

# Documentação de Testes Automatizados

SmartCities API: SpecFlow + NUnit

Gabri - Maio 2025

# Cenários BDD

- 1. Health Check retorna 200
- 2. Listar todas as cidades (positiva)
- 3. Buscar cidade inexistente (negativa)
- 4. Listar todos os acientes (positiva)
- 5. Buscar acidente inexistente (negativa)

# HealthCheck.feature

- # language: pt-BR
- Funcionalidade: Verificar saúde da API
- Cenário: Health check retorna 200
- Dado que o serviço está rodando em "https://localhost:5000"
- Quando faço uma requisição GET no endpoint "/health"
- Então o código de resposta deve ser 200
- E o corpo da resposta deve conter "status": "UP"

# CitiesApi.feature

- # language: pt-BR
- Funcionalidade: Gestão de Cidades
- Cenário: Listar todas as cidades (positiva)
  - Dado que a API está disponível em "https://localhost:5000"
  - Quando faço GET em "/cities"
  - Então o status code deve ser 200
  - E a resposta deve ser um array JSON não vazio
- Cenário: Buscar cidade inexistente (negativa)
  - Dado que a API está disponível em "https://localhost:5000"
  - Quando faço GET em "/cities/9999"
  - Então o status code deve ser 404

# Acidente.feature

- # language: pt-BR
- Funcionalidade: Gestão de Acidentes
  - Para monitorar incidentes de trânsito
  - Como usuário da API
  - Quero ver a lista de acidentes e detalhes de um acidente
- Cenário: Listar todos os acidentes
  - Dado que a API está disponível em "https://localhost:5000"
  - Quando faço GET em "/acidentes"
  - Então o status code deve ser 200
  - E a resposta deve ser um array JSON não vazio
  - E a resposta deve obedecer ao schema "acidentes-schema.json"
- Cenário: Buscar acidente inexistente
  - Dado que a API está disponível em "https://localhost:5000"
  - Quando faço GET em "/acidentes/9999"
  - Então o status code deve ser 404

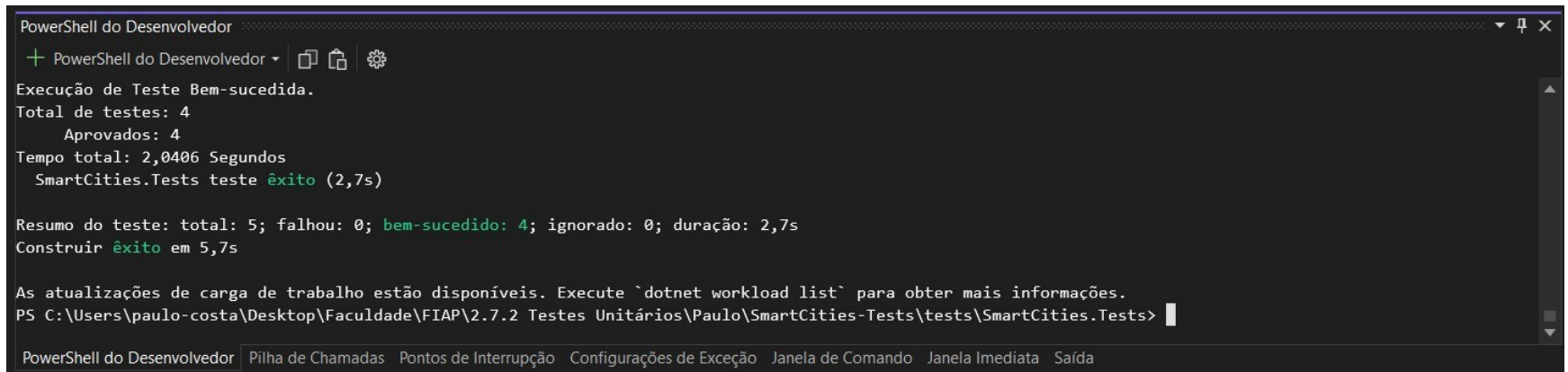
# Evidência de Execução

- Execução de Teste Bem-sucedida:
- Total de testes: 4
- Aprovados: 4
- Tempo total: ~1.3s

# Evidência de Execução

```
Resumo do teste: total: 5; falhou: 0; bem-sucedido: 4; ignorado: 0; duração: 1,3
s
Construir êxito em 2,6s

As atualizações de carga de trabalho estão disponíveis. Execute `dotnet workload
list` para obter mais informações.
```



```
PowerShell do Desenvolvedor
+ PowerShell do Desenvolvedor
Execução de Teste Bem-sucedida.
Total de testes: 4
  Aprovados: 4
Tempo total: 2,0406 Segundos
  SmartCities.Tests teste êxito (2,7s)

Resumo do teste: total: 5; falhou: 0; bem-sucedido: 4; ignorado: 0; duração: 2,7s
Construir êxito em 5,7s

As atualizações de carga de trabalho estão disponíveis. Execute `dotnet workload list` para obter mais informações.
PS C:\Users\paulo-costa\Desktop\Faculdade\FIAP\2.7.2 Testes Unitários\Paulo\SmartCities-Tests\tests\SmartCities.Tests>
```

PowerShell do Desenvolvedor | Pilha de Chamadas | Pontos de Interrupção | Configurações de Exceção | Janela de Comando | Janela Imediata | Saída

# Evidência de Execução

The screenshot displays a GitHub Actions workflow run interface. At the top, the workflow name is 'fix: livingdoc test-assembly apontar para a dll #13', preceded by a green checkmark icon. To the right of the name are buttons for 'Re-run all jobs' and a three-dot menu. The left sidebar contains navigation links: 'Summary' (selected), 'Jobs', 'Run details', 'Usage', and 'Workflow file'. Under 'Jobs', the job 'build-and-test' is highlighted with a green checkmark. The main area shows the details for the 'build-and-test' job, which 'succeeded now in 59s'. A search bar labeled 'Search logs' is present. Below the job name is a list of 15 steps, each with a green checkmark icon and a duration. The steps are: 'Set up job' (1s), 'Run actions/checkout@v4' (1s), 'Setup .NET' (0s), 'Restore API' (12s), 'Build API' (10s), 'Run API (background)' (12s), 'Restore tests' (4s), 'Build tests' (2s), 'Execute tests' (5s), 'Publish LivingDoc' (7s), 'Upload Report' (1s), 'Post Setup .NET' (0s), 'Post Run actions/checkout@v4' (1s), and 'Complete job' (0s).

← CI

✓ fix: livingdoc test-assembly apontar para a dll #13

Re-run all jobs ...

Summary

Jobs

✓ build-and-test

Run details

Usage

Workflow file

**build-and-test**  
succeeded now in 59s

Search logs

- > ✓ Set up job 1s
- > ✓ Run actions/checkout@v4 1s
- > ✓ Setup .NET 0s
- > ✓ Restore API 12s
- > ✓ Build API 10s
- > ✓ Run API (background) 12s
- > ✓ Restore tests 4s
- > ✓ Build tests 2s
- > ✓ Execute tests 5s
- > ✓ Publish LivingDoc 7s
- > ✓ Upload Report 1s
- > ✓ Post Setup .NET 0s
- > ✓ Post Run actions/checkout@v4 1s
- > ✓ Complete job 0s



# Repositório no Github

<https://github.com/paulokrg/Fase7-Cap2-TestesAutomatizados>