



Testes Baseados em Máquinas de Estados Finitos

Prof. Dr. Valério Gutemberg



Discente: Moisés, Rafael Augusto, Alessandro, Guilherme Cadete



Disciplina de Testes de Software



Curso de Sistemas para Internet



O que é uma Máquinas de Estados Finitos (FSM)?

- **FSM:** É um modelo matemático composto por um número finito de estados, transições entre esses estados, e ações que podem ocorrer em cada estado. Significa, em resumo, que é uma máquina que só pode ter um estado por vez e precisa de uma forma de transicionar entre estados.

Exemplo de código simples

```
portao.py 1, U ✕  
codigo > portao.py > ...  
1 fechado = 'Fechado'  
2 aberto = 'Aberto'  
3  
4 eventos = ['inserir_chave', 'abrir_portao', 'fechar_portao']  
5  
6 from pymodel.FSM import FSM  
7  
8 estado_inicial = fechado  
9  
10 def inserir_chave(estado):  
11     if estado == fechado:  
12         return aberto  
13     return estado  
14  
15 def abrir_portao(estado):  
16     if estado == aberto:  
17         return aberto  
18     return estado  
19  
20 def fechar_portao(estado):  
21     if estado == aberto:  
22         return fechado  
23     return estado  
24  
25 estados = {fechado, aberto}  
26 acoes = {inserir_chave, fechar_portao, abrir_portao}  
27 transicoes = {  
28     (fechado, inserir_chave): aberto,  
29     (aberto, fechar_portao): fechado,  
30     (aberto, abrir_portao): aberto,  
31 }  
32  
33 modelo = FSM(estados, acoes, transicoes)  
34 + |
```



Tecnologias de CI/CD

Jenkins

- Plataforma open-source para automação de processos.
- Amplamente usada para CI/CD.
- Suporta integração com várias ferramentas de teste.

GitHub Actions

- Integra CI/CD diretamente no repositório do GitHub.
- Suporte nativo a PyTest, Go Test e Rust Test.
- Configuração simplificada com workflows YAML.



Travis CI

- Serviço baseado na nuvem para CI/CD.
- Suporte a múltiplas linguagens de programação.
- Integração fácil com repositórios GitHub.



Ferramentas de Teste

PyTest (Python)

- Framework de testes para Python.
- Suporte a testes unitários e de integração.
- Integração fácil com Jenkins, GitHub Actions e Travis CI.

Go Test (Go)

- Ferramenta nativa de testes para Go.
- Executa testes unitários, benchmarks e de integração.
- Totalmente suportado nas pipelines CI/CD.



Rust Test (Rust)

- Framework nativo para testar aplicações Rust.
- Fácil integração em pipelines de CI/CD.
- Verifica segurança, desempenho e consistência do código.



Exemplo de Pipeline CI/CD (GitHub Actions)

```
name: CI Pipeline
on:
  push:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2
      - name: Configurar Python
        uses: actions/setup-python@v2
        with:
          python-version: 3.x
      - name: Instalar Dependências
        run: pip install pytest
      - name: Executar Testes
        run: pytest
```




Conclusão

O CI/CD automatiza e acelera o processo de desenvolvimento.

A combinação com ferramentas de teste garante a qualidade do software.

Integração eficiente entre Jenkins, GitHub Actions e Travis CI com PyTest, Go Test e Rust Test



INSTITUTO FEDERAL
Rio Grande do Norte

Obrigado!

Dúvidas?