CI/CD con GitHub Actions: Pruebas, Reportes y Notificaciones

Este repositorio demuestra un pipeline CI con **GitHub Actions** que cubre: 1) **Clonación del repositorio**
2) **Ejecución de pruebas automatizadas** (unitarias)
3) **Generación de reportes** de resultados y cobertura
4) **Notificaciones** (opcional por Slack)

---

1) Workflow (YAML)

Archivo: `.github/workflows/main.yml`

“‘yaml name: CI - Build, Test, Coverage & Notify

on: push: branches: [“**”] pull_request: branches: [“**”]

permissions: contents: read checks: write pull-requests: write

jobs: ci: runs-on: ubuntu-latest steps: # (1) CLONACIÓN - name: Checkout repository uses: actions/checkout@v4

```
  # SDK .NET
  - name: Setup .NET
    uses: actions/setup-dotnet@v4
    with:
      dotnet-version: '9.0.x'

  # Restore & Build (opcional pero recomendado)
  - name: Restore
    run: dotnet restore MauiClimaDemo.sln

  - name: Build
    run: dotnet build MauiClimaDemo.sln --configuration Release --no-restore

  # (2) PRUEBAS + COBERTURA
  - name: Test (with coverage)
    run: |
      dotnet test MauiClimaDemo.Tests/MauiClimaDemo.Tests.csproj \
        --configuration Release --no-build \
        --logger "trx;LogFileName=test_results.trx" \
        --results-directory TestResults \
        --collect "XPlat Code Coverage"

  # (3) REPORTES (TRX + Cobertura HTML/XML)
  - name: Publish test results
    uses: EnricoMi/publish-unit-test-result-action@v2
    if: always()
    with:
      files: "TestResults/**/*.trx"

  - name: Generate coverage report (ReportGenerator)
    uses: danielpalme/ReportGenerator-GitHub-Action@5.4.12
    with:
      reports: "TestResults/**/coverage.cobertura.xml"
      targetdir: "coveragereport"
      reporttypes: "HtmlInline;Cobertura;MarkdownSummaryGithub"
```

```
- name: Upload artifacts (reports & trx)
  uses: actions/upload-artifact@v4
  with:
    name: test-and-coverage-reports
    path: |
      TestResults/**/*.trx
      TestResults/**/coverage.cobertura.xml
      coveragereport/**

# (4) NOTIFICACIONES (Slack opcional)
# Requiere Secrets: SLACK_BOT_TOKEN y SLACK_CHANNEL_ID
- name: Notify Slack
  if: always()
  uses: slackapi/slack-github-action@v2.0.0-rc.2
  with:
    method: chat.postMessage
    token: ${{ secrets.SLACK_BOT_TOKEN }}
    payload: |
      channel: ${{ secrets.SLACK_CHANNEL_ID }}
      text: ":rocket: *CI* en *${{ github.ref_name }}* terminó con *${{ job.status }}* (run: ${{ githu
```

git add README.md git commit -m “Add README with pipeline design, diagram and explanation” git push

git add README.md git commit -m “Add README with pipeline design, diagram and explanation” git push

git push

1) Instalar herramientas

sudo apt-get update -y sudo apt-get install -y pandoc npm npm install -g @mermaid-js/mermaid-cli

2) Guardar el diagrama Mermaid a archivo (.mmd)

cat > diagram.mmd «‘EOF’ flowchart LR A[Push o Pull Request] –> B[Checkout del repositorio] B –> C[Setup .NET SDK] C –> D[Restore & Build] D –> E[dotnet test + cobertura] E –> F[Publicar resultados (TRX) en Summary/PR] E –> G[Generar reporte HTML de cobertura] F –> H[Subir artefactos (TRX + cobertura + HTML)] G –> H E –> I[Notificación (Slack o GitHub)] H –> J[Fin]