



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**PRACTICA DE LABORATORIO 05**

Curso: Calidad de software

Docente: Ing. Patrick Cuadros Quiroga

Integrantes:

**Yanqui Chambilla, Rodrigo Jesus (2017059555)**

**Tacna – Perú**

**2020**

## 1. Procedimiento

Se utilizara el framework CoreBDD. Instalar la plantilla de proyecto dotnet a traves de dotnet new -i corebdd.projecttemplate

```
C:\Users\Randi\carpeta de prueba>dotnet new -i corebdd.projecttemplate
Determinando los proyectos que se van a restaurar...
C:\Program Files\dotnet\sdk\5.0.100\Sdks\Microsoft.NET.Sdk\targets\Microsoft.NET.EolTargetFrameworks.targets(28,5): warning: El framework de destino 5.0 no es compatible con este técnico y no recibirá actualizaciones de seguridad en el futuro. Para obtener más información sobre la directiva de destino, consulte https://aka.ms/etp.
C:\Users\Randi\templateengine\dotnetcli\v5.0.100\scratch\restore.csproj]
Se ha restaurado C:\Users\Randi\templateengine\dotnetcli\v5.0.100\scratch\restore.csproj (en 467 ms).
```

Templates	Short Name	Language	Tags
Console...	console	[C#], F#, VB	Common/Console
Class Library...	classlib	[C#], F#, VB	Common/Library
WPF App...	wpf	[C#], VB	Common/WPF
WPF Class Library...	wpflib	[C#], VB	Common/WPF
WPF Custom Control Library...	wpfcustomcontrollib	[C#], VB	Common/WPF
WPF User Control Library...	wpfusercontrollib	[C#], VB	Common/WPF
Windows Forms...	winforms	[C#], VB	Common/WinForms
Windows Forms Custom Control Library...	winformscontrollib	[C#], VB	Common/WinForms
Windows Forms Class Library...	winformslib	[C#], VB	Common/WinForms
Worker...	worker	[C#], F#	Common/Worker/Web
CoreBDD...	corebdd	[C#]	CoreBDD/xUnit/BDD/TDD
Unit Test...	mstest	[C#], F#, VB	Test/MSTest
Unit Test...	nunit	[C#], F#, VB	Test/NUnit
Unit Test...	nunit-test	[C#], F#, VB	Test/NUnit
Unit Test...	xunit	[C#], F#, VB	Test/xUnit
Razor Component...	razorcomponent	[C#]	Web/ASP.NET

Luego cree una nueva carpeta para su proyecto de prueba y ejecute dotnet new corebdd  
Alternativamente, puede agregar CoreBDD a un proyecto de prueba xUnit existente a través del paquete nuget dotnet add package CoreBDD

```
C:\Users\Randi\carpeta de prueba>dotnet new corebdd
The template "CoreBDD Test Project Template" was created successfully.

C:\Users\Randi\carpeta de prueba>dotnet add package CoreBdd
Determinando los proyectos que se van a restaurar...
Writing C:\Users\Randi\AppData\Local\Temp\tmpA5E2.tmp
Info : Agregando PackageReference para el paquete "CoreBdd" al proyecto "C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj".
Info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/corebdd/index.json
Info : Restaurando paquetes para C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj...
Info : CACHE https://api.nuget.org/v3-flatcontainer/corebdd.specgeneration/index.json
Info : El paquete "CoreBdd" es compatible con todos los marcos de trabajo especificados del proyecto.
Info : Se actualizó PackageReference para la versión "1.0.0" del paquete "CoreBdd" en el archivo "C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj".
Info : Ejecutando restauración...
Info : Generación de archivo MSBuild C:\Users\Randi\carpeta de prueba\obj\carpeta de prueba.csproj
Info : Generación de archivo MSBuild C:\Users\Randi\carpeta de prueba\obj\carpeta de prueba.csproj
Info : Escribiendo el archivo de recursos en el disco. Ruta de acceso: C:\Users\Randi\carpeta de prueba\obj\carpeta de prueba.csproj
Log : Se ha restaurado C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj (en 557 ms).

C:\Users\Randi\carpeta de prueba>
```

También hay una herramienta de línea de comandos opcional para ejecutar pruebas con salida personalizada, clases de prueba de andamios (características / escenarios / pasos) y generación de código bidireccional (pruebas de Gherkin a CoreBDD y viceversa). Hay más documentación disponible sobre la CLI en la parte inferior de esta página. dotnet tool install -g corebdd.commandline

```
C:\Users\Randi\carpeta de prueba>
C:\Users\Randi\carpeta de prueba>dotnet tool install -g corebdd.commandline
Puede invocar la herramienta con el comando siguiente: corebdd
La herramienta "corebdd.commandline" (versión '1.0.0') se instaló correctamente.

C:\Users\Randi\carpeta de prueba>
```

## 2. Escribir pruebas de CoreBDD

Siguiendo el ejemplo habitual de la calculadora, podemos comenzar con el siguiente modelo para probar

```
Calculator.cs X
C: > Users > Randi > carpeta de prueba > Model > Calculator.cs
1 namespace carpeta_de_prueba
2 {
3     public class Calculator
4     {
5         public string Equation { get; set; }
6         public int Total { get; set; }
7         public int Add(int x, int y) => x + y;
8         public int Subtract(int x, int y) => x - y;
9         public Calculator Key(int value) { Total = value; Equation = value.ToString(); return this; }
10        public Calculator Add(int value) { Total += value; Equation += " + " + value; return this; }
11        public Calculator Subtract(int value) { Total -= value; Equation += " - " + value; return this; }
12        public Calculator Multiply(int value) { Total *= value; Equation += " x " + value; return this; }
13        public Calculator Divide(int value) { Total /= value; Equation += " / " + value; return this; }
14        public Calculator Clear() { Total = 0; Equation = ""; return this; }
15    }
16 }
```

Podemos definir una Característica para cotejar un conjunto de escenarios derivando de la clase base Especificacion y decorando con el atributo Característica. Tenga en cuenta que ambos constructores son necesarios para admitir los diferentes estilos de sintaxis de prueba.

```
CalculatorFeature.cs X
CalculatorFeature.cs
1 using CoreBDD;
2
3 namespace carpeta_de_prueba
4 {
5     [Feature("Calculator",
6         @"In order to avoid silly mistakes
7         As a math idiot
8         I want to be told the sum of two numbers")]
9     public class CalculatorFeature : Specification
10    {
11        public CalculatorFeature(SpecFixture context):base(context)
12        {
13        }
14
15        public CalculatorFeature()
16        {
17        }
18    }
19
20 }
```

Una vez que hemos creado nuestra Característica base, tenemos varios sabores diferentes para escribir pruebas, primero podemos generar un escenario por clase (similar a las pruebas de estilo Cucumber) con un metodo para cada paso de Dado / Cuando / Entonces. Para hacer esto, simplemente herede de la nueva clase de entidad, decore con un atributo de ejemplo y proporcione los metodos Given, When, Then que se ejecutaran en orden

```
AddTwoNumbers.cs X
C: > Users > Randi > carpeta de prueba > Features > Calculator > Scenarios > AddTwoNumbers.cs
1  using CoreBDD;
2
3  namespace carpeta_de_prueba
4  {
5      [Example("Add two numbers")]
6      public class AddTwoNumbers : CalculatorFeature
7      {
8          readonly Calculator calc;
9
10         public AddTwoNumbers(SpecFixture state)
11             : base(state) => calc = new Calculator();
12
13         [Given("I have entered {0} into the calculator", 1)]
14         public void GivenStep(int first) => Context.Given.First = first;
15
16         [And("I have also entered {0} into the calculator", 2)]
17         public void AndStep(int second) => Context.Given.Second = second;
18
19         [When("I press add")]
20         public void WhenStep() => Context.When = calc.Add(Context.Given.First, Context.Given.Second);
21
22         [Then("the result should be {0}", 3)]
23         public void ThenStep(int result) => Context.Result.ShouldBe(result);
24     }
25 }
```

Tambien puede definir escenarios en un solo metodo utilizando delegados para cada uno de los pasos y permitiendo que se definan multiples escenarios dentro de la misma clase

```
5  {
6      public class AdvancedCalculator : CalculatorFeature
7      {
8          Calculator calculator;
9
10         [Scenario("Multiply two numbers")]
11         public void MultiplyTwoNumbers()
12         {
13
14             Given("I have a calculator", () => calculator = new Calculator());
15             When("I key in 10", () => calculator.Key(10));
16             And("I key in 5 and press multiply", () => calculator.Multiply(5));
17             Then("It sets the Total to 50", () => calculator.Total.ShouldBe(50));
18             And("It sets the equation to 10 x 5", () => calculator.Equation.ShouldBe("10 x 5"));
19         }
20
21         [Scenario("Divide two numbers")]
22         public void DivideTwoNumbers()
23         {
24             Given("I have a calculator", () => calculator = new Calculator());
25             When("I key in 10", () => calculator.Key(10));
26             And("I key in 5 and press divide", () => calculator.Divide(5));
27             Then("It sets the Total to 2", () => calculator.Total.ShouldBe(2));
28         }
29     }
30 }
```



La sintaxis basada en metodos tambien admite pruebas basadas en datos, utilizando xUnit InlineData (los escenarios basados en clases aun no admiten pruebas basadas en datos).

```
[ScenarioOutline("Divide two numbers")]
References
public void DivideTwoNumbers()
{
    Given("I have a calculator", () => calculator = new Calculator());
    When("I key in 10", () => calculator.Key(10));
    And("I key in 5 and press divide", () => calculator.Divide(5));
    Then("It sets the Total to 2", () => calculator.Total.ShouldBe(2));
}
[Examples(10, 2, 5)]
[Examples(20, 4, 5)]
References
public void DivideTwoNumbers(int number, int divideby, int result)
{
    Given($"I have a calculator", () => calculator = new Calculator());
    When($"I key in {number}", () => calculator.Key(number));
    And($"I key in {divideby} and press divide", () => calculator.Divide(divideby));
    Then($"It sets the Total to {result}", () => calculator.Total.ShouldBe(result));
    //And($"It sets the equation to {number} / {divideby}", () => calculator.Equation.ShouldBe($"{number} / {divideby}"));
}
```

Puede generar especificaciones de Gherkin de sus pruebas utilizando la biblioteca de extensiones CoreBDD.SpecGeneration , ya sea llamando desde una aplicacion o herramienta de linea de comandos y pasando la ruta al ensamblado que contiene las pruebas, o conectando su proyecto de prueba para generar las especificaciones despues de prueba de funcionamiento. Para hacer esto ultimo, primero haga referencia a la biblioteca CoreBDD.SpecGeneration

```
PS C:\Users\Randi> cd "carpeta de prueba"
PS C:\Users\Randi\carpeta de prueba> dotnet add package CoreBDD.SpecGeneration
Determinando los proyectos que se van a restaurar...
Writing C:\Users\Randi\AppData\Local\Temp\tmp211.tmp
info : Agregando PackageReference para el paquete "CoreBDD.SpecGeneration" al proyecto "C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/corebdd.specgeneration/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/corebdd.specgeneration/index.json 1010 ms
info : Restaurando paquetes para C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj...
info : El paquete "CoreBDD.SpecGeneration" es compatible con todos los marcos de trabajo especificados del proyecto
info : Se actualizó PackageReference para la versión "1.0.0" del paquete "CoreBDD.SpecGeneration" en el archivo "C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj".
info : Ejecutando restauración...
info : Escribiendo el archivo de recursos en el disco. Ruta de acceso: C:\Users\Randi\carpeta de prueba\obj\project.assets.json
info : Se ha restaurado C:\Users\Randi\carpeta de prueba\carpeta de prueba.csproj (en 477 ms).
PS C:\Users\Randi\carpeta de prueba>
```

Cree una clase de Fixture dentro de su proyecto de prueba y llame a GenerateSpecs.OutputFeatureSpecs dentro del metodo Dispose, pasando el ensamblado (o ruta al ensamblado) y la carpeta de salida para las especificaciones generadas.

```

GenerateSpecsFixture.cs X
C:\Users\Randi\carpeta de prueba\Fixtures\GenerateSpecsFixture.cs
1 using System;
2 using System.IO;
3 using CoreBDD.SpecGeneration;
4 using Xunit;
5
6 namespace carpeta_de_prueba
7 {
8     [CollectionDefinition("CoreBDD")]
9     public class Collection : ICollectionFixture<GenerateSpecsFixture> { }
10
11     public class GenerateSpecsFixture : IDisposable
12     {
13         public void Dispose()
14         {
15             //var specsFolder = Directory.GetParent(System.Reflection.Assembly.GetExecutingAssembly().Location).Parent;
16             //GenerateSpecs.OutputFeatureSpecs(this.GetType().Assembly.Location, specsFolder + @"Specs");
17             GenerateSpecs.OutputFeatureSpecs(this.GetType().Assembly.Location, @"..\..\Specs");
18         }
19     }
20 }

```

Cuando las pruebas terminan de ejecutarse, se genera un archivo FeatureName.feature en la carpeta Especificaciones del proyecto de prueba xUnit. Genera especificaciones de Gherkin para la característica y los escenarios relacionados. Ejemplo CalculatorFeature.feature :

```

CalculatorFeature.cs CalculatorFeature.feature X
C:\Users\Randi\carpeta de prueba\Specs\CalculatorFeature.feature
1 # This file is auto-generated, any changes made to this file will be lost
2
3
4 Feature: Calculator
5     In order to avoid silly mistakes
6     As a math idiot
7     I want to be told the sum of two numbers
8
9 Scenario: Add two numbers
10     Given I have entered 1 into the calculator
11     And I have also entered 2 into the calculator
12     When I press add
13     Then the result should be 3
14
15 Scenario: Multiply two numbers
16     Given I have a calculator
17     When I key in 10
18     And I key in 5 and press multiply
19     Then it sets the total to 50
20     And It sets the equation to 10 x 5
21
22 Scenario: Divide two numbers
23
24 Scenario: Subtract two numbers
25     Given I have entered 5 into the calculator
26     And I have also entered 2 into the calculator
27     When I press minus
28     Then the result should be 3

```

### 3. Herramienta de linea de comando

La herramienta de linea de comandos facilita la ejecucion de tareas como la ejecucion de pruebas con la salida de estilo Gherkin, la generacion de archivos de prueba de caracterissticas y escenarios predeterminados y la generacion de archivos de funciones de Gherkin a partir de pruebas existentes o la generacion de pruebas a partir de archivos

```
S C:\Users\Randi\carpeta de prueba> mkdir demobdd

Directorio: C:\Users\Randi\carpeta de prueba

Mode                LastWriteTime         Length Name
----                -
05/12/2020      18:50                demobdd

S C:\Users\Randi\carpeta de prueba> cd demobdd
S C:\Users\Randi\carpeta de prueba\demobdd> dotnet new corebdd
The template "CoreBDD Test Project Template" was created successfully.
S C:\Users\Randi\carpeta de prueba\demobdd> corebdd test

Unhandled Exception: System.Reflection.TargetInvocationException: Exception has been thrown by the target of invocation.
   at System.IO.Enumeration.FileSystemEnumerator`1.CreateDirectoryHandle(String path, Boolean ignoreNotFound)
   at System.IO.Enumeration.FileSystemEnumerator`1..ctor(String directory, EnumerationOptions options)
   at System.IO.Enumeration.FileSystemEnumerable`1..ctor(String directory, FindTransform transform, EnumerationOptions options)
   at System.IO.Enumeration.FileSystemEnumerableFactory.UserFiles(String directory, String expression, EnumerationOptions options)
   at System.IO.Directory.InternalEnumeratePaths(String path, String searchPattern, SearchTarget searchTarget)
   at CoreBDD.CommandLine.Tools.Analyzer.FindCoreBDDAssemblies.Find(String path) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 105
   at CoreBDD.CommandLine.Tools.TestRunner.Run(Boolean generateSpecs, String path, String output) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 97
   at CoreBDD.CommandLine.Program.OnExecute() in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 31
--- End of inner exception stack trace ---
   at System.RuntimeMethodHandle.InvokeMethod(Object target, Object[] arguments, Signature sig, Boolean coerced, Boolean wrapExceptions)
   at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture)
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.Invoke(MethodInfo method, Object[] args) in C:\Users\Steven\Source\CoreBDD\src\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 86
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.OnExecute(ConventionContext context) in C:\Users\Steven\Source\CoreBDD\src\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 64
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.<>c__DisplayClass0_0.<<ApplyConvention>>b__0() in C:\Users\Steven\Source\CoreBDD\src\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 25
-- End of stack trace from previous location where exception was thrown --
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.<>c__DisplayClass126_0.<OnExecute>b__0() in C:\Users\Steven\Source\CoreBDD\src\Extensions\CommandLineUtils\CommandLineApplication.cs:line 31
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.Execute[TApp](CommandLineContext context) in C:\Users\Steven\Source\CoreBDD\src\Extensions\CommandLineUtils\CommandLineApplication.cs:line 97
   at CoreBDD.CommandLine.Program.Main(String[] args) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 31
S C:\Users\Randi\carpeta de prueba\demobdd>
```

Ejecute pruebas y luego genere archivos .feature de Gherkin en una ubicacion especifica  
corebdd test specs output ./Specs



```

S C:\Users\Randi\carpeta de prueba\demobdd>
S C:\Users\Randi\carpeta de prueba\demobdd> corebdd test --specs --output ./Specs

Unhandled Exception: System.Reflection.TargetInvocationException: Exception has been thrown by the target of an invocation.
the path 'C:\Users\Randi\carpeta de prueba\demobdd\bin\Debug'.
   at System.IOEnumeration.FileSystemEnumerator`1.CreateDirectoryHandle(String path, Boolean ignoreNotFound)
   at System.IOEnumeration.FileSystemEnumerator`1..ctor(String directory, EnumerationOptions options)
   at System.IOEnumeration.FileSystemEnumerable`1..ctor(String directory, FindTransform transform, EnumerationOptions options)
   at System.IOEnumeration.FileSystemEnumerableFactory.UserFiles(String directory, String expression, EnumerationOptions options)
   at System.IO.Directory.InternalEnumeratePaths(String path, String searchPattern, SearchTarget searchTarget, EnumerationOptions options)
   at CoreBDD.CommandLine.Tools.Analyzer.FindCoreBDDAssemblies.Find(String path) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Tools\Analyzer.cs:line 11
   at CoreBDD.CommandLine.Tools.TestRunner.Run(Boolean generateSpecs, String path, String output) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Tools\TestRunner.cs:line 11
   at CoreBDD.CommandLine.Program.OnExecute() in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 11
--- End of inner exception stack trace ---
   at System.RuntimeMethodHandle.InvokeMethod(Object target, Object[] arguments, Signature sig, Boolean constructor, Boolean wrapExceptions)
   at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo info)
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.Invoke(MethodInfo method, Object instance, Object[] arguments) in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 86
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.OnExecute(ConventionContext context) in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 64
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.<>c__DisplayClass0_0.<<Apply>b__0>d.MoveNext() in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 25
-- End of stack trace from previous location where exception was thrown ---
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.<>c__DisplayClass126_0.<OnExecute>b__0() in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 105
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.Execute[TApp](CommandLineContext context) in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 31
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.Execute[TApp](IConsole console, String[] args) in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 97
   at CoreBDD.CommandLine.Program.Main(String[] args) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 11
S C:\Users\Randi\carpeta de prueba\demobdd>

```

**Andamio de una clase de entidad CoreBDD llamada 'Iniciar sesion' en la carpeta actual corebdd generate feature name login namespace demobdd**

**Andamio de una clase de escenario CoreBDD llamada 'LoginToWebsite' en la funcion 'Iniciar sesion' corebdd generate scenario name LoginToWebsite feature login namespace demobdd**

```

S C:\Users\Randi\carpeta de prueba\demobdd> corebdd generate scenario --name LoginToWebsite --feature login --namespace demobdd
S C:\Users\Randi\carpeta de prueba\demobdd> corebdd test

Unhandled Exception: System.Reflection.TargetInvocationException: Exception has been thrown by the target of an invocation. ----> Sys
the path 'C:\Users\Randi\carpeta de prueba\demobdd\bin\Debug'.
   at System.IOEnumeration.FileSystemEnumerator`1.CreateDirectoryHandle(String path, Boolean ignoreNotFound)
   at System.IOEnumeration.FileSystemEnumerator`1..ctor(String directory, EnumerationOptions options)
   at System.IOEnumeration.FileSystemEnumerable`1..ctor(String directory, FindTransform transform, EnumerationOptions options)
   at System.IOEnumeration.FileSystemEnumerableFactory.UserFiles(String directory, String expression, EnumerationOptions options)
   at System.IO.Directory.InternalEnumeratePaths(String path, String searchPattern, SearchTarget searchTarget, EnumerationOptions options)
   at CoreBDD.CommandLine.Tools.Analyzer.FindCoreBDDAssemblies.Find(String path) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Tools\Analyzer.cs:line 11
   at CoreBDD.CommandLine.Tools.TestRunner.Run(Boolean generateSpecs, String path, String output) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Tools\TestRunner.cs:line 11
   at CoreBDD.CommandLine.Program.OnExecute() in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 47
--- End of inner exception stack trace ---
   at System.RuntimeMethodHandle.InvokeMethod(Object target, Object[] arguments, Signature sig, Boolean constructor, Boolean wrapExceptions)
   at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo info)
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.Invoke(MethodInfo method, Object instance, Object[] arguments) in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 86
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.OnExecute(ConventionContext context) in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 64
   at McMaster.Extensions.CommandLineUtils.Conventions.ExecuteMethodConvention.<>c__DisplayClass0_0.<<Apply>b__0>d.MoveNext() in C:\Users\Steven\Source\Extensions\CommandLineUtils\Conventions\ExecuteMethodConvention.cs:line 25
-- End of stack trace from previous location where exception was thrown ---
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.<>c__DisplayClass126_0.<OnExecute>b__0() in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 105
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.Execute[TApp](CommandLineContext context) in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 31
   at McMaster.Extensions.CommandLineUtils.CommandLineApplication.Execute[TApp](IConsole console, String[] args) in C:\Users\Steven\Source\Extensions\CommandLineUtils\CommandLineApplication.cs:line 97
   at CoreBDD.CommandLine.Program.Main(String[] args) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 11
S C:\Users\Randi\carpeta de prueba\demobdd>

```

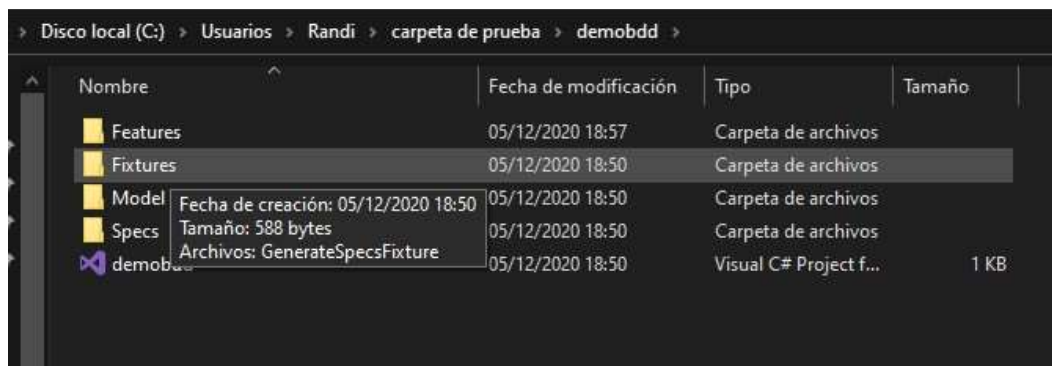
**Scaffold CoreBDD Pruebas de archivos existentes '.feature' de pepinillo, especificando la ubicacion de los archivos de características y la carpeta de destino para las pruebas generadas. - Si ha seguido utilizando el ejemplo de prueba 'corebdd', elimine la carpeta 'Caracteristicas' (dejando la carpeta Especificaciones con archivos .feature intactos) y luego ejecute: corebdd generate tests path ./Specs output ./Features namespace demobdd**

```

   at CoreBDD.CommandLine.Program.Main(String[] args) in C:\Users\Steven\Source\CoreBDD\src\CoreBDD.CommandLine\Program.cs:line 11
S C:\Users\Randi\carpeta de prueba\demobdd> corebdd generate scenario --name LoginToWebsite --feature login --namespace demobdd
S C:\Users\Randi\carpeta de prueba\demobdd> corebdd generate tests --path ./Specs --output ./Features --namespace demobdd
S C:\Users\Randi\carpeta de prueba\demobdd>

```

Ahora debería tener los apendices de prueba regenerados utilizando los escenarios de archivo .feature.



The screenshot shows a Windows File Explorer window with the address bar indicating the path: > Disco local (C:) > Usuarios > Randi > carpeta de prueba > demobdd >. The main area displays a list of files and folders. A tooltip is visible over the 'Model' folder, showing its creation date and size. The 'Fixtures' folder is highlighted.

Nombre	Fecha de modificación	Tipo	Tamaño
Features	05/12/2020 18:57	Carpeta de archivos	
Fixtures	05/12/2020 18:50	Carpeta de archivos	
Model	05/12/2020 18:50	Carpeta de archivos	
Specs	05/12/2020 18:50	Carpeta de archivos	
demobdd	05/12/2020 18:50	Visual C# Project f...	1 KB

Tooltip for 'Model':  
Fecha de creación: 05/12/2020 18:50  
Tamaño: 588 bytes  
Archivos: GenerateSpecsFixture