la captura es el JSON generado para el ambiente .Net.

A continuación, editaremos los datos de la base de datos para verificar que el test está efectivamente ejecutando sobre el json y no sobre la base de datos:

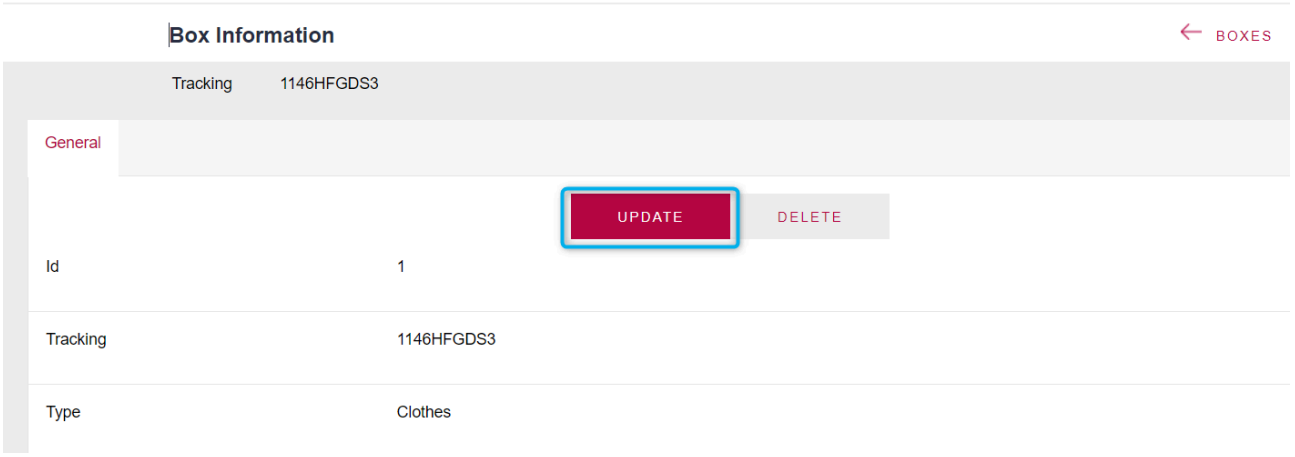
7) Hacer clic en *“Containers”* y luego clic sobre el link *“00128”* del contenedor de Id 1 para ver la información del contenedor.



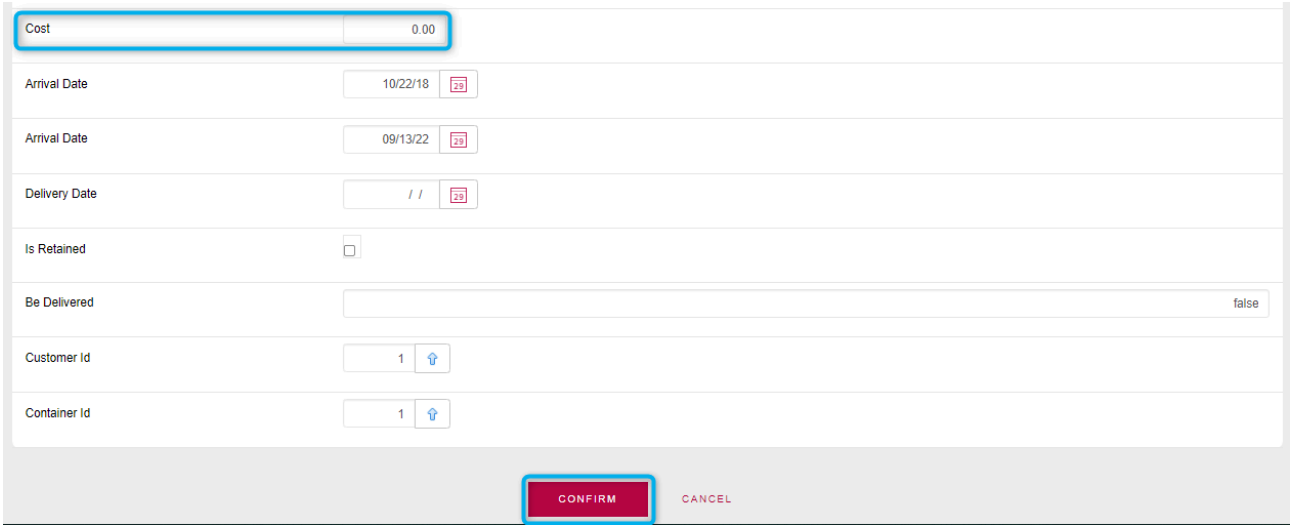
8) Hacer clic en la pestaña Box. Podrá observar en la columna Cost que todas las cajas tienen los valores seteados por el Proc *SetCostToBoxesInContainer*.



9) Seleccionar la Box 1 y hacer clic en Update

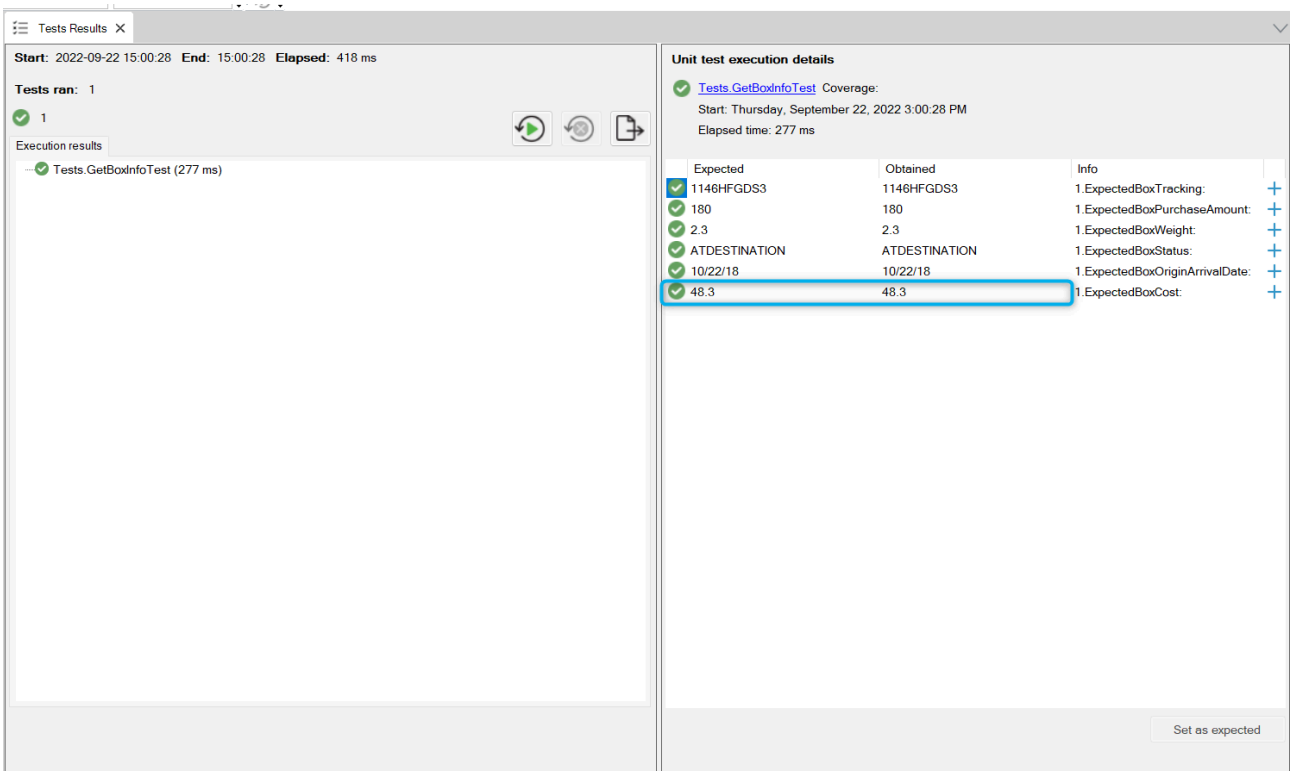


10) En la celda Cost cambiar el valor 48.30 por 0.00 y hacer clic en Confirm



11) Correr el test nuevamente haciendo clic en “Run Again” Esta vez se ejecutará con los datos grabados mediante mock.

Al terminar la ejecución del *test*, se mostrarán los resultados en el panel *Tests Results.* Vemos que el costo ‘48.3’, es obtenido desde el archivo grabado con mock, y no desde la base de datos editada anteriormente.



***Comentarios adicionales al práctico**:*

La funcionalidd es tiene como principal objetivo aislar los resultados del tests del estado de la base de datos

El archivo de mock es util o sirve siempre y cuando las sentencias que genera GeneXus durante la ejecución del test sean las mismas que las que se grabaron en el archivo de mock.

En caso de que exista una diferencia entre las sentencias que genera GeneXus durante la ejecución y las sentiencias grabadas en el archivo Mock, la ejecución se hará sobre la base de datos

abstracta

MONTEVIDEO - URUGUAY	Sarmiento 2465, 11300	+598 2711 0561
SAN FRANCISCO - USA	100 Pine St., Ste. 1250, CA 94111	+1 415 745 3678
LONDRES - UK	71-75 Shelton Street, WC2H 9JQ	+44 203 696 6682