UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

CURSO DE ENGENHARIA INFORMÁTICA

RELATÓRIO DE ESTÁGIO PROFISSIONAL

Desenvolvimento de um sistema integrado e customizado de monitoramento de um circuito de televisão fechado distribuído para a ALTEL.

Autor:

Aiton António Cumbi

Supervisor da instituição:

Eng. Frederico Muianga

Supervisor da faculdade:

Eng. Rúben Manhiça

Maputo, Janeiro de 2024

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

CURSO DE ENGENHARIA INFORMÁTICA

RELATÓRIO DE ESTÁGIO PROFISSIONAL

Desenvolvimento de um sistema integrado e customizado de monitoramento de um circuito de televisão fechado distribuído para a ALTEL.

Autor:

Aiton António Cumbi

Supervisor da instituição:

Eng. Frederico Muianga

Supervisor da faculdade:

Eng. Rúben Manhiça

Maputo, Janeiro de 2024

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

CURSO DE ENGENHARIA INFORMÁTICA

TERMO DE ENTREAGA DO RELATÓRIO DE ESTÁGIO PROFISSIONAL

Declaro que Aiton António Cumbi, entregou no dia ____/____/2024, as ____ copias do seu relatório de estágio profissional, com referência **2023EIEPN6** intitulado: Desenvolvimento de um sistema integrado e customizado de monitoramento de um circuito de televisão fechado distribuído para ALTEL.

Maputo, _____ de Janeiro de 2024

A chefe da secretaria:

_____

Agradecimentos

Gostaria de agradecer, em primeiro lugar, aos meus pais, pois sem eles não seria capaz de frequentar este curso. Gostaria também de agradecer aos meus ex-colegas, André Come, Castigo Dramuce, Frederico Muianga, Júlio Langa e Valério Macumbuia, pelo incentivo e moral na realização deste relatório e ao longo do curso. Gostaria também de agradecer ao meu supervisor Eng.º Ruben Manhiça que disponibilizou o seu precioso tempo para me auxiliar na elaboração deste relatório.

Gostaria também de agradecer ao Sr. Manuel Gaivão pela oportunidade de estagiar na ALTEL e ser o líder do projecto relacionado com o presente relatório, e por partilhar o seu conhecimento e experiência para me auxiliar no desenvolvimento da minha carreira.

A todos os meus agradecimentos.

Resumo

Apresenta-se de seguida o relatório relativo as actividades realizadas durante o período de estágio profissional na empresa ALTEL soluções globais de comunicação, com o principal objectivo de descrever o processo de desenvolvimento de um sistema de monitorização para um sistema de CCTV distribuído, para uma organização cliente da ALTEL. O actual sistema de monitorização de CCTV da organização depara-se com várias limitações, como a dificuldade no acesso remoto às transmissões de vídeo em directo, a dependência em tecnologia desactualizada e descontinuada, a impossibilidade de múltiplas sessões de utilizador e de transmissão simultânea. O projecto visa desenvolver um sistema de monitorização de CCTV mais seguro e moderno, que seja multiplataforma, incluindo Android, iOS e Web, escalável, uma vez que permite múltiplas sessões de utilizador e transmissão em simultâneo, e orientado para a segurança, aplicando as melhores práticas de desenvolvimento e empregando protocolos seguros para comunicação.

O sistema proposto foi construído com base na arquitectura cliente-servidor com recurso à linguagem de programação TypeScript, recorrendo também à utilização de bibliotecas UI JavaScript, como React e React-native, para as aplicações cliente. A metodologia de desenvolvimento seguiu uma abordagem incremental e foi efectuada uma pesquisa bibliográfica a fim de adquirir conhecimentos sobre tópicos relevantes, como protocolos de comunicação em dispositivos de segurança, protocolos de transmissão de vídeo e boas práticas de código seguro.

Pretende-se que o sistema proposto resolva as limitações do sistema de monitorização existente e forneça uma solução mais moderna, segura e flexível para o acesso às imagens em directo do sistema de CCTV.

palavras-chave: CCTV, sistema de gestão de vídeo, sistema de monitorização, ONVIF, RTSP, HLS, TypeScript, NodeJS

Abstract

The following is a report on the activities carried out during the professional internship period at ALTEL Global Communication Solutions, with the main objective of describing the process of developing a monitoring system for a distributed CCTV system for an organization that is a client of ALTEL. The organization's current CCTV monitoring system faces several limitations, such as the difficulty in remotely accessing live video transmissions, reliance on outdated and discontinued technology, the impossibility of multiple user sessions and simultaneous transmission. The project aims to develop a more secure and modern CCTV monitoring system that is multi-platform, including Android, iOS, and Web, scalable, as it allows multiple user sessions and simultaneous transmission, and security-oriented, applying best development practices and employing secure protocols for communication.

The proposed system was built based on client-server architecture using the TypeScript programming language, and using JavaScript UI libraries, such as React and React-native, for the client applications. The development methodology followed an incremental approach and bibliographic research was conducted to acquire knowledge on relevant topics, such as communication protocols in security devices, video transmission protocols and good practices for secure code.

The proposed system is intended to resolve the limitations of the existing monitoring system and provide a more modern, secure, and flexible solution for accessing live images from the CCTV system.

keywords: CCTV, video management system, monitoring system, ONVIF, RTSP, HLS, TypeScript, NodeJS

Índice

Agradecimentos i
Resumo ii
Abstract iii
Índice iv
Lista de figuras ix
Lista de tabelas xi
Lista de acrónimos e abreviaturas xii
Glossário xiii
1 Capítulo I - Introdução 1
1.1 Contextualização 1
1.2 Definição do problema 2
1.3 Justificativa 2
1.4 Objectivos 3
1.4.1 Geral 3
1.4.2 Específicos 3
1.5 Metodologia 4
1.5.1 Análise de requisitos. 4
1.5.2 Revisão de literatura. 4
1.5.3 Desenvolvimento do sistema. 4
2 Capítulo II - Revisão de literatura. 7
2.1 Circuito de televisão fechado (CCTV) 7
2.1.1 Câmeras de segurança: 8
2.1.2 Dispositivo de gravação 8

2.1.3 Dispositivo de apresentação (Monitor) 8
2.1.4 Cablagem e infra-estrutura 9
2.2 Sistema de CCTV distribuído 9
2.3 Sistema de monitorização de CCTV 9
2.4 Sistema de gestão de vídeo (VMS) 10
2.4.1 Comparação de soluções de gestão de vídeo 10
2.5 Padrões industriais para sistemas de videovigilância. 13
2.5.1 Open network video interface forum - ONVIF 14
2.5.2 Physical Security Interoperability Alliance - PSIA. 15
2.6 Real time streaming protocol (RTSP) 17
2.6.1 Real time streaming protocol em sistemas de CCTV 17
2.7 Transmissão de vídeo em directo em plataformas modernas 17
2.7.1 HTTP Live Streaming - HLS 18
2.7.2 Web Real Time Communication - WebRTC 18
2.7.3 Dynamic adaptive streaming over HTTP - DASH 19
2.8 Comparação de “*tech stacks*” para desenvolvimento de software 20
2.8.1 MERN – MongoDB, ExpressJS, React, NodeJS 20
2.8.2 LAMP – Linux, Apache, MySQL, PHP 22
3 Capítulo III - Caso de estudo 24
3.1 Apresentação da instituição de estágio 24
3.1.1 Missão 24
3.1.2 Visão 25
3.1.3 Princípios 25
3.1.4 Valores 25
3.2 Serviços prestados pela ALTEL 26

3.3 Horário de trabalho 26
3.4 Estrutura da instituição 27
3.5 Descrição das actividades desempenhadas 27
3.5.1 Actividades rotineiras 27
3.5.2 Projectos participados 28
3.6 Situação actual 29
3.7 Constrangimentos apresentados 30
4 Capítulo IV – Descrição da solução 31
4.1 Recolha e análise de requisitos 31
4.1.1 Problemas actuais 31
4.1.2 Requisitos de software 31
4.1.3 Requisitos funcionais 32
4.1.4 Requisitos não funcionais 33
4.2 Concepção do sistema (design) 33
4.2.1 Ferramentas utilizadas para o desenvolvimento do sistema (prototipagem) 34
4.2.2 Descrição dos actores do sistema 35
4.2.3 Diagrama de casos de uso 35
4.2.4 Diagrama de classes 36
4.2.5 Arquitectura do sistema 37
4.3 Desenvolvimento do sistema 39
4.3.1 Ferramentas de codificação 39
4.3.2 Base de dados 41
4.3.3 Servidor de backend (API) 41
4.3.4 Servidor de media (Transmissão das imagens de vídeo das câmeras de vigilância) 44

4.3.5 Aplicações front-end 45
4.3.6 Ferramentas utilizadas no desenvolvimento do servidor de backend 46
4.4 Testes de código 47
4.4.1 Testes unitários e de integração 47
4.4.2 Testes de API 47
4.5 Implementação (Deployment) 48
4.5.1 Implementação da aplicação Web: 50
4.5.2 Implementação da API de backend (pública e privada) 52
4.5.3 Implementação da base de dados 55
4.6 Manutenção do código 55
4.6.1 Contínua monitorização para a mitigação de vulnerabilidades no código fonte e dependências 55
4.6.2 Qualidade do código fonte 56
5 Capítulo V - Conclusões e Recomendações 57
5.1 Conclusão 57
5.2 Recomendações 58
Referências bibliográficas 59
6 Anexos 66
6.1 Anexo 1 - Diagrama de casos de estudo (versão extensa) 66
6.1.1 Descrição dos casos de uso 66
6.2 Anexo 2 – Diagrama de classes e diagramas de sequência 70
6.2.1 Diagramas de sequência 70
6.3 Anexo 3 – Interfaces de utilizador 72
6.3.1 Aplicação Web 72
6.3.2 Aplicação móvel 79

6.4 Anexo 4 – Plano de actividades 87

Lista de figuras

Figura 1: Interacção cliente servidor WSDL (Fonte: (Awati et al., 2014, 742)) 14
Figura 2:Modelo de troca de mensagens com dispositivos ONVIF (Fonte: (Awati et al., 2014, 742)). 15
Figura 3:: Exemplo de uma arquitectura PSIA. (Fonte: PSIA service model v3.0). 16
Figura 4: Estrutura da instituição de estágio (fonte: elaboração própria) 27
Figura 5: Diagrama de casos de uso (compacto, referir ao anexo 1 para a versão extensa) 36
Figura 6: Diagrama de classes compacto (referir ao anexo 2 para a versão extensa) . 37
Figura 7: Arquitectura do sistema (fonte: elaboração própria) 39
Figura 8: Fluxo de autenticação: 43
Figura 9:Endpoints da API 44
Figura 10: Arquitectura do servidor de media 45
Figura 12: dashboard postman 47
Figura 13: Instalação do Internet Information Services (IIS) (Fonte: Elaboração própria) 50
Figura 14: Criação de um certificado HTTPS auto assinado (Fonte: Elaboração própria) 51
Figura 15: Compilação do website (Fonte: Elaboração própria) 51
Figura 16: Configuração do website no IIS (Fonte Elaboração própria)..... 52
Figura 17: Configuração da virtualização em cascata em uma máquina virtual (Fonte: Elaboração própria)..... 53
Figura 18: Compilação do contentor docker..... 53
Figura 19: Contentores na consola docker (Fonte: elaboração própria) 54
Figura 20: Consola do contentor docker com a Implementação da API (Fonte: Elaboração própria)..... 54
Figura 21: MongoDB compass (Fonte: Elaboração própria) 55
Figura 11: dashboard synk-code..... 56
Figura 22: Diagrama de casos de uso, versão extensa 66
Figura 23: Diagrama de sequência de login (Fonte: Elaboração própria) 70

Figura 24: Diagrama de sequência de visualização da transmissão em directo (Fonte: elaboração própria) 71
Figura 25: Ecrã de login (Fonte: Elaboração própria)..... 72
Figura 26: Ecrã da transmissão em directo (Fonte: elaboração própria)..... 73
Figura 27: Ecrã da transmissão em directo (Fonte Elaboração própria) 73
Figura 28: Ecrã de gestão de utilizadores (Fonte: elaboração própria)..... 74
Figura 29: Ecrã de adição de utilizadores (fonte elaboração própria) 74
Figura 30: Ecrã de edição de utilizadores (fonte elaboração própria) 75
Figura 31: Ecrã da gestão de câmeras de segurança (Fonte: Elaboração própria) 75
Figura 32: Ecrã de adição de uma câmera de segurança (Fonte: Elaboração própria) 76
Figura 33: Ecrã de edição de uma câmera de segurança (Fonte: elaboração própria) 76
Figura 34: Ecrã de gestão de localizações (Fonte: elaboração própria) 77
Figura 35: Ecrã de adição de localizações (Fonte: Elaboração própria) 77
Figura 36: Ecrã de edição de uma localização (Fonte: Elaboração própria)..... 78
Figura 37: Ecrã de login (móvel) (Fonte: elaboração própria)..... 79
Figura 38: Ecrã da troca de password (móvel) (Fonte: elaboração própria) 80
Figura 39: Ecrã inicial (móvel) (Fonte: elaboração própria)..... 81
Figura 40: Ecrã da lista de localizações (móvel) (Fonte: elaboração própria)..... 82
Figura 41: Ecrã de listagem das câmeras de segurança (móvel) (Fonte: elaboração própria)..... 83
Figura 42: Ecrã da transmissão em directo das câmeras de segurança (móvel) (Fonte: elaboração própria)..... 84
Figura 43: Ecrã da transmissão em directo das câmeras de segurança (móvel-pressionado) (Fonte: elaboração própria) 85
Figura 44: Ecrã do perfil de utilizador (móvel) (Fonte: elaboração própria)..... 86

Lista de tabelas

Tabela 1:Comparação dos sistemas de gestão de vídeo (fonte: (VMS - Video Management Solution Review, Comparison, Best Products, Implementations, Suppliers., n.d.) 11
Tabela 2: comparação dos sistemas de gestão de vídeo (continuação) (fonte: (VMS - Video Management Solution Review, Comparison, Best Products, Implementations, Suppliers., n.d.) 13
Tabela 4: Constrangimentos apresentados 30
Tabela 5: Requisitos funcionais (fonte: elaboração própria) 32
Tabela 6: Requisitos não funcionais (fonte: elaboração própria) 33
Tabela 7: Funcionalidades da API privada 42
Tabela 8: Funcionalidades da API pública 42
Tabela 9: Descrição do UC_001 67
Tabela 10: Descrição do UC_002 67
Tabela 11: Descrição do UC_003 68
Tabela 12: Descrição do UC_004 69
Tabela 13: Descrição do UC_005 69

Lista de acrónimos e abreviaturas

API	Application programming interface
CCTV	Closed circuit television
CLI	Command line interface (Interface de linha de comando)
DVR	Digital video recorder
HLS	HTTP live streaming
HTTP/HTTPS	Hypertext transfer protocol / secure
IDE	Integrated development environment (Ambiente de desenvolvimento integrado)
JSON	JavaScript Object Notation
JWT	JSON Web Token
NAS	Network attached storage
NVR	Network video recorder
ONVIF	Open network video interface forum
PSIA	Physical security interoperability alliance
RTP	Real time protocol (protocolo em tempo real)
RTSP	Real time streaming protocol
SOAP	Simple object access protocol
TCP	transmission control protocol
UDP	User datagram protocol
UI	User interface (Interface de utilizador)
UML	Unified modeling language (Linguagem universal de modelação)
URI	Unique resource identifier (identificador de recurso único)
UTP	Untwisted shielded pair
UX	User experience (Experiência de utilizador)
VMS	Video management system
WSDL	Web services description language
XML	Extensible markup language

Glossário

Framework - Uma framework é uma colecção de código pré-escrito que fornece uma base para a criação de softwares.

API (application programming interface) - Uma API é um conjunto de regras e especificações que possibilitam que diferentes componentes de software comuniquem e troquem dados entre si.

Base de dados - uma base de dados é um conjunto organizado de informações estruturadas, ou dados.

Tech Stack - A totalidade de um sistema ou aplicação informática, incluindo tanto o front-end como o back-end.

Back-end - É o componente que acomoda a lógica do negócio em uma aplicação , que não é directamente interagida com o utilizador final.

Front-End – Refere-se à interface de utilizador de uma aplicação, que é directamente interagida pelo utilizador final.

1 Capítulo I - Introdução

1.1 Contextualização

Nos últimos tempos, a procura de soluções avançadas de vigilância e segurança aumentou significativamente em vários sectores, de acordo com a pesquisa efectuada por (Fortune Business Insights, 2023) em 2021, o tamanho do mercado global de câmeras de vigilância de CCTV foi avaliado em 31,88 bilhões de dólares (USD), e espera-se que até 2029 este apresente um crescimento até os 105,20 bilhões.

Os sistemas de CCTV evoluíram muito para além das suas funções tradicionais, tornando-se ferramentas indispensáveis para monitorizar, gravar e analisar actividades em diversos ambientes. Um dos principais componentes dos sistemas de CCTV é o software de monitorização, que permite o acesso ao sistema, as imagens capturadas por estes sistemas e as gravações armazenadas pelo gravador.

No entanto, apesar do uso abrangente dos sistemas de CCTV, ocasionalmente organizações encontram desafios ao tentar adaptar softwares de monitoramento de CCTV para atender às suas necessidades específicas. Normalmente o software de monitorização de CCTV é fornecido com o próprio sistema de CCTV, o que restringe a possibilidade de customização, que por sua vez são susceptíveis de conduzir a falhas críticas operacionais, de segurança e privacidade. Há ainda a possibilidade do uso de uma solução de terceiros, mas às vezes estas não se alinham perfeitamente com as necessidades exclusivas de cada ambiente.

Resulta assim do conjunto destas limitações a necessidade do desenvolvimento de um sistema de monitorização de CCTV ou de gestão de vídeo customizado, que possa ser adaptado às necessidades específicas da organização.

O presente relatório tem como visão descrever as actividades desempenhadas na

empresa ALTEL Soluções Globais de Comunicação, enquanto na qualidade de estagiário e apresentar o processo de desenvolvimento de um sistema de monitorização de CCTV para uma organização cliente da ALTEL.

1.2 Definição do problema

A ALTEL deparou-se com o desafio de desenvolver uma solução de monitorização de um sistema de CCTV para um dos seus clientes, o sistema de CCTV em questão encontra-se instalado em várias localizações geográficas, o que, por sua vez, implica que o sistema de monitorização permita várias sessões de utilizadores, vários utilizadores a acederem a transmissão de vídeo de uma câmera de vigilância e o acesso remoto fora da rede local (a partir da internet). O actual sistema de monitorização de CCTV não só tem inobservado estes requisitos, bem como acresce ainda outros inconvenientes; Em primeiro lugar, o sistema de monitorização só pode ser acedido através do agora obsoleto e descontinuado navegador web internet Explorer, além disso, recorre ao HTTP como protocolo de transmissão de dados, o que, por sua vez, introduz um risco de segurança, uma vez que os dados em trânsito não são encriptados. O outro aspecto apresentado é a privacidade, tendo sido solicitado que as transmissões em directo de vídeo só estivessem disponíveis para indivíduos autorizados e que não fossem armazenados ou transmitidos por provedores de serviços em nuvem de terceiros, assim como a capacidade de limitar o acesso a determinadas localizações geográficas a utilizadores específicos.

1.3 Justificativa

O projecto inicial focava no desenvolvimento de uma aplicação móvel personalizada para o acesso à transmissão das imagens de vídeo das câmeras de segurança. Contudo, com o desenvolvimento do projecto e a evolução das necessidades, percebeu-se a necessidade de uma solução mais abrangente. Esta mudança foi impulsionada por vários factores cruciais, entre eles: A adaptação multiplataforma - um sistema de

monitorização eficiente deve operar tanto em dispositivos móveis quanto em computadores pessoais, o que permite a monitorização da transmissão de forma conveniente e eficaz. Substituição de tecnologia obsoleta - foi expressa a necessidade de substituição do sistema de monitorização actual por uma solução modernizada pois este apresenta limitações em termos de funcionalidade, escalabilidade e segurança. Preocupações com a privacidade em softwares de terceiros - com uma solução desenvolvida internamente, assegura-se o controlo completo sobre a transmissão e armazenamento de dados. Personalização e flexibilidade - um software desenvolvido internamente oferece a vantagem da adaptação às necessidades específicas, assegurando a integração à infra-estrutura e aos processos operacionais da organização. Preparação para o futuro - o investimento numa solução interna representa mais do que a resolução das necessidades imediatas, mas é também um investimento nas futuras capacidades e necessidades de segurança e vigilância da organização.

1.4 Objectivos

1.4.1 Geral

Desenvolver um sistema customizado de monitorização para um circuito de televisão fechado e distribuído.

1.4.2 Específicos

- Caracterizar o actual panorama de acesso às imagens de vídeo em tempo real no sistema de CCTV distribuído.
- Compreender o princípio de funcionamento dos protocolos de comunicação em sistemas de CCTV.
- Desenvolver uma ferramenta multiplataforma de transmissão de imagens de vídeo em directo que seja compatível com Android, iOS e Web.

1.5 Metodologia

Para atingir os objectivos enunciados neste relatório, é necessário adoptar e implementar um processo estruturado. Assim sendo, apresentam-se de seguida as etapas e respectivas metodologias.

1.5.1 Análise de requisitos.

Serão realizadas entrevistas com os utilizadores finais através de reuniões online para recolher todas as necessidades e problemas actuais, o software de monitorização de CCTV existente será também analisado para recolher as características e requisitos a implementar.

1.5.2 Revisão de literatura.

Será realizada uma extensa revisão da literatura para recolher informações relevantes sobre temas como o funcionamento das comunicações para as câmeras de CCTV, quais os protocolos utilizados pelas câmeras de CCTV para transmitir imagens de vídeo através de redes de computadores e quais os protocolos multiplataforma de transmissão de vídeo em directo que podem ser utilizados para transmitir imagens de vídeo de câmeras de segurança.
Esta revisão será composta por uma pesquisa bibliográfica de livros, artigos e relatórios em fontes académicas de renome, como JSTOR, Google scholar, ResearchGate e IEEExplore.

1.5.3 Desenvolvimento do sistema.

O desenvolvimento do sistema seguirá a metodologia em cascata, que é concebido para orientar a progressão de um projecto de desenvolvimento através de diferentes etapas meticulosamente definidas. À semelhança do que sucede na descida constante da água numa cascata, esta abordagem segue uma estrutura faseada e progressiva, em que

cada fase serve de pré-requisito para a fase seguinte. Assim, seguem-se as fases de composição da metodologia em cascata:

Análise de requisitos

A fase de iniciação do projecto envolve uma análise aprofundada dos requisitos do projecto. Esta fase preliminar requer uma compreensão abrangente das necessidades e expectativas do cliente. Através de uma análise minuciosa, delineia-se um plano detalhado das características e funcionalidades esperadas para o produto final.

Concepção do sistema (Design)

Após a definição dos requisitos, o foco passa para a concepção do sistema. Esta fase implica a formulação de um plano de arquitectura consolidado que delineia a estrutura e os módulos do sistema. As especificações de concepção, que incluem detalhes técnicos, são meticulosamente traçadas, de modo a fornecer um guião para os sucessivos esforços de desenvolvimento.

Desenvolvimento do sistema

Subsequentemente, o projecto passa para a fase de desenvolvimento, em que o sistema concebido torna-se um produto tangível. Esta fase inclui as actividades de codificação e programação, que transformam as construções teóricas da fase de concepção numa realidade funcional.

Testes

A integridade e a funcionalidade do sistema implementado são sujeitas à rigorosos controlos da fase de teste. São aplicados vários procedimentos de teste para identificar e rectificar quaisquer discrepâncias ou anomalias. Esta fase garante que o sistema cumpre as especificações predefinidas e não contém erros ou avarias.

Implementação

Após a finalização e o teste exaustivo do sistema, este passa à fase de implementação. Esta fase envolve a disponibilização do sistema aos utilizadores finais, marcando o

culminar do processo de desenvolvimento. Significa a transição do desenvolvimento para a produção.

Manutenção

Esta fase envolve o apoio contínuo e a melhoria do sistema implementado. Uma vez utilizado, o sistema pode deparar-se com problemas ou necessitar de actualizações para se adaptar às novas necessidades. A fase de manutenção aborda estas questões, ao resolver erros, introduzir melhorias e garantir que o sistema se mantém eficaz e eficiente ao longo do tempo. Implica um ciclo contínuo de monitorização, aperfeiçoamento e adaptação do software para garantir a sua funcionalidade e relevância sustentadas.

2 Capítulo II - Revisão de literatura.

A seguinte revisão de literatura aborda tópicos relacionados com os objectivos do presente relatório. É feita a análise do que são sistemas de CCTV, softwares de monitorização de CCTV, sistemas de gestão de vídeo (VMS), de que forma a comunicação com sistemas de CCTV opera em redes de computadores e que protocolos são utilizados para comunicar, transmitir e obter imagens em directo de câmeras de segurança.

2.1 Circuito de televisão fechado (CCTV)

Segundo (Dempsey, 2007, 78) circuito de televisão fechado é um sistema de vídeo privado para monitorização da segurança num edifício, estabelecimento ou área geográfica. (Deisman, 2003) por sua vez, define circuito de televisão fechado como sistemas de vigilância electrónica que utilizam câmeras de vídeo, ligadas por meio de um circuito "fechado" (ou não radiodifundido), para captar, recolher, registar e/ou transmitir informações visuais sobre o estado dos acontecimentos num determinado espaço ao longo do tempo.
(Damjanovski, 2005, 321) centra-se no carácter definitivo dos sistemas de CCTV descrevendo-os como um sistema de televisão destinado apenas a um conjunto restrito e específico de espectadores que se opõem à televisão de radiodifusão.

Considerando as interpretações dos referidos autores, podemos concluir que circuito fechado de televisão é a utilização da tecnologia de vigilância cujo objectivo centra-se na monitorização e registo de actividades numa determinada área. O principal objectivo do CCTV é proporcionar segurança, protecção e dissuasão contra a incidência de condutas ilícitas no espaço em que se encontra.

Os sistemas de CCTV são normalmente compostos pelos seguintes elementos:

2.1.1 Câmeras de segurança:

São responsáveis pela captação de imagens em directo e pela transmissão das mesmas para um dispositivo central de gravação, normalmente um DVR ou NVR. As câmeras de segurança podem ser analógicas ou digitais, fixas ou móveis. As câmeras analógicas transmitem sinais analógicos e as câmeras convertem o sinal analógico em digital e os transmitem como sinais digitais. As câmeras fixas apenas monitorizam e gravam a actividade numa área específica, enquanto as móveis podem ser ajustadas à distância para cobrir uma área maior.

2.1.2 Dispositivo de gravação

Encarregue de armazenar, processar e exibir as imagens captadas pelas câmeras de segurança. Os dispositivos de gravação de CCTV podem ser de dois tipos: DVR (gravador de vídeo digital) ou NVR (gravador de vídeo em rede). Regra geral, este dispositivo é também onde o software de monitorização de CCTV está instalado e pode ser acedido. Os DVRs são normalmente instalados em conjunto com câmeras de segurança analógicas, uma vez que a sua principal função consiste em converter o sinal analógico recebido pelas câmeras em digital, para que possa ser visualizado e gravado em sistemas digitais. Os NVRs, por outro lado, são normalmente combinados com câmeras digitais e não armazenam as gravações neles próprios, mas sim num dispositivo de armazenamento ligado à rede (NAS). Neste cenário, toda a conversão do sinal é efectuada pela câmera de segurança e, em geral, o dispositivo de gravação funciona como um dispositivo de gestão central ao qual todas as câmeras de segurança se ligam, quer através de cabos coaxiais, quer através de redes IP utilizando cabos Ethernet.

2.1.3 Dispositivo de apresentação (Monitor)

Utilizados para visualizar as imagens em directo ou as gravações captadas pelas câmeras de segurança. Os monitores são normalmente instalados em locais estratégicos, como salas de controlo ou centros de operações.

2.1.4 Cablagem e infra-estrutura

Sistemas de CCTV implicam a existência de uma infra-estrutura de cablagem adequada, que pode ser composta por cabos coaxiais, UTP ou fibra óptica para transmitir os sinais de vídeo captados. Dependendo da sua dimensão e distribuição, podem ainda passar por equipamento de rede, como routers e switches, a fim de assegurar a gestão do fluxo de dados.

2.2 Sistema de CCTV distribuído

A expressão CCTV distribuído permite inferir que se refere a um sistema de CCTV disperso ou distribuído por múltiplas localizações geográficas, tais como diferentes instalações, cidades, províncias ou mesmo países. Num sistema de CCTV distribuído instalam-se múltiplas câmeras de segurança em cada espaço de observação sendo posteriormente as imagens capturadas transmitidas a um servidor de gestão centralizado o qual permite acesso às transmissões em directo de cada localização.

2.3 Sistema de monitorização de CCTV

O software de monitorização de CCTV refere-se à tecnologia aplicacional que permite a monitorização de diferentes locais ou regiões utilizando câmeras de circuito de televisão fechado (CCTV) (Doni, 2019). Recorre à utilização de redes informáticas e da Internet para permitir a monitorização numa área ampla, ou mesmo à escala global (Kang Min Jae et al., 2017). O software inclui normalmente funcionalidades como a detecção de objectos, a emissão de sinais de alerta e o processamento de imagens para detectar e monitorizar de forma eficaz a presença de indivíduos ou objectos no local de monitorização (Park Sung Ha et al., 2020).

2.4 Sistema de gestão de vídeo (VMS)

Um sistema de gestão de vídeo é uma plataforma baseada em software destinada à gestão e controlo de câmeras de segurança, dispositivos de gravação e outros componentes de segurança. Os sistemas de gestão de vídeo são normalmente utilizados por empresas, governos e outras organizações que necessitam de monitorização de segurança e vigilância em larga escala. (iSARSOFT, 2023)

2.4.1 Comparação de soluções de gestão de vídeo

Sistema de gestão de vídeo			
Características	**Eocortex**	**Omnicast**	**Wisenet Wave**
Dispositivos suportados	5000	6000	5000
Escalabilidade	Requer licenciamento	Requer licenciamento	Sim
Gestão centralizada	Requer licenciamento	Requer licenciamento	Sim
Dispositivos IP suportados	Câmera DVR NVR Microfone Telefone de entrada	Câmera DVR NVR Altifalantes Microfone Telefone de entrada	Câmera DVR NVR Altifalantes Microfone Telefone de entrada
Quantidade de câmeras por servidor	Ilimitada / Requer licenciamento	Ilimitada / Requer licenciamento	128
Restrição de largura de banda	Ilimitada	200Mb / Câmera	Ilimitada

Sessões de utilizador	Ilimitada / Requer licenciamento	Ilimitada / Requer licenciamento	Ilimitada
Versão móvel	Móvel e Web	Não suportado	Móvel e Web
Transmissão de vídeo em dispositivos moveis	Suportado	Não suportado	Apenas transmissões ao vivo
Versão gratuita	Não	Não	Apenas transmissões ao vivo
Licenciamento	Por dispositivo	Por dispositivo	Por dispositivo
Quantidade de transmissões em simultâneo	Ilimitada / Requer licenciamento	Ilimitada	2
Protocolos suportados	HTTP MJPEG ONVIF PSIA	ONVIF RTSP NTP (native video transport)	HLS HTTP HTTPS MJPEG ONVIF RTSP WebM
Modelo de Renovação de licença	Subscrição Anual	Subscrição anual	Subscrição anual

*Tabela 1:Comparação dos sistemas de gestão de vídeo (fonte: (*VMS - Video Management Solution Review, Comparison, Best Products, Implementations, Suppliers*., n.d.)*

Sistema de gestão de vídeo			
Características	Luxriot evo	Milestone Xprotect	Network Optix
Dispositivos suportados	5000	7000	5000
Escalabilidade	Requer licenciamento	Requer licenciamento	Sim
Gestão centralizada	Requer licenciamento	Requer licenciamento	Sim
Dispositivos IP suportados	Câmera DVR NVR	Câmera DVR NVR Altifalantes Microfone Telefone de entrada	Câmera DVR NVR Altifalantes Microfone Telefone de entrada
Quantidade de câmeras por servidor	Ilimitada / Requer Licenciamento	Ilimitada / Requer Licenciamento	128
Restrição de largura de banda	Ilimitada	Ilimitada	Ilimitada
Sessões de utilizador	Ilimitada / Requer Licenciamento	Ilimitadas	Ilimitadas
Versão móvel	Não suportado	Móvel e Web	Móvel e Web
Transmissão de vídeo em dispositivos moveis	Suportado	Suportado	Suportado
Versão gratuita	Apenas 9 sessões de 2Mp	Sim	Apenas para transmissões ao vivo

Licenciamento	Por sessão	Por dispositivo	Por dispositivo
Quantidade de transmissões em simultâneo	Ilimitada / Requer Licenciamento	Ilimitadas	2
Protocolos suportados	HTTP ONVIF PSIA RTSP	HTTP HTTPS ONVIF PSIA RTSP	HLS HTTP HTTPS MJPEG ONVIF RTSP WebM
Modelo de Renovação de licença	Subscrição bianual, 4 anos ou 10 anos	Subscrição anual	Subscrição anual

*Tabela 2: comparação dos sistemas de gestão de vídeo (continuação) (fonte: (*VMS - Video Management Solution Review, Comparison, Best Products, Implementations, Suppliers.*, n.d.)*

Após definir os elementos fundamentais dos sistemas de CCTV, passamos agora a abordar a dinâmica da comunicação nestes sistemas. A ênfase recai sobre a transmissão de vídeo em directo e os procedimentos para estabelecer comunicação com uma câmera de segurança, bem como os protocolos utilizados por estas para transmitir imagens ao vivo.

Antes de podermos aceder às imagens em directo de uma câmera de segurança, é necessário identificá-la na rede. Afortunadamente existem padrões ou especificações de sistemas de segurança desenvolvidos e frequentemente adoptados pelos fabricantes.

2.5 Padrões industriais para sistemas de videovigilância.

Actualmente, existem dois principais padrões industriais para segurança e vigilância, o ONVIF e o PSIA, decorrentes da necessidade existencial de protocolos para a

comunicação, e interoperabilidade entre equipamentos de segurança de diferentes tecnologias, plataformas e fabricantes.

2.5.1 Open network video interface forum - ONVIF

ONVIF, acrónimo de Open Network Video Interface Fórum, consiste num consórcio global aberto da indústria que se centra na normalização dos protocolos de comunicação para produtos de segurança física baseados em IP. Fundado em 2008 pela *Axis communications*, *Bosh Security Systems* e *Sony Corporation*, o ONVIF estabelece um modelo de rede para a comunicação em interfaces, e uma estrutura de dados para o intercâmbio de mensagens. Concentra-se na descoberta automática de dispositivos e na transferência de meta dados mantendo a sua integridade, utilizando tecnologias como serviços web e o protocolo de transmissão em tempo real (RTSP). (Awati et al., 2014, 741)

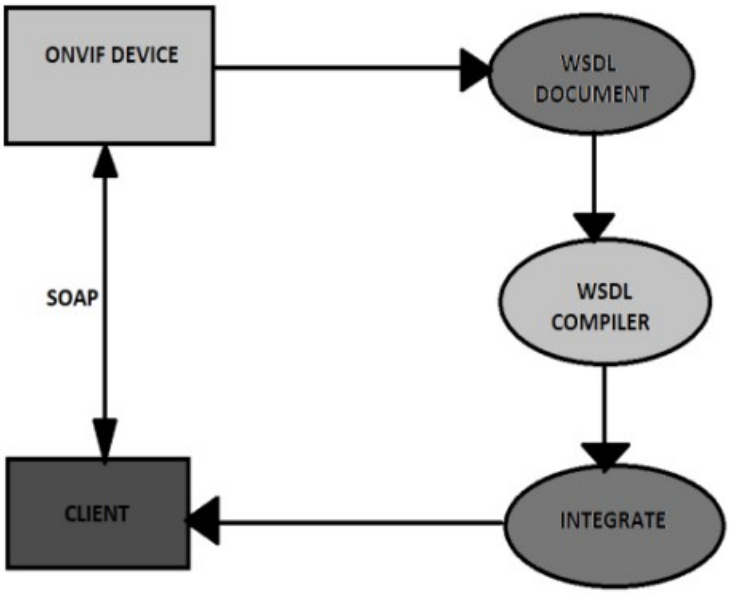

Figura 1: Interacção cliente servidor WSDL (Fonte: (Awati et al., 2014, 742))

As normas utilizadas na integração de aplicações com dispositivos ONVIF são, WSDL, SOAP, XML e RTSP. A linguagem de descrição de serviços Web (WSDL) é empregue na descrição dos serviços suportados pelo dispositivo ONVIF. A comunicação por via de mensagens entre os clientes e dispositivos ONVIF efectua-se através do protocolo SOAP, o XML é utilizado como a sintaxe de modelação das mensagens e o RTSP é usado na transmissão das imagens de vídeo do dispositivo.

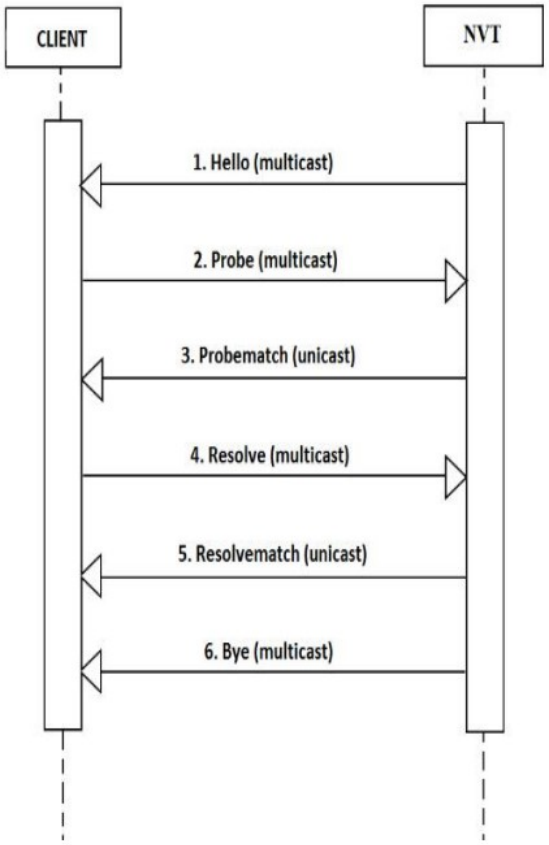

Figura 2:Modelo de troca de mensagens com dispositivos ONVIF (Fonte: (Awati et al., 2014, 742)).

2.5.2 Physical Security Interoperability Alliance - PSIA.

À semelhança do ONVIF, PSIA, acrónimo de Physical Security Interoperability Alliance, trata-se de um consórcio global de empresas fabricantes de produtos de segurança física e integradores de sistemas. Proposto por Cisco Systems, IBM, Panasonic e Pelco em 2008, tem como objectivo promover a interoperabilidade de equipamentos e sistemas de

segurança com suporte IP em todo o ecossistema de segurança. (Chia-Hsu Kuo et al., 2013)

No entanto, contrariamente à sua contraparte baseada em WSDL e SOAP, o PSIA utiliza RestFul APIs na comunicação entre os seus dispositivos compatíveis.

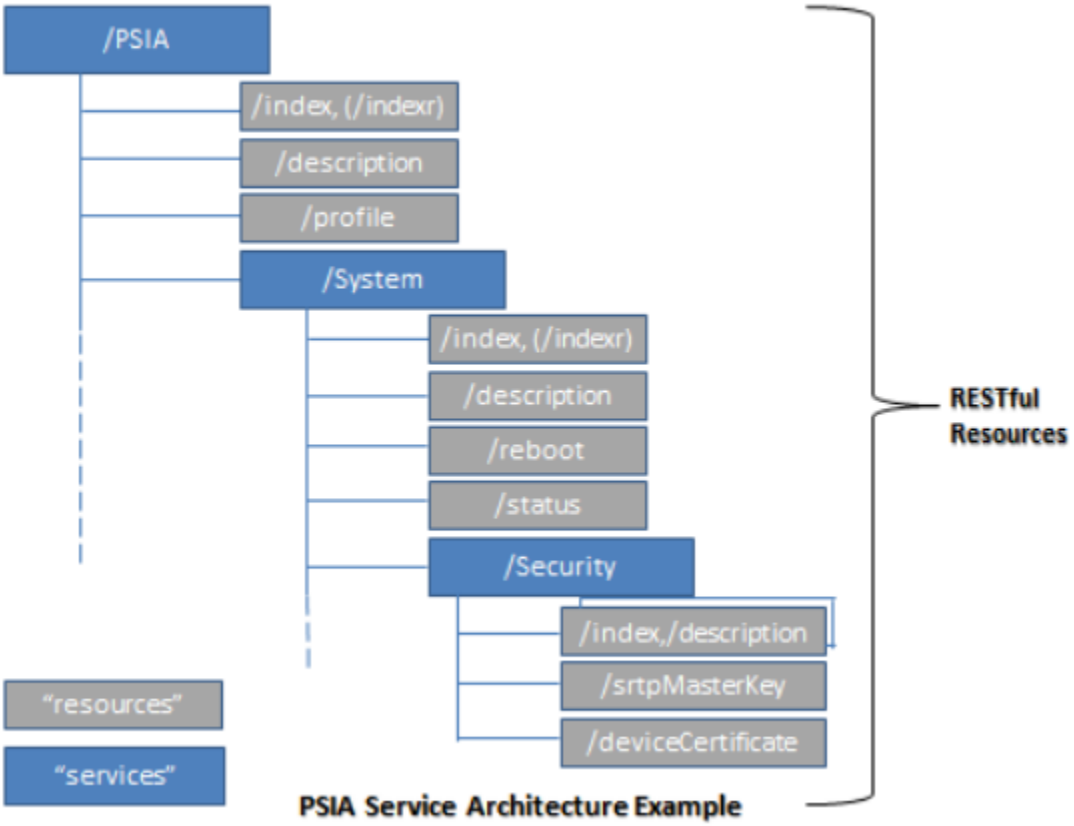

Figura 3:: Exemplo de uma arquitectura PSIA. (Fonte: PSIA service model v3.0).

Conforme supra-referido, são os dois principais padrões de indústria para a comunicação e integração com dispositivos de segurança baseados em IP, pelo que, para os objectivos deste relatório, utilizar-se-á um destes padrões com o objectivo de estabelecer a comunicação inicial com uma câmera de segurança, a fim de obter o identificador único de recursos (URI) para a transmissão de imagens em directo da câmera de segurança e controlos adicionais para rotação, inclinação e ampliação.

2.6 Real time streaming protocol (RTSP)

O Real Time Streaming Protocol, ou RTSP, é um protocolo a nível de aplicação que permite o controlo da entrega de dados com propriedades em tempo real. O RTSP proporciona uma estrutura extensível para permitir a transmissão controlada e a medida de dados em tempo real, tais como áudio e vídeo. As fontes de dados podem incluir tanto transmissões de dados em directo quanto ficheiros armazenados. Este protocolo destina-se a controlar múltiplas sessões de transmissão de dados, providenciando um meio de escolher a transmissão seja por UDP, UDP multicast, TCP, e fornecer um meio para escolher mecanismos de entrega baseados no RTP. (Rao et al., 1998)

2.6.1 Real time streaming protocol em sistemas de CCTV

O RTSP desempenha um papel crucial nos sistemas modernos de CCTV, particularmente naqueles onde são utilizadas câmeras de vigilância baseadas em IP. Pois a transmissão das imagens de vídeo em directo ocorre através do RTSP. O RTSP possibilita a transmissão controlada de dados em tempo real, incluindo fluxos de áudio e vídeo. Isto é especialmente pertinente no contexto dos sistemas de CCTV, onde a transmissão atempada e eficiente de videovigilância é fundamental.

Para além disso, o RTSP permite a gestão de sessões, possibilitando que vários utilizadores tenham acesso e visualizem a mesma transmissão de vídeo em simultâneo. Este recurso é fundamental em situações em que vários operadores precisam monitorar a mesma área, como em grandes espaços públicos ou instalações de infra-estrutura crítica.

2.7 Transmissão de vídeo em directo em plataformas modernas

Nesta secção, apresentar-se-ão os protocolos comumente utilizados para a transmissão de conteúdos multimédia em plataformas modernas, como Web, Android e iOS.

2.7.1 HTTP Live Streaming - HLS

O HLS é um protocolo de transmissão de multimédia desenvolvido a partir do protocolo de comunicação HTTP. Este protocolo é da iniciativa da Apple Inc. e foi implementado nas plataformas de produtos da Apple Inc., como o sistema operativo iOS do iPod e do iPhone. Este protocolo de transmissão em fluxo contínuo aplica-se principalmente à transmissão em tempo real ou à transmissão de multimédia pré-codificada. Ao contrário do tradicional fluxo de dados multimédia HTTP, o HLS separa o ficheiro multimédia numa série de segmentos multimédia, em que cada segmento multimédia faz parte do ficheiro multimédia originalmente completo. Quando o cliente solicita a transmissão de ficheiros multimédia, o reprodutor tem a possibilidade de escolher a taxa de bits de transmissão adequada de acordo com as condições de largura de banda da rede. Além disso, o HLS fornece a técnica de encriptação de pacotes de rede AES-128, pelo que os fornecedores de informação multimédia podem proteger eficazmente as suas redes. (Chin-Feng Lai et al., 2013)

2.7.1.1 Plataformas suportadas:

- iOS (dispositivos Apple)
- Android (com leitores de terceiros)
- Navegadores Web

2.7.2 Web Real Time Communication - WebRTC

A WebRTC (Web Real-Time Communication) consiste numa nova tecnologia Web que permite que navegadores e aplicações móveis tenham funcionalidades como chamadas de áudio/vídeo, conversação, e partilha de ficheiros P2P (peer-to-peer), não necessitando de software ou plugins adicionais de terceiros. Publicada como tecnologia de código aberto pela Google em Maio de 2011, inclui componentes fundamentais para a comunicação em tempo real na Web. (Sredojev et al., 2015)
Os principais componentes da API WebRTC são:

- **MediaStream**
    - Possibilita que um navegador Web aceda à câmera e ao microfone.
- **RTCPeerConnection**
    - Estabelece chamadas de áudio ou vídeo.
- **RTCDataChannel**
    - Permite aos navegadores enviar dados através de ligações peer-to-peer.

2.7.2.1 Plataformas suportadas

- Navegadores da Web modernos incluindo Chrome, Firefox e Edge.

2.7.3 Dynamic adaptive streaming over HTTP - DASH

O MPEG-DASH, acrónimo para "Dynamic Adaptive Streaming over HTTP", é um método de transmissão semelhante ao HLS (HTTP Live Streaming). O seu funcionamento consiste na decomposição dos vídeos em partes mais pequenas e na sua codificação em vários níveis de qualidade. Isto permite um fluxo contínuo em diferentes níveis de qualidade, permitindo aos utilizadores alternar entre diferentes níveis durante a transmissão de um vídeo. Uma vez que se baseia em HTTP, qualquer servidor de origem pode ser configurado para servir fluxos MPEG-DASH.(*What Is MPEG-DASH?*, n.d.)

2.7.3.1 Plataformas suportadas:

- Navegadores Web.
- Android.
- iOS (com reprodutores de terceiros).

2.8 Comparação de "*tech stacks*" para desenvolvimento de software

Nesta secção pretende-se realizar a comparação das diferentes "*tech stacks*", para o desenvolvimento de aplicações "full stack". Existem actualmente várias tecnologias de apoio no desenvolvimento de software, porém, iremos abordar apenas duas das mais comuns nomeadamente, MERN e LAMP.

2.8.1 MERN – MongoDB, ExpressJS, React, NodeJS

Actualmente uma das escolhas mais populares para o desenvolvimento de softwares com base nas linguagens JavaScript e TypeScript, o MEAN é um pacote de tecnologias de código aberto que inclui o MongoDB, o ExpressJS, a livraria React e o ambiente de execução NodeJS.

2.8.1.1 MongoDB

Uma base de dados No SQL que utiliza um formato JSON (JavaScript Object Notation) para armazenar dados. A sua perfeita compatibilidade com outras partes da stack baseadas em JavaScript asseguram uma excelente performance e celeridade. O MongoDB não possui esquemas, o que o torna altamente flexível, e é construído sobre uma arquitectura de expansão horizontal que lhe permite lidar com grandes volumes de dados. (*MEAN and MERN Stacks*, 2022)

2.8.1.2 ExpressJS

É uma framework de aplicações Web de back-end que é executada a partir do Node.js. Em suma, trata-se de um conjunto de ferramentas destinadas a controlar o fluxo de trabalho entre o cliente e a base de dados, a fim de assegurar a transferência dos dados. O Express é utilizado para criar APIs, fazer a gestão de pedidos HTTP e apresentar um roteamento básico.(*MEAN and MERN Stacks*, 2022)

2.8.1.3 React

A camada superior do MERN é o React.js, uma framework JavaScript declarativa para criar aplicações dinâmicas do lado do cliente com HTML. O React permite criar interfaces complexas por meio de componentes simples, conectá-los a dados no servidor back-end e renderizá-los como HTML.(*What Is The MERN Stack?*, 2023)

2.8.1.4 NodeJS

consiste num ambiente de execução JavaScript de back-end que permite que o código JavaScript seja executado fora do navegador. Na sequência da sua criação, tornou-se finalmente possível programar código JavaScript do lado do servidor, permitindo um ciclo de desenvolvimento completo utilizando apenas JavaScript. (*MEAN and MERN Stacks*, 2022)

2.8.1.5 Vantagens

- Linguagem de programação uniforme, maior eficiência operacional com menos recursos.
- Reutilização abrangente de código.
- Elevado desempenho e velocidade.
- Conjunto de ferramentas gratuitas e de código fonte aberto.

2.8.1.6 Desvantagens

- Insuficiência em termos de processamento intensivo no “*back-end*”

2.8.2 LAMP – Linux, Apache, MySQL, PHP

O LAMP stack é um conjunto de quatro tecnologias de software diferentes utilizadas por desenvolvedores para criar sites e aplicações Web. LAMP é um acrónimo para o sistema operativo, Linux; o servidor Web, Apache; o servidor de bases de dados, MySQL; e a linguagem de programação, PHP. (*What Is a LAMP Stack?*, 2023)

2.8.2.1 Linux

O Linux é um sistema operacional de código aberto que pode ser instalado e configurado para atender a diversas necessidades de aplicação. Posicionado no primeiro nível da pilha LAMP, o Linux oferece suporte a outros componentes nas camadas superiores. (*What Is a LAMP Stack?*, 2023)

2.8.2.2 Apache

O servidor web Apache, um software de código aberto, é parte integrante da pilha LAMP. Actuando como intermediário entre os arquivos do site e um navegador, o módulo Apache facilita a troca de informações usando o protocolo HTTP. (*What Is a LAMP Stack?*, 2023)

2.8.2.3 MySQL

O MySQL consiste num sistema de gestão de bases de dados relacionais de código aberto e constitui a terceira camada da pilha da LAMP. No modelo LAMP, o MySQL é utilizado para armazenar, gerir e consultar informações em bases de dados relacionais. (*What Is a LAMP Stack?*, 2023)

2.8.2.4 PHP

O PHP, cuja sigla significa PHP: Hypertext Preprocessor (Pré-processador de hipertexto), é a quarta e última camada para o LAMP stack. É uma linguagem de scripts que permite que os websites executem processos dinâmicos. Um processo dinâmico consiste em informações num software que mudam constantemente. Os

desenvolvedores Web incorporam a linguagem de programação PHP em HTML para exibir informações actualizadas ou em tempo real em websites. Utilizam PHP para permitir que o servidor Web, a base de dados e o sistema operativo processem de forma coesa os pedidos dos navegadores.(*What Is a LAMP Stack?*, 2023)

2.8.2.5 Vantagens

- Baixo custo de implementação.
- Comunidade de desenvolvimento extensa, o que implica facilidade e acesso ao suporte técnico.
- Flexibilidade e fiabilidade

2.8.2.6 Desvantagens

- Menos flexível em relação a outras stacks que não usem bases de dados relacionais
- Requer um conhecimento aprofundado na configuração e implementação dos serviços.

3 Capítulo III - Caso de estudo

Na próxima secção pretende-se fazer a apresentação da instituição de estágio, e a descrição das actividades exercidas durante o período de estágio profissional, incluindo uma descrição detalhada das actividades diárias (rotineiras) e a participação em diversos projectos.

O estagiário foi integrado ao departamento de informática da instituição de estágio, em que desempenhou um papel duplo ao actuar simultaneamente como técnico de informática (TI) e engenheiro de software.

3.1 Apresentação da instituição de estágio

A ALTEL soluções globais de comunicação, anteriormente designada por ALCATEL Moçambique, foi fundada em 2003 para atender à crescente demanda por soluções de comunicação de voz e dados em ambientes corporativos. Em 2014 a ALTEL tornou-se parte do Grupo Meridian 32 quando teve 98% do seu capital social adquirido por este, a ALTEL expandiu a sua oferta tecnológica, e consolidou-se como uma das empresas líder no sector das tecnologias de informação em Moçambique.

Com uma ampla experiência no mercado, a ALTEL conta com diversos clientes incluindo, instituições governamentais, empresas privadas, instituições financeiras, indústrias e instituições no sector de saúde. Reconhecida pela sua qualidade de serviço, inovação no fornecimento de tecnologia de ponta e compromisso com a satisfação do cliente, a ALTEL destaca-se como uma referência no mercado tecnológico moçambicano, com mais de 800 clientes satisfeitos.

3.1.1 Missão

A ALTEL tem como a missão, ser líder de referência no mercado nacional na área das TICs, pela qualidade e inovação dos serviços e soluções que proporciona aos seus clientes, contribuindo para o seu crescimento sustentável.

3.1.2 Visão

Proporcionar aos seus clientes soluções ambiciosas e compatíveis com os seus requisitos, sendo reconhecida como um parceiro de confiança, capaz de acompanhar e promover a evolução das suas necessidades.

3.1.3 Princípios

Garantir um ambiente para a operacionalização eficaz e eficiente dos processos que permita aos Colaboradores o desenvolvimento das suas competências, a sua criatividade e a sua motivação para benefício comum;
Cumprir os requisitos legislativos, normativos e regulamentares aplicáveis;
Aperfeiçoar e manter continuamente o SGQ, sensibilizando, formando e envolvendo todos os Colaboradores e todas as partes envolvidas que se considere relevante;
Analisar e melhorar constantemente a eficácia e a eficiência do SGQ, com vista à satisfação dos Clientes e outras partes interessadas.

3.1.4 Valores

- Liderança.
- Ética e profissionalismo.
- Trabalho em equipa.
- Competência.
- Comprometimento.

3.2 Serviços prestados pela ALTEL

A ALTEL oferece os seguintes serviços na área de tecnologias de informação.

Área	Serviços
Datacenters	Construção, operação e gestão de datacenters
Segurança	Protecção de endpoints, virtualização, redes e sistemas de prevenção de intrusão, controlo de acesso e vídeo vigilância (CCTV)
Comunicação	Voz através de IP (VoIP), videoconferência
Virtualização	Computação em nuvem, virtualização de desktops, recuperação em caso de desastres e consolidação de servidores
Rede	Redes sem fios (WIFI), roteamento de sistemas de cablagem estruturada, rádio transmissão VHF e UHF e links multicanal

3.3 Horário de trabalho

O período laboral designado pela empresa para os colaboradores administrativos é delineado a seguir:

- De segunda-feira a sexta-feira: entrada as 08:00 e saída das 17:00 com um período de 1h:30m de intervalo a partir das 12:30.

3.4 Estrutura da instituição

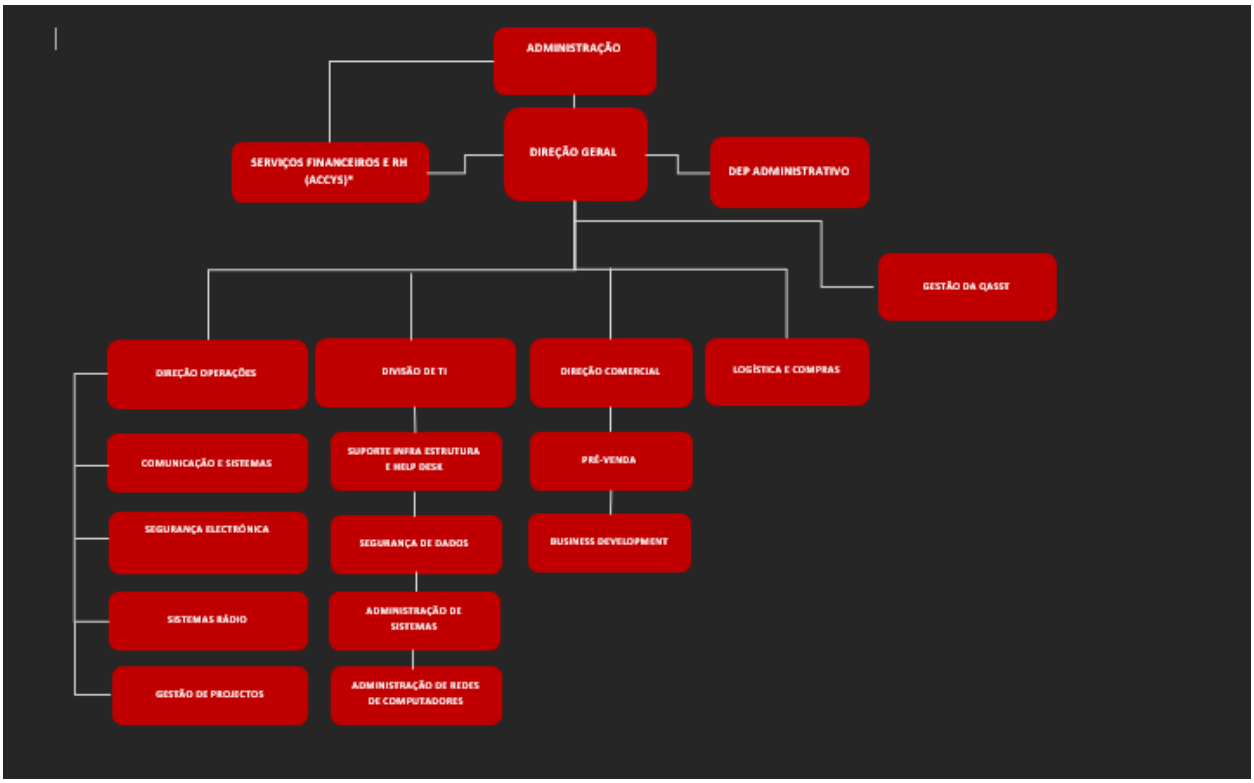

Figura 4: Estrutura da instituição de estágio (fonte: elaboração própria)

3.5 Descrição das actividades desempenhadas

Na presente secção serão apresentadas todas as actividades realizadas durante o estágio profissional, incluindo as tarefas diárias e os projectos participados. Numa primeira fase, foi necessário conhecer a estrutura e os procedimentos operacionais da instituição, tendo esta acção ocorrido nas duas primeiras semanas de estágio.

3.5.1 Actividades rotineiras

As actividades diárias eram principalmente compostas por tarefas relacionadas com a administração de sistemas e a segurança da rede.

3.5.1.1 Administração de firewall

Esta actividade consistia principalmente no monitoramento e gestão da firewall da instituição de estágio. A actividade era composta por tarefas como:

- Monitoramento de eventos de segurança.
- Criação de regras e políticas de acesso.
- Configuração e manutenção de links de acesso de provedores de serviços de internet.
- Administração e configuração de acessos remotos.
- Administração e configuração de serviços publicados.

3.5.1.2 Administração de sistemas

Consistiu na validação da operacionalidade dos servidores e dos serviços configurados nestes, bem como na criação e atribuição de acessos a utilizadores. Também foram executadas tarefas de criação e implementação de máquinas virtuais, gestão do sistema de vídeo vigilância e controlo de acesso.

3.5.2 Projectos participados

3.5.2.1 Implementação de um sistema de gestão documental

Este projecto consistiu na implementação de um sistema de gestão documental, na infra-estrutura tecnológica do grupo Meridian 32 ao qual faz parte a instituição de estágio. O sistema em questão tem como o objectivo a digitalização do fluxo de processamento de documentos nos vários departamentos de cada uma das empresas do grupo.

A função desempenhada neste projecto, esteve ligada a gestão do projecto, desde a planificação, recolha dos requisitos, escolha da solução e interacção com o fornecedor da solução escolhida. Foi também desempenhado o papel de implementador que esteve ligado a criação e configuração da máquina virtual em que o sistema seria instalado, e a implementação (deployment) do sistema de gestão documental.

3.5.2.2 Migração e actualização da infra-estrutura de segurança de rede e estações de trabalho

Este projecto consistiu na instalação e configuração de uma nova firewall Fortigate 100F na infra-estrutura do grupo, de modo a substituir a firewall Fortigate 100D em uso que se encontrava no fim do período de vida útil. E na instalação e configuração da solução de protecção de estações de trabalho e servidores, sophos XDR.

Para a implementação da nova firewall, as tarefas consistiram na replicação das configurações, políticas de acesso e serviços da Fortigate 100D para a nova firewall Fortigate 100F. Para a configuração da protecção a estações de trabalho e servidores, as tarefas consistiram na instalação do sophos XDR em 70 estações de trabalho e em 24 servidores e configuração dos serviços de protecção, scans e políticas de acesso a conteúdos na internet no servidor de gestão centralizada.

3.6 Situação actual

Nesta secção pretende-se apresentar o caso de estudo do projecto objecto do relatório, que se trata de uma empresa moçambicana do sector da logística e de transportes, esta organização em questão é cliente da ALTEL, e devido a compromissos de confidencialidade e protecção da privacidade, será designada pelo pseudónimo ILS.

A ILS manifestou recentemente o seu descontentamento com o actual sistema de CCTV em vigor, centrando-se especificamente nos desafios relacionados com o acesso em directo às imagens das câmeras de vigilância. A principal preocupação centra-se em questões de fiabilidade e acessibilidade, uma vez que o sistema actual impõe limitações severas ao acesso às imagens em directo. Actualmente, os utilizadores estão limitados a aceder às imagens em directo das câmeras de vigilância, uma de cada vez. Isto não só resulta em ineficiências operacionais, mas também levanta preocupações sobre a disponibilidade do sistema.

Uma complicação adicional surge do requisito de que o acesso em directo às imagens das câmeras só pode ser realizado através de um computador pessoal que utilize o agora obsoleto navegador Internet Explorer e o protocolo HTTP que é inerentemente inseguro. Isto não só limita a acessibilidade do utilizador, como também apresenta riscos de segurança numa época em que a segurança cibernética é de extrema importância.

A ILS também manifestou o seu desejo de permitir o acesso remoto a imagens de câmeras de vigilância em directo via Internet, através de dispositivos móveis. Isto decorre da necessidade de se adaptar às exigências em evolução da vigilância, incluindo a capacidade de aceder de forma segura e conveniente a imagens de câmeras em directo a partir de diferentes locais. Essa adaptação à tecnologia moderna e às expectativas dos usuários é crucial para atender às preocupações da ILS e garantir um sistema de vigilância mais eficiente e seguro.

3.7 Constrangimentos apresentados

São apresentados os seguintes constrangimentos:

No.	Constrangimento
1	O sistema de monitoramento não permite múltiplas sessões de utilizadores.
2	O sistema de monitoramento não permite o acesso simultâneo à transmissão de vídeo de uma câmera de vigilância.
3	O sistema não consta com uma versão para dispositivos móveis.
4	O sistema no seu estado actual não permite o acesso remoto a partir da internet a transmissão de vídeo das câmeras de vigilância
5	O sistema só é suportado pelo navegador web internet Explorer.

Tabela 3: Constrangimentos apresentados

4 Capítulo IV – Descrição da solução

De facto, não restam dúvidas de que os sistemas de gestão de vídeo apresentados podem satisfazer a maioria das necessidades apresentadas pela ILS, porém, devido às preocupações levantadas pela ILS, especialmente no que se refere a softwares de terceiros e questões de privacidade, surge então a necessidade de desenvolver um sistema personalizado de monitorização de CCTV. Consequentemente, torna-se necessário adoptar uma metodologia para o desenvolvimento deste sistema. A descrição da solução apresentada neste capítulo seguirá a metodologia de desenvolvimento em cascata, e as secções em seguida estarão estruturadas com base nas suas seis fases, nomeadamente: Recolha e análise de requisitos, concepção (design), desenvolvimento, testes, implementação e manutenção.

4.1 Recolha e análise de requisitos

4.1.1 Problemas actuais

- O Sistema de monitorização actual é obsoleto e se baseia em tecnologia descontinuada.
- O acesso ao sistema é limitado apenas a computadores pessoais com o navegador Internet Explorer.
- O sistema limita o número de sessões de utilizador e não permite a transmissão de imagens de vídeo em simultâneo.
- O sistema não permite o acesso remoto.
- Necessidade da monitorização multiplataforma , incluindo android e iOS.

4.1.2 Requisitos de software

Os requisitos de software são uma descrição minuciosa da funcionalidade, dos serviços e das restrições operacionais de um sistema. Estes podem incluir: Requisitos funcionais, Requisitos não funcionais, Serviços fornecidos, e Restrições operacionais. (Pal, 2018)

4.1.3 Requisitos funcionais

Os requisitos funcionais de um software referem-se às funções ou características consideradas essenciais para o funcionamento normal do sistema. Estes requisitos são especificações claras e detalhadas das funcionalidades que o software deve oferecer. Incluem a descrição de todas as acções e processos que o software deve ser capaz de realizar, abarcando tanto as interacções do utilizador com o sistema como as operações internas necessárias para processar dados e produzir resultados. (Bigelow, 2020)

Código	**Requisito funcional**
RF01	O sistema só deverá permitir o acesso a utilizadores autenticados e devidamente autorizados
RF02	O sistema deve permitir a visualização das imagens ao vivo das câmeras de CCTV
RF03	A versão do sistema para dispositivos móveis deve permitir o acesso remoto as imagens ao vivo das câmeras de CCTV a partir da internet
RF04	O sistema deve ser capaz de suportar múltiplas sessões de utilizador
RF05	O sistema deve permitir a transmissão em simultâneo das câmeras de CCTV
RF06	O sistema deve permitir uma gestão integrada dos utilizadores, incluindo funções como a adição, actualização e exclusão
RF07	O sistema deve permitir uma gestão integrada das câmeras de segurança, incluindo funções como a adição, actualização e exclusão
RF08	O sistema deve permitir uma gestão integrada das localizações geográficas, incluindo funções como a adição, actualização e exclusão
RF09	O sistema deve ser compatível com as tecnologias modernas como navegadores e sistemas operativos móveis actuais

Tabela 4: Requisitos funcionais (fonte: elaboração própria)

4.1.4 Requisitos não funcionais

Os requisitos não funcionais definem a eficácia, usabilidade e experiência do utilizador (UX) de um produto de software; normalmente, não afectam a funcionalidade subjacente do sistema. Na maioria dos casos, o software pode desempenhar a sua função pretendida, mesmo quando não cumpre com os seus requisitos não funcionais. (Bigelow, 2020)

Código	**Requisito não funcional**
RNF01	A versão móvel do sistema deve permitir se possível a autenticação por biometria
RNF02	O sistema deve apresentar uma interface simples, fácil e intuitiva de usar.
RNF03	O sistema deve garantir a autenticação e autorização por privilégios, permitindo apenas que utilizadores autorizados executem acções
RNF04	O sistema deve apresentar o estado das câmeras de vigilância (operacional, não operacional)

Tabela 5: Requisitos não funcionais (fonte: elaboração própria)

4.2 Concepção do sistema (design)

O sistema proposto é composto por três (3) componentes principais, nomeadamente, uma interface de programação de aplicações (API), que abrange toda a camada lógica e de negócio. É nesta camada onde realiza-se a gestão de activos como utilizadores, localizações e câmeras. Nesta camada também são realizadas operações chave como a autenticação e autorização. Outro componente é a interface de front-end que inclui as aplicações móveis e web. Neste componente realiza-se a apresentação dos dados fornecidos pela API e a interacção com o utilizador, o front-end permite a visualização da transmissão em directo das câmeras de vigilância, bem como acessos a painéis de gestão e configuração dos vários activos e componentes do sistema. Por fim encontramos o servidor de media, este é responsável pela transcodificação da transmissão em directo das câmeras de vigilância que usam o protocolo RTSP, para os protocolos suportados pelas plataformas web, android e iOS.

4.2.1 Ferramentas utilizadas para o desenvolvimento do sistema (prototipagem)

4.2.1.1 Linguagem de modelação

Como linguagem de modelação optou-se pela Linguagem de Modelação Unificada (UML, do inglês Unified Modeling Language) que constitui uma linguagem gráfica destinada à visualização, especificação, construção e documentação dos artefactos de um sistema intensivo de software. A UML oferece um método padrão para a elaboração dos projectos de um sistema, incluindo tanto elementos conceptuais, como processos de negócios e funções do sistema, quanto elementos concretos, abrangendo instruções de linguagens de programação, esquemas de base de dados e componentes de software reutilizáveis. Esta linguagem surge como um instrumento fundamental na engenharia de software, facilitando a comunicação entre os profissionais da área e a materialização de visões técnicas em implementações efectivas. (Sparx Systems, 2023)

4.2.1.2 Ferramenta de diagramação

No contexto do desenvolvimento do projecto, a escolha recaiu sobre a ferramenta LucidChart para a realização de tarefas de modelação. Esta decisão foi fundamentada na capacidade abrangente do LucidChart de suportar a criação de todos os tipos de diagramas UML, oferecendo uma interface de utilizador intuitiva e recursos avançados de colaboração.

4.2.1.3 Ferramenta de desenho de interfaces e prototipagem (Figma)

A selecção do Figma como ferramenta para o desenho das interfaces front-end das aplicações web e móveis baseia-se nas suas capacidades como editor gráfico de vector e plataforma de prototipagem para projectos de design. Caracterizando-se principalmente pelo seu funcionamento baseado em navegador web, o Figma facilita a colaboração e a partilha de projectos de design em tempo real.

4.2.2 Descrição dos actores do sistema

Um actor de sistema é um meio que interage com outros sistemas num ambiente. Pode ser um indivíduo, uma organização ou um sistema externo que interage com uma aplicação ou sistema. Os actores são objectos externos que produzem ou consomem dados. (AKKA, 2013)

O sistema conta com os seguintes actores:

- **Administrador**: O administrador é responsável pela configuração do sistema e gestão dos activos (utilizadores, localizações e câmeras de vigilância)
- **Operador**: O operador tem a capacidade de aceder ao sistema e visualizar imagens em directo das câmeras de vigilância. (Akka, 2023)

4.2.3 Diagrama de casos de uso

Um diagrama de casos de uso é uma representação gráfica das interacções entre um sistema e os seus utilizadores. Trata-se de um tipo de diagrama comportamental que modela a funcionalidade de um sistema. O principal propósito de um diagrama de casos de uso é capturar os requisitos funcionais de um sistema. (Mahr, 2022) Os diagramas de casos de uso descrevem as funções de alto nível e o âmbito de um sistema. Identificam as interacções entre o sistema e os seus actores. Os casos de uso e actores nos diagramas de casos de uso descrevem o que o sistema faz e como os actores o utilizam.(Waykar, 2015) Os diagramas de casos de uso são utilizados em projectos grandes e complexos para que os desenvolvedores possam compreender facilmente os requisitos do sistema. Eles também actuam como um dispositivo de comunicação entre os diferentes interessados no projecto. (IBM Corporation, 2023)

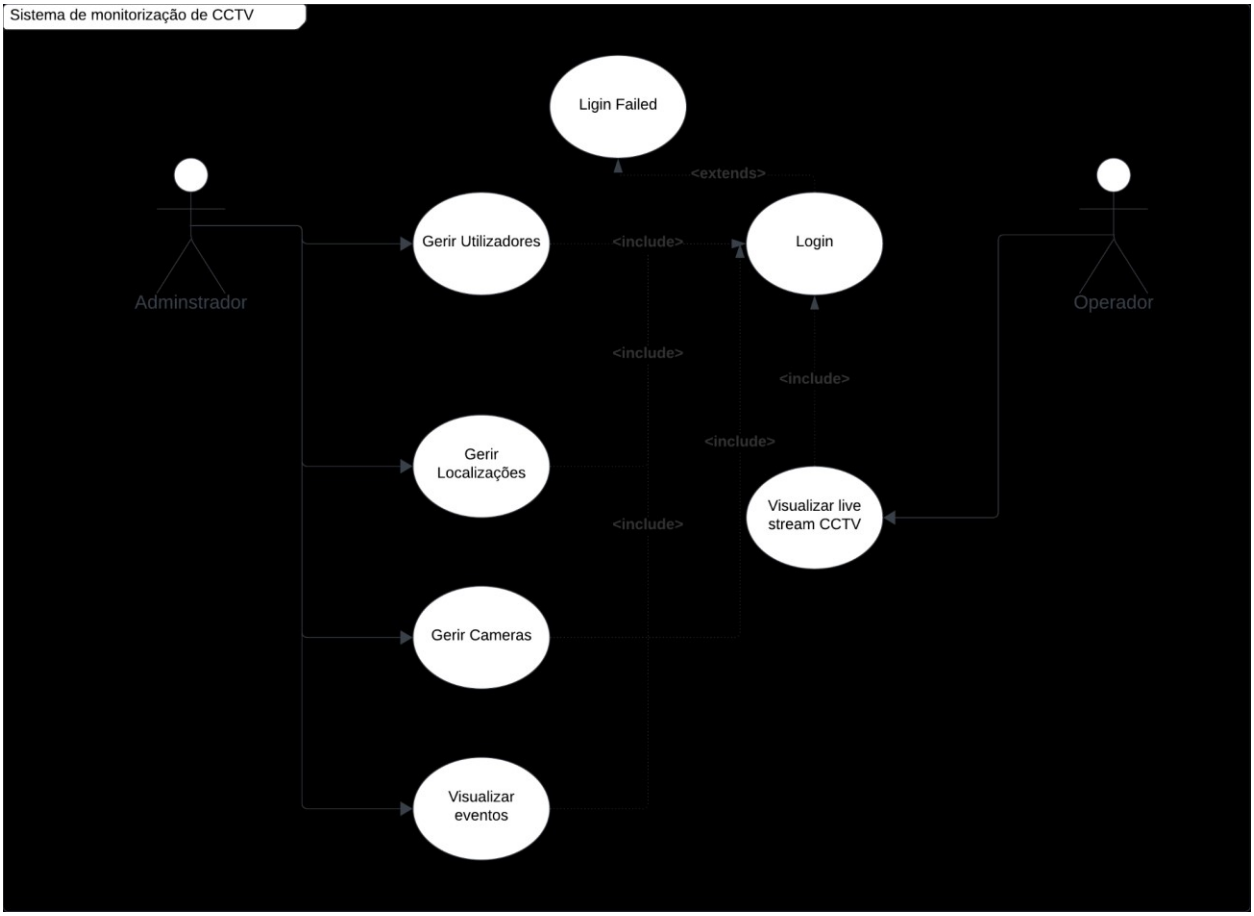

Figura 5: Diagrama de casos de uso (compacto, referir ao anexo 1 para a versão extensa)

4.2.4 Diagrama de classes

Um diagrama de classes, no contexto do desenvolvimento e desenho de software, é um tipo de diagrama de estrutura estática que descreve a estrutura de um sistema ao apresentar as classes do sistema, os seus atributos, operações (ou métodos) e as relações entre objectos. O diagrama de classes é uma parte fundamental da Linguagem de Modelagem Unificada (UML), que é uma linguagem padrão para modelar a estrutura e o comportamento de sistemas de software.(Boustedt, 2010)

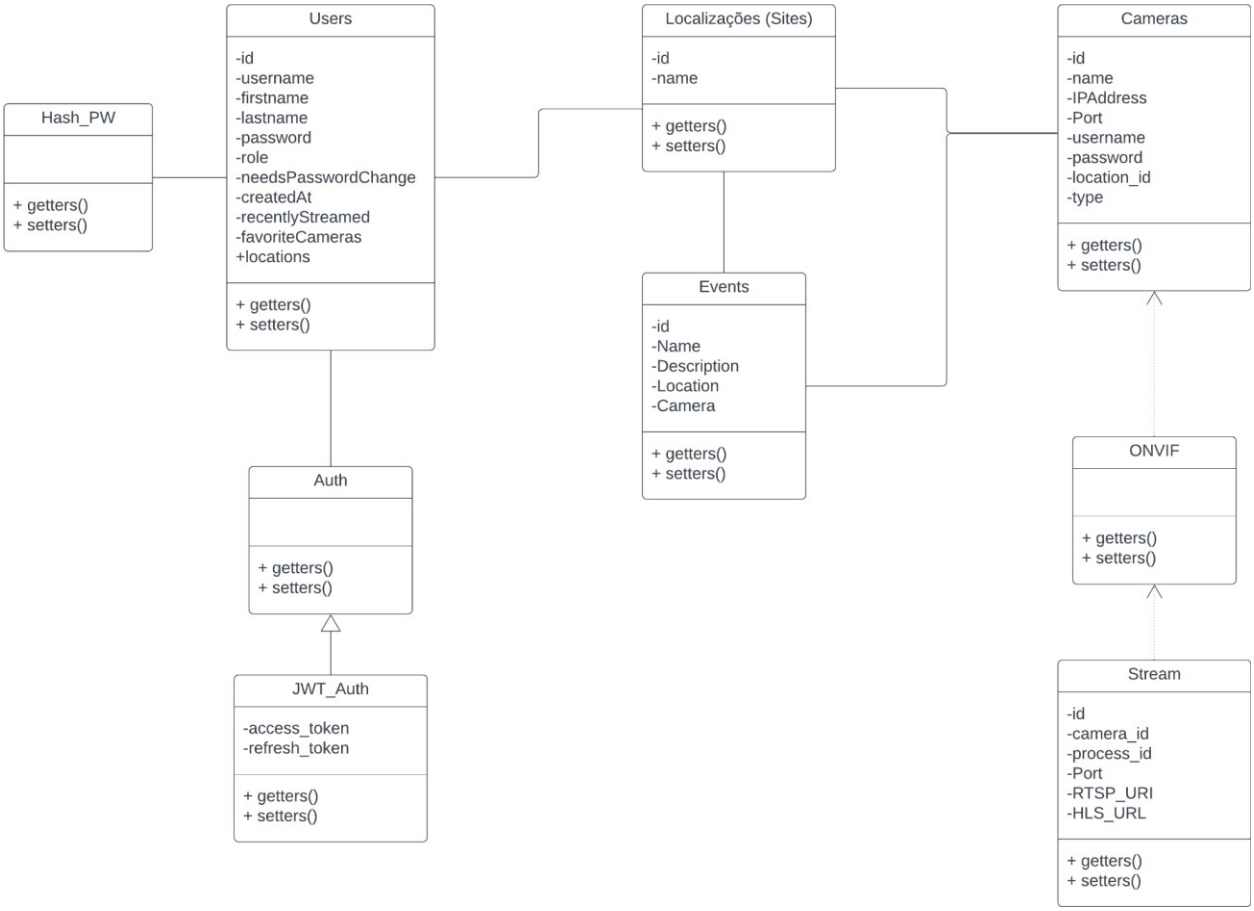

Figura 6: Diagrama de classes compacto (referir ao anexo 2 para a versão extensa)

4.2.5 Arquitectura do sistema

A proposta de solução é baseada na arquitectura de três (3) camadas e com o modelo cliente-servidor, que segundo (Natesan, 2019) é um modelo de computação em que servidores centrais fornecem e estruturam recursos e serviços, que são depois acedidos e utilizados por computadores clientes. Este modelo envolve frequentemente um ou vários clientes ligados a um servidor central através de uma ligação à Internet ou a uma rede de computadores.

Na arquitectura de três (3) camadas o sistema é composto por três (3) camadas distintas, apresentação, lógica e dados, cada uma destas camadas é normalmente instalada num

servidor independente, responsável por fornecer os serviços específicos relativos à sua função. Porém, para o presente caso, serão instaladas numa única máquina virtual e cada camada será implantada recorrendo a um contentor Docker.

4.2.5.1 Camada de apresentação

A camada de apresentação, que serve de interface com o utilizador e de camada de comunicação da aplicação, é o local onde o utilizador final interage com a mesma. O seu propósito principal é mostrar e recolher informações do utilizador. Esta camada superior pode operar através de um navegador da web, como uma aplicação para ambiente de trabalho (desktop), ou numa interface gráfica de utilizador (GUI). Normalmente, as camadas de apresentação em ambiente web são desenvolvidas com base em HTML, CSS e JavaScript, enquanto os aplicativos para ambiente de trabalho podem ser escritos em várias linguagens, dependendo da plataforma específica. (*IBM*, 2023.)

4.2.5.2 Camada lógica

A camada lógica também conhecida como camada de aplicação, constitui o núcleo do sistema. É nesta camada onde os dados recolhidos na camada de apresentação são processados, muitas vezes em combinação com outros dados provenientes da camada de dados. Este processamento é realizado seguindo a lógica de negócios, que se baseia num conjunto definido de regras empresariais. Além disso, a camada lógica tem a capacidade de adicionar, remover ou alterar os dados na camada de dados.
Em termos de desenvolvimento, a camada lógica é comumente desenvolvida com base em linguagens de programação como Python, Java, Ruby ou PHP. A comunicação com a camada de dados é efectuada através de chamadas de API.(*IBM*, 2023.)

4.2.5.3 Camada de dados

A camada de dados, também conhecida como camada de base de dados, é a camada onde todas as informações processadas pela camada lógica são armazenadas e geridas. Esta camada pode ser constituída por um sistema de gestão de bases de dados

relacional, com PostgreSQL, MySQL, MariaDB ou alternativamente um servidor de base de dados NoSQL, tal como Cassandra, CouchDB ou MongoDB.

Num sistema que usa a arquitectura de três camadas, toda a comunicação é intermediada pela camada lógica. As camadas de apresentação e dados não tem a capacidade de comunicação directa entre si, assegurando que todas as interacções e trocas de dados sejam processadas e geridas de forma centralizada pela camada lógica.(*IBM*, 2023)

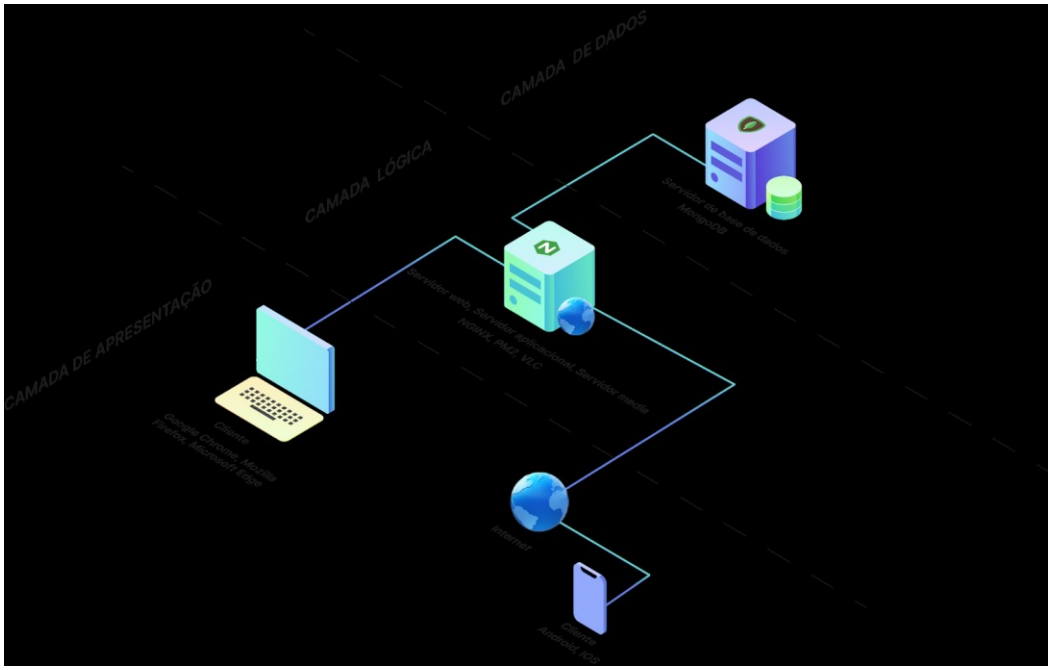

Figura 7: Arquitectura do sistema (fonte: elaboração própria)

4.3 Desenvolvimento do sistema

4.3.1 Ferramentas de codificação

Entende-se por ferramentas de codificação os instrumentos de auxílio aos programadores na manutenção ou desenvolvimento de aplicações. Tais instrumentos podem variar entre frameworks, bibliotecas, editores de código, IDEs, sistemas e plataformas.

4.3.1.1 Linguagem de programação

Na construção do sistema, incluindo os componentes de back-end e front-end, optou-se pela utilização da linguagem de programação TypeScript. Segundo (Microsoft, 2023), TypeScript é uma linguagem de programação orientada a objectos que se distingue pelo seu sistema rigoroso na definição e verificação de tipos de dados. Esta linguagem fundamenta-se no JavaScript, ampliando as suas capacidades com ferramentas mais sofisticadas para programação em diversos contextos e escalas. A escolha desta linguagem para o desenvolvimento do sistema é baseada na capacidade de combinar a flexibilidade do JavaScript com um maior controlo sobre a estrutura do código, oferecendo assim benefícios significativos em termos de segurança, manutenção e clareza do código.

4.3.1.2 Controlo de versões

Conforme descrito por (Chacon & Straub, 2014, 16), o controlo de versões é um sistema que regista as alterações efectuadas num ou mais ficheiros ao longo do tempo, permitindo o acesso a estados anteriores dos ficheiros para análise ou recuperação. No âmbito do desenvolvimento deste projecto, foram seleccionadas as ferramentas de controlo de versões reconhecidas pela sua eficiência, nomeadamente o Git (Command Line Interface - CLI), o GitHub Desktop e o GitHub.

O Git, através da sua interface de linha de comandos (CLI), proporciona uma gestão precisa do historial (histórico) de versões dos ficheiros, crucial para o tratamento das complexidades no desenvolvimento de software. Por seu lado, o GitHub Desktop oferece uma interface gráfica de utilizador, simplificando a gestão de versões para uma abordagem menos técnica. O GitHub, enquanto plataforma baseada na nuvem, actua como repositório central para o armazenamento e partilha de código.

4.3.1.3 Editor de código fonte

O editor de código fonte escolhido para o desenvolvimento do sistema foi o Microsoft Visual Studio Code. O Visual Studio Code é um editor de código-fonte simples, mas robusto o qual pode ser executado no ambiente de trabalho em Windows, macOS e Linux. Inclui suporte integrado para JavaScript, TypeScript e Node.js para além de um ecossistema diversificado de extensões para outras linguagens e ambientes de execução como C++, C#, Java, Python, PHP, Go, .NET. (*Documentation for Visual Studio Code*, 2023)

4.3.2 Base de dados

No que diz respeito à base de dados do sistema, a escolha incidiu sobre a MongoDB. Esta escolha deveu-se às características distintivas da MongoDB enquanto base de dados não relacional e orientada a documentos. Esta tecnologia destaca-se pelo suporte de armazenamento de dados num formato semelhante ao JSON (JavaScript object notation), proporcionando uma abordagem intuitiva e flexível à gestão de dados. O sistema MongoDB foi concebido com um modelo de dados altamente flexível, ideal para armazenar e manipular dados não estruturados. Este modelo permite uma gestão eficiente dos dados, apoiada por funcionalidades avançadas de indexação e replicação, e é acessível através de APIs ricas e intuitivas.

A decisão por trás da escolha da MongoDB é amplamente justificada pela sua facilidade de implementação e pelo suporte que oferece para um desenvolvimento flexível.

4.3.3 Servidor de backend (API)

Incorporando uma abordagem centrada na segurança, o sistema de monitorização CCTV integrará dois servidores API de backend distintos. A escolha desta estratégia decorre da necessidade de compartimentar as funcionalidades - uma adaptada à aplicação Web interna e outra dedicada à aplicação móvel. Conforme ilustrado na Figura 6, a primeira API actua como servidor privado (interno), acessível apenas dentro da rede local da

organização, garantindo uma maior segurança. Por outro lado, a segunda API funciona como servidor público (externo), destinando-se ao acesso através da Internet. Esta segregação permite um controlo mais granular das acções e capacidades do servidor público, mitigando potenciais riscos de segurança.

Abaixo as tabelas de distribuição das funcionalidades com base na exposição da API.

No.	Funcionalidade
1	Autenticação / Autorização
2	Gestão de utilizadores (CRUD)
3	Gestão de localizações (CRUD)
4	Gestão de câmeras (CRUD)
5	Visualização de eventos
6	Visualização das imagens em directo de CCTV

Tabela 6: Funcionalidades da API privada

No.	Funcionalidade
1	Autenticação / Autorização
2	Visualização de eventos
3	Visualização das imagens em directo de CCTV

Tabela 7: Funcionalidades da API pública

O servidor de backend é responsável pela ligação entre o repositório de dados (base de dados) e o front-end composto pelas interfaces de utilizador. O ciclo de vida dos dados é assegurado por operações CRUD - criação, recuperação, actualização e exclusão - facilitadas por endpoints dedicados da API que empregam os métodos HTTP correspondentes, designadamente "GET" para recuperação de dados, "POST" para publicação de dados, "PUT" para actualização de dados e "DELETE" para remoção de dados. A autenticação baseia-se nas credenciais do utilizador, designadamente um nome de utilizador e uma palavra-passe, sendo que, depois de estabelecida com êxito a sessão do utilizador, a autorização (acesso aos endpoints da API) realiza-se através de um Json Web Token (JWT) assinado e de duração limitada, o qual é emitido na sequência de uma autenticação bem-sucedida por parte do utilizador.

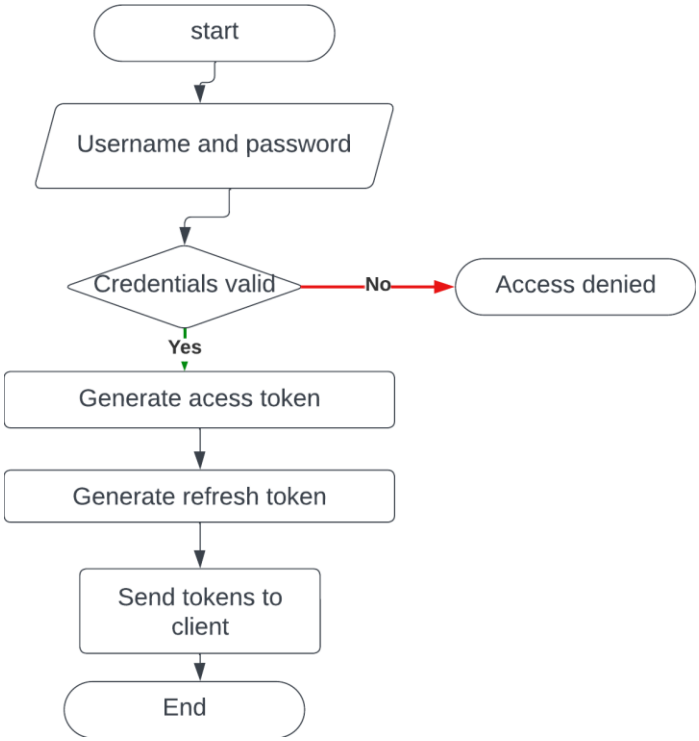

Figura 8: Fluxo de autenticação:

Figura 9:Endpoints da API

4.3.4 Servidor de media (Transmissão das imagens de vídeo das câmeras de vigilância)

Dado o facto de o protocolo RTSP não ser suportado nativamente pelos navegadores Web e pelos dispositivos móveis, surgiu a necessidade de implementar uma solução de transmissão multiplataforma. O protocolo de streaming escolhido foi o HLS, que é maioritariamente suportado pelos três sistemas-alvo (Web, Android e iOS). A razão da escolha do protocolo HLS em vez de qualquer outro protocolo de streaming prende-se com as restrições impostas aos dispositivos iOS, em que a única forma nativa de visualizar uma transmissão em directo é através do protocolo internamente utilizado, o HLS. Além disso, a opção de não implementar o RTSP, que é suportado nas plataformas alvo através de plug-ins e software de terceiros, resulta no facto de a transcodificação e a conversão do vídeo serem feitas pelo cliente, o que, por sua vez, pode levar a problemas de desempenho em dispositivos inferiores. Por conseguinte, a transmissão de imagens em directo das câmeras CCTV será feita a partir de um servidor multimédia separado, que receberá a transmissão RTSP da câmera especificada e, em seguida, utilizará o software de código aberto VLC com o apoio do software também de código

aberto ffmpeg para transcodificar a transmissão de vídeo RTSP para uma transmissão de vídeo HLS que, por sua vez, pode ser visualizada nas plataformas Web, Android e iOS.

As conversões no servidor de media são efectuadas a pedido, quando um novo pedido é efectuado ao endpoint da API do servidor. De igual modo, o servidor multimédia também será construído com base no ambiente de execução NodeJS e será composto por alguns endpoints API que podem ser chamados para iniciar ou parar a conversão de uma transmissão em directo, além disso, os dados relacionados com a conversão serão armazenados numa base de dados Redis.

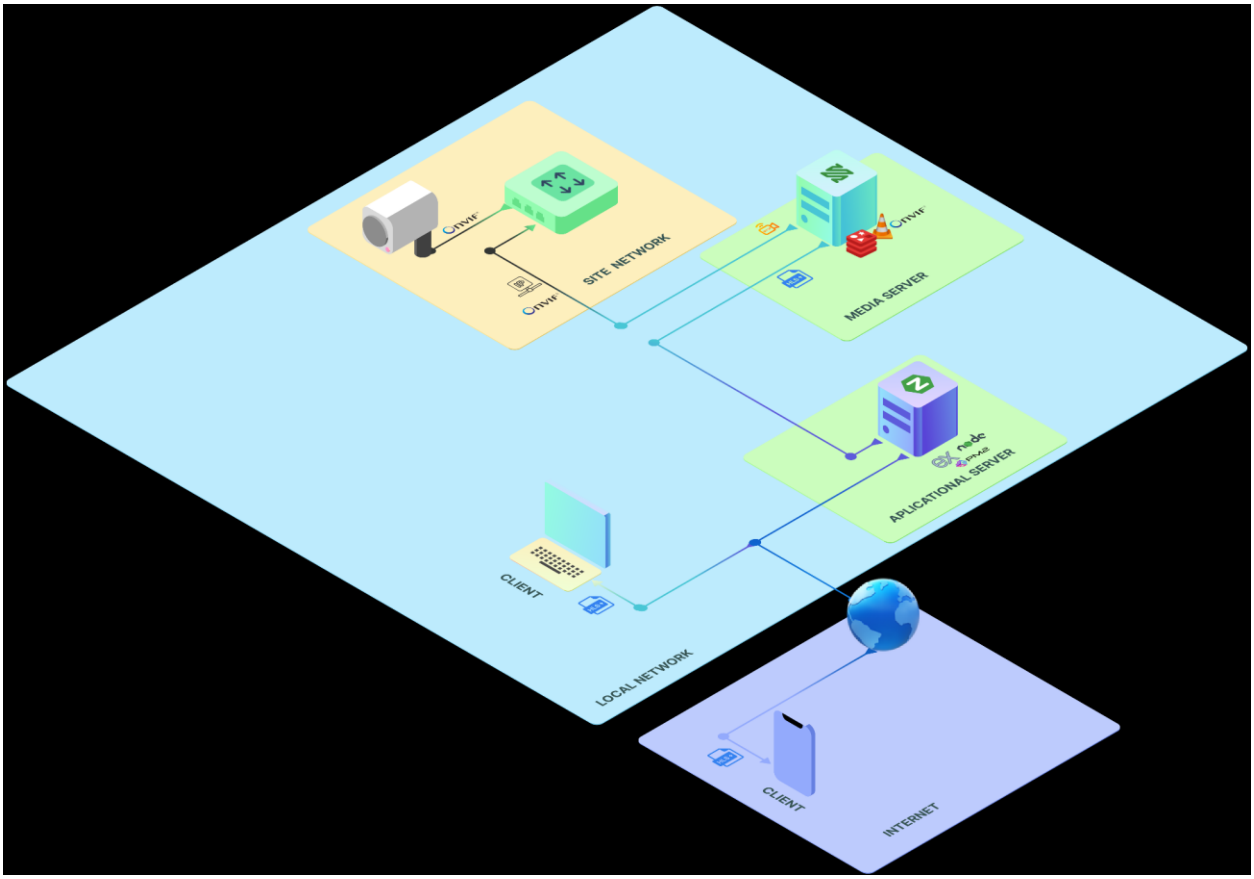

Figura 10: Arquitectura do servidor de media

4.3.5 Aplicações front-end

O sistema é composto por duas aplicações front-end. Em primeiro lugar, uma aplicação de página única baseada em Web que será construída com base na biblioteca de UI

React do JavaScript. Esta aplicação só estará disponível na rede local e o seu principal objectivo será fornecer uma interface ao utilizador na qual este possa configurar e gerir todas as funcionalidades relacionadas com o sistema de monitorização, permitirá também a transmissão em directo das câmeras de segurança e contará ainda com uma interface na qual servirá de centro para as transmissões em directo numa sala de controlo. A outra aplicação corresponderá à versão móvel do sistema de monitorização, que será também desenvolvida a partir de uma biblioteca UI JavaScript react-native. A versão móvel do sistema de monitorização será utilizada principalmente para acesso remoto às imagens em directo das câmeras. Esta versão do sistema não permitirá quaisquer funcionalidades de gestão ou configuração, uma vez que, por questões de segurança, essas funcionalidades não estarão disponíveis a partir da Internet.

As interfaces das aplicações front-end poderão ser encontradas no anexo 3 do presente relatório.

4.3.6 Ferramentas utilizadas no desenvolvimento do servidor de backend

4.3.6.1 Ambiente de execução (Runtime environment)

Para o ambiente de execução, a escolha recaiu sobre o NodeJS. O Node.js é um projecto de código aberto criado com base no Google Chrome JavaScript Engine, o qual funciona como uma plataforma robusta para o desenvolvimento de aplicações JavaScript do lado do servidor, o que permite que sejam executadas independentemente dos navegadores Web. (IBM, 2021)

4.3.6.2 Framework

Optou-se pela framework Express JS para o desenvolvimento da API. Express JS é uma framework de aplicações web, caracterizada por sua natureza minimalista e flexibilidade. Oferece um conjunto robusto de ferramentas para o desenvolvimento de aplicações móveis e web, facilitando a criação de soluções eficientes e escaláveis (OpenJS Foundation, 2017).

4.4 Testes de código

4.4.1 Testes unitários e de integração

Para testes automatizados, de integração e unitários, a escolha recaiu sobre a framework de testes JavaScript JEST, uma framework robusta e amplamente utilizada, para avaliar e validar meticulosamente a funcionalidade do código fonte. A decisão deveu-se à reputação da JEST em relação ao facto de fornecer um conjunto de testes abrangente bem como à sua excelente integração com vários projectos JavaScript.

4.4.2 Testes de API

A fim de testar os endpoints da API, a ferramenta escolhida para o efeito foi o Postman, uma plataforma destinada à construção, teste e utilização de API, que oferece um conjunto de ferramentas que simplificam todo o ciclo de vida de uma API.

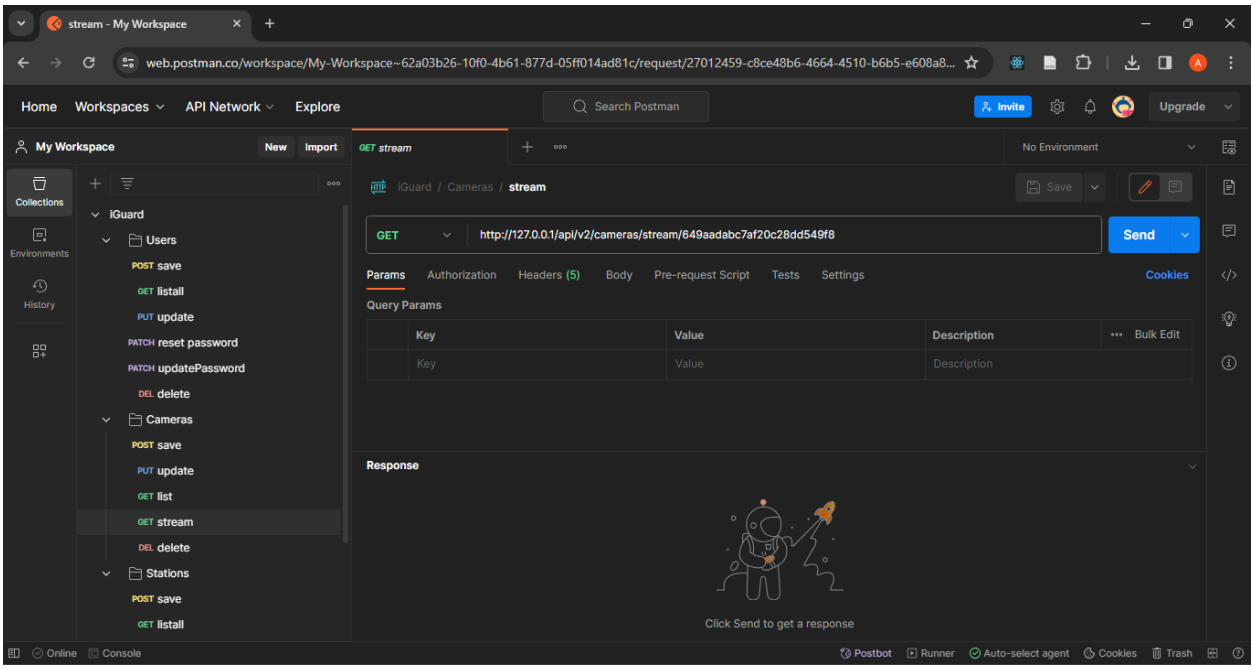

Figura 11: dashboard postman

4.5 Implementação (Deployment)

Tendo toda a lógica do sistema sido programada, avaliada e testada, o próximo passo crucial é a implementação do sistema. A fase de implementação envolve a configuração de cinco componentes-chave:

Aplicação Web (disponível apenas na rede local):
A aplicação Web será implementada numa máquina virtual Windows Server 2022 dedicada, utilizando o servidor Web IIS (Internet Information Services) da Microsoft.

API de backend privada (incluindo um servidor multimédia):
Este componente também será implantado em uma máquina virtual do Windows Server 2022, utilizando um contentor Docker e o gestor de processos pm2.

API de backend público (incluindo um servidor multimédia):
À semelhança do backend privado, a API de backend público e o servidor multimédia serão implementados numa máquina virtual Windows Server 2022 separada, utilizando um contentor Docker e o gestor de processos pm2. O servidor em questão estará na zona desmilitarizada (DMZ) da rede da organização

Aplicação móvel:
A aplicação móvel será instalada em dispositivos móveis nas plataformas Android e iOS.

Base de dados:
A base de dados, será implantada numa máquina virtual dedicada com o Windows Server 2022.

A arquitectura escolhida para a Implementação é a arquitectura multi-tier, em que cada componente reside no seu servidor específico. Estes servidores são máquinas virtuais, que executam o sistema operativo Windows Server 2022.

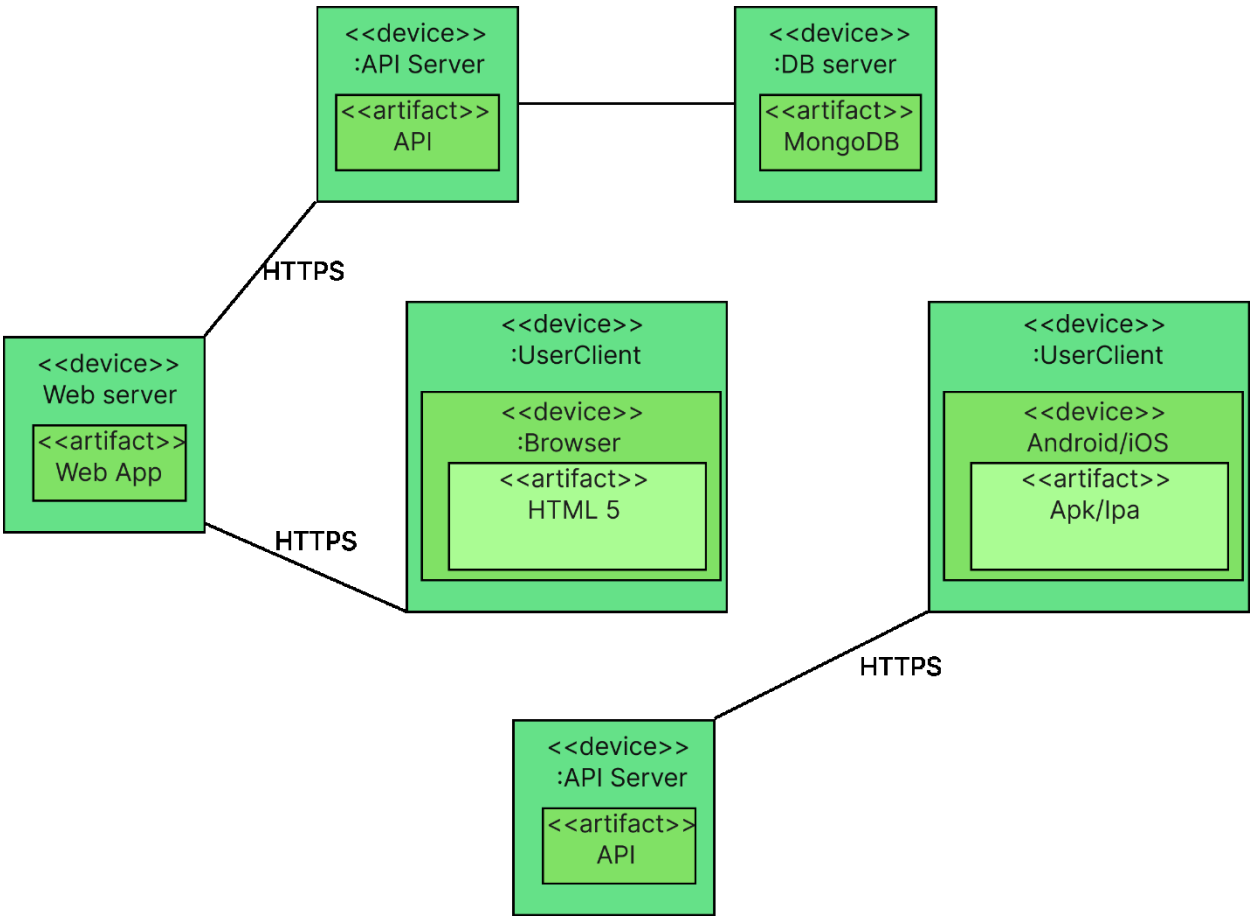

Figura 12: Diagrama de implementação

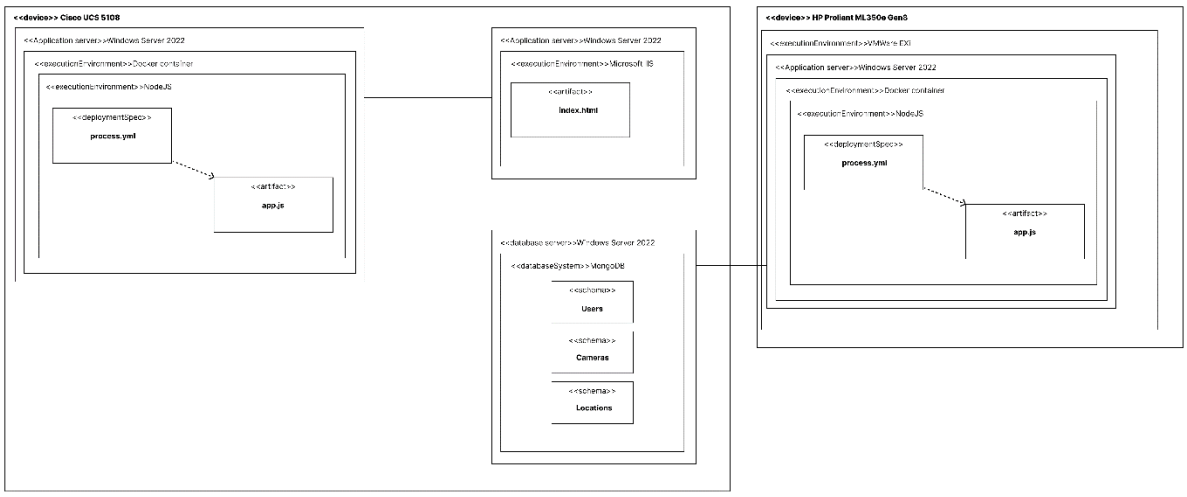

Figura 13: Diagrama de implementação específico

4.5.1 Implementação da aplicação Web:

A aplicação Web será executada numa máquina virtual Windows Server 2022, alojada pelo servidor Web IIS (Internet Information Services) da Microsoft. O IIS é um conjunto abrangente de servidores Web da Microsoft, que inclui um servidor Web, um servidor FTP, um servidor NNTP, entre outros (Rossberg, 2006, 307).

4.5.1.1 Configuração do IIS:

Em primeiro lugar, antes de proceder com a implementação da aplicação web, é necessário instalar e configurar a função Internet Information Services (IIS) no servidor. Para tal usamos a consola de gestão do servidor do Windows Server:

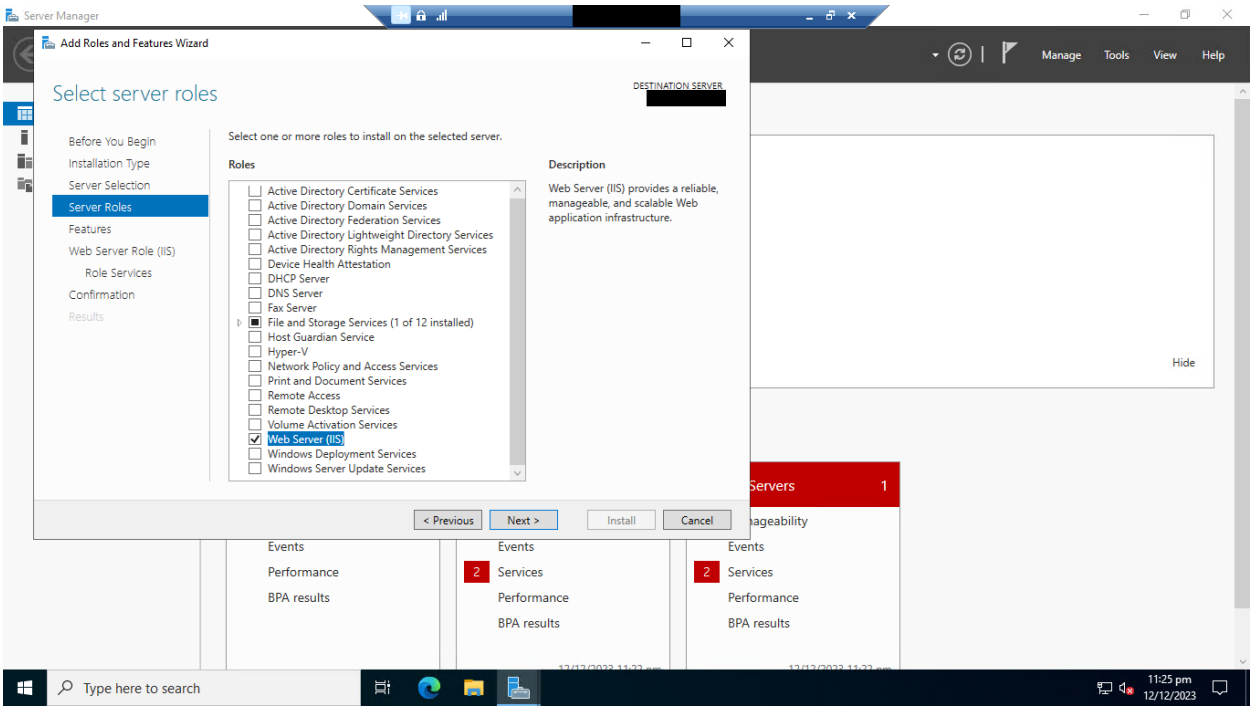

Figura 14: Instalação do Internet Information Services (IIS) (Fonte: Elaboração própria)

Após a instalação, procede-se à Implementação do website na consola do IIS. Porém, primeiro é necessário implementar um certificado HTTPS para o website, neste caso será utilizado um certificado auto assinado que será posteriormente implementado na autoridade de certificação da organização, entretanto, por enquanto, este será utilizado no estado em que se encontra.

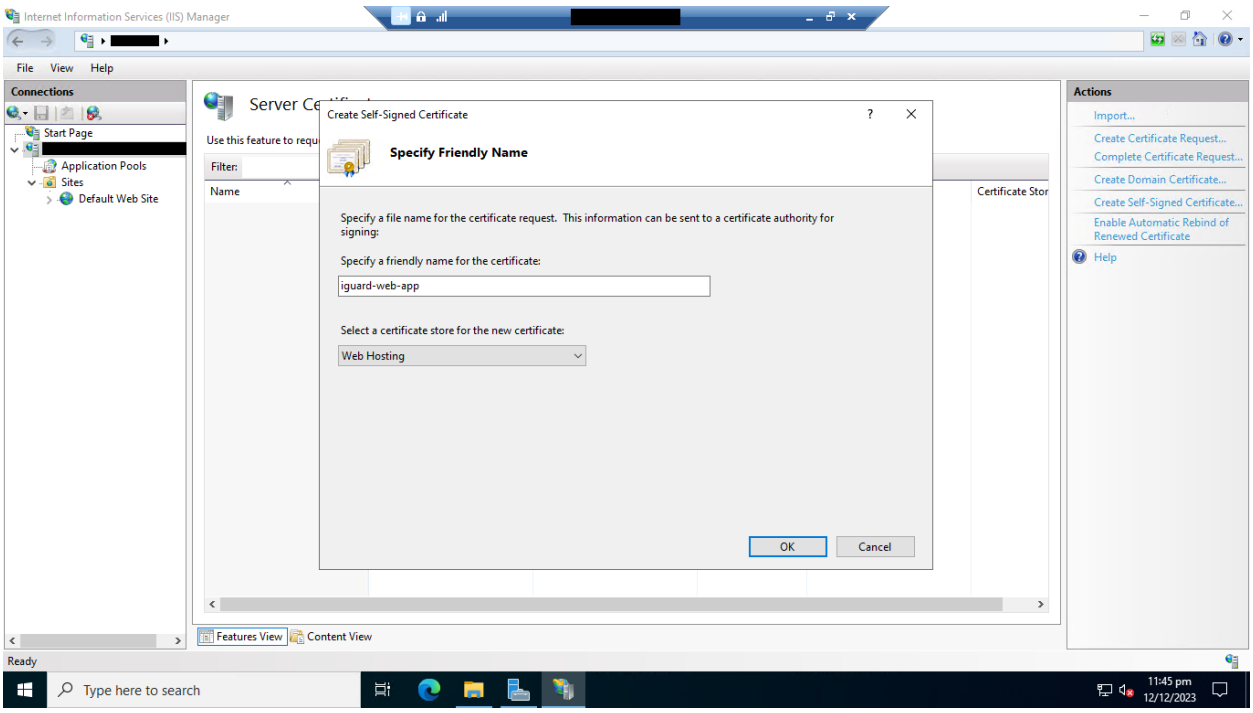

Figura 15: Criação de um certificado HTTPS auto assinado (Fonte: Elaboração própria)

Por fim, podemos implantar o site depois de compilá-lo usando a ferramenta de interface de linha de comando para aplicativos react, vite. Dessa forma, o código e os arquivos TypeScript serão comprimidos, e o código será ofuscado em produção.

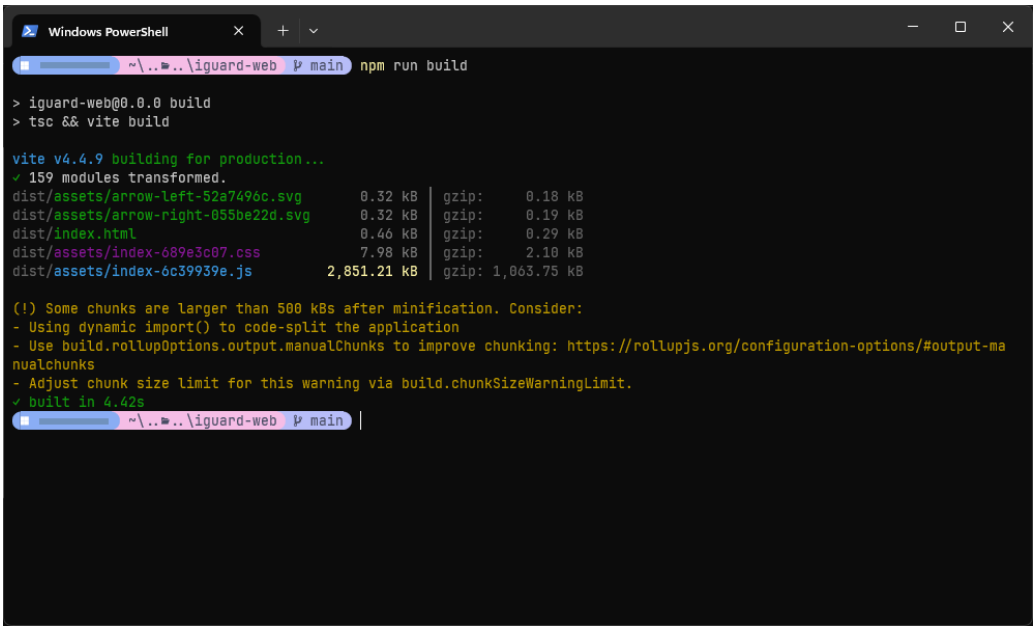

Figura 16: Compilação do website (Fonte: Elaboração própria)

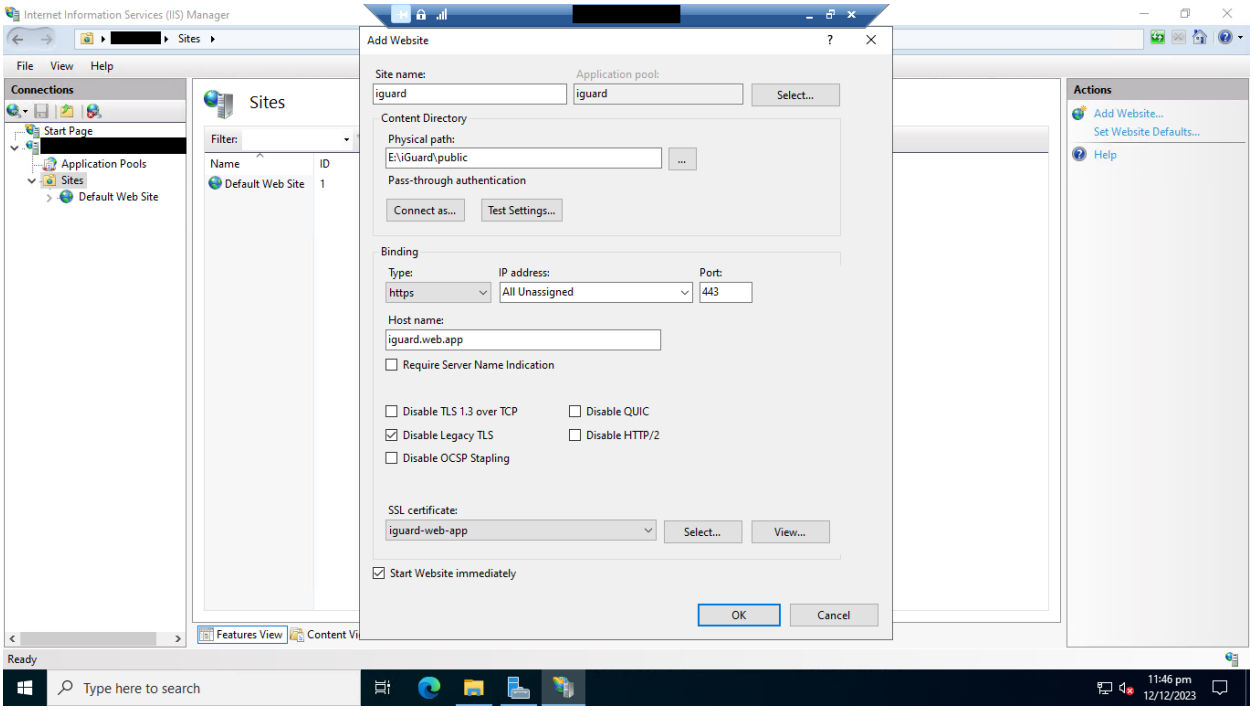

Figura 17: Configuração do website no IIS (Fonte Elaboração própria)

4.5.2 Implementação da API de backend (pública e privada)

A implementação da API backend seguirá o mesmo processo tanto para a API pública como para a API privada. A implementação seguirá a instalação e a configuração do software Docker Desktop, a fim de simplificar a configuração e a Implementação dos contentores Docker, que serão o principal ambiente de execução das API.

A escolha de utilizar contentores Docker para o ambiente de execução, apesar de as API estarem em servidores completamente diferentes, visa garantir o isolamento e uma superfície de risco limitada em caso de exposição.

4.5.2.1 Configuração do Docker

Antes de procedermos com a instalação é necessário realizar uma configuração na máquina virtual, pois por padrão a virtualização não é inicialmente permitida em máquinas virtuais. Para tal é necessário activar a virtualização aninhada.

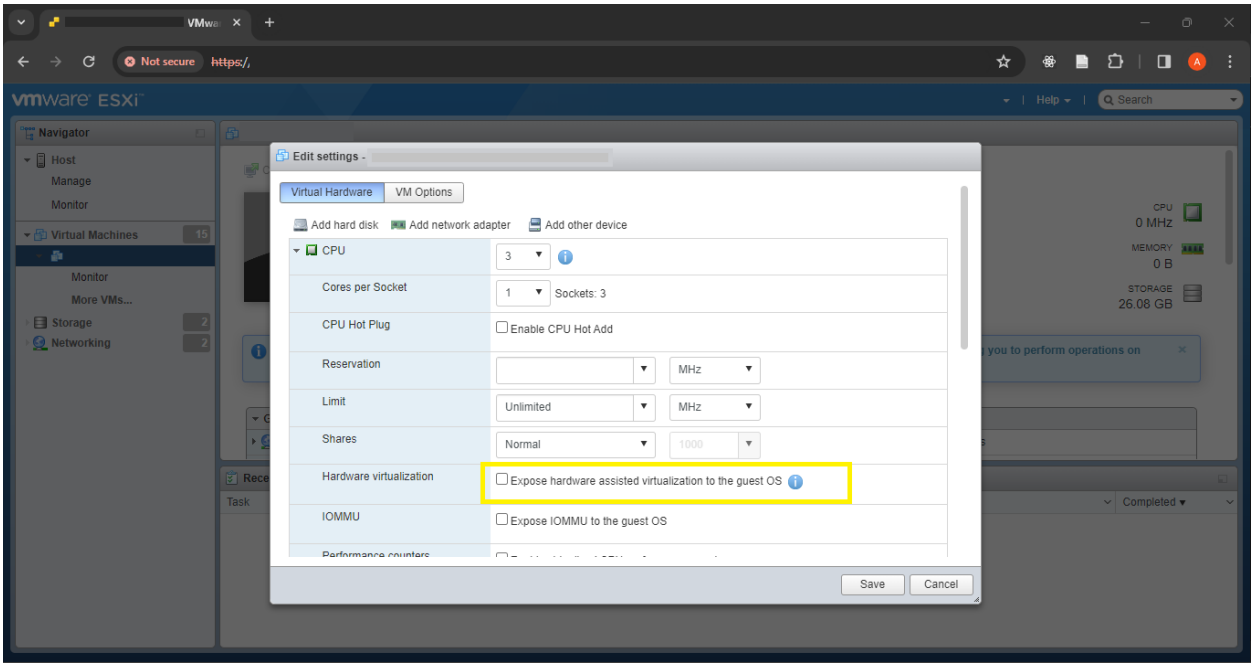

Figura 18: Configuração da virtualização em cascata em uma máquina virtual (Fonte: Elaboração própria)

Tendo activado a virtualização aninhada e prosseguido com a instalação do Docker, o passo seguinte envolve a compilação e a Implementação do contentor Docker através de um ficheiro Docker Compose.

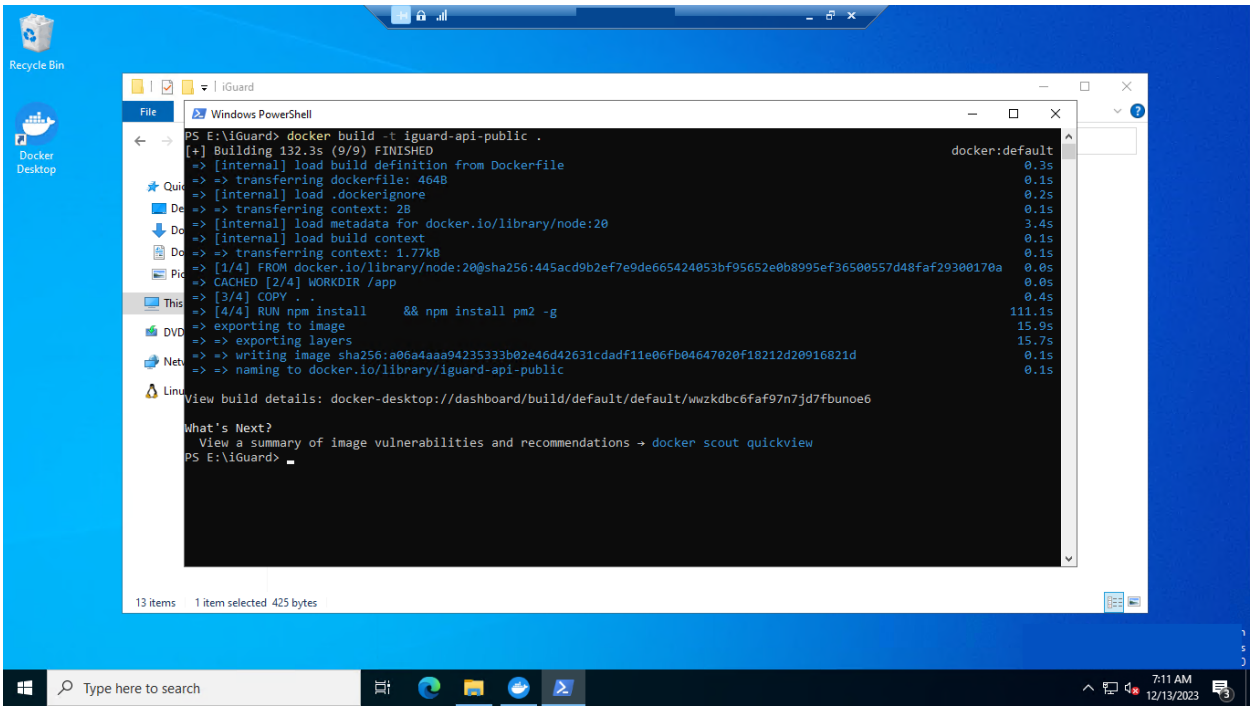

Figura 19: Compilação do contentor docker

Após a criação da imagem do contentor Docker, o passo seguinte envolve a sua implementação através da consola Docker.

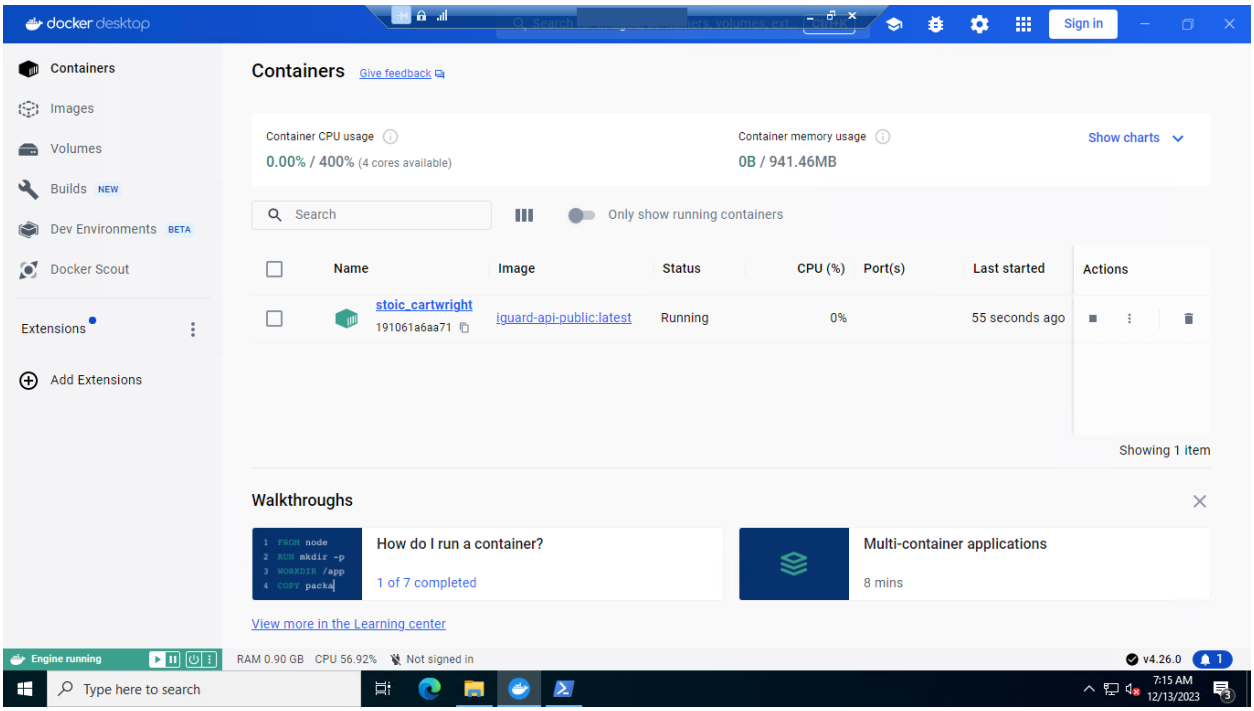

Figura 20: Contentores na consola docker (Fonte: elaboração própria)

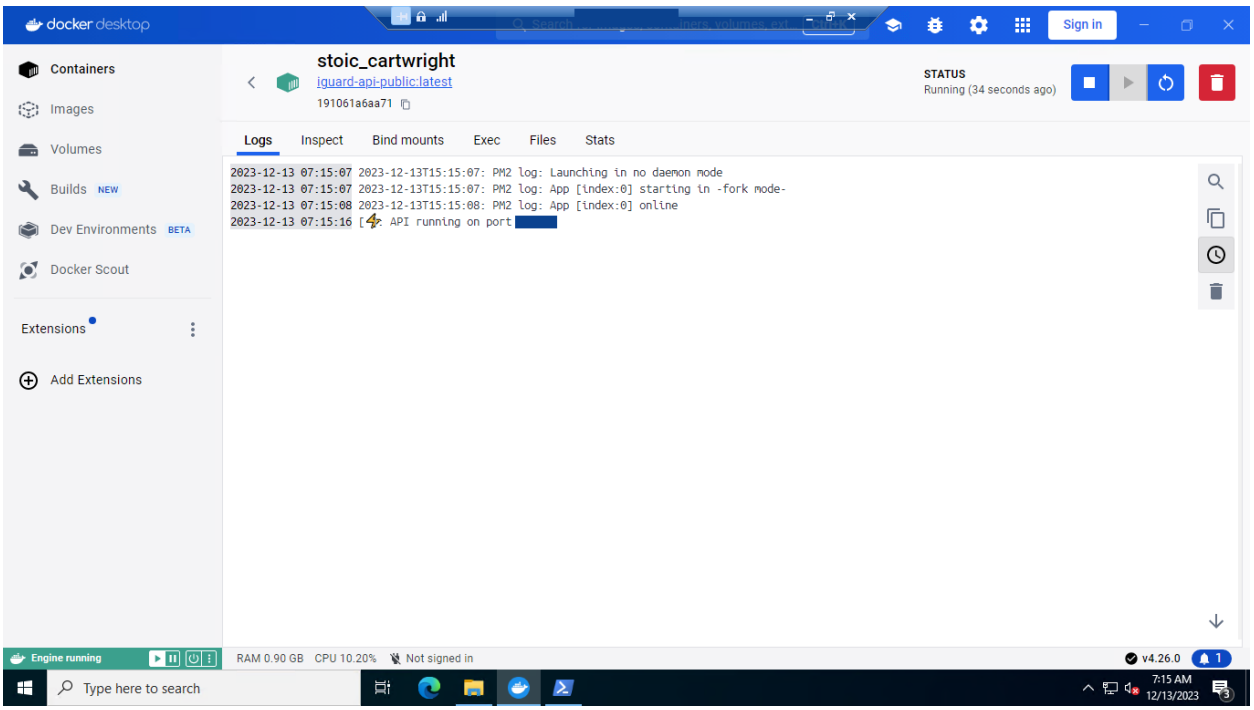

Figura 21: Consola do contentor docker com a Implementação da API (Fonte: Elaboração própria)

4.5.3 Implementação da base de dados

A base de dados será instalada num servidor dedicado, recorrendo ao MongoDB Community Server. As tarefas de gestão serão realizadas através da consola MongoDB Compass.

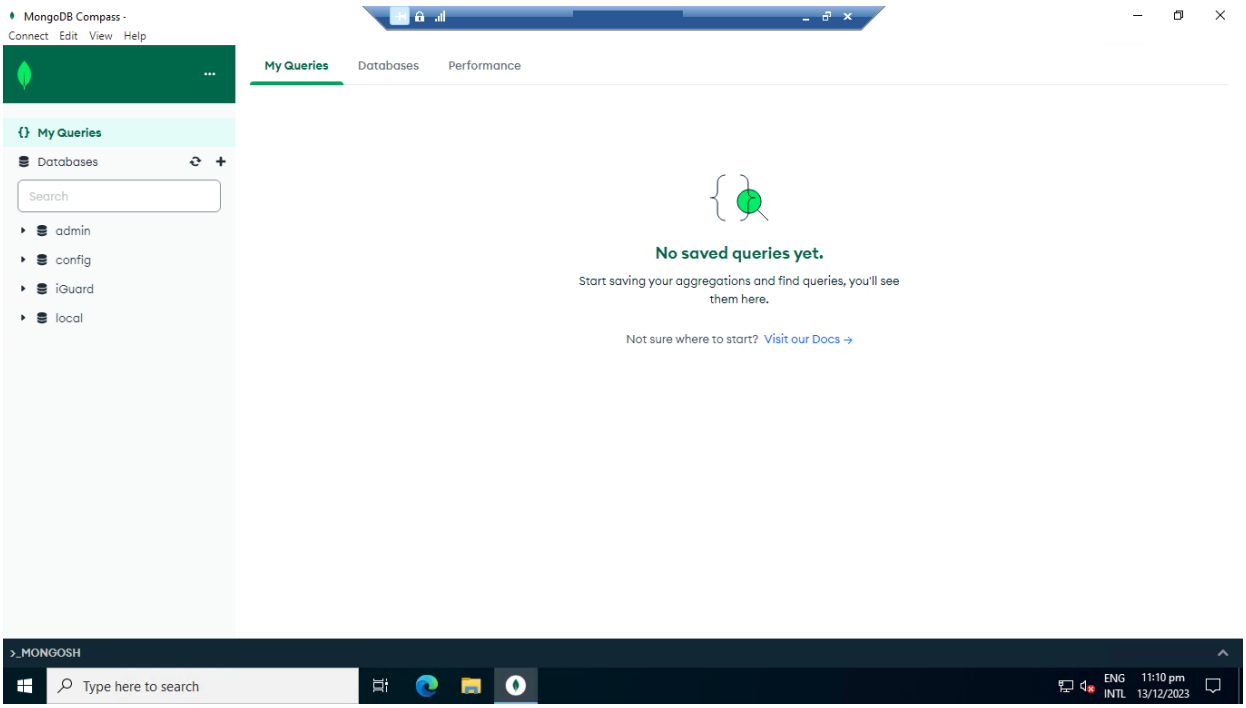

Figura 22: MongoDB compass (Fonte: Elaboração própria)

4.6 Manutenção do código

4.6.1 Contínua monitorização para a mitigação de vulnerabilidades no código fonte e dependências

Para a monitorização contínua, controlo e mitigação de vulnerabilidades no código fonte ou nas dependências do sistema, foi utilizada a plataforma Snyk-code.

Snyk-code é uma plataforma de segurança para desenvolvedores. Integrando-se directamente com as ferramentas de desenvolvimento, fluxos de trabalho e pipelines de automação, tornando mais fácil para as equipas encontrar, priorizar e corrigir

vulnerabilidades de segurança no código, dependências, contentores e infra-estrutura como código. (Synk, 2023)

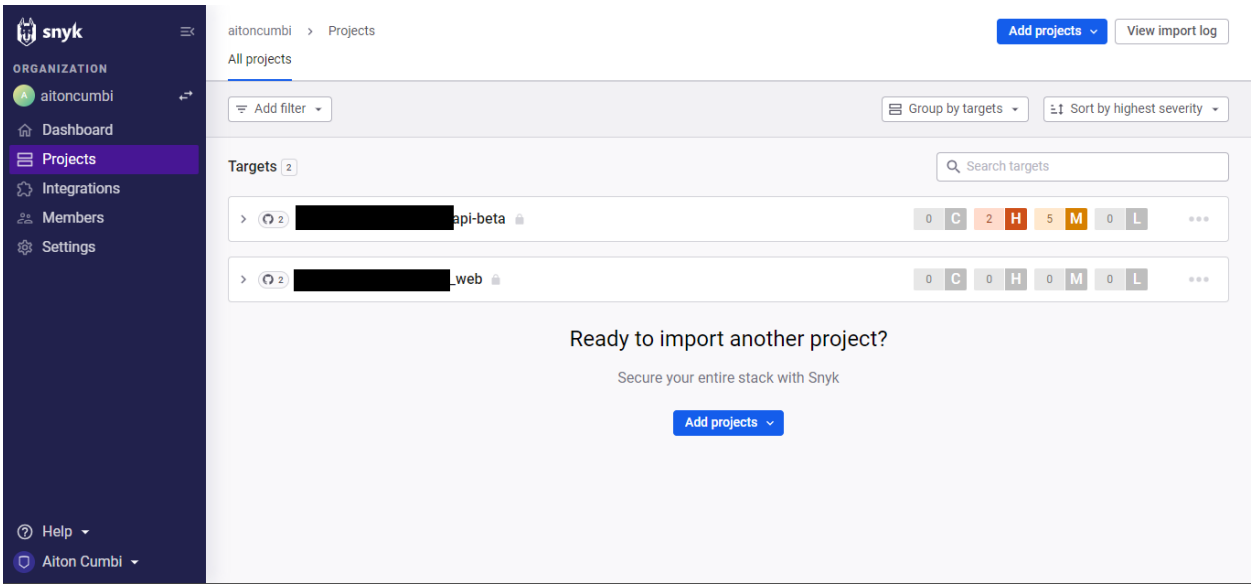

Figura 23: dashboard synk-code

4.6.2 Qualidade do código fonte

Para a ferramenta de controlo da qualidade do código, optou-se pela plataforma de código aberto SonarQube, que é utilizada para a inspecção contínua da qualidade do código. Para além de analisar o código, a plataforma fornece relatórios que identificam erros, vulnerabilidades, deficiências de código e outros potenciais problemas.

5 Capítulo V - Conclusões e Recomendações

5.1 Conclusão

O presente relatório apresentou a concepção e desenvolvimento de um sistema de monitorização personalizado para um sistema de CCTV distribuído. O sistema foi concebido para substituir o sistema de monitorização actual e colmatar as suas deficiências, tais como a falta de suporte multiplataforma, preocupações com a privacidade, escalabilidade limitada, fiabilidade em tecnologia obsoleta e ultrapassada e concepção inflexível.

O sistema proposto resolve estas deficiências utilizando uma interface moderna baseada na Web que é compatível com os navegadores modernos, empregando a transmissão segura de dados e a encriptação através do protocolo HTTPS. O sistema é também escalável, uma vez que foi concebido para suportar um grande número de utilizadores e de câmeras de segurança. Foi também concebido para ser facilmente adaptável à evolução das necessidades.

O sistema foi concebido utilizando uma combinação de componentes e plataformas de software de código aberto e comercial. A interface baseada na Web foi desenvolvida utilizando React, a versão móvel do sistema foi desenvolvida utilizando React-native e a API de backend foi construída sobre o ambiente de execução NodeJS com o apoio da framework ExpressJS. A transmissão de vídeo em directo multiplataforma foi feita através do protocolo HLS e a comunicação com as câmeras foi feita utilizando a norma ONVIF.

Na generalidade, o desenvolvimento do sistema de monitorização CCTV revelou-se um grande sucesso. O sistema colmatou as deficiências do sistema de monitorização precedente e tem sido largamente aceite pela organização.

5.2 Recomendações

Durante o desenvolvimento do sistema, foram observados alguns cenários que podem melhorar a funcionalidade do sistema e aumentar a sua segurança, tais como:

Implementação de uma ferramenta de monitorização contínua do desempenho e da segurança.

Exploração da possibilidade de recorrer à inteligência artificial para melhorar as capacidades do sistema, tais como a detecção de objectos e pessoas em áreas perigosas ou restritas, e a detecção de acidentes ou descarrilamentos.

Referências bibliográficas

Akka. (2023). *Actor Systems • Akka Documentation*. Documentation. Retrieved December 7, 2023, from https://doc.akka.io/docs/akka/current/general/actor-systems.html

Awati, Y., Bhintade, A., Gutal, R., & Taware, S. (2014, Março). International Journal of Emerging Technology and Advanced Engineering. *Onvifsense: ONVIF Network Device Accessibility Application*, *4*(3), 6. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=69933145cbb0758c80ce36e03c5b2ffac29a7e7a

Bigelow, S. J. (2020, September 30). *Functional vs. nonfunctional requirements in software engineering*. TechTarget. Retrieved December 7, 2023, from https://www.techtarget.com/searchsoftwarequality/answer/Functional-and-nonfunctional-requirements

Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.

Damjanovski, V. (2005). *CCTV: Networking and Digital Technology*. Elsevier Science.

Deisman, W. (2003, January). (PDF) CCTV: Literature Review and Bibliography. *ResearchGate*. https://www.researchgate.net/publication/254152990_CCTV_Literature_Review_and_Bibliography

Dempsey, J. (2007). *Introduction to Private Security*. Cengage Learning.

Doni, F. R. (2019, August 2). *Monitoring Kamera CCTV Untuk Perangkat Mobile Dengan Menggunakan Smartphone*. e Journal BSI. Retrieved December 4, 2023, from https://ejournal.bsi.ac.id/ejurnal/index.php/jtk/article/download/5368/pdf

Fortune Business Insights. (2023, Janeiro). *CCTV Camera Market Size, Growth | Global Report [2022-2029]*. Fortune Business Insights. Retrieved Julho 12, 2023, from https://www.fortunebusinessinsights.com/cctv-camera-market-107115

iSARSOFT. (2023, June 24). *What is a Video Management System (VMS)? VMS Meaning*. Isarsoft. Retrieved December 3, 2023, from https://www.isarsoft.com/knowledge-hub/vms

Mahr, N. (2022, April 20). *Use Case Diagram, Document & Template | Overview & Examples*. Study.com. Retrieved December 7, 2023, from https://study.com/learn/lesson/use-case-diagram-template.html

Microsoft. (2023). TypeScript: JavaScript With Syntax For Types. Retrieved November 13, 2023, from https://www.typescriptlang.org/

OpenJS Foundation. (2017). Express - Node.js web application framework. Retrieved December 8, 2023, from https://expressjs.com/

Organization | Physical Security Interoperability Alliance. (n.d.). Physical Security Interoperability Alliance |. Retrieved December 5, 2023, from https://psialliance.org/about/organization/

Pal, S. (2018). A Critical review on software requirements specification: quality perspective. *International Journal of Engineering Science*.

Quirchmayer, G., Basl, J., You, I., Xu, L., & Weippl, E. (Eds.). (2012). *Multidisciplinary Research and Practice for Informations Systems: IFIP WG 8.4,*

8.9, TC 5 International Cross Domain Conference and Workshop on Availability, Reliability, and Security, CD-ARES 2012, Prague, Czech Republic, August 20-24, 2012, Proceedings. Springer Berlin Heidelberg.

Sparx Systems. (2023). *UML Tutorial - Unified Modelling Language*. Sparx Systems. Retrieved December 7, 2023, from https://sparxsystems.com/resources/tutorials/uml/part1.html.

Synk. (2023). Snyk | Developer security | Develop fast. Stay secure. | Snyk. Retrieved November 12, 2023, from https://snyk.io/

Velastin, S. A., & Remagnino, P. (Eds.). (2006). *Intelligent Distributed Video Surveillance Systems*. Institution of Engineering and Technology.

Castello, C. C., Fan, J., Chou, T.-S., & Kuo, H.-M. (2008). Integration and Implementation of Secured IP Based Surveillance Networks. *2008 IEEE Asia-Pacific Services Computing Conference*, 117–122. https://doi.org/10.1109/APSCC.2008.39

Chia-Hsu Kuo, Huan-Ming Hsu, Shu-Chun Ho, & Wen-Tin Lee. (2013). Universal Middleware Bridge System for IP cam networking. *2013 International Symposium on Next-Generation Electronics*, 291–295. https://doi.org/10.1109/ISNE.2013.6512347

Chin-Feng Lai, Han-Chieh Chao, Ying-Xun Lai, & Jiafu Wan. (2013). Cloud-assisted real-time transrating for http live streaming. *IEEE Wireless Communications*, *20*(3), 62–70. https://doi.org/10.1109/MWC.2013.6549284

Documentation for Visual Studio Code. (2023). https://code.visualstudio.com/docs

Fu Jianhui, & Wang Dong. (2019a). *Network video storage device and method with face recognition and analysis function*.

Fu Jianhui, & Wang Dong. (2019b). *Network video storage device and method with face recognition and analysis function*.

Ganesan, V. (2022). Effective IP Camera Video Surveillance With Motion Detection and Cloud Services. *International Journal of Innovative Science and Research Technology*, *7*(1), 219.

IBM. (2021, September 9). *IBM Documentation*. https://www.ibm.com/docs/en/i/7.2?topic=languages-nodejs

IBM Documentation. (2021, March 4). https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case

IEEE Approved Draft Standard for Adaptive Streaming. (2018). *IEEE P1857.7/D3, July 2018*, 1–70.

Kang Min Jae, Lee Jin Ho, Jang In Hwan, & Kim Jeom Sik. (2017). *CCTV CCTV monitoring system*.

Le, H. T., Nguyen, T., Ngoc, N. P., Pham, A. T., & Thang, T. C. (2018). HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(9), 2423–2427. https://doi.org/10.1109/TCSVT.2018.2850740

Moonsin, S., & Anusas-amornkul, T. (2017). Bandwidth and buffer-based (B2)-dynamic adaptive streaming over HTTP. *2017 10th International Conference on Ubi-Media Computing and Workshops (Ubi-Media)*, 1–6. https://doi.org/10.1109/UMEDIA.2017.8074144

Mu Kesong, She Yongtao, & Sun Yanlong. (2019). *A video monitoring method of an ONVIF protocol under an Internet network*.

[No title found]. (n.d.). *International Journal of Emerging Technology and Advanced Engineering*.

O que é arquitetura de três camadas (tiers) | IBM. (n.d.). Retrieved December 8, 2023, from https://www.ibm.com/br-pt/topics/three-tier-architecture

Organization | Physical Security Interoperability Alliance. (n.d.). Retrieved December 5, 2023, from https://psialliance.org/about/organization/

Park Sung Ha, Rim Chae Ook, Park Sung Jong, Kim Eui Soo, Jo Soo Yeun, & Kim Hyun Jeong. (2020). *Cctv cctv monitoring system for detecting the invasion in the exterior wall of building and method thereof*.

Pelco. (n.d.). *What is an ONVIF Camera? Guide to Protocols & Profiles*. Pelco. Retrieved December 5, 2023, from https://www.pelco.com/blog/onvif-guide

Peres, M. (2012, October 31). Padrões de Interoperabilidade em Sistemas de CFTV IP. *GuiadoCFTV*. https://www.guiadocftv.com.br/artigos/2012/10/padroes-de-interoperabilidade-em-sistemas-de-cftv-ip/

PSIA and ONVIF: Measuring Video Standards - asmag.com. (n.d.). Retrieved December 5, 2023, from https://www.asmag.com/showpost/9020.aspx

Rao, A., Lanphier, R., & Schulzrinne, H. (1998). *Real Time Streaming Protocol (RTSP)* (Request for Comments RFC 2326). Internet Engineering Task Force. https://doi.org/10.17487/RFC2326

Rossi, M. (2019, February 18). *Understanding ONVIF and its limitations*. CCTV Camera World Knowledge Base. https://www.cctvcameraworld.com/onvif-security-camera-compatibility/

Senst, T., Eiselein, V., Badii, A., Einig, M., & Keller, I. (2013, July 1). *A decentralized Privacy-sensitive Video Surveillance Framework*. 2013 18th International Conference on Digital Signal Processing, DSP 2013. https://doi.org/10.1109/ICDSP.2013.6622765

Sivertsen Clas Gerhard, & Torkehagen Paal Fure. (2020). *Network video transmitter and receiver display system with auto-adjustable power and remote host wakeup*.

Sonalika Vishwakarma, Harsh Ghosalkar, Rohan Benere, & Yogita Chavan. (2021). *Night Motion Detection and Alert System*. http://dx.doi.org/10.48175/IJARSCT-1643

Sredojev, B., Samardzija, D., & Posarac, D. (2015). WebRTC technology overview and signaling solution design and implementation. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1006–1009. https://doi.org/10.1109/MIPRO.2015.7160422

Understanding RTSP on Security Cameras: Benefits and Limitations | SecurityBros. (2023, March 10). https://securitybros.com/understanding-rtsp-on-security-cameras-benefits-and-limitations/

What is MPEG-DASH? | HLS vs. DASH. (n.d.). Cloudflare. Retrieved December 8, 2023, from https://www.cloudflare.com/learning/video/what-is-mpeg-dash/

Xiong Lisha. (2019). *Network marketing video live display system*.

6 Anexos

6.1 Anexo 1 - Diagrama de casos de estudo (versão extensa)

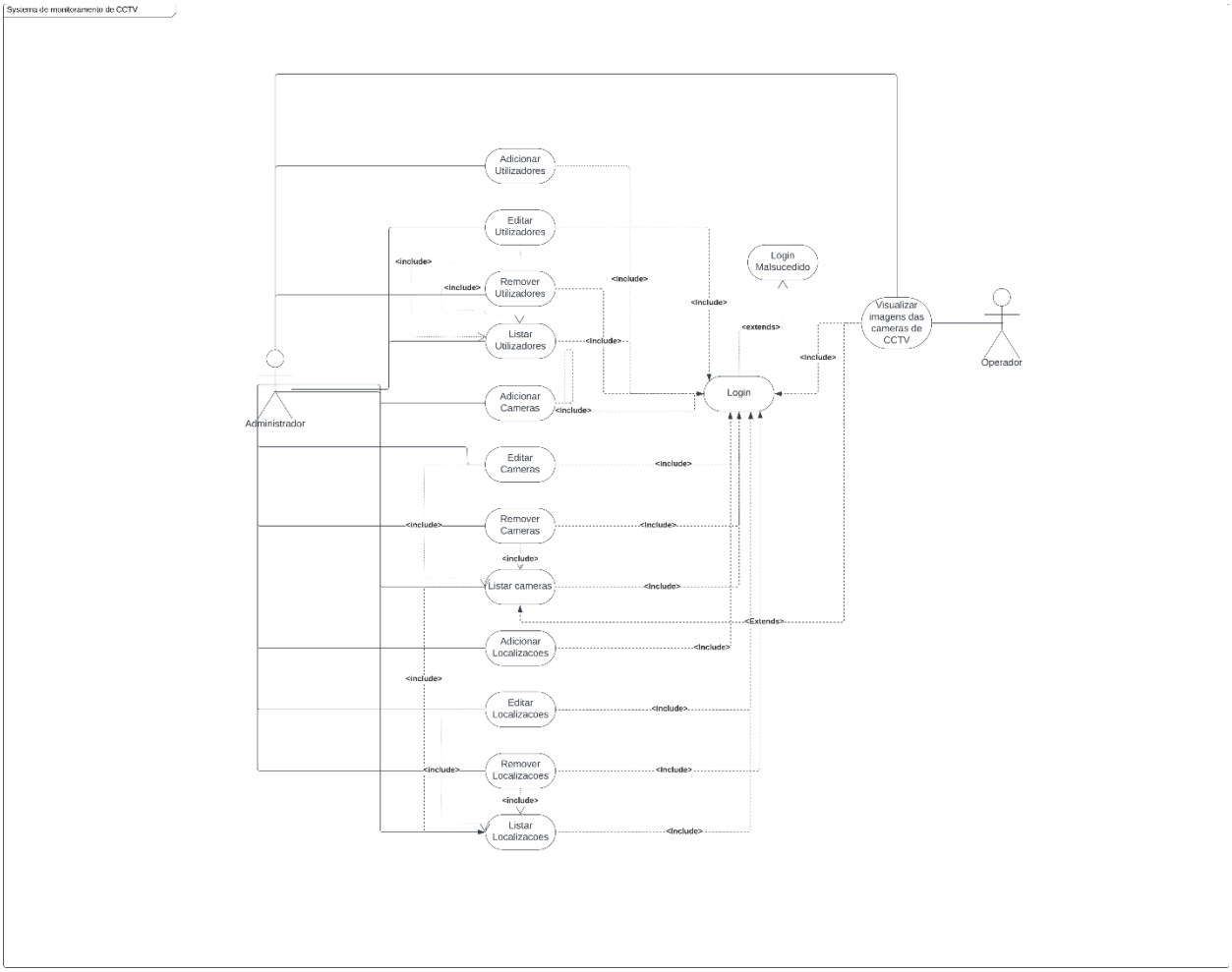

Figura 24: Diagrama de casos de uso, versão extensa

6.1.1 Descrição dos casos de uso

6.1.1.1 [UC_001] Login

ID	UC_001
Nome	Login
Descrição	Permite ao utilizador se autenticar ao sistema, para aceder a funções relevantes de acordo com o seu papel.
Actor	Administrador, operador

Pré-condições	O utilizador tem uma conta válida no sistema; O utilizador consegue aceder ao sistema
Pós-condições	O sistema apresenta ao utilizador uma página inicial
Fluxo	1. O utilizador acede ao sistema 2. O utilizador preenche o nome de utilizador e a senha no formulário 3. O utilizador pressiona o botão login 4. O sistema apresenta a página inicial

Tabela 8: Descrição do UC_001

6.1.1.2 [UC_002] Gerir utilizadores

ID	UC_002
Nome	Gerir utilizadores
Descrição	Permite ao utilizador realizar operações de criação, actualização, listagem e remoção de utilizadores
Actor	Administrador
Pré-condições	O utilizador tem uma conta válida no sistema; O utilizador consegue aceder ao sistema; O utilizador se autentica no sistema O utilizador tem a função de administrador;
Pós-condições	Utilizadores são criados, actualizados, listados e removidos
Fluxo	1. O utilizador acede ao sistema 2. O utilizador autentica-se no sistema 3. O utilizador seleciona a opção utilizadores 4. O utilizador é apresentado a lista de utilizadores 5. O utilizador selecciona a opção
	6 - Adicionar utilizador: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Editar utilizador: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Remover utilizador 7 - O utilizador selecciona o utilizador na lista de utilizadores 8 - o utilizador selecciona a opção apagar

Tabela 9: Descrição do UC_002

6.1.1.3 [UC_003] Gerir localizações

ID	UC_003
Nome	Gerir localizações
Descrição	Permite ao utilizador realizar operações de criação, actualização, listagem e remoção de localizações geográficas
Actor	Administrador
Pré-condições	O utilizador tem uma conta válida no sistema; O utilizador consegue aceder ao sistema; O utilizador se autentica no sistema O utilizador tem a função de administrador;
Pós-condições	Localizações geográficas são criadas, actualizadas, listadas e removidas
Fluxo	1. O utilizador acede ao sistema 2. O utilizador autentica-se no sistema 3. O utilizador selecciona a opção utilizadores 4. O utilizador é apresentado a lista de localizações 5. O utilizador selecciona a opção
	6 - Adicionar localização: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Editar localização: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Remover utilizador 7 - O utilizador selecciona a localização na lista de localizações 8 - o utilizador selecciona a opção apagar

Tabela 10: Descrição do UC_003

6.1.1.4 [UC_004] Gerir câmeras

ID	UC_004
Nome	Gerir câmeras
Descrição	Permite ao utilizador realizar operações de criação, actualização, listagem e remoção de câmeras de segurança
Actor	Administrador
Pré-condições	O utilizador tem uma conta válida no sistema; O utilizador consegue aceder ao sistema; O utilizador se autentica no sistema O utilizador tem a função de administrador;

Pós-condições	câmeras de segurança são criadas, actualizadas, listadas e removidas
Fluxo	1. O utilizador acede ao sistema 2. O utilizador autentica-se no sistema 3. O utilizador selecciona a opção utilizadores 4. O utilizador é apresentado a lista de câmeras de segurança 5. O utilizador selecciona a opção
	6 - Adicionar câmera: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Editar câmera: 7 - O utilizador preenche os campos do formulário; 8 - O utilizador pressiona o botão guardar;
	6 - Remover câmera 7 - O utilizador selecciona a câmera na lista de câmeras de segurança 8 - o utilizador selecciona a opção apagar

Tabela 11: Descrição do UC_004

6.1.1.5 [UC_005] Visualizar live stream CCTV

ID	UC_005
Nome	Visualizar live stream CCTV
Descrição	Permite ao utilizador visualizar uma ou múltiplas transmissões em directo das câmeras de vigilância
Actor	Administrador, operador
Pré-condições	O utilizador tem uma conta válida no sistema; O utilizador consegue aceder ao sistema; O utilizador tem acesso a lista das câmeras de segurança
Pós-condições	O sistema apresenta ao utilizador a transmissão em directo da câmera de vigilância
Fluxo	1. O utilizador acede ao sistema 2. O utilizador preenche o nome de utilizador e a senha no formulário 3. O utilizador pressiona o botão login 4. O sistema apresenta a página inicial 5. O utilizador selecciona a opção stream 6. O utilizador selecciona uma das janelas de reprodução 7. O utilizador selecciona a câmera de segurança na lista apresentada 8. A transmissão inicia

Tabela 12: Descrição do UC_005

6.2 Anexo 2 – Diagrama de classes e diagramas de sequência

6.2.1 Diagramas de sequência

6.2.1.1 Login

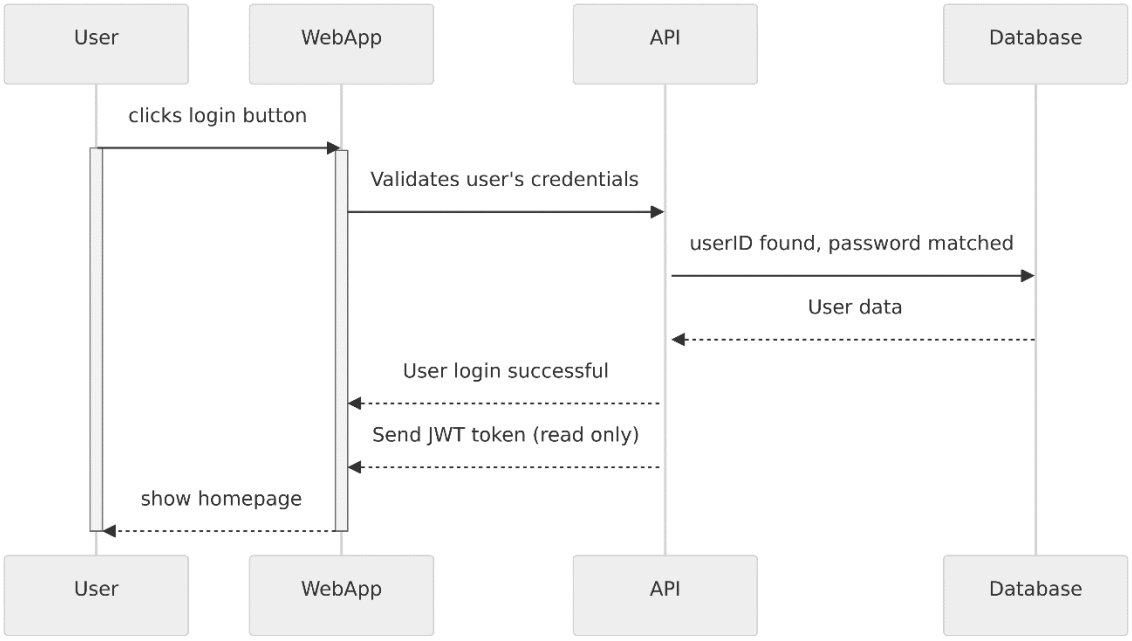

Figura 25: Diagrama de sequência de login (Fonte: Elaboração própria)

6.2.1.2 Visualizar a transmissão em directo

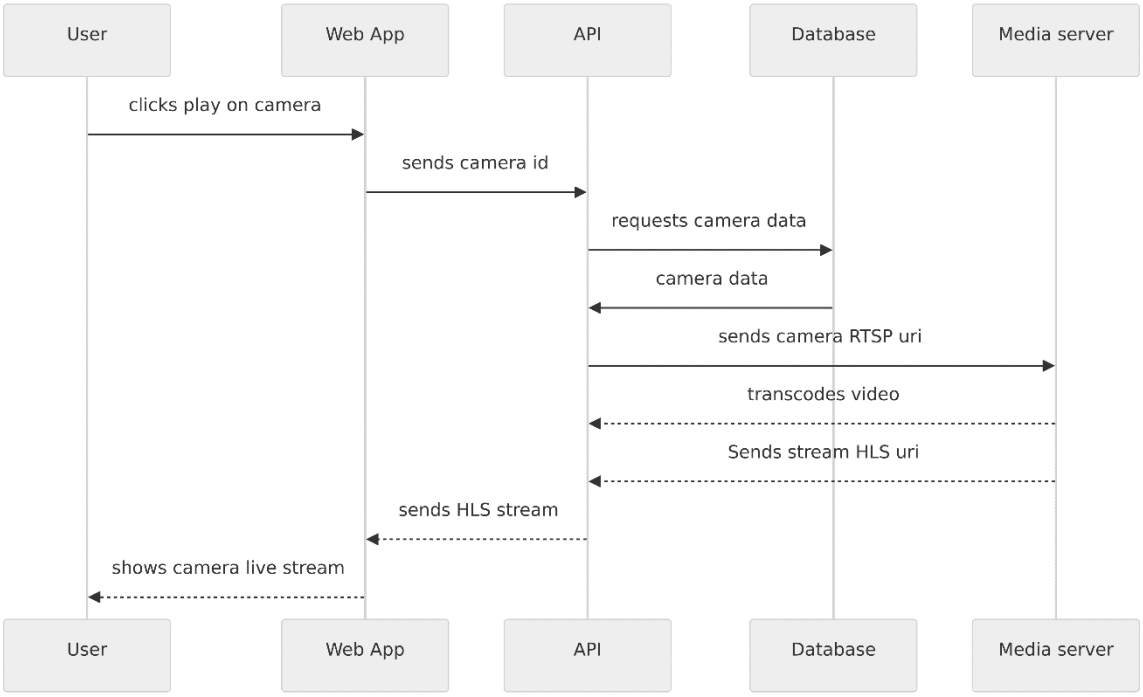

Figura 26: Diagrama de sequência de visualização da transmissão em directo (Fonte: elaboração própria)

6.3 Anexo 3 – Interfaces de utilizador

6.3.1 Aplicação Web

6.3.1.1 Ecrã de login

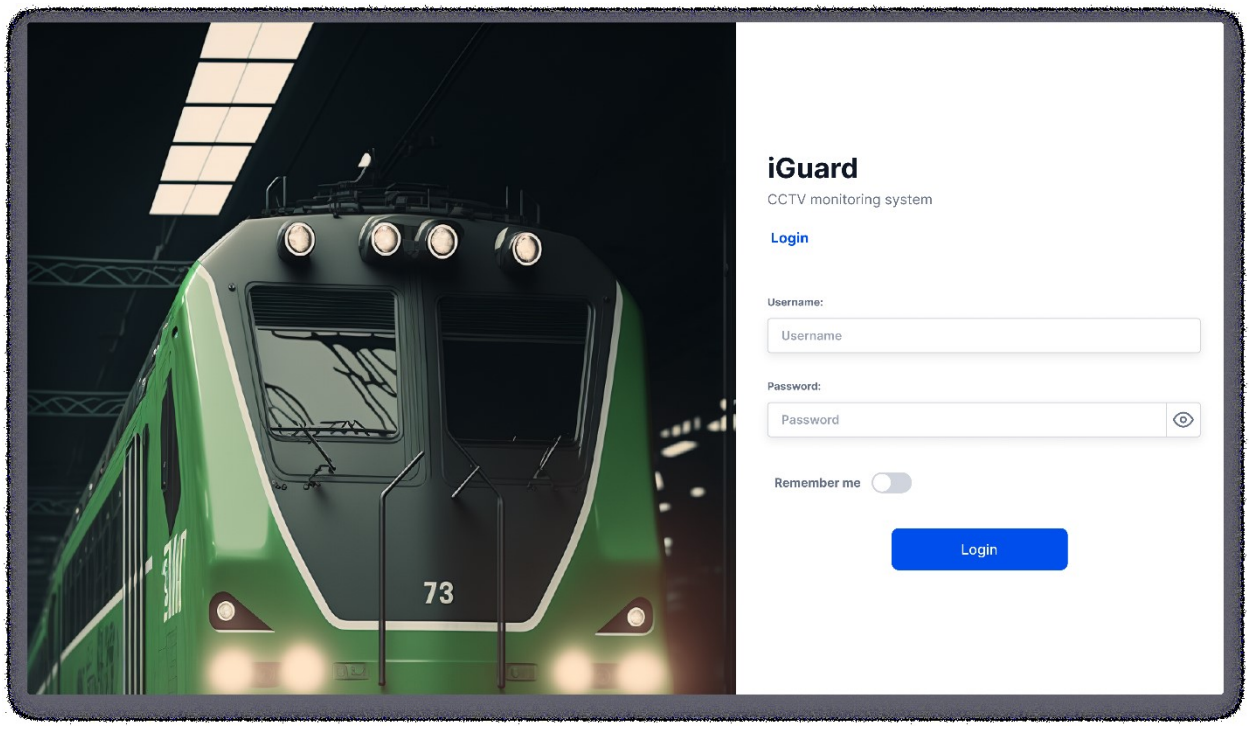

Figura 27: Ecrã de login (Fonte: Elaboração própria)

6.3.1.2 Ecrã da transmissão em directo pt.1

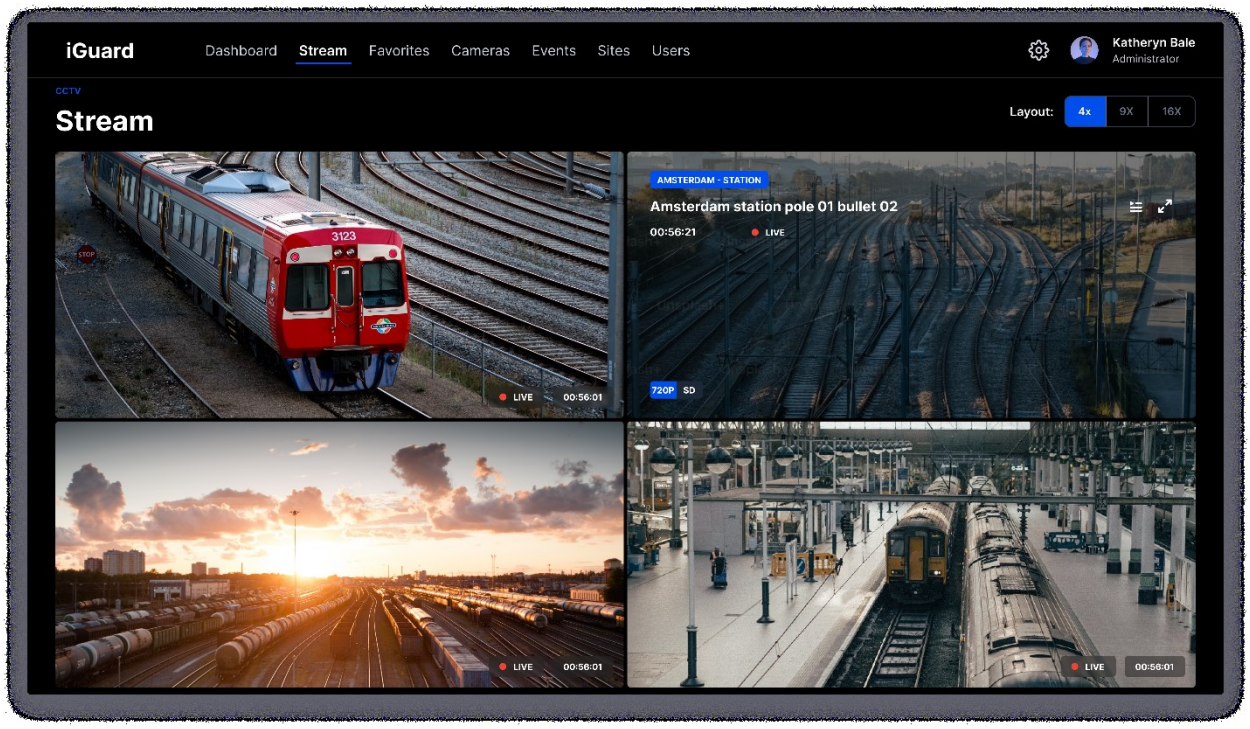

Figura 28: Ecrã da transmissão em directo (Fonte: elaboração própria)

6.3.1.3 Ecrã da transmissão em directo pt.2

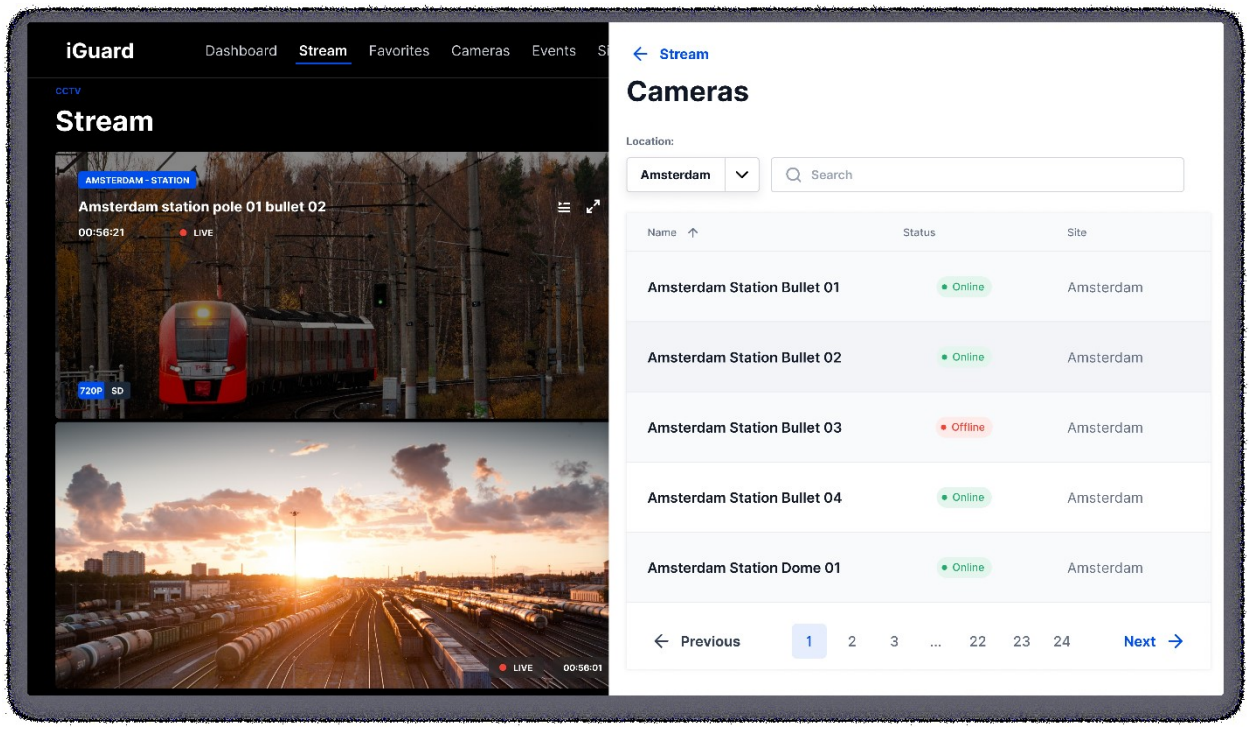

Figura 29: Ecrã da transmissão em directo (Fonte Elaboração própria)

6.3.1.4 Gestão de utilizadores

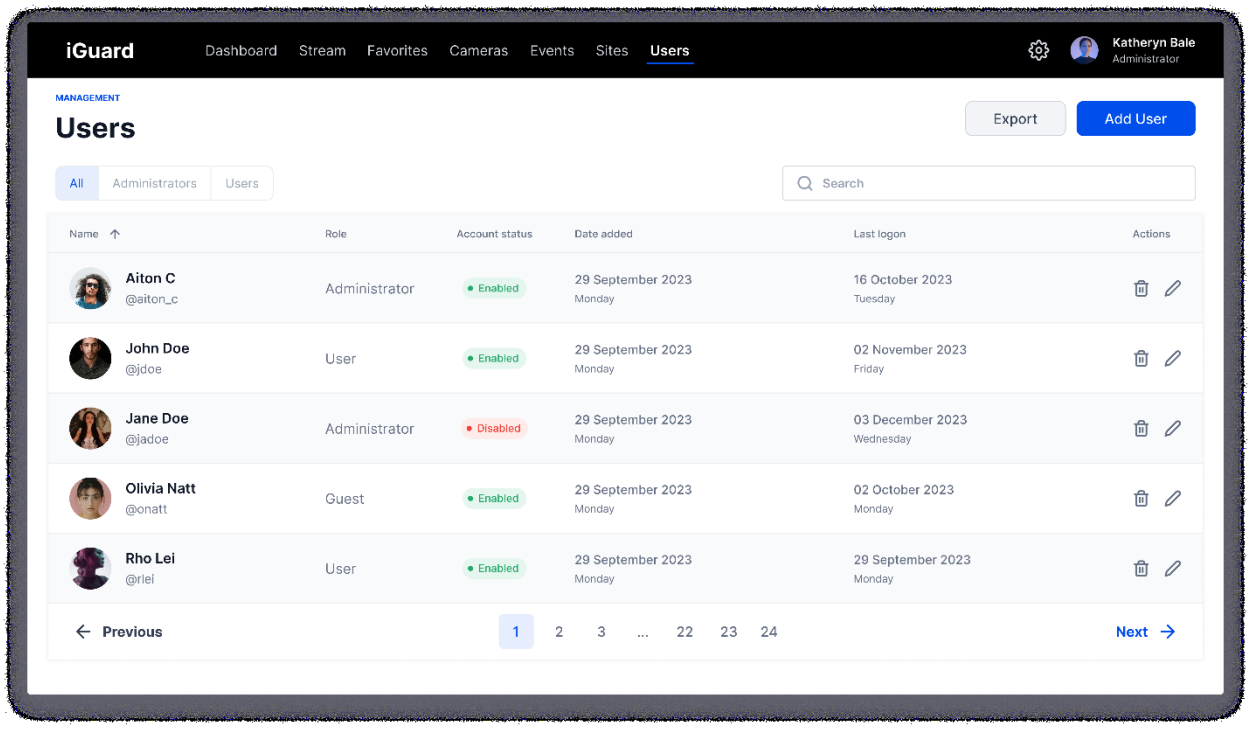

Figura 30: Ecrã de gestão de utilizadores (Fonte: elaboração própria)

6.3.1.5 Adicionar utilizador

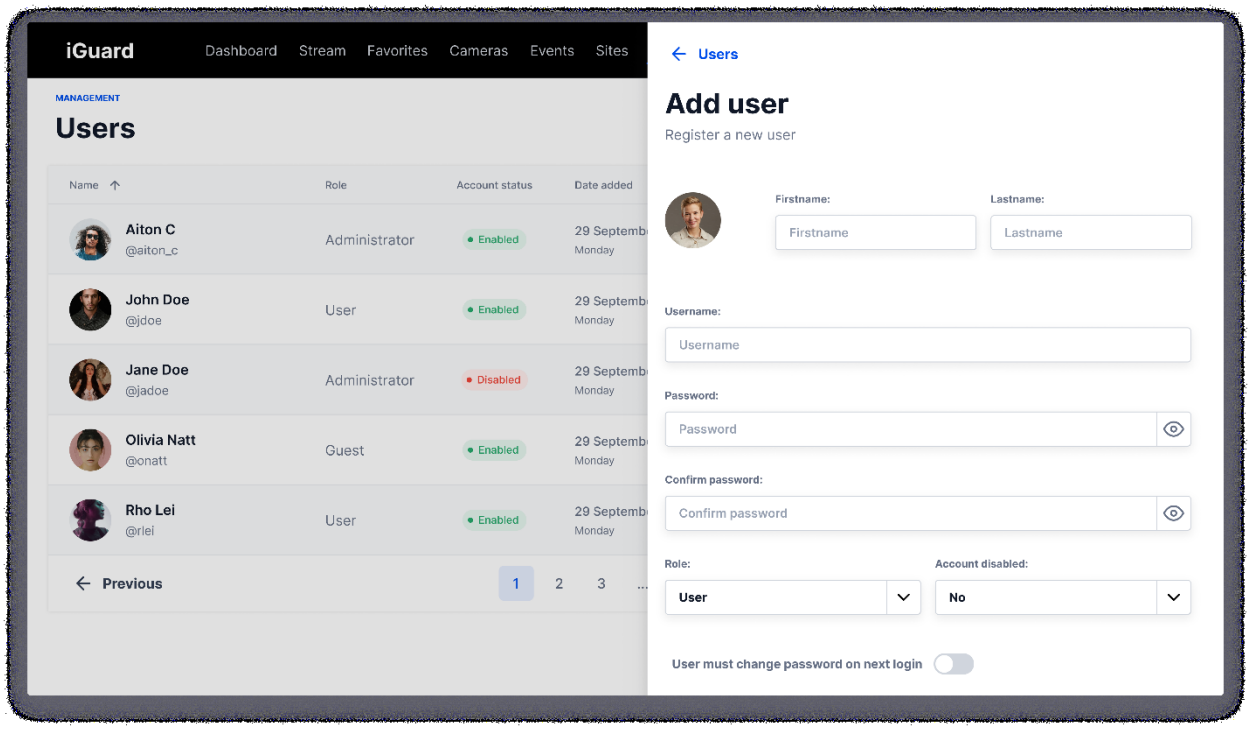

Figura 31: Ecrã de adição de utilizadores (fonte elaboração própria)

6.3.1.6 Editar utilizador

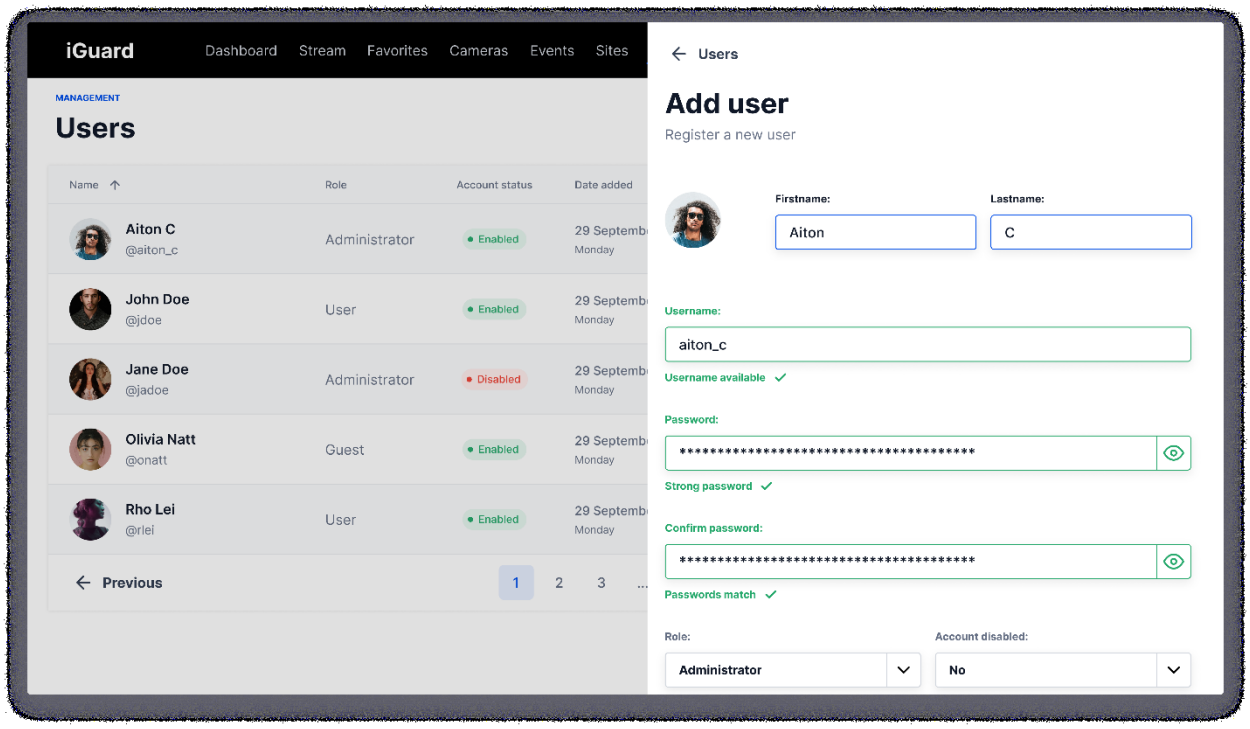

Figura 32: Ecrã de edição de utilizadores (fonte elaboração própria)

6.3.1.7 Gestão de câmeras

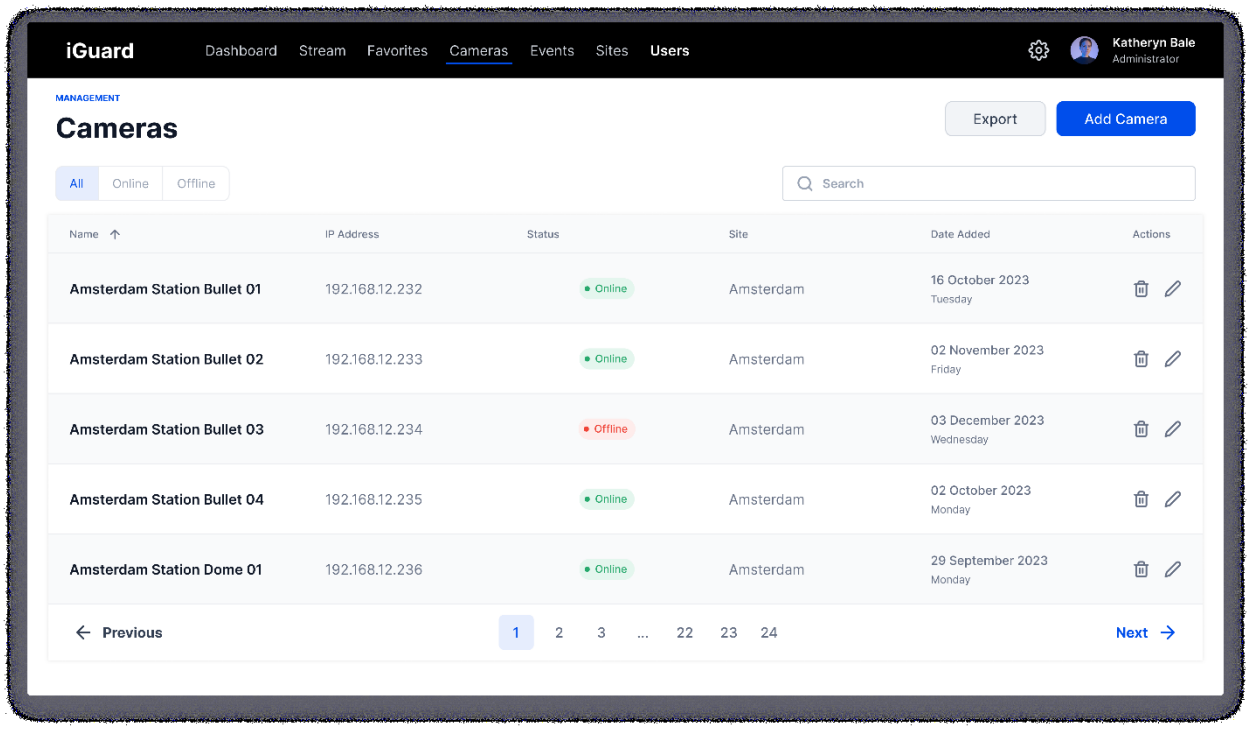

Figura 33: Ecrã da gestão de câmeras de segurança (Fonte: Elaboração própria)

6.3.1.8 Adicionar câmera

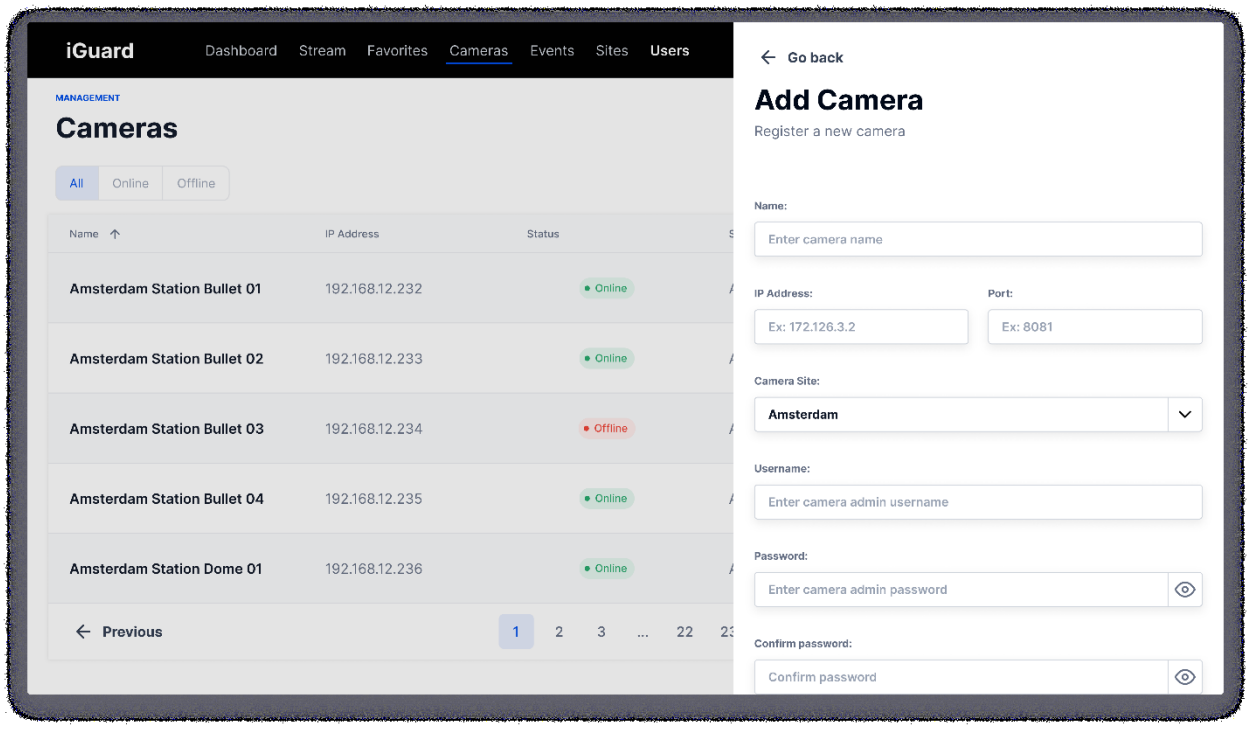

Figura 34: Ecrã de adição de uma câmera de segurança (Fonte: Elaboração própria)

6.3.1.9 Editar câmera de segurança

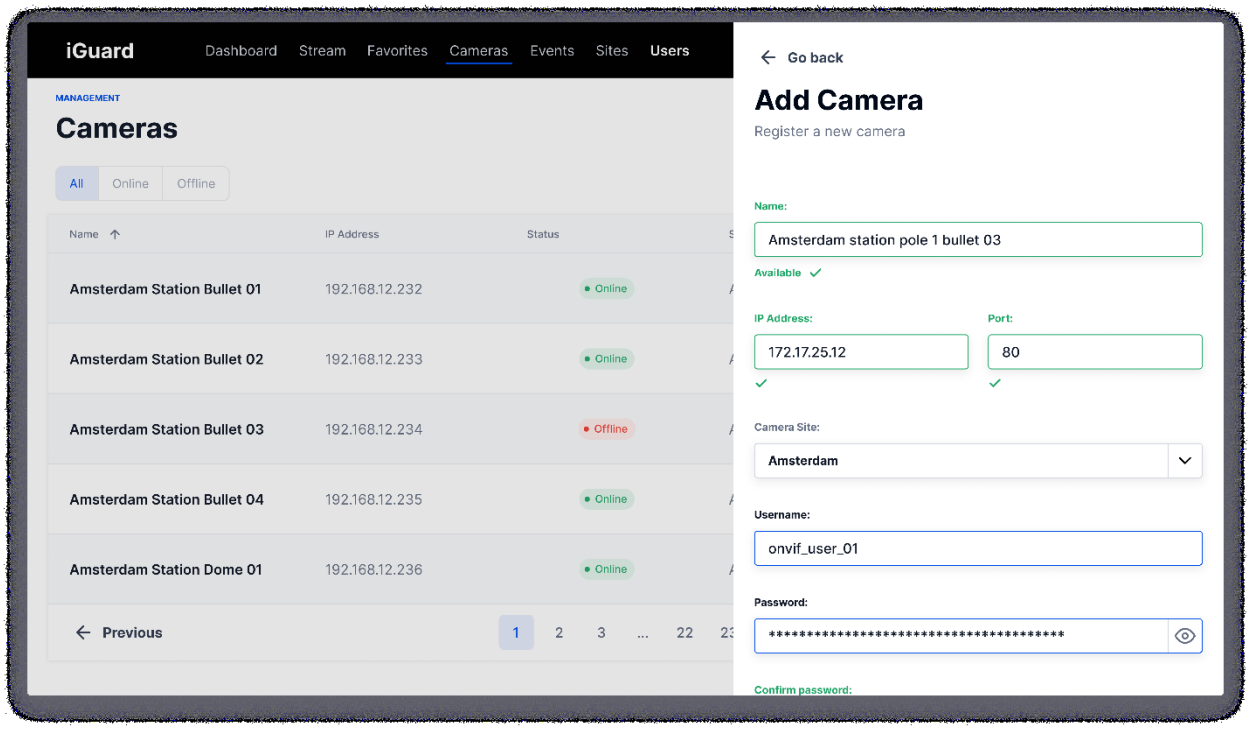

Figura 35: Ecrã de edição de uma câmera de segurança (Fonte: elaboração própria)

6.3.1.10 Gestão de localizações

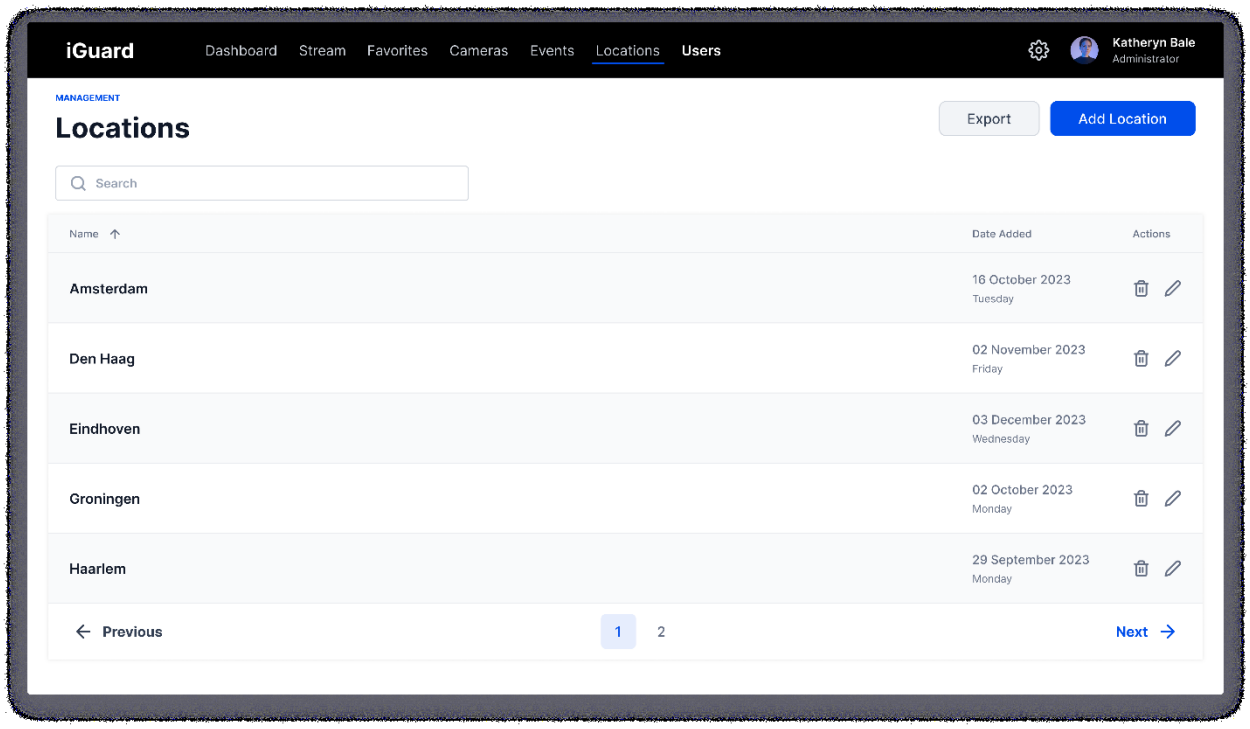

Figura 36: Ecrã de gestão de localizações (Fonte: elaboração própria)

6.3.1.11 Adicionar uma localização

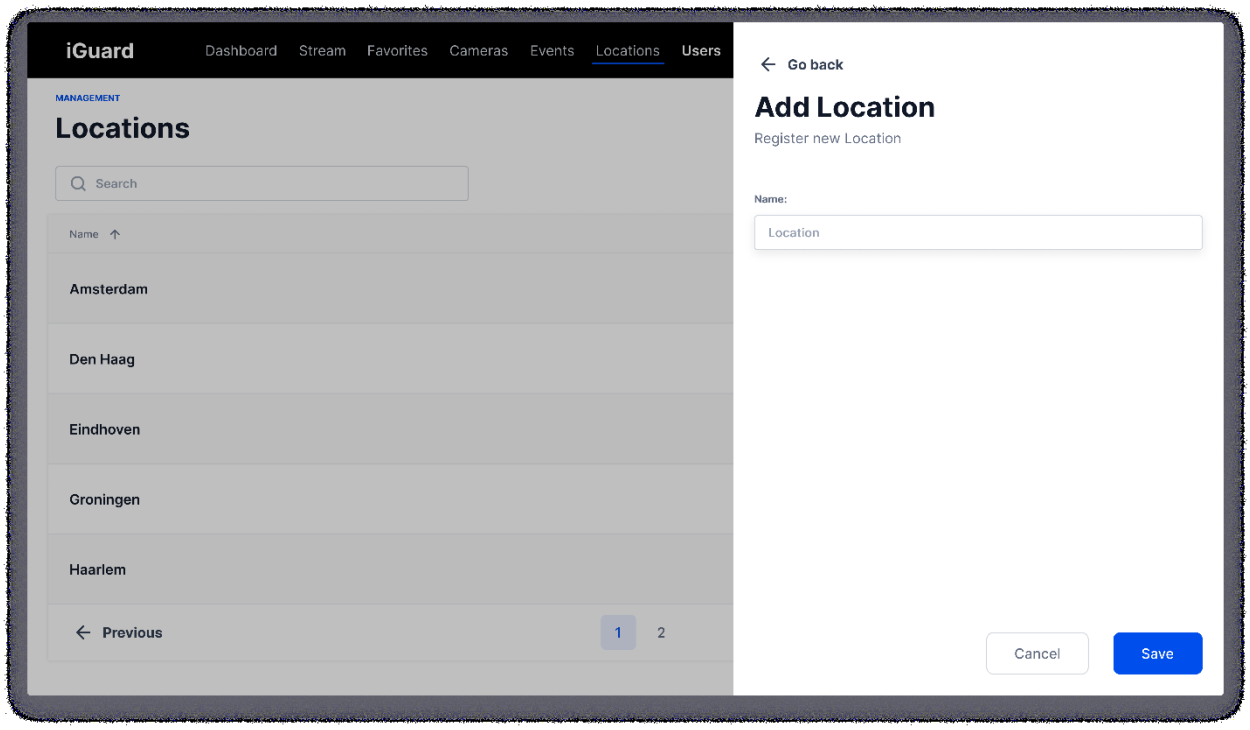

Figura 37: Ecrã de adição de localizações (Fonte: Elaboração própria)

6.3.1.12 Editar uma localização

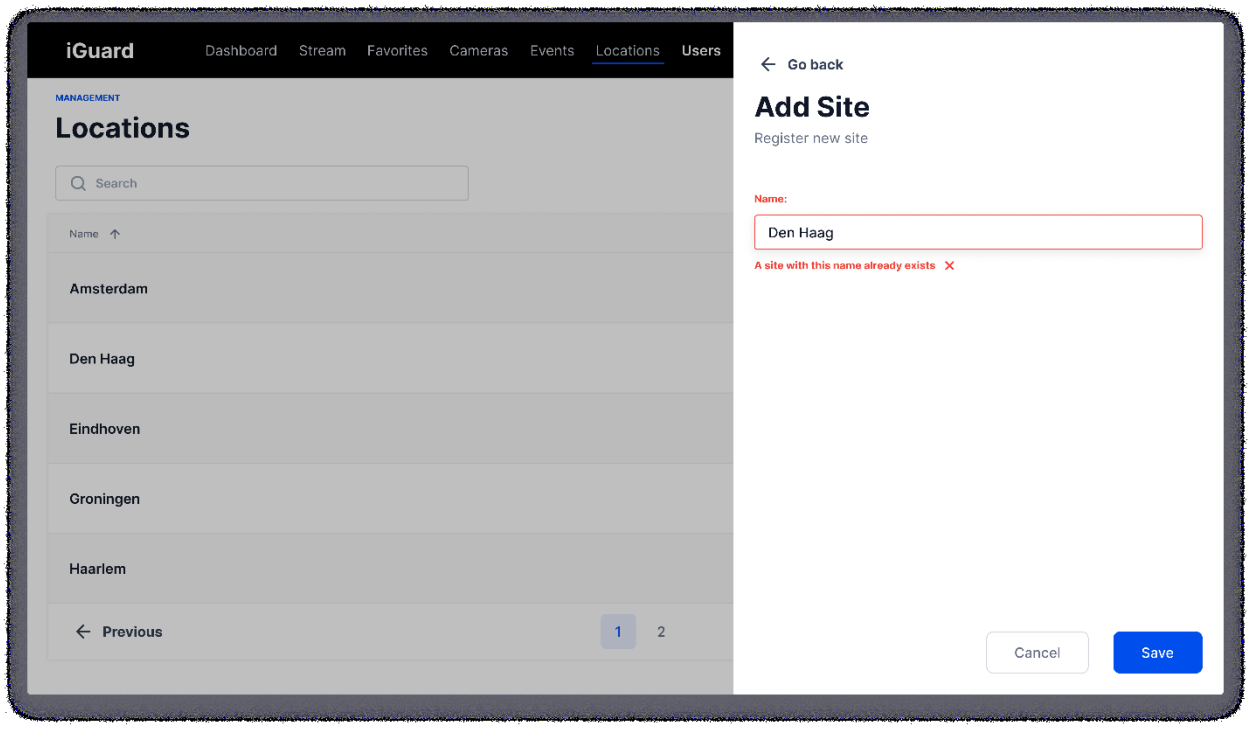

Figura 38: Ecrã de edição de uma localização (Fonte: Elaboração própria)

6.3.2 Aplicação móvel

6.3.2.1 Ecrã de login

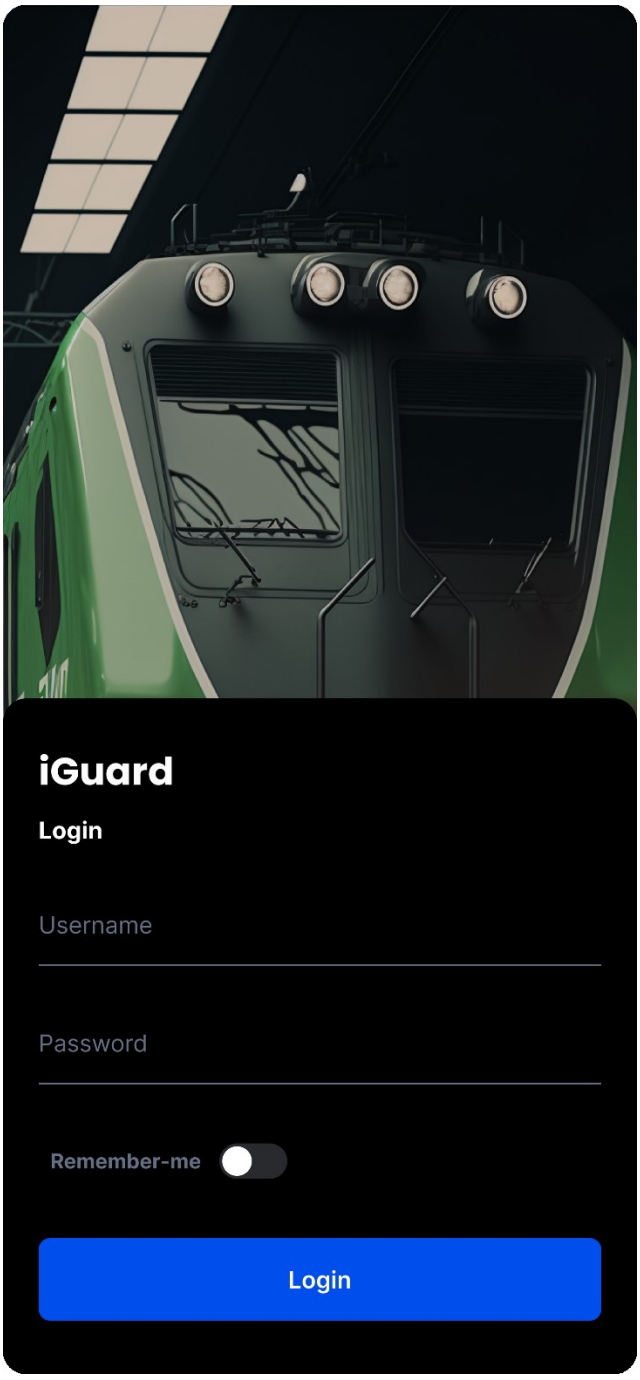

Figura 39: Ecrã de login (móvel) (Fonte: elaboração própria)

6.3.2.2 Ecrã da troca/actualização da palavra-passe de utilizador

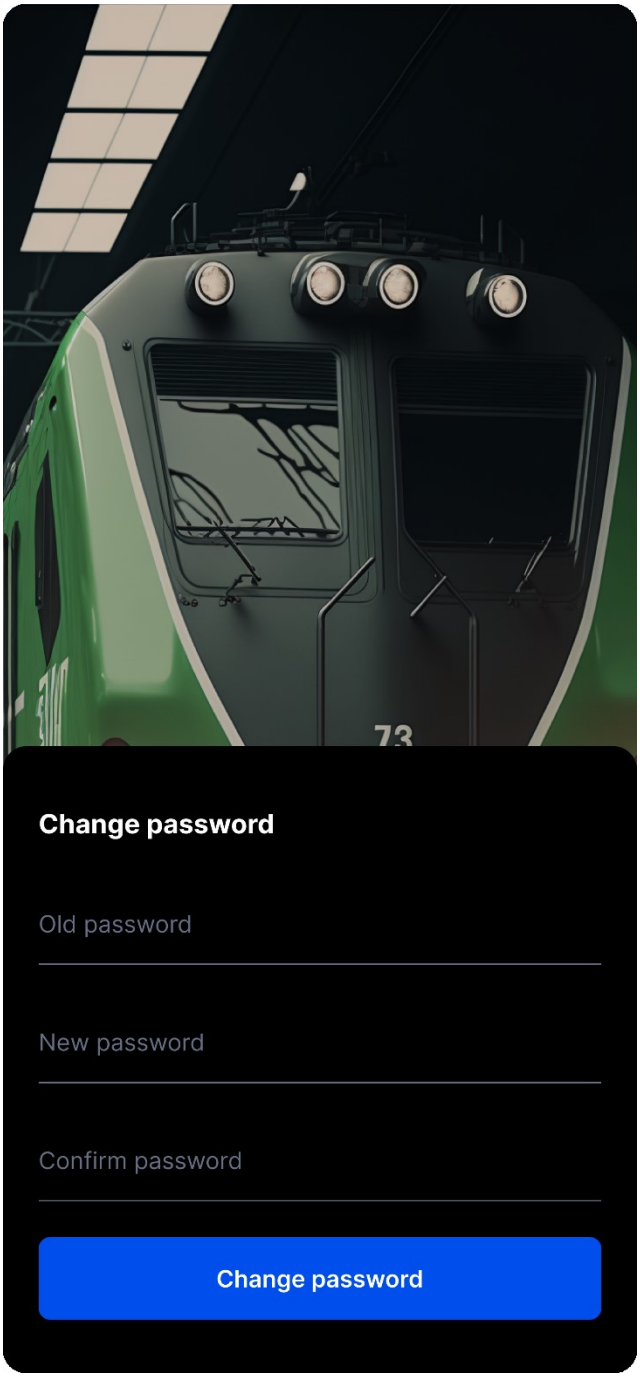

Figura 40: Ecrã da troca de password (móvel) (Fonte: elaboração própria)

6.3.2.3 Ecrã inicial

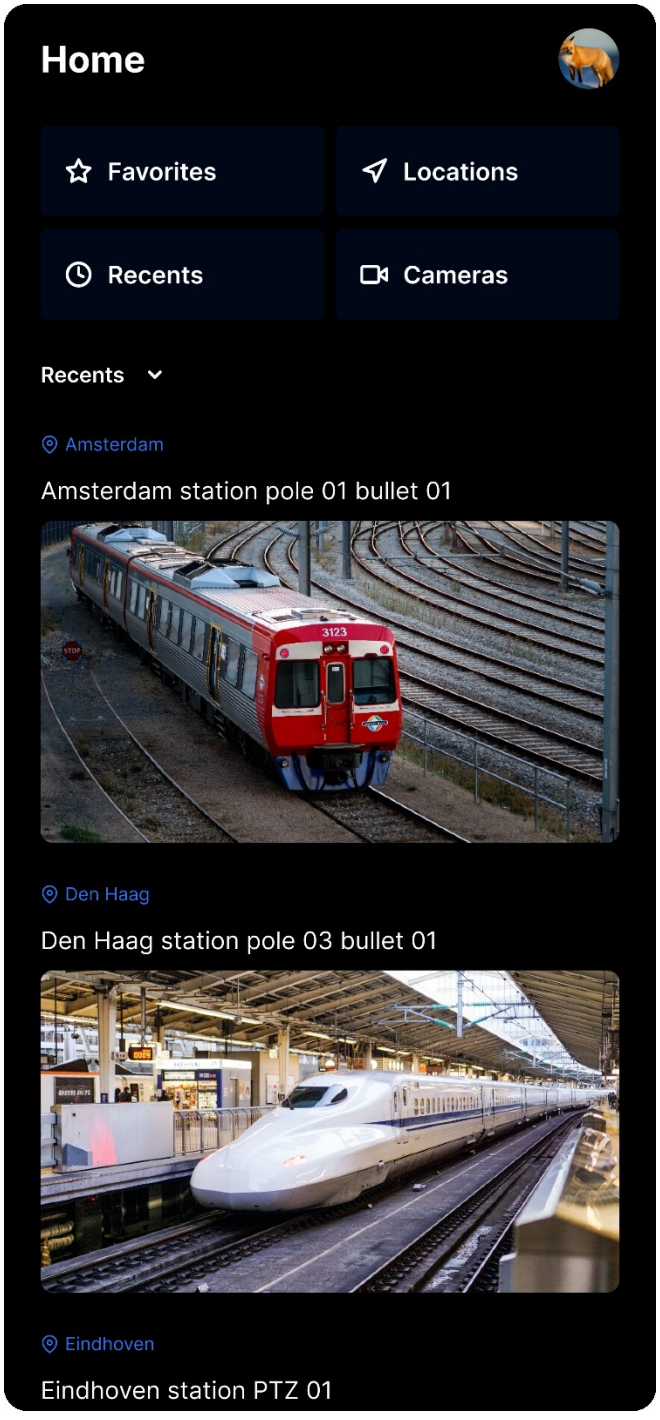

Figura 41: Ecrã inicial (móvel) (Fonte: elaboração própria)

6.3.2.4 Listagem das localizações

Figura 42: Ecrã da lista de localizações (móvel) (Fonte: elaboração própria)

6.3.2.5 Listagem das câmeras

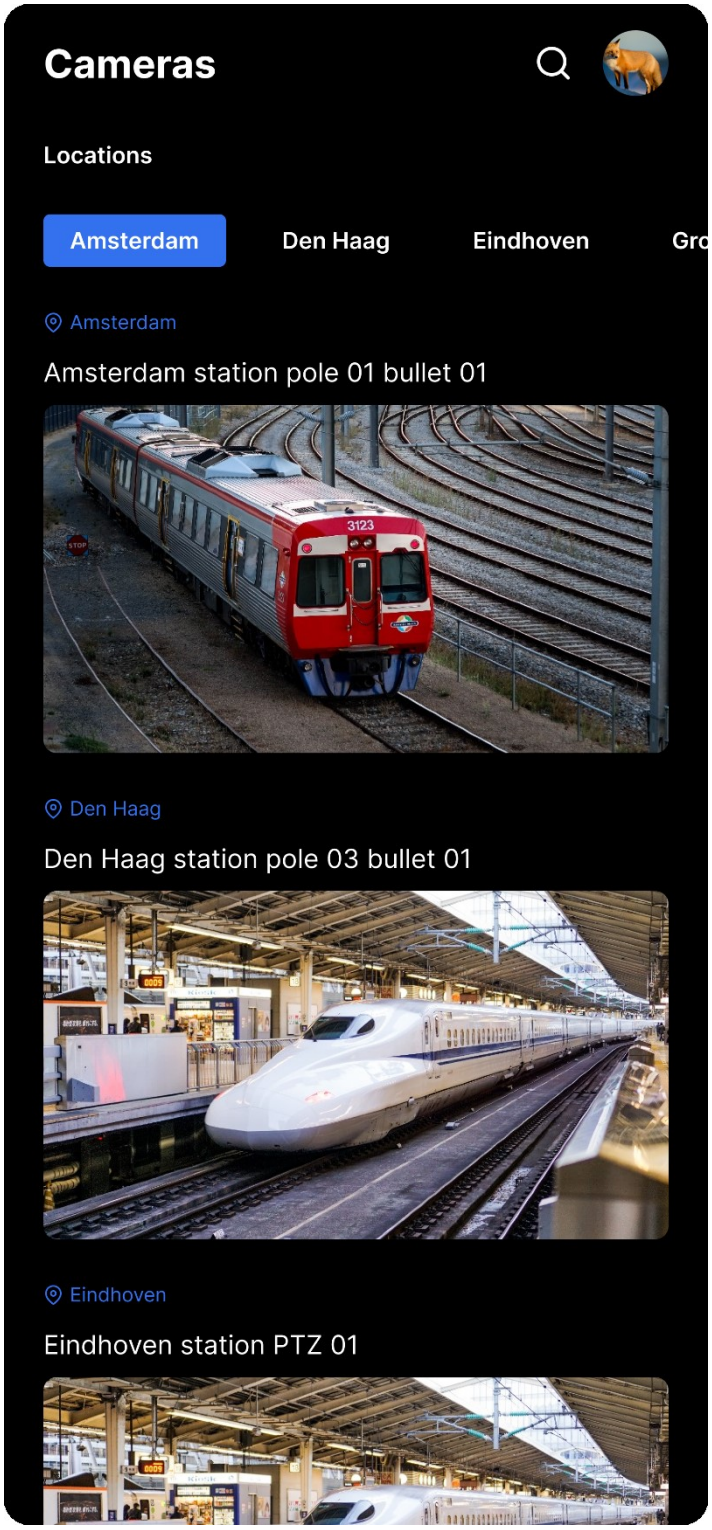

Figura 43: Ecrã de listagem das câmeras de segurança (móvel) (Fonte: elaboração própria)

6.3.2.6 Transmissão em directo das câmeras de segurança

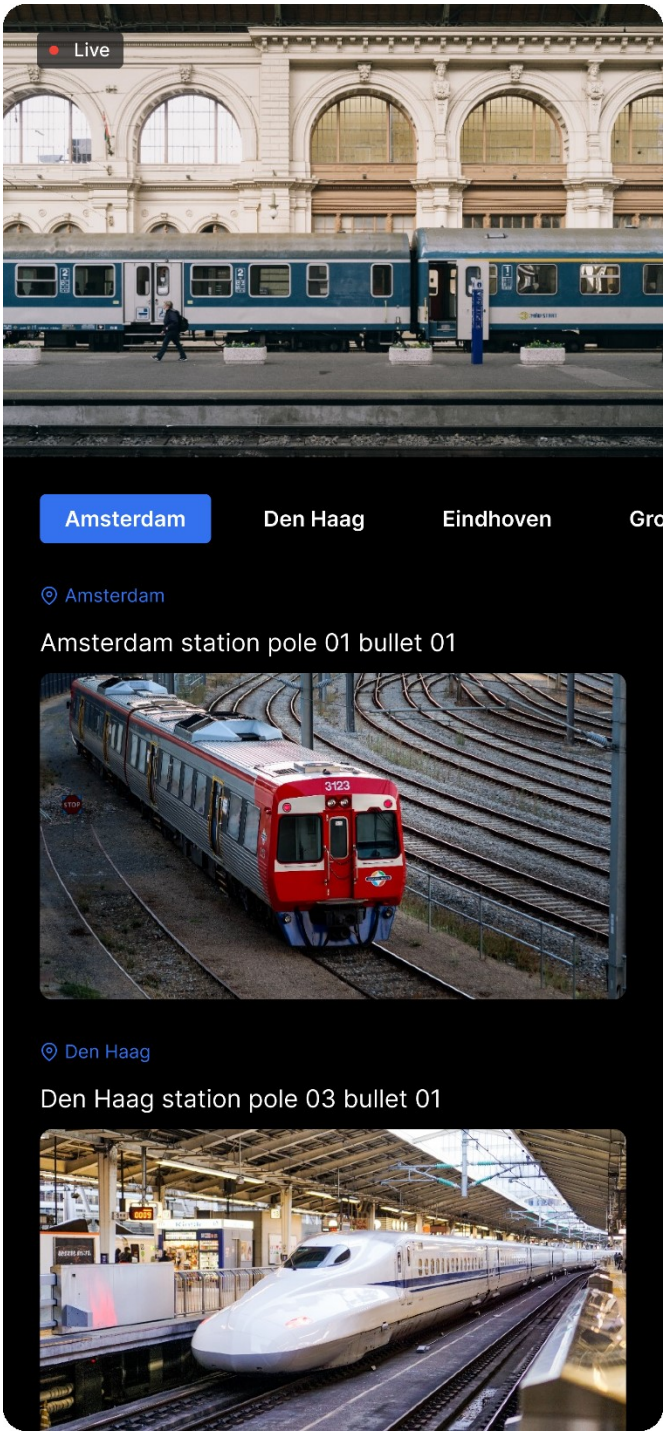

Figura 44: Ecrã da transmissão em directo das câmeras de segurança (móvel) (Fonte: elaboração própria)

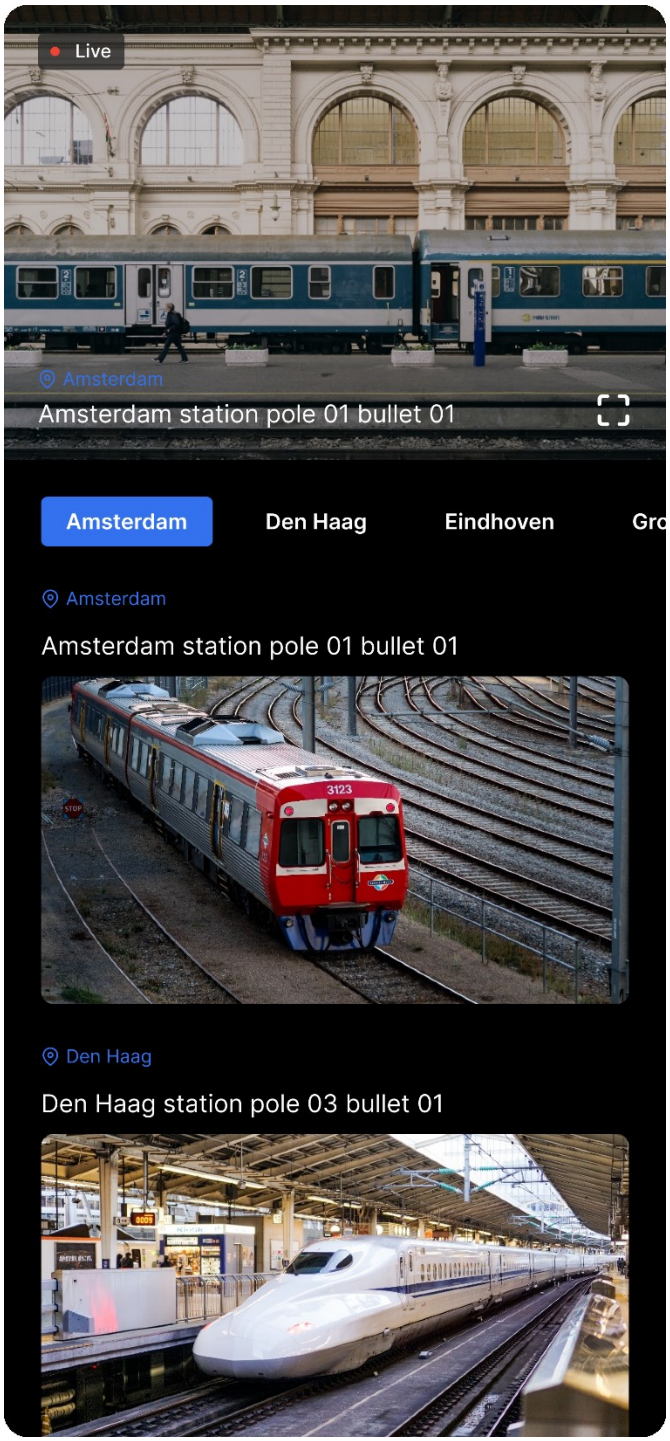

Figura 45: Ecrã da transmissão em directo das câmeras de segurança (móvel-pressionado) (Fonte: elaboração própria)

6.3.2.7 Perfil de utilizador

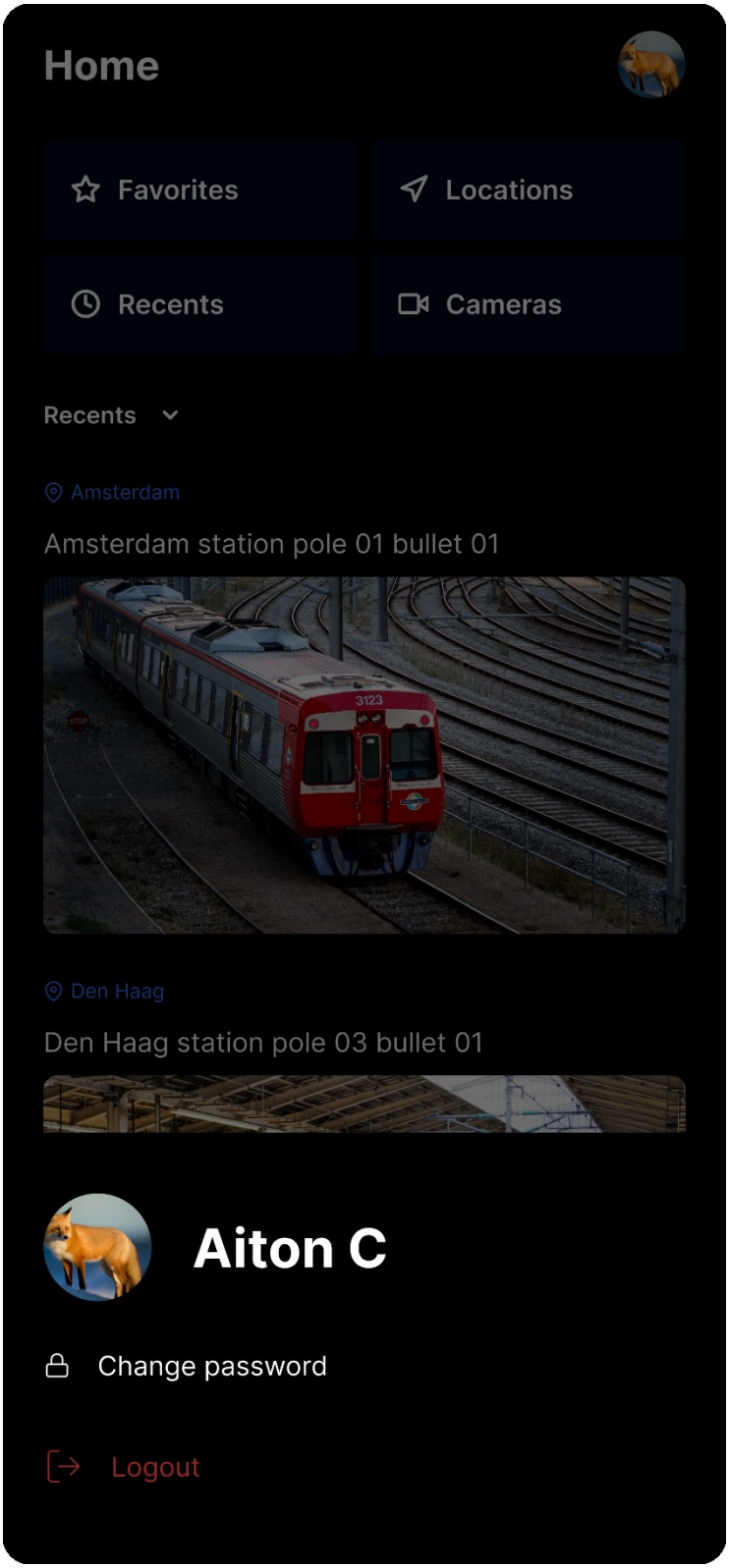

Figura 46: Ecrã do perfil de utilizador (móvel) (Fonte: elaboração própria)

6.4 Anexo 4 – Plano de actividades

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

PLANO DE ACTIVIDADES DE ESTÁGIO PROFISSIONAL

Estudante: Cumbi, Aiton António

Referência do tema:
Tema: Desenvolvimento de um sistema integrado e customizado de monitoramento de um circuito de televisão fechado distribuído para a ALTEL.
Instituição de estágio: ALTEL Soluções Globais de Comunicação

N°	**Actividade**	**Agosto**				**Setembro**				**Outubro**				**Novembro**			
		1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª
1.	**Instituição de estágio**																
1.1.	Reunião com o supervisor da instituição	X															

1.2.	Introdução e orientação na instituição de estágio (Ambientação)		X	X													
2.	**Desenvolvimento da aplicação para o acesso remoto as câmeras de segurança.**																
2.1.	Reunião com a organização cliente da instituição de estágio				X												
2.2.	Levantamentos dos requisitos de software				X												
2.3.	Criação do documento de requisitos de software (SRS)				X												
2.4.	Criação do "*statement of work*" para a actividade de desenvolvimento da aplicação.				X												
2.5.	Desenho da arquitectura de software da solução					X											
2.6.	Desenho das interfaces frontend móvel & Web					X	X										
2.7.	Desenvolvimento e testagem da API de backend				X	X	X	X	X								
2.8.	Desenvolvimento e testagem da interface frontend móvel						X	X									
2.9.	Desenvolvimento e testagem da interface frontend web								X	X	X	X	X				
2.10.	Criação e entrega da versão de testes da aplicação a organização												X				

2.11.	Correcções e patches da aplicação													X	X		
2.12.	Entrega da versão final da aplicação															X	
3.	**Elaboração do relatório de estágio profissional**																
3.1.	Introdução															X	
3.2.	Definição do problema				X												
3.3.	Hipóteses					X											
3.4.	Objectivos				X												
3.5.	Metodologia					X											
3.6.	Apresentação da instituição de estágio		X														
3.7.	Revisão de literatura						X	X									
3.8.	Caso de estudo (situação actual)					X											
3.9.	Proposta de soluções									X							
3.10.	Apresentação de resultados													X			
3.11.	Conclusões e recomendações														X		
3.12.	Referencias bibliográficas														X		
4.	Submissão do relatório																X
5.	Criação dos slides de apresentação															X	
6.	Apresentação e defesa																X

SUPERVISORES

	Nome	Assinatura
Da UEM	Eng. Rúben Manhiça	
Da Instituição	Eng. Frederico Muianga	

Maputo, aos____ de Agosto de 2023