O projeto FHIRUT (FHIR Unit Test Suite) tem como objetivo automatizar testes de conformidade de recursos FHIR, comparando resultados esperados com os obtidos por meio do validador oficial HL7. A combinação de Angular (frontend) e .NET (backend) é ideal para esse sistema devido à capacidade de criar interfaces ricas e processar validações complexas de forma eficiente.

2. Benefícios do Angular para o FHIRUT

2.1. Dashboard Interativo para Relatórios

Visualização de resultados em tabelas, gráficos e cards.
Filtros dinâmicos para navegar entre testes aprovados/reprovados.
Destaque de diferenças entre resultados esperados e obtidos.

2.2. Componentização Reutilizável

TestSummaryComponent: Exibe estatísticas gerais (quantidade de sucessos, falhas, avisos).
TestDetailComponent: Mostra detalhes de um teste específico (instância FHIR, erros encontrados).
DiffHighlighterComponent: Comparação lado a lado entre JSON esperado e obtido.

2.3. Consumo de API .NET

Chamadas HTTP para executar testes e recuperar relatórios:

```
this.http.post('/api/tests/run', { testFiles: ['patient-001.yml'] })
  .subscribe(report => this.displayReport(report));
```

3. Benefícios do .NET para o FHIRUT

3.1. Processamento de Casos de Teste (YAML/JSON)

Leitura de arquivos YAML com YamlDotNet.
Manipulação de JSON FHIR com System.Text.Json.

3.2. Integração com Validador FHIR (validator_cli.jar)

Execução do validador Java em paralelo:

```
var process = new Process
{
    StartInfo = new ProcessStartInfo
    {
        FileName = "java",
        Arguments = $"-jar validator_cli.jar -ig {profilePath} -output {outputPath}
{inputPath}"
    }
};
process.Start();
if (!process.WaitForTimeout(TimeSpan.FromSeconds(30)))
    throw new TimeoutException("Validação excedeu o tempo limite.");
```

3.3. Comparação de Resultados

Avaliação de FHIRPath (ex: OperationOutcome.issue.count() = 0).

Matching de erros/warnings com resultados esperados.

3.4. API REST para o Frontend

Endpoints como:

POST /api/tests/run → Dispara a execução dos testes.

GET /api/reports/{id} → Retorna relatório em JSON/HTML.

4. Fluxo do Sistema

Frontend (Angular):

Usuário seleciona testes (.yml) e clica em "Executar".

Envia requisição para o backend.

Backend (.NET):

Processa YAML, executa validator_cli.jar e compara resultados.

Gera relatório estruturado.

Frontend (Angular):

Exibe relatório com gráficos, tabelas e detalhes.