

## RELATÓRIO TÉCNICO – MATHQUIZ

### 1. ARQUITETURA DO SISTEMA

O projeto *mathquiz* segue uma arquitetura monolítica, adequada para aplicações de linha de comando. Foi projetado com uma estrutura modular interna, aplicando princípios de orientação a objetos.

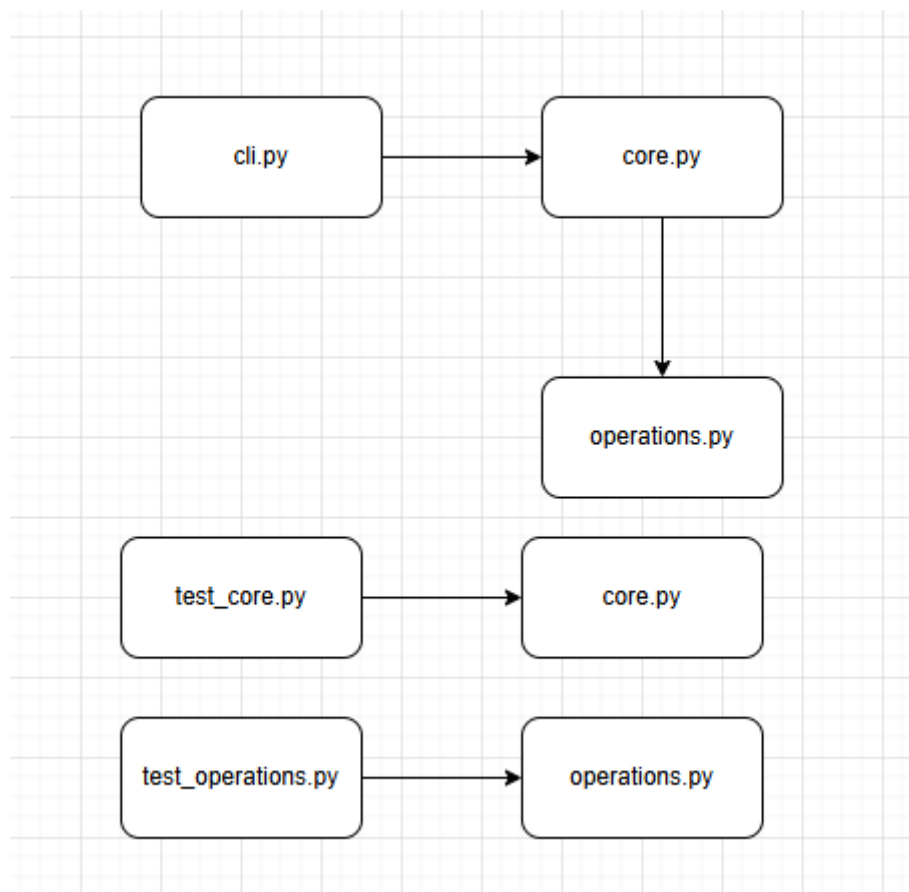
#### Estrutura e Diagrama:

mathquiz/

- └─ cli.py      # Interface de linha de comando com *argparse*
- └─ core.py      # Lógica do quiz: fluxo principal, coleta e verificação de respostas
- └─ operations.py # Estratégias de operação matemática (*Strategy Pattern*)
- └─ \_\_init\_\_.py

tests/

- └─ test\_core.py    # Testes para o quiz e perguntas
- └─ test\_operations.py # Testes para as estratégias de operação



### Observação:

Foi utilizada o **Strategy Pattern** que permite trocar operações (+, -, \*, /) sem alterar a lógica principal do sistema.

## 2. DECISÕES DE DESIGN

- **Padrão Strategy:** aplicado em “operations.py” para cada operação matemática.
- **Separação de responsabilidades:** cada componente (CLI, lógica, operações) em módulo separado.
- **CLI com *argparse*:** facilita a configuração dinâmica do quiz.
- **Testes automatizados:** com *pytest*, cobrindo operações e perguntas.

## 3. COMO EXECUTAR

### Instalar o projeto:

```
pip install -e .
```

### Executar CLI:

```
python -m mathquiz.cli --num 5 --ops + - * / --min 1 --max 10 --decimals --places 2
```

### Rodar testes:

```
pytest
```

### Verificar estilo com ruff:

```
ruff check .
```

## 4. POSSÍVEIS EXTENSÕES

- **Níveis de dificuldade\*\*:** definir faixas e operadores diferentes por dificuldade.
- **Exportação de relatório:** suporte a formatos como CSV, JSON.
- **Modo estudo:** com feedback explicativo por questão.
- **Suporte multilíngue:** internacionalização via arquivos `.po``.
- **Histórico de sessões:** salvar desempenho do usuário para progresso contínuo.

## 5. PRINT DE USO REAL

```
(venv) PS C:\Users\stefanini\Desktop\Teste_CLI_Adenilson> python -m mathquiz.cli --num 2 --ops + - --min 1 --max 10 --decimals --places 2
Iniciando o quiz matemático!

Questão 1: 7.67 + 7.09 = ?
> 5
Errado. Resposta correta: 14.76

Questão 2: 4.05 - 7.05 = ?
> 6
Errado. Resposta correta: -3.0

Resultado Final
-----
Total: 2 questões
Corretas: 0
Incorretas: 2
% Acerto: 0.00%
Tempo total: 4.95s
Tempo médio por questão: 2.47s
Resultados salvos em 'resultados.csv'.
Quiz concluído! Obrigado por participar.
```

```
• (venv) PS C:\Users\stefanini\Desktop\Teste_CLI_Adenilson> python -m mathquiz.cli --num 3 --ops + - --min 1 --max 10
Iniciando o quiz matemático!

Questão 1: 4 - 4 = ?
> 0
Correto!

Questão 2: 1 + 5 = ?
> 6
Correto!

Questão 3: 4 + 7 = ?
> 11
Correto!

Resultado Final
-----
Total: 3 questões
Corretas: 3
Incorretas: 0
% Acerto: 100.00%
Tempo total: 14.05s
Tempo médio por questão: 4.68s
Resultados salvos em 'resultados.csv'.
Quiz concluído! Obrigado por participar.
```

```
• (venv) PS C:\Users\stefanini\Desktop\Teste_CLI_Adenilson> pytest
>>
===== test session starts =====
platform win32 -- Python 3.10.1, pytest-8.4.0, pluggy-1.6.0
rootdir: C:\Users\stefanini\Desktop\Teste_CLI_Adenilson
configfile: pyproject.toml
plugins: anyio-4.9.0, langsmith-0.3.42
collected 5 items

tests\test_core.py . [ 20%]
tests\test_operations.py .... [100%]

===== 5 passed in 0.04s =====
```

```
• (venv) PS C:\Users\stefanini\Desktop\Teste_CLI_Adenilson> ruff check .
>>
All checks passed!
○ (venv) PS C:\Users\stefanini\Desktop\Teste_CLI_Adenilson> █
```

**Observação:** Este projeto contou com o apoio de ferramentas de inteligência artificial, como o GitHub Copilot e o ChatGPT, que foram utilizadas para acelerar o desenvolvimento, fornecer sugestões de código e auxiliar na resolução de problemas.