

# Manual de Uso de la Aplicación PIBImpact

**Versión:** 1.0

**Última actualización:** 2025-04-23

## 1. Introducción

PIBImpact es un sistema en C# diseñado para:

- Cargar datos desde archivos CSV (países, sectores económicos, aranceles).
- Simular el impacto de cambios en aranceles sobre el Producto Interno Bruto (PIB).
- Paralelizar cálculos para mejorar el rendimiento.
- Analizar resultados y medir tiempos de ejecución.
- Generar gráficos y reportes de salida.

Este manual explica paso a paso cómo instalar, configurar y usar la aplicación.

## 2. Requisitos Previos

1. **.NET SDK 6.0 (o superior)** instalado en el equipo.
2. **Visual Studio 2022** (o VS Code con extensión C#).
3. Archivos CSV con la estructura:
  - **Países.csv:** CódigoISO,Nombre
  - **Sectores.csv:** Nombre,Descripción
  - **Aranceles.csv:** CódigoISO,Pais,NombreSector,Tasa
4. Acceso de lectura a carpeta donde residan los CSV.

## 3. Instalación

1. Clona el repositorio GitHub:  
[https://github.com/1707rosa/pib\\_impact\\_progparalela.git](https://github.com/1707rosa/pib_impact_progparalela.git)
2. Abre la solución Carga-datos.sln en Visual Studio.
3. Restaura paquetes NuGet y construye la solución (Build → Build Solution).

#### 4. Estructura del Proyecto

/ [Root]

- └─ Carga-datos.sln       # Solución de Visual Studio
- └─ src/               # Proyecto principal
  - | └─ PIBImpact.csproj   # Archivo de proyecto
  - | └─ Program.cs       # Punto de entrada
  - | └─ Models/          # Clases de modelo
    - | | └─ Pais.cs
    - | | └─ SectorEconomico.cs
    - | | └─ Arancel.cs
    - | | └─ Resultado.cs
  - | └─ Data/           # Carga y manejo de datos
    - | | └─ DataLoader.cs
  - | └─ Simulation/      # Lógica de simulación
    - | | └─ ImpactSimulator.cs
    - | | └─ ParaleloCalculator.cs
  - | └─ Analysis/        # Análisis y generación de salida
    - | | └─ Graficador.cs
    - | | └─ ReporteGenerador.cs
  - | └─ UI/            # Interfaz de usuario (WPF o consola)
    - | | └─ MainWindow.cs
  - | └─ Services/       # Servicios transversales
    - | | └─ PerformanceAnalyzer.cs
- └─ tests/            # Pruebas unitarias (xUnit)
  - | └─ PIBImpact.Tests.csproj
  - | └─ DataLoaderTests.cs

```
| └─ SimulationTests.cs
└─ docs/          # Documentación técnica
    └─ implementacion_tecnica.md
```

## 5. Uso de la Aplicación

### 5.1 Ejecución desde Consola

Si tu UI es de consola, sigue estos pasos:

1. Abre una terminal y navega a la carpeta src/:

```
cd src
```

2. Ejecuta:

```
dotnet run --project PIBImpact.csproj -- "ruta/Paises.csv" "ruta/Sectores.csv"
"ruta/Aranceles.csv"
```

3. La aplicación:

- Carga los tres CSV.
- Ejecuta la simulación (paralela).
- Muestra por pantalla un resumen de resultados.
- Guarda gráficos y reportes en output/.

### 5.2 Ejecución desde Interfaz Gráfica (WPF)

1. Ejecuta el proyecto PIBImpact.csproj con depuración (F5).
2. En la ventana principal:
  - Haz clic en "**Cargar Datos**" y selecciona los tres CSV.
  - Presiona "**Ejecutar Simulación**".
  - Observa la barra de progreso (paralelización).
  - Tras finalizar, aparecerán gráficos y tabla de resultados.
  - Para generar reporte PDF, haz clic en "**Exportar Reporte**".

## 6. Detalles Funcionales

### 6.1 DataLoader

- Métodos:
  - LoadPaises(string csvPath): lee y devuelve lista de Pais.
  - LoadSectores(string csvPath): lee y devuelve lista de SectorEconomico.
  - LoadAranceles(string csvPath): lee y devuelve lista de Arancel.

### 6.2 ImpactSimulator

- Simular(...):
  1. Recibe colecciones de países, sectores y aranceles.
  2. Para cada combinación, calcula el nuevo PIB base vs modificado.
  3. Devuelve lista de Resultado.
  4. Internamente invoca a ParaleloCalculator para paralelizar bucles.

### 6.3 ParaleloCalculator

- RunInParallel<TInput, TResult>(IEnumerable<TInput> items, Func<TInput, TResult> work): distribuye el cálculo entre hilos disponibles y sincroniza resultados.

### 6.4 Análisis y Salida

- **Graficador:** crea gráficos de barras o líneas comparando PIB antes y después.
- **ReporteGenerador:** compone un documento PDF/HTML con resumen, gráficos e indicadores de rendimiento.

## 7. Carpeta output/

Tras ejecutar simulación, se genera:

- resultados.csv: tabla con todas las instancias de Resultado.
- grafico\_pib.png: gráfico visual.
- reporte\_impacto.pdf: reporte final.

## 8. Pruebas Unitarias

1. Navega a tests/:

`cd tests`

2. Ejecuta pruebas:

`dotnet test`

3. Verifica que todos los tests (DataLoader, Simulation) pasen correctamente.

## 9. Solución de Problemas

- **Error de lectura CSV:** revisa que la ruta sea correcta y el delimitador sea ,.
- **Simulación lenta:** ajusta el grado de paralelismo o revisa que no haya cuellos de botella I/O.
- **Fallo al generar PDF:** verifica que la librería de reporte (iTextSharp, etc.) esté referenciada.

## 10. Soporte

Para dudas, reportes de errores o sugerencias, abre un *issue* en el repositorio GitHub o contacta a **equipo@tudominio.com**.