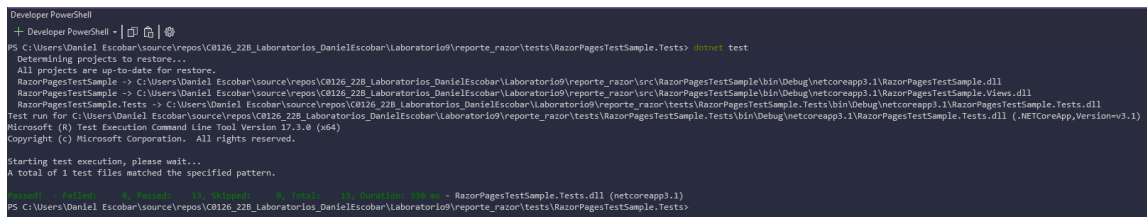


Laboratorio 9 - reporte_razor

Enlace de mi repositorio:

https://github.com/DanielEscobar19/C0126_22B_Laboratorios_DanielEscobar

Imagen mostrando que se ejecutan los test del proyecto



```
Developer PowerShell
PS C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\tests\RazorPagesTestSample.Tests> dotnet test
Determining projects to restore...
All projects are up-to-date for restore.
RazorPagesTestSample -> C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\src\RazorPagesTestSample\bin\Debug\netcoreapp3.1\RazorPagesTestSample.dll
RazorPagesTestSample -> C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\src\RazorPagesTestSample\bin\Debug\netcoreapp3.1\RazorPagesTestSample.Views.dll
RazorPagesTestSample.Tests -> C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\tests\RazorPagesTestSample.Tests\bin\Debug\netcoreapp3.1\RazorPagesTestSample.Tests.dll
Test run for C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\tests\RazorPagesTestSample.Tests\bin\Debug\netcoreapp3.1\RazorPagesTestSample.Tests.dll (.NETCoreApp,Version=v3.1)
Microsoft (R) Test Execution Command Line Tool Version 17.3.0 (x64)
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.
PS C:\Users\Daniel_Escobar\source\repos\C0126_22B_Laboratorios_DanielEscobar\Laboratorio9\reporte_razor\tests\RazorPagesTestSample.Tests>
```

Preguntas:

A) En el tercer bloque de código, explique brevemente el propósito de la línea de código:

```
var db = new AppDbContext(Utilities.TestDbContextOptions())
```

Esta línea de código permite crear una base de datos nueva cada vez que se prueba el método. De esta forma las pruebas pueden ser atómicas porque se crea una base provisional desde 0. Así si una prueba anterior modificó la base de datos las siguientes pruebas no se ven afectadas. Además, se puede modificar esta base de datos provisional para que tenga datos necesarios para la prueba sin necesidad de modificar la base de datos de testing o de producción.

B) En el quinto bloque de código, explique brevemente ¿Por qué los casos de prueba unitarios tienen la etiqueta [Fact] antes de la firma del método de prueba?

Esta etiqueta es la que se utiliza en el framework xUnit para indicar/marcar los métodos que van a ser probados. Podríamos decir que es el equivalente a [TestMethod] en MSTest.

Fact siempre revisa que los assert den verdadero y no varía según el tipo de datos, por otro lado existe theory solo dan dar verdadero con un conjunto particular de datos.

C) Para los siguientes dos casos de prueba:

DeleteMessageAsync_MessagesDeleted_WhenMessagesFound()
DeleteMessageAsync_NoMessagesDeleted_WhenMessagesNotFound()

Indique cuál método del programa están probando, y explique brevemente el objetivo de cada uno de estos casos de prueba.

DeleteMessageAsync_MessagesDeleted_WhenMessagesFound()

Este método prueba que cuando se elimina un mensaje de la base de datos solo se elimine dicho mensaje y que no se vean afectados el resto de los mensajes. Para esto el método elimina un mensaje de la base de datos y revisa en el assert que los mensajes que quedan en las base no se hayan modificado. Para esto saca una lista con los mensajes que hay en la base antes de la eliminación ignorando el que se va a eliminar y luego de la eliminación del mensaje compara los mensajes que hay en la base con la lista que creó antes de borrar el mensaje. Si las dos listas son iguales, la prueba es exitosa.

DeleteMessageAsync_NoMessagesDeleted_WhenMessagesNotFound()

En este caso, el método verifica que si se trata de eliminar un mensaje que no existe que la base de datos no sea alterada de ninguna forma. Para esto crea una lista con los mensajes de la base antes de la eliminación, luego realiza la eliminación del mensaje que no existe y finalmente saca la lista de mensajes después de la eliminación y la compara con la lista de mensajes antes de la eliminación. Si las dos listas son iguales, la prueba es exitosa.

Casos de prueba en el PI:

En el tutorial se enseña como hacer unit test del data access layer (DAL). En nuestro caso se realiza testing de los métodos de los handlers porque estos son los que manejan el acceso a los datos.

Nombres clase y método
Clase: EstructuraOrgHandler Método: EliminarPuesto(PuestoModel puestoEliminar)
Nombre del caso de prueba
EliminarPuesto_PuestoEsEliminado_CuandoPuestoEsEncontrado()
Objetivo de la prueba
Verifica que un puesto es eliminado de la base cuando se llama al método de eliminar y este no afecta el resto de puestos de la base.
Nombre de clase reemplazable con un mock
PuestoModel Además, para poder realizar la prueba, antes de probar el método de borrar el método de testing inserta en la base de datos un puesto ficticio de prueba. Luego de esto elimina de la base los datos ficticios.

Nombres clase y método
Clase: EstructuraOrgHandler Método: ActualizarPuesto(string nombrePuesto, PuestoModel puestoInsertar)
Nombre del caso de prueba
ActualizarPuesto_PuestoEsActualizado_CuandoPuestoEsEncontrado()
Objetivo de la prueba
Verifica que un puesto es actualizado de la base cuando se llama al método de actualizar y esta actualización no afecta el resto de puestos de la base.
Nombre de clase reemplazable con un mock
PuestoModel Este método recibe un <i>string nombrePuesto</i> el cual representa el nombre antes de la modificación que es utilizado para encontrar el puesto en la base. Este también se puede reemplazar con un mock.

Además, para poder realizar la prueba, antes de probar el método de actualizar el método de testing inserta en la base de datos un puesto ficticio de prueba. Luego de esto elimina de la base los datos ficticios.

Nombres clase y método
Clase: Handler Método: obtenerNegocioDeAnalisis(DateTime fechaAnalisis)
Nombre del caso de prueba
SeRetornaNegocio_CuandoSeEncuentraAnalisis()
Objetivo de la prueba
Verifica que se retorna el modelo negocio correspondiente según la fecha de análisis que se recibe. Solo se debe retornar un negocio y que el análisis con la fecha que recibe por parámetro si pertenezca al negocio retornado.
Nombre de clase reemplazable con un mock
<div>DateTime fechaAnalisis</div> <p>Además, para poder realizar la prueba, antes de probar el método de obtener negocio se va a insertar un negocio con un análisis ficticio para poder saber qué resultados esperar. Luego de esto elimina de la base los datos ficticios.</p>

Nombres clase y método
Clase: EstructuraOrgHandler Método: ObtenerListaDePuestos(DateTime fechaAnalisis)
Nombre del caso de prueba
SeRetornaListaPuestos_CuandoSeEncuentraAnalisis()
Objetivo de la prueba
Verifica que se retorna una lista de puestos con todos los puestos de un análisis. Solo se debe retornar los puestos del análisis con la fecha indicada en el parámetro del método y no se debe modificar ningún dato dentro de la base.

Nombre de clase reemplazable con un mock
<div>DateTime fechaAnálisis</div> <p>Además, para poder realizar la prueba, antes de probar el método de obtener puestos el método de testing va insertar un análisis con puestos fijos para poder saber qué resultados esperar. Luego de esto elimina de la base los datos ficticios.</p>