# Relatório Final — Projeto UC00606

## 1. Introdução e objetivos do projeto

O presente projeto tem como objetivo o desenvolvimento de uma plataforma de autenticação resiliente, com mecanismos de deteção e bloqueio de ataques de força bruta. O sistema implementa hashing seguro de palavras-passe com PBKDF2-HMAC-SHA256 e salt único por utilizador, um mecanismo de lockout com backoff exponencial, e uma análise automática de logs para identificar comportamentos suspeitos com base em heurísticas predefinidas.

## 2. Descrição do funcionamento do sistema

O sistema é composto por dois módulos principais: (1) o módulo de autenticação e (2) o módulo de análise. O primeiro permite criar utilizadores, efetuar login e registar todas as tentativas em ficheiro CSV, incluindo timestamp, utilizador, IP e resultado. O segundo módulo, implementado no ficheiro analytics.py, processa os registos e aplica regras de deteção de padrões de ataque, atualizando o ficheiro blacklist.json com IPs temporariamente ou permanentemente bloqueados.

## 3. Estrutura dos ficheiros e lógica principal

A estrutura do projeto é organizada em vários ficheiros Python:  
- auth.py: Implementa o hashing PBKDF2-HMAC-SHA256 e verificação de passwords.  
- storage.py: Gere os utilizadores e estados de lockout em ficheiros JSON.  
- login\_cli.py: Interface de linha de comando para criação de utilizadores e login, incluindo o registo em logs e aplicação de lockout.  
- analytics.py: Analisa o ficheiro de logs, aplica heurísticas e gera a blacklist.json.  
- generate\_logs.py: Gera dados de teste automáticos.  
- flowchart.mmd: Fluxograma geral do funcionamento.

## 4. Heurísticas de deteção e bloqueio

Foram implementadas três heurísticas principais de deteção de comportamentos suspeitos:  
1. 10 falhas de autenticação em 5 minutos → bloqueio temporário de 1 hora.  
2. 30 falhas em 24 horas → bloqueio permanente.  
3. 5 utilizadores distintos atacados pelo mesmo IP em 10 minutos → bloqueio temporário de 1 hora.  
Estas regras são avaliadas continuamente e os resultados armazenados em blacklist.json com timestamps legíveis (since\_human e until\_human).

## 5. Validação das bibliotecas utilizadas e análise de CVEs

Foi realizada uma auditoria de segurança às dependências do projeto, utilizando as ferramentas pip-audit e safety, ambas baseadas nas bases de dados NVD e MITRE CVE.  
  
Resultados obtidos:  
- pip-audit: Detetou inicialmente uma vulnerabilidade na versão 25.0.1 do pip (GHSA-4xh5-x5gv-qwph). Após atualização para a versão 25.3, não foram encontradas vulnerabilidades.  
- safety: Após correção de compatibilidades, foi executado com sucesso e não reportou vulnerabilidades conhecidas.  
  
Conclusão: Nenhuma vulnerabilidade crítica foi detetada nas dependências utilizadas.

## 6. Testes realizados e resultados

Foram realizados testes unitários e funcionais para validar a integridade do sistema:  
- Criação e autenticação de utilizadores.  
- Teste de lockout com backoff exponencial após falhas consecutivas.  
- Simulação de logs automáticos com generate\_logs.py (258 registos).  
- Execução de analytics.py para deteção de IPs suspeitos.  
  
Resultados: Foram corretamente detetados e bloqueados dois IPs (203.0.113.200 e 198.51.100.250) por cumprirem as heurísticas de ataque distribuído. O ficheiro blacklist.json foi atualizado com as datas humanas (since\_human e until\_human) correspondentes ao bloqueio.

## 7. Conclusão e possíveis melhorias

O projeto cumpre integralmente os requisitos definidos no enunciado, apresentando um sistema de autenticação seguro, resiliente e capaz de detetar e mitigar ataques automatizados.  
  
Como possíveis melhorias futuras, sugerem-se:  
- Integração com uma base de dados real (PostgreSQL ou SQLServer) em vez de ficheiros JSON.  
- Implementação de dashboard visual (Grafana ou Streamlit) para análise em tempo real.  
- Integração com APIs externas para reporte automático de IPs maliciosos.  
- Suporte a autenticação multifator (MFA) para reforçar a segurança do login.