



*Las Americas Institute of Technology*

**NOMBRE:** Enjher Javier Agüero Ovalles

**MATRÍCULA:** 2023-1113

**CARRERA:** Tecnólogo en Desarrollo de Software

**MATERIA:** Programación Paralela

**TEMA:** Documentación Pruebas Unitarias

## **HOLA**

En su recorrido este documento albergará un reporte de pruebas unitarias realizadas en el proyecto. Las pruebas fueron ejecutadas utilizando el framework xUnit para verificar el comportamiento esperado de varias clases.

### **//Prueba 1: Correcto Uso de Pais.cs**

Con esta prueba verificamos que el modelo Pais.cs funciona correctamente. O sea, la prueba valida que el objeto pueda almacenar valores de entrada como un nombre y un PIB sin generar errores.

#### **Resultado esperado:**

- El nombre no debe estar vacío
- El PIB debe ser un valor positivo

#### **Resultado obtenido:**

- El objeto Pais se creó correctamente con los valores proporcionados.

Con esto podemos concluir en que la prueba valida la creación del objeto con nombre y PIB. En esta prueba se utilizó "China" como nombre, pero cualquier valor de país (como "República Dominicana") debería funcionar igualmente si cumple con las condiciones mencionadas.

## //Prueba 2: PibProjectionService

Con esta prueba verificamos que el método `ProyectarPibAjustado5Years` funciona correctamente para calcular el PIB ajustado y proyectado durante 5 años, utilizando una tasa de arancel y una tasa de crecimiento.

### **Resultado esperado:**

- El primer valor debe ser el PIB ajustado
- Los siguientes 4 valores deben ser la proyección anual con crecimiento aplicado

### **Resultado obtenido:**

- El método calculó correctamente el PIB ajustado y proyectó los valores durante 5 años.

### **Conclusión:**

La prueba valida que el cálculo y la proyección se realizan correctamente, considerando la tasa de arancel y crecimiento.

## 2.1: Excepción con Valor Negativo

En esta prueba verificamos que el método `ProyectarPibAjustado5Years` lance una excepción (`ArgumentException`) cuando se le pase un valor negativo para el PIB original, ya que el PIB no puede ser negativo.

### Resultado esperado:

- Se debe lanzar una `ArgumentException` cuando el valor de `pibOriginal` sea negativo.

### Resultado obtenido:

- La prueba pasó correctamente, lanzando la excepción como se esperaba.

### Conclusión:

Esta prueba valida que el sistema maneja correctamente los valores negativos de PIB, asegurando que no se realicen cálculos con un PIB inválido.

## **2.2: Crecimiento Cero**

Esta prueba verifica que el método `ProyectarPibAjustado5Years` funcione correctamente cuando la tasa de crecimiento sea cero. En este caso, el PIB ajustado debe permanecer constante durante los 5 años.

### **Resultado esperado:**

- El PIB ajustado debe mantenerse igual durante los 5 años, ya que la tasa de crecimiento es cero.

### **Resultado obtenido:**

- La prueba pasó correctamente, y se verificó que el PIB proyectado para los 5 años se mantuvo constante al valor de PIB ajustado.

### **Conclusión:**

Esta prueba valida que el método maneja correctamente el caso en el que no hay crecimiento, asegurando que el PIB ajustado no cambie cuando la tasa de crecimiento es cero.

## //Prueba 3: Graficar Proyección PIB

Con esta prueba verificamos que el método GraficarProyeccionPIB sea capaz de generar el gráfico de la proyección del PIB y guardarlo en una imagen en la ruta especificada.

### **Resultado esperado:**

- El gráfico de la proyección del PIB debe generarse correctamente.
- El archivo de imagen debe ser guardado en la ruta especificada.

### **Resultado obtenido:**

- El archivo de imagen se guardó correctamente en la ruta temporal especificada.

### **Conclusión:**

Esta prueba asegura que la función de graficado y almacenamiento de imágenes funciona como se espera.

## //Prueba 4: Grafico Paralelo y Secuencial

Esta prueba valida el método que se encarga de representar gráficamente los tiempos de ejecución secuencial y paralelo usando ZedGraph mediante una gráfica.

### **Resultados esperados:**

- Se genera una gráfica compuesta por dos puntos, uno secuencial y el otro paralelo.

### **Resultado obtenido:**

- Se generó una barra en el gráfico.

### **Conclusión:**

Esta prueba valida que el método visualiza adecuadamente los datos comparativos de rendimiento.

## //Prueba 5: Cálculo del PIB Ajustado por Aranceles

Esta prueba verifica que el método CalcularPibAjustado calcula correctamente el impacto de los aranceles sobre el PIB.

### Resultado esperado:

- El valor ajustado del PIB debe corresponder a la fórmula:  
$$\text{pibOriginal} * \text{Math.Exp}(-1.5 * \text{tasaArancel})$$

### Resultado obtenido:

- El valor retornado por el método fue exactamente igual al calculado manualmente con la fórmula, con una precisión de dos decimales.

### Conclusión:

Esta prueba valida que la lógica matemática implementada para calcular el impacto de los aranceles sobre el PIB es precisa y consistente con el modelo esperado.



## Advertencias y Errores Reportados:

- El paquete Newtonsoft.Json 9.0.1 tiene una vulnerabilidad de seguridad de alta gravedad, según un aviso en GitHub. Lo recomendable es actualizar a una versión más reciente de Newtonsoft.Json (Solucionado)
- El paquete ZedGraph 5.1.7 que se está usando no es del todo compatible con .NET 8.0, que es la plataforma de destino del proyecto.

```
% A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.csproj : warning NU1701: El paquete "ZedGraph 5.1.7" se restauró con ".NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8, .NETFramework,Version=v4.8.1" en lugar de la plataforma de destino del proyecto "net8.0-windows7.0". Puede que el paquete no sea totalmente compatible con el proyecto.
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.Tests.csproj : warning NU1701: El paquete "Newtonsoft.Json 9.0.1" tiene una vulnerabilidad de gravedad alta conocida, https://github.com/advisories/GHSA-Scrp-9c3c-p9vr
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.Tests.csproj : warning NU1701: El paquete "ZedGraph 5.1.7" se restauró con ".NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8, .NETFramework,Version=v4.8.1" en lugar de la plataforma de destino del proyecto "net8.0-windows7.0". Puede que el paquete no sea totalmente compatible con el proyecto.
PIBImpact correcto con 3 advertencias (7.8s) => src\bin\Debug\net8.0-windows\PIBImpact.dll
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.csproj : warning NU1701: El paquete "ZedGraph 5.1.7" se restauró con ".NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8, .NETFramework,Version=v4.8.1" en lugar de la plataforma de destino del proyecto "net8.0-windows7.0". Puede que el paquete no sea totalmente compatible con el proyecto.
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\Analysis\Verificador.cs(38,127): warning CS8632: La anotación para tipos de referencia que aceptan valores NUL solo debe usarse en el código dentro de un contexto de anotaciones "Nullable".
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\Simulation\Parallelismo.cs(19,23): warning CS8632: La anotación para tipos de referencia que aceptan valores NUL solo debe usarse en el código dentro de un contexto de anotaciones "Nullable".
Error:
An assembly specified in the application dependencies manifest (testhost.deps.json) was not found:
  package: 'Newtonsoft.Json', version: '13.0.1'
  path: 'lib\netstandard2.0\Newtonsoft.Json.dll'
El proceso del host de pruebas para fuentes 'A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\bin\Debug\net8.0-windows\PIBImpact.dll' finalizó con el siguiente error: Error:
An assembly specified in the application dependencies manifest (testhost.deps.json) was not found:
  package: 'Newtonsoft.Json', version: '13.0.1'
  path: 'lib\netstandard2.0\Newtonsoft.Json.dll'
Consulte los registros de diagnóstico para obtener más información.
PIBImpact prueba correcto con 1 errores (8.8s)
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\bin\Debug\net8.0-windows\PIBImpact.dll : error TESTRUNABORT: Serie de pruebas anulada.
PIBImpact.Tests correcto con 2 advertencias (1.1s) => tests\bin\Debug\net8.0-windows\PIBImpact.Tests.dll
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.Tests.csproj : warning NU1701: El paquete "Newtonsoft.Json 9.0.1" tiene una vulnerabilidad de gravedad alta conocida, https://github.com/advisories/GHSA-Scrp-9c3c-p9vr
A:\Curso C# Básico\pib_impact_prog\pib_impact_prog\src\PIBImpact.Tests.csproj : warning NU1701: El paquete "ZedGraph 5.1.7" se restauró con ".NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8, .NETFramework,Version=v4.8.1" en lugar de la plataforma de destino del proyecto "net8.0-windows7.0". Puede que el paquete no sea totalmente compatible con el proyecto.
xunit.net 00:00:00.000 xunit.net VSTest Adapter v2.4.1+1b04f5d02b (64-bit .NET 8.0.15)
xunit.net 00:00:00.003 Discovering: PIBImpact.Tests
xunit.net 00:00:00.007 Discovered: PIBImpact.Tests
xunit.net 00:00:00.007 Starting: PIBImpact.Tests
xunit.net 00:00:00.023 Finished: PIBImpact.Tests
xunit.net 00:00:00.000 xunit.net VSTest Adapter v2.4.1+1b04f5d02b (64-bit .NET 8.0.15)
xunit.net 00:00:00.000 Discovering: PIBImpact.Tests
xunit.net 00:00:00.003 Discovered: PIBImpact.Tests
xunit.net 00:00:00.007 Starting: PIBImpact.Tests
xunit.net 00:00:00.011 Finished: PIBImpact.Tests
PIBImpact.Tests prueba realizado correctamente (9.5s)

Resumen de pruebas: total: 4; con errores: 0; correcto: 4; omitido: 7.3 s
Compilación error con 1 errores y 8 advertencias en 17.2s
```