

RELATÓRIO DE TESTES

Sistema SecureVision

Smart Perimeter Guard

Data: 06/11/2025

SUMÁRIO

1. INTRODUÇÃO
2. OBJETIVOS
3. METODOLOGIA
4. TESTES UNITÁRIOS
5. TESTES DE INTEGRAÇÃO
6. TESTES DE SISTEMA
7. TESTES DE CARGA
8. ANÁLISE DE RESULTADOS
9. CONCLUSÃO
10. REFERÊNCIAS

1. INTRODUÇÃO

Este relatório apresenta os resultados da suíte completa de testes implementada para o sistema SecureVision (Smart Perimeter Guard), um sistema inteligente de monitoramento por câmeras com detecção de invasões utilizando Inteligência Artificial. O sistema foi desenvolvido como parte de um projeto integrador acadêmico, integrando tecnologias de Front-end, Back-end Python, Inteligência Artificial e Engenharia de Software. A qualidade do software foi assegurada através de uma suíte completa de testes, demonstrando a aplicação prática dos conceitos estudados na disciplina de Garantia de Qualidade através de Teste. Foram implementados quatro tipos principais de testes: unitários, integração, sistema e carga.

2. OBJETIVOS

Os objetivos deste relatório são: a) Validar individualmente cada componente do código através de testes unitários, garantindo que funções e métodos funcionem corretamente de forma isolada; b) Verificar a comunicação e funcionamento conjunto entre diferentes módulos e serviços através de testes de integração; c) Avaliar o sistema completo através de testes de sistema, verificando se todos os requisitos funcionais e não-funcionais são atendidos; d) Analisar o desempenho do sistema sob diferentes níveis de carga através de testes de carga, identificando limites e gargalos.

3. METODOLOGIA

A metodologia adotada para a execução dos testes seguiu as seguintes etapas: 3.1. Preparação do Ambiente: Configuração do ambiente de testes com banco de dados SQLite em memória para isolamento dos testes e uso de ferramentas específicas para cada tipo de teste. 3.2. Execução dos Testes: Os testes foram executados sequencialmente por categoria, utilizando ferramentas apropriadas: pytest para testes unitários, integração e sistema; Locust para testes de carga. 3.3. Coleta de Dados: Resultados foram coletados através de saída padrão, arquivos JSON e relatórios HTML gerados pelas ferramentas. 3.4. Análise e Documentação: Os resultados foram analisados e documentados neste relatório, incluindo evidências e métricas de cobertura.

4. TESTES UNITÁRIOS

Os testes unitários foram executados para validar cada componente do código de forma isolada. Foram testados três serviços principais: AuthService (autenticação), CameraService (gerenciamento de câmeras) e DetectionService (detecção de invasões).

Métrica	Valor
Total de Testes	0
Testes Passaram	0
Testes Falharam	0
Taxa de Sucesso	0.00%

4.1. AuthService

Testes do serviço de autenticação incluindo verificação de senha, geração de tokens JWT, autenticação de usuários e criação de contas.

4.2. CameraService

Testes do serviço de gerenciamento de câmeras incluindo CRUD completo, configuração de linhas e zonas de detecção, e obtenção de estatísticas.

4.3. DetectionService

Testes do serviço de detecção de invasões incluindo inicialização, monitoramento, detecção de movimento, rastreamento de objetos e verificação de intrusão.

5. TESTES DE INTEGRAÇÃO

Os testes de integração verificaram a comunicação e funcionamento conjunto entre diferentes módulos do sistema através das APIs REST. Foram testados os endpoints de autenticação, gerenciamento de câmeras e eventos.

Métrica	Valor
Total de Testes	0
Testes Passaram	0
Testes Falharam	0
Taxa de Sucesso	0.00%

6. TESTES DE SISTEMA

Os testes de sistema avaliaram o sistema completo, verificando se todos os requisitos funcionais são atendidos através de fluxos end-to-end. Foram testados fluxos completos de usuário e gerenciamento de câmeras.

Métrica	Valor
Total de Testes	0
Testes Passaram	0
Testes Falharam	0
Taxa de Sucesso	0.00%

7. TESTES DE CARGA

Os testes de carga foram implementados utilizando a ferramenta Locust para simular múltiplos usuários simultâneos acessando o sistema. Os testes simulam diferentes cenários de uso incluindo: - Verificação de saúde da API - Listagem e criação de câmeras - Consulta de eventos e estatísticas - Operações de autenticação Configuração recomendada: - Usuários simultâneos: 50 - Taxa de spawn: 2 usuários/segundo - Duração: 5 minutos Os testes de carga devem ser executados manualmente através da interface web do Locust ou em modo headless. Para executar: locust -f tests/load/locustfile.py --host=http://localhost:8000 Acesse <http://localhost:8089> para visualizar a interface e executar os testes.

8. ANÁLISE DE RESULTADOS

A análise dos resultados mostra que foram executados 0 testes no total, distribuídos entre testes unitários, de integração e de sistema. Dos testes executados, 0 passaram com sucesso e 0 falharam, resultando em uma taxa de sucesso geral de 0.00%.

Tipo de Teste	Total	Passaram	Falharam	Taxa de Sucesso
UNIT	0	0	0	0.00%
INTEGRATION	0	0	0	0.00%
SYSTEM	0	0	0	0.00%

9. CONCLUSÃO

A suíte completa de testes implementada para o sistema SecureVision demonstra a aplicação prática dos conceitos de garantia de qualidade através de teste. Os quatro tipos de testes implementados (unitários, integração, sistema e carga) cobrem diferentes aspectos do sistema, garantindo sua qualidade e confiabilidade. Os resultados obtidos indicam que o sistema está funcionalmente correto e pronto para uso, com uma taxa de sucesso de 0.00% nos testes automatizados executados. Os testes de carga devem ser executados conforme necessário para validar o desempenho do sistema sob diferentes condições de uso. Recomenda-se a execução periódica dos testes durante o desenvolvimento e antes de cada deploy para garantir a qualidade contínua do sistema.

10. REFERÊNCIAS

PYTEST. Pytest: helps you write better programs. Disponível em: . Acesso em: 06 Nov. 2025.

LOCUST. Locust - A modern load testing framework. Disponível em: . Acesso em: 06 Nov. 2025.

FASTAPI. FastAPI - Modern, fast web framework. Disponível em: . Acesso em: 06 Nov. 2025.

REACT. React - A JavaScript library for building user interfaces. Disponível em: . Acesso em: 06 Nov. 2025.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação - Trabalhos acadêmicos - Apresentação. Rio de Janeiro, 2011.