

Ejercicio 1: Corrección y cumplimiento del código C#

1. Problemas Detectados por SonarQube

Durante el análisis estático del proyecto LaMalaCalculadora con SonarQube, se identificaron varios problemas relacionados con fiabilidad, mantenibilidad y buenas prácticas de programación. Entre los principales hallazgos se encontraron:

- **Uso de goto**
El uso de goto provoca saltos incontrolados en el flujo del programa, lo cual dificulta el mantenimiento y puede generar comportamientos inesperados.
- **Comparación incorrecta de valores flotantes**
Se detectaron comparaciones directas con valores tipo double, lo cual no es confiable debido a la naturaleza de los números de punto flotante.
- **Manejo deficiente de división y módulo por cero**
La versión original utilizaba ajustes improvisados (como sumar 0.0000001) en lugar de validar correctamente el divisor.
- **Falta de validación de entrada**
No se manejaban correctamente escenarios donde el usuario ingresaba valores no numéricos.
- **Código duplicado y estructuras complejas**
Había lógica repetida en operaciones, manejo de historial y parseo, lo cual afectaba la mantenibilidad.

2. Cómo se corrigieron los problemas

Se aplicó un proceso de refactorización eliminando las malas prácticas y centralizando la lógica para hacer el código más seguro y legible.

- **Eliminación de goto**
Se reemplazó por un ciclo while con un flujo estructurado mediante switch, mejorando la claridad y el orden del programa.
- **Comparación apropiada para valores flotantes**
Se implementó el método:

```
private static bool IsApproximatelyZero(double value, double epsilon = 1e-9)
{
    return Math.Abs(value) < epsilon;
}
```

Este método evita comparaciones exactas y mejora la fiabilidad numérica.

- Manejo correcto de división/módulo por cero
Ahora se valida explícitamente antes de realizar la operación:

```
if (op == "/" && IsApproximatelyZero(b))
{
    Console.WriteLine("No se puede dividir por cero.");
    return;
}
```

Esto evita crashes y resultados incorrectos.

- Validación robusta de entradas
Se creó el método:

```
private static bool TryParseDouble(string text, out double value)
```

que sanitiza la entrada y usa double.TryParse con cultura invariable.

- Refactorización de la clase ShoddyCalc
La lógica matemática se centralizó utilizando switch expresivo:

```
return op switch
{
    "+" => a + b,
    "-" => a - b,
    "*" => a * b,
    "/" => a / b,
    "^" => Math.Pow(a, b),
    "%" => a % b,
    _ => double.NaN
}
```

```
};
```

Esto hace el código más limpio, extensible y fácil de mantener.

3. Mejoras logradas en la aplicación

Gracias a las correcciones aplicadas y verificadas mediante análisis en SonarQube, la aplicación ahora cumple con mejores prácticas de desarrollo y estándares de calidad.

- Maneja entradas inválidas

El programa detecta valores no numéricos y previene errores de ejecución.

- Evita divisiones y operaciones inválidas

Se valida correctamente la división y el módulo por cero.

- Uso de estructuras claras

Se eliminaron los goto y se reorganizó el flujo con un switch dentro de un ciclo bien estructurado.

- Código más legible y mantenible

La lógica está dividida en métodos pequeños y reutilizables, con validaciones y mensajes claros para el usuario.

- Historial mejorado

El historial se maneja con una clase dedicada, evitando dependencias globales y mejorando la organización del código.

PROCESO

1. Pimero ejecutar el comando para iniciar el programa (CMD)

```
C:\Users\User\Downloads\sonarqube-25.10.0.114319\sonarqube-25.10.0.114319\bin\windows-x86-64
```

Luego el StartSonar.ba

2. Comando 1 — Iniciar análisis

```
cd "C:\Users\User\Downloads\LaMalaCalculadora"
```

```
Powershell: dotnet sonarscanner begin /k:"LaMalaCalculadora"
```

```
/d:sonar.host.url="http://localhost:9000"
```

```
/d:sonar.token="sqp_7de4c56594ac6d724a4660180324f739e2c3ceb5"
```

```
PS C:\Users\User\Downloads\LaMalaCalculadora> cd "C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora"
PS C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora> dotnet sonarscanner begin /k:"LaMalaCalculadora" /d:sona
r.host.url="http://localhost:9000" /d:sonar.token="sqp_7de4c56594ac6d724a4660180324f739e2c3ceb5"
SonarScanner for .NET 11.0
Using the .NET Core version of the Scanner for .NET
Pre-processing started.
Preparing working directories...
18:51:44.674 Updating build integration targets...
18:51:44.793 Using SonarQube v25.10.0.114319.
18:51:44.995 Fetching analysis configuration settings...
18:51:47.505 Provisioning analyzer assemblies for cs...
18:51:47.507 Installing required Roslyn analyzers...
18:51:47.508 Processing plugin: csharp version 10.15.0.120848
18:51:48.581 Provisioning analyzer assemblies for vbnet...
18:51:48.584 Installing required Roslyn analyzers...
18:51:48.586 Processing plugin: vbnet version 10.15.0.120848
18:51:48.603 Incremental PR analysis: Base branch parameter was not provided.
18:51:48.606 Cache data is empty. A full analysis will be performed.
18:51:48.651 Pre-processing succeeded.
PS C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora> |
```

3. Comando 2 — Build del proyecto

```
Powershell: dotnet build
```

```
Windows PowerShell
(https://rules.sonarsource.com/csharp/RSPEC-907) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\BadCalc_
VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(126,21): warning S907: Remove this use of 'goto'
. (https://rules.sonarsource.com/csharp/RSPEC-907) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\BadCalc_
VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(139,21): warning S907: Remove this use of 'goto'
. (https://rules.sonarsource.com/csharp/RSPEC-907) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\BadCalc_
VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(180,13): warning S907: Remove this use of 'goto'
. (https://rules.sonarsource.com/csharp/RSPEC-907) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\BadCalc_
VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(33,23): warning S2325: Make 'DoIt' a static meth
od. (https://rules.sonarsource.com/csharp/RSPEC-2325) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\BadC
alc_VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(133,25): warning S1481: Remove the unused local
variable 'tpl'. (https://rules.sonarsource.com/csharp/RSPEC-1481) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalc
uladora\BadCalc_VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(135,25): warning S1481: Remove the unused local
variable 'uin'. (https://rules.sonarsource.com/csharp/RSPEC-1481) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalc
uladora\BadCalc_VeryBad.csproj]
C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\Program.cs(136,25): warning S1481: Remove the unused local
variable 'sys'. (https://rules.sonarsource.com/csharp/RSPEC-1481) [C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalc
uladora\BadCalc_VeryBad.csproj]
  41 Advertencia(s)
    0 Errores

Tiempo transcurrido 00:00:04.82

Hay actualizaciones de carga de trabajo disponibles. Ejecute "dotnet workload list" para obtener más información.
PS C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora> |
```

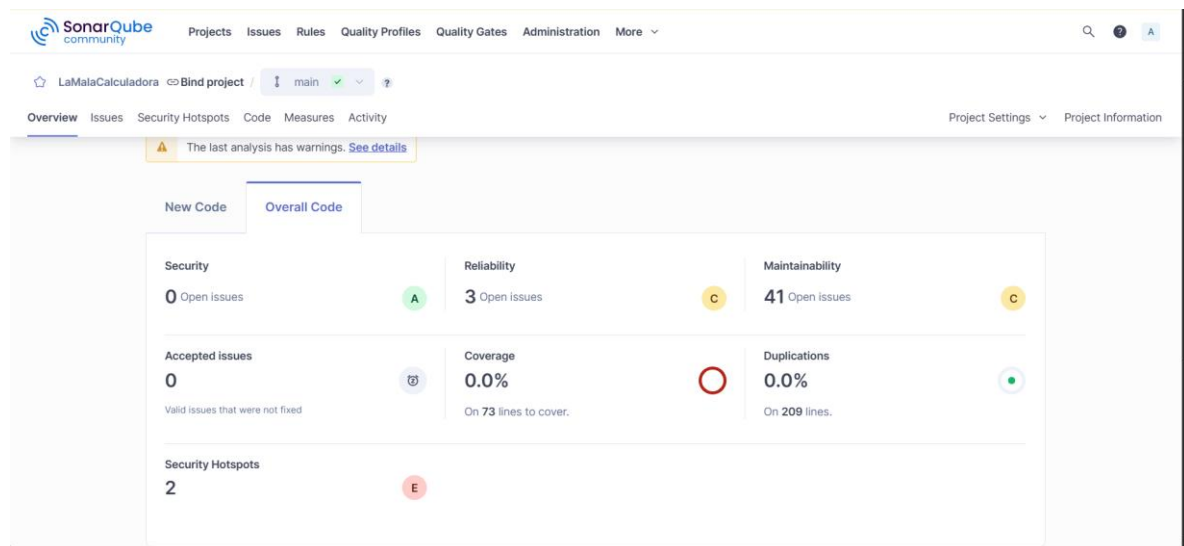
4. Comando 3 — Cerrar el análisis

Powershell: dotnet sonarscanner end

/d:sonar.token="sqp_7de4c56594ac6d724a4660180324f739e2c3ceb5"

```
Windows PowerShell
18:54:41.571 INFO: Using Git CLI to retrieve untracked files
18:54:41.731 WARNING: WARN: Retrieving only language associated files, make sure to run the analysis inside a git repository to make use of inclusions specified via "sonar.text.inclusions"
18:54:41.734 INFO: Starting the text and secrets analysis
18:54:41.741 INFO: 7 source files to be analyzed for the text and secrets analysis
18:54:41.845 INFO: 7/7 source files have been analyzed for the text and secrets analysis
18:54:41.853 INFO: Sensor TextAndSecretsSensor [text] (done) | time=2084ms
18:54:41.855 INFO: ----- Run sensors on project
18:54:42.5 INFO: Sensor C# Telemetry processor [csharp]
18:54:42.515 INFO: Sensor C# Telemetry processor [csharp] (done) | time=13ms
18:54:42.517 INFO: Sensor C# Telemetry Json processor [csharp]
18:54:42.586 INFO: Sensor C# Telemetry Json processor [csharp] (done) | time=71ms
18:54:42.588 INFO: Sensor C# [csharp]
18:54:42.628 INFO: Importing results from 8 proto files in 'C:\Users\User\Downloads\LaMalaCalculadora\LaMalaCalculadora\sonarqube\out\0\output-cs'
18:54:43.462 INFO: Importing 1 Roslyn report
18:54:43.761 INFO: Found 1 MSBuild C# project: 1 MAIN project.
18:54:43.762 INFO: Sensor C# [csharp] (done) | time=1177ms
18:54:43.763 INFO: Sensor Analysis Warnings import [csharp]
18:54:43.766 INFO: Sensor Analysis Warnings import [csharp] (done) | time=2ms
18:54:43.767 INFO: Sensor C# File Caching Sensor [csharp]
18:54:43.78 INFO: Sensor C# File Caching Sensor [csharp] (done) | time=15ms
18:54:43.781 INFO: Sensor Zero Coverage Sensor
18:54:43.812 INFO: Sensor Zero Coverage Sensor (done) | time=34ms
18:54:43.813 INFO: ----- Gather SCA dependencies on project
18:54:43.822 INFO: Dependency analysis skipped
18:54:43.824 INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
18:54:43.835 INFO: CPD Executor Calculating CPD for 1 file
18:54:43.874 INFO: CPD Executor CPD calculation finished (done) | time=31ms
```

5. Resultado del análisis Inicial



6. Resultado del análisis Final

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

More

LaMalaCalculadora

Bind project

main

Overview

Issues

Security Hotspots

Code

Measures

Activity

Project Settings

Project Information

New Code

Overall Code

Security

0 Open issues

A

Accepted issues

0

Valid issues that were not fixed

B

Security Hotspots

0

A

Reliability

0 Open issues

A

Coverage

0.0%

On 86 lines to cover.

Maintainability

3 Open issues

A

Duplications

0.0%

On 283 lines.