CURSO DE CIÊNCIAS DA COMPUTAÇÃO

INSTITUTO SUPERIOR POLITÉCNICO DA CAÁLA

AUTORES:

MOISÉS CHONGOLOLA JOÃO

KIAKO GARCIA

JÚNIOR MAIENA

TITULO :

FACEALERT

TURMA 401

SEMESTRE _1º

DOCENTE

_____-

CAÁLA/2026

INTRODUÇÃO

O FaceAlert é um sistema de monitorização inteligente baseado em visão computacional, desenvolvido em Python com Flask, OpenCV e a biblioteca face_recognition. O sistema captura vídeo em tempo real, deteta rostos, compara com uma base de dados de pessoas cadastradas e gera eventos classificados como **CONHECIDO** ou **DESCONHECIDO**, apresentados num dashboard web.

O desenvolvimento foi conduzido segundo a metodologia **Extreme Programming (XP)**, que privilegia:

- entregas incrementais de software funcional;
- histórias de utilizador como requisito central;
- simplicidade do código;
- testes contínuos;
- refatoração permanente.

Esta documentação descreve o projeto de acordo com os artefactos oficiais do XP: papéis, user stories, planeamento por iterações, critérios de aceitação e evidência de implementação no código.

2. VISÃO DO PRODUTO (XP Product Vision)

2.1 Problema

Ambientes como residências, igrejas e escritórios dependem de vigilância humana, sujeita a falhas e ausência de registos estruturados.

2.2 Proposta de Valor

O FaceAlert automatiza a vigilância, criando:

- identificação contínua de pessoas;
- registo com evidência (snapshot + hora);
- diferenciação entre autorizados e desconhecidos;
- painel de decisão em tempo real.

2.3 Tecnologias do Código

- Backend: Flask (rotas /monitor, /people, /dashboard, /logout)
- IA: OpenCV + face_recognition
- Persistência: SQLite
- Frontend: HTML + CSS + Jinja2

3.PAPÉIS XP DO PROJETO

Papel XP	Equivalente no Face Alert	Responsabilidades
Custumer	Administrador	Define requisitos e valida funcionalidades
User	Morador	Utilizador Final do Sistema
System	Módulo	Executa Reconhecimento e alertas
Programmer	Desenvolvedor	Implementa Código
Test	Desenvolvedor /Cliente	Valida Critério

4. USER STORIES (ARTEFACTO CENTRAL XP)

US1 – Cadastro de moradores

Como Administrador
Quero registar moradores
Para criar base de reconhecimento

- Critério: formulário grava dados no banco
- Evidência no código: rota /people + tabela de pessoas

US2 – Registo de rosto

Como Morador
Quero capturar meu rosto pela webcam
Para ser reconhecido

- Critério: imagem gravada e embedding gerado

US3 – Captura em tempo real

Como Sistema
Quero ler frames da câmara
Para processar deteção

- Evidência: rota /monitor com stream OpenCV

US4 – Deteção de desconhecidos

Como Sistema
Quero comparar rostos com base
Para classificar conhecidos/desconhecidos

- Critério: cálculo de distância de embeddings

US5 – Emissão de alerta

Como Sistema
Quero registar evento e snapshot
Para notificar segurança

- Evidência: tabela events + campo snapshot

US7 – Login

Como Utilizador
Quero autenticar
Para proteger acesso

- Evidência: sessão Flask e rota /logout

US8 – Dashboard

Como Administrador
Quero ver KPIs e últimos alertas
Para decisão rápida

- Evidência direta no teu código HTML:
    - {{ people_count }}
    - {{ events_count }}
    - loop {% for e in recent %}

5. PLANEAMENTO POR ITERAÇÕES (XP Planning Game)

Iteração 1 – Base

- Setup Flask
- Acesso à câmara
- US1, US2

Iteração 2 – IA

- Reconhecimento
- US3, US4

Iteração 3 – Eventos

- Persistência
- US5

Iteração 4 – Interface

- Dashboard
- US7, US8

6. CRITÉRIOS DE ACEITAÇÃO (XP Acceptance Tests)

Historia	Test
Us3	Frame exibido em /monitor
Us4	Desconhecido marcado vermelho
Us5	Snapshot salvo
Us8	Dashboard mostra contadores

7. RASTREABILIDADE COM O CÓDIGO

Dashboard ↔ US8

O template implementa exatamente os requisitos:

- KPI de pessoas:
  {{ people_count }}
- KPI de eventos:
  {{ events_count }}

8. PRÁTICAS XP APLICADAS

8.1 Simplicidade

- Rotas pequenas
- Template direto
- Sem over-engineering

8.2 Feedback

- Testes com rostos reais
- Ajuste de confiança

8.3 Refatoração

- CSS responsivo
- tratamento onerror na imagem
- formatação de confiança

8.4 Integração Contínua

- cada iteração resultou em sistema executável

9. TESTES REALIZADOS

- Teste de câmara indisponível
- Teste de rosto conhecido
- Teste de desconhecido
- Teste do dashboard vazio
- Teste de sessão/logout

CONCLUSÃO

O desenvolvimento do **FaceAlert** demonstrou, de forma prática, a eficácia da metodologia **Extreme Programming (XP)** na construção de sistemas baseados em visão computacional. A utilização de histórias de utilizador como principal artefacto de requisitos permitiu que cada funcionalidade fosse implementada com foco direto nas necessidades reais do utilizador, garantindo rastreabilidade entre **requisito → código → teste → interface**.

A abordagem incremental do XP possibilitou que o sistema evoluísse de forma segura: iniciou-se com a captura de vídeo, avançou para o reconhecimento facial, posteriormente para o registo de eventos e, por fim, para a criação do dashboard decisional. Cada iteração resultou num produto executável, validado através de critérios de aceitação claros, como a identificação de desconhecidos, gravação de snapshots e apresentação dos KPIs no painel.

A integração entre o backend em Flask, os algoritmos de reconhecimento com OpenCV e face_recognition, e a interface em Jinja2 comprovou a conformidade entre a documentação XP e o código implementado. Elementos como a exibição dinâmica de people_count, events_count e da lista recent evidenciam que as histórias de utilizador foram materializadas diretamente na aplicação funcional.

Do ponto de vista académico e técnico, o projeto permitiu:

- aplicar conceitos reais de engenharia de software ágil;
- compreender a importância de testes contínuos e refatoração;
- integrar desenvolvimento web;
- produzir um sistema com valor prático para segurança.

Conclui-se que o FaceAlert não é apenas um protótipo, mas uma solução extensível, construída segundo boas práticas do XP, capaz de apoiar cenários reais de vigilância e controlo de acesso. A metodologia adotada garantiu qualidade, simplicidade e alinhamento entre expectativas do utilizador e implementação, validando o Extreme Programming como abordagem adequada para projetos de natureza evolutiva e experimental.