

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Mateus Rocha Resende

**Sistema de Videomonitoramento Seguro com
Acesso Remoto Utilizando VPN, DNS
Dinâmico e Software Livre**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Mateus Rocha Resende

Sistema de Videomonitoramento Seguro com Acesso Remoto Utilizando VPN, DNS Dinâmico e Software Livre

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Luís Fernando Faina

Coorientador: João Henrique de Sousa Pereira

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Mateus Rocha Resende

Sistema de Videomonitoramento Seguro com Acesso Remoto Utilizando VPN, DNS Dinâmico e Software Livre

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 07 de maio de 2025:

Luís Fernando Faina

Orientador

João Henrique de Sousa Pereira

Coorientador

Paulo Rodolfo da Silva Leite Coelho

Convidado 1

Rafael Pasquini

Convidado 2

Uberlândia, Brasil

2025

Resumo

Fornecedores de sistemas de videomonitoramento modernos buscam oferecer dispositivos que facilitem a instalação e uso do produto para os consumidores. Porém, essa praticidade frequentemente resulta em vulnerabilidades de segurança inerentes, como configurações padrão inseguras e dependência de serviços em nuvem de terceiros, comprometendo a privacidade e o controle do usuário sobre os dados. Adicionalmente, a natureza proprietária de muitas soluções comerciais limita a customização e a integração, e o acesso remoto muitas vezes carece de mecanismos robustos de segurança. Este trabalho propõe e implementa um sistema de videomonitoramento seguro, que garante acesso remoto e controle pelo usuário, abordando as limitações de privacidade e segurança encontradas em soluções comerciais. A solução envolveu a integração do *hardware* disponível (Raspberry Pi, duas câmeras IP) com *software* livre, utilizando ZoneMinder como NVR para gerenciamento centralizado de vídeo, WireGuard para acesso remoto via VPN segura, e DNS Dinâmico (DuckDNS) para resolução de nomes de domínio associados a IPs públicos variáveis. A implementação resultou em um sistema funcional, permitiu o monitoramento e controle remoto seguro das câmeras em diferentes cenários de rede (local ou por VPN em rede externa). A análise comparativa destacou vantagens em controle de dados e privacidade frente a alternativas comerciais dependentes de servidores em nuvem. A solução cumpre os requisitos de segurança e controle, mantendo os dados sob domínio do usuário, ainda que o processo de configuração demande um nível de complexidade e trabalho significativamente superior ao observado em sistemas comerciais.

Palavras-chave: Videomonitoramento, Acesso Remoto, Segurança, Rede Virtual Privada, Câmera IP.

Lista de ilustrações

Figura 1 – Diagrama de um sistema NVR básico	12
Figura 2 – Diagrama de uma Rede VPN	13
Figura 3 – Diagrama de Rede do Sistema	18
Figura 4 – Controles PTZ e visualização de imagens da câmera AXIS	19
Figura 5 – Controles PTZ e Visualização de Imagens da Câmera Wanscam	20
Figura 6 – Ferramenta Raspberry Pi Imager	22
Figura 7 – Personalização da Instalação do Raspberry Pi OS	22
Figura 8 – Configuração da Câmera AXIS 214 PTZ no Zoneminder	24
Figura 9 – Configuração da Câmera Wanscam FR4020A2 no Zoneminder	24
Figura 10 – Interface Web do WireGuard VPN	26
Figura 11 – Painel de Gerenciamento de Domínios DuckDNS	27
Figura 12 – Interface do roteador para reservar endereço IP estático	28
Figura 13 – Diagrama do <i>Port Forwarding</i> e tabela NAT	30
Figura 14 – Interface do roteador para configurar o <i>Port Forwarding</i>	30
Figura 15 – Interface de configuração de VLAN no roteador	31
Figura 16 – Interface de configuração da <i>bridge (br1)</i> no roteador	32
Figura 17 – Configuração de acesso para o cliente WireGuard	33
Figura 18 – Etapas de configuração inicial do zmNinja	34
Figura 19 – Diagrama de conexão dos clientes ao servidor	35
Figura 20 – Perfil VPN WireGuard habilitado e conectado	36
Figura 21 – Visão geral das imagens das câmeras no zmNinja	36
Figura 22 – Visão individual e controle das câmeras no zmNinja	37
Figura 23 – Diagrama para conexão do cliente dentro da rede doméstica	37
Figura 24 – Lista de dispositivos encontrados na Rede Local	38
Figura 25 – Diagrama para conexão do cliente fora da rede doméstica sem VPN	38
Figura 26 – Tentativa de conexão ao Zoneminder pelo navegador sem sucesso	39
Figura 27 – Diagrama para conexão do cliente fora da rede doméstica com VPN	39
Figura 28 – Diagrama para conexão do cliente na rede da Universidade com VPN	40
Figura 29 – Registros de tentativa de conexão à VPN sem sucesso	40
Figura 30 – Diagrama da configuração das Câmeras IP Intelbras	41
Figura 31 – Visualização individual da câmera no aplicativo <i>Mibo Smart</i>	42
Figura 32 – Visão geral das imagens das câmeras disponíveis no aplicativo <i>Mibo Smart</i>	42
Figura 33 – Funcionalidades de videomonitoramento oferecidas por serviço pago	43
Figura 34 – Diagrama da conexão P2P em câmeras IP	45

Lista de tabelas

Tabela 1 – Características das Câmeras AXIS 214 PTZ e Wanscam FR4020A2 . .	20
Tabela 2 – Especificações do microcomputador Raspberry Pi 3B	21

Lista de abreviaturas e siglas

CFTV	Circuito Fechado de Televisão
DNS	<i>Domain Name System</i>
DDNS	<i>Dynamic DNS</i>
DVR	<i>Digital Video Recorder</i>
FTP	<i>File Transfer Protocol</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IPSec	<i>Internet Protocol Security</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
NAPT	<i>Network Address Port Translation</i>
NAT	<i>Network Address Translation</i>
NVR	<i>Network Video Recorder</i>
P2P	<i>Peer-to-peer</i>
PTZ	<i>Pan-Tilt-Zoom</i>
RTSP	<i>Real Time Streaming Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SSH	<i>Secure Shell</i>
UID	<i>Unique Identifier</i>
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>
VLAN	<i>Virtual Local Area Network</i>
VPN	<i>Virtual Private Network</i>
WAN	<i>Wide Area Network</i>

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.2	Organização desse Trabalho de Conclusão de Curso	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	Fundamentação Teórica	11
2.1.1	Câmeras IP <i>Standalone</i>	11
2.1.2	NVR (Network Video Recorder)	11
2.1.3	Redes VPN (Virtual Private Network)	12
2.1.4	DNS (Domain Name System)	13
2.1.5	Raspberry Pi	14
2.1.6	ZoneMinder	15
2.1.7	Docker	15
2.2	Trabalhos Correlatos	16
3	DESENVOLVIMENTO	18
3.1	Descrição Geral do Sistema de Monitoramento	18
3.2	Configuração das Câmeras PTZ IP	19
3.2.1	Câmera AXIS 214 PTZ	19
3.2.2	Câmera Wanscam FR4020A2	20
3.3	Configuração do Servidor	21
3.3.1	Configuração do Raspberry Pi	21
3.3.2	Configuração do Zoneminder	23
3.3.3	Configuração do WireGuard VPN	24
3.3.4	Configuração do DuckDNS	26
3.4	Configuração do Roteador	28
3.4.1	Alocação de Endereços IP Estáticos	28
3.4.2	Conceitos de NAT, NAPT e <i>Port Forwarding</i>	29
3.4.3	Configuração do <i>Port Forwarding</i>	29
3.4.4	Configuração opcional da VLAN (<i>Virtual Local Area Network</i>)	30
3.5	Configuração do Cliente	32
3.5.1	Configuração do Cliente Wireguard (VPN)	32
3.5.2	Configuração do Cliente zmNinja (ZoneMinder)	33
4	RESULTADOS	35
4.1	Operação Padrão do Sistema de Videomonitoramento	35

4.2	Videomonitoramento pela WAN ou pela LAN	37
4.2.1	Cliente dentro da Rede Doméstica	37
4.2.2	Cliente fora da Rede Doméstica sem VPN	38
4.2.3	Cliente fora da Rede Doméstica com VPN via Rede Celular	39
4.2.4	Cliente fora da Rede Doméstica com VPN via Rede UFU	40
4.3	Comparação a um Sistema Comercial de Videomonitoramento	41
4.3.1	Configuração e Funcionalidades	41
4.3.2	Protocolo Peer-to-Peer para Câmeras IP	43
4.3.3	Arquitetura do Sistema de Videomonitoramento Intelbras	44
5	CONCLUSÃO	46
5.1	Trabalhos Futuros	47
	REFERÊNCIAS	48

1 Introdução

Com o avanço tecnológico no mercado de videomonitoramento e da necessidade de recursos de segurança para residências ou estabelecimentos comerciais, cada vez mais consumidores buscam um sistema de câmeras com facilidade de instalação e acesso remoto pela Internet. Para atender essa demanda, empresas oferecem Câmeras IP atreladas a serviços em nuvem e aplicativos *web* que facilitam a configuração e uso desse equipamento por pessoas sem conhecimento técnico, além de exigir menor investimento quando comparado a Sistemas de Videomonitoramento tais como NVR (*Network Video Recorder*). No entanto, tais sistemas nem sempre garantem a segurança de acesso, pois, p.ex., Câmeras IP possuem vulnerabilidades e são configuradas de forma insegura por padrão dos fabricantes, aumentando o risco de ataques pela Internet e deixando ao usuário a responsabilidade pela configuração segura dos dispositivos (SILVA, 2018). Neste contexto, esse trabalho propõe um sistema seguro de acesso remoto às imagens de câmeras conectadas a uma rede doméstica ou organização de pequeno ou médio porte com controle de acesso de usuários, garantia de acesso restrito e seguro por VPN (*Virtual Private Network*) e maior segurança quanto ao repositório das imagens e/ou vídeos.

Do ponto de vista comercial, o ponto relevante neste trabalho é a combinação e junção de diferentes soluções de segurança e de videomonitoramento desenvolvidos por diferentes fabricantes para concepção de um sistema seguro de acesso remoto às imagens e vídeos de câmeras conectadas a uma rede doméstica ou organização de pequeno ou médio porte por meio da Rede Internet, permitindo que o usuário tenha segurança e garantia de acesso ao sistema e conteúdo gerado no sistema. Do ponto de vista acadêmico, o tema permite a combinação de diferentes tecnologias e soluções de código aberto para videomonitoramento por VPN por meio da Rede Internet e oferece ao discente o aprendizado dessas tecnologias bem como a proposição de como, por meio de sistemas de *software*, podem ser agrupadas formando um novo sistema.

O trabalho Liranzo e Hayajneh (2017) mostra a importância do tema ao destacar os problemas de segurança e privacidade presentes em Câmeras IP conectadas a serviços em nuvem, como a falta de criptografia e as vulnerabilidades presentes nessas câmeras, que deixam abertura para ataques que quebram a confidencialidade das imagens. Já os trabalhos Stürmer e Ahlert (2020) e Bravo (2015) indicam que é possível ter bons resultados na construção de um sistema de videomonitoramento de pequeno porte com o uso de *hardware* e *software* semelhantes aos que serão utilizados durante o trabalho.

1.1 Objetivos

Esse trabalho teve como objetivo propor a criação de um sistema de videomonitoramento de pequeno e médio porte para acesso de imagens de Câmeras IP em tempo real de forma remota com garantia de segurança e privacidade de seus usuários. Nesse contexto, descreve-se na sequência alguns dos objetivos específicos:

- permitir acesso ao sistema de videomonitoramento por meio de uma Rede VPN acessível somente por usuários devidamente autorizados;
- configurar e disponibilizar um serviço para controle e monitoramento das câmeras que possibilite utilizar funcionalidades das câmeras, como PTZ e visão noturna;
- garantir que a eventual mudança de IP público do roteador doméstico ou da empresa de pequeno ou médio porte não seja impeditivo de acesso para seus usuários.

1.2 Organização desse Trabalho de Conclusão de Curso

O Capítulo 2 apresenta a fundamentação teórica de conceitos e tecnologias essenciais para o entendimento do trabalho e trabalhos correlatos utilizados na pesquisa e construção do sistema. O Capítulo 3 detalha a organização dos componentes de *hardware* e *software* e o processo de configuração do sistema de videomonitoramento. O Capítulo 4 expõe os resultados obtidos com o sistema implementado, comparando-o com soluções comerciais e analisando seu desempenho em diferentes cenários de rede. Por fim, o Capítulo 5 apresenta a conclusão do trabalho, pontuações sobre o que foi realizado e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Esse capítulo apresenta a explicação de conceitos e tecnologias essenciais para o entendimento do trabalho. O capítulo apresenta também trabalhos correlatos que foram utilizados na pesquisa e construção do sistema.

2.1 Fundamentação Teórica

2.1.1 Câmeras IP *Standalone*

Câmeras IP são dispositivos de videomonitoramento que se conectam a uma Rede Local ou Internet, e possuem um servidor *web* interno para visualização das imagens e controle remoto por um usuário com acesso à rede (VIEIRA, 2023). Esses dispositivos possuem variados modelos, como Câmeras fixas ou PTZ (*Pan-Tilt-Zoom*), e funcionalidades, como: opções para diferentes resoluções de imagem, visão noturna e configurações para acesso remoto e segurança no aplicativo *web*.

Apesar da facilidade de instalação de Câmeras IP comuns e disponíveis no mercado, ainda existem preocupações quanto a segurança no seu uso no modo *standalone*. Neste contexto, destacam-se algumas preocupações: i) uso de senha padrão no painel de controle da Câmera; ii) vulnerabilidade a ataques de força bruta contra credenciais de login tendo por base dicionário de senhas; iii) vulnerabilidades associadas ao *firmware* decorrentes da não atualização dos mesmos; iv) serviço de segurança da Câmera não configurado ou configurado indevidamente (SILVA, 2018).

Ao comparar uma Câmera IP no modo *standalone* a Sistemas de Videomonitoramento do tipo NVR (*Network Video Recorder*), são notáveis as limitações: i) baixa capacidade de armazenamento de vídeo; ii) dificuldade de integração com alarmes e sistemas de controle de acesso; iii) ausência de um controle e gerenciamento centralizado para administrar um sistema com múltiplas câmeras.

2.1.2 NVR (*Network Video Recorder*)

Com o avanço da tecnologia em câmeras e em redes e a necessidade do uso de videomonitoramento para segurança residencial e empresarial, o uso de Câmeras IP em sistemas CFTV (*Circuito Fechado de Televisão*) e NVR se popularizou. Diferentemente das Câmeras IP, o NVR é um sistema de videomonitoramento com mais funcionalidades que permite: i) centralização do controle de um conjunto de Câmeras IP conectadas à mesma LAN (*Local Area Network*); ii) gravação do vídeo gerado pelas câmeras em um

disco rígido; iii) acesso remoto às gravações ([SAMSARA, 2021](#)).

Embora os NVRs ofereçam vantagens sobre Câmeras IP no modo *standalone* e Sistemas Analógicos como os DVRs (*Digital Video Recorder*), ainda existem lacunas em suas implementações: i) compatibilidade restrita a modelos específicos de Câmeras IP; ii) escalabilidade demanda alto investimento financeiro; iii) acesso remoto pode criar brechas de segurança e permitir acessos não autorizados. Diante de tais lacunas e vulnerabilidades, [Stürmer e Ahlert \(2020\)](#) demonstra que é viável a criação de um sistema que utiliza uma Câmera IP em conjunto com outras ferramentas de *hardware* e *software* como alternativa de baixo custo a um Sistema NVR Tradicional.

Um sistema NVR básico, exemplificado na Figura 1, é composto por uma câmera IP responsável pela captura e transmissão do vídeo para o gravador de vídeo em rede. O NVR realiza o armazenamento das imagens recebidas e é conectado por cabo Ethernet tanto com a câmera IP quanto com o roteador da rede. Essa configuração possibilita que as imagens gravadas sejam acessadas por dispositivos autorizados, seja na rede local ou de forma remota, conforme a política de acesso estabelecida ([LEARNCCTV, 2019](#)).

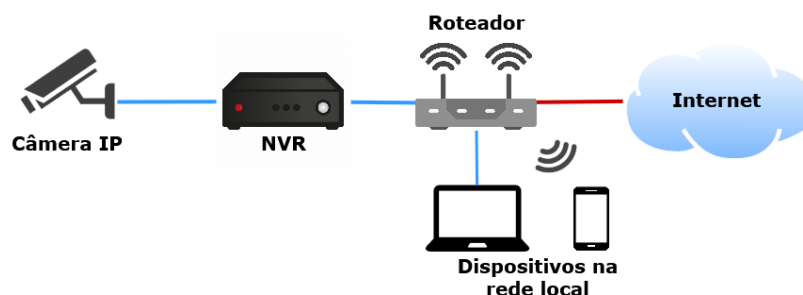


Figura 1 – Diagrama de um sistema NVR básico. Adaptado de [LearnCCTV \(2019\)](#)

2.1.3 Redes VPN (Virtual Private Network)

Uma Rede Privada Virtual (VPN) tem por objetivo interligar redes corporativas ou domésticas de forma segura através da Internet, além de fornecer acesso seguro a usuários remotos ([KUROSE; ROSS, 2005](#)). Para isso, as VPNs utilizam protocolos de segurança como o IPSec (Internet Protocol Security) para criar túneis criptografados entre um dispositivo e um servidor garantindo a confidencialidade, integridade e autenticidade dos dados transmitidos. Além da segurança proporcionada por Redes VPN, uma outra vantagem refere-se ao custo de implantação, ou seja, o serviço substitui a necessidade de instalação e manutenção de uma estrutura de uma infraestrutura de rede física para interconexão dos dispositivos.

Diante da necessidade da segurança e proteção de dados trocados entre diferentes redes, é natural que haja o aumento do uso de VPNs. Nesse contexto, os avanços são

frequentes, tais como: i) algoritmos de criptografia mais eficientes; ii) desenvolvimento de protocolos VPN com melhor desempenho, p.ex., WireGuard. De acordo com [Donenfeld \(2017\)](#), o WireGuard propõe a facilidade de configuração, simplicidade e rapidez de conexão, implementação do estado-da-arte da criptografia e um protocolo *connection-less*: após o *handshake* inicial não é necessário o estabelecimento de uma conexão.

A Figura 2 apresenta um exemplo de rede VPN baseada no protocolo WireGuard. Nesse cenário, um computador remoto, configurado como *peer*, conecta-se à rede interna de uma organização por meio da Internet, estabelecendo um túnel criptografado com o servidor WireGuard localizado na rede local. A comunicação entre o *peer* remoto e o servidor VPN é realizada através de interfaces virtuais, utilizando criptografia ponta a ponta para assegurar a confidencialidade dos dados transmitidos. Uma vez autenticado e conectado, o dispositivo remoto pode acessar recursos e interagir com outros dispositivos da rede interna, como se estivesse fisicamente presente no ambiente local, respeitando as regras e permissões configuradas na VPN ([SAUKKONEN, 2020](#)).

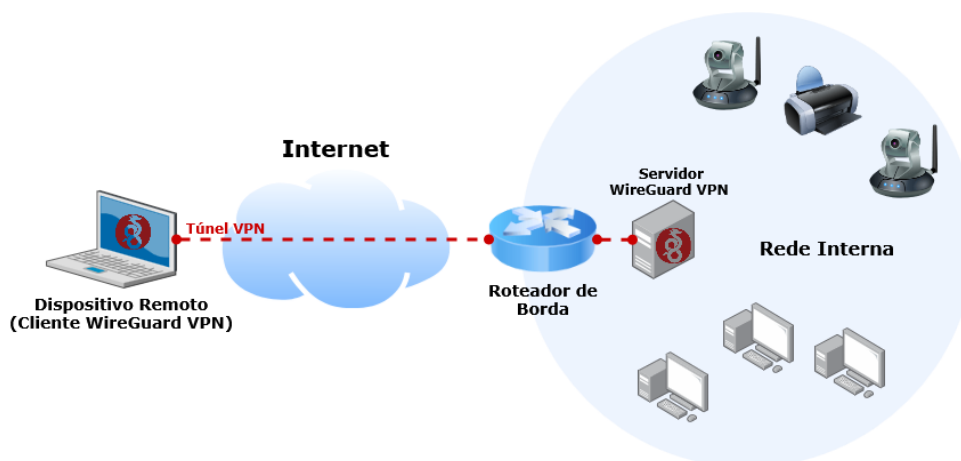


Figura 2 – Diagrama de uma Rede VPN.

2.1.4 DNS (Domain Name System)

O Sistema de Nomes de Domínio (DNS) é um sistema distribuído globalmente que traduz nomes de domínio legíveis por humanos (p.ex.: `facom.ufu.br`) em endereços IP numéricos necessários para a comunicação na rede de computadores que constitui a Internet ([MOCKAPETRIS, 1987](#)). Desde sua criação na década de 1980, o DNS tornou-se um componente fundamental para a arquitetura da Internet, substituindo o sistema inicial de arquivos de *hosts* centralizados (`HOSTS.TXT`) ([BONASTRE; VEÀ, 2019](#)). O sistema distribuído e hierárquico do DNS permite maior escalabilidade, confiabilidade e resiliência, contribuindo para a expansão da Internet.

O DNS Dinâmico (*Dynamic DNS* ou DDNS) é uma extensão do DNS, formalizada por [Vixie et al. \(1997\)](#), e introduzida para atender as necessidades crescentes de dina-

mesmo na associação entre nomes de domínio e endereços IP, acompanhando o aumento da mobilidade dos dispositivos e a popularização de conexões residenciais com endereçamento dinâmico ([PADMANABHAN, 2020](#)). O DDNS permite a atualização automática e em tempo real de registros de mapeamento de domínio para endereços IP. Sua operação fundamenta-se na interação entre um agente cliente, usualmente localizado no *host* ou *gateway* de rede interna, responsável por detectar alterações no endereço IP público dinamicamente atribuído, e um servidor DDNS. Mediante autenticação, o cliente requisita a atualização do registro DNS correspondente, garantindo a associação persistente entre o nome de domínio e o endereço IP atual.

Com o passar dos anos, foram incorporadas ao DNS diversas melhorias voltadas à segurança e à privacidade das comunicações. Entre essas inovações, destacam-se mecanismos como o DNSSEC (*Domain Name System Security Extensions*) ([BAU; MITCHELL, 2010](#)), voltado para proteger a integridade e a autenticidade das informações fornecidas pelo DNS, bem como a utilização de protocolos como DNS sobre HTTPS (DoH) ([HOFFMAN; MCMANUS, 2018](#)) e DNS sobre TLS (DoT) ([HU et al., 2016](#)), que promovem a criptografia do tráfego entre clientes e servidores DNS. No contexto do DNS Dinâmico, a popularização de provedores como DynDNS, No-IP e DuckDNS possibilitou o acesso remoto simplificado a dispositivos residenciais e servidores de pequeno porte, permitindo que serviços como câmeras IP, servidores *web* domésticos e sistemas de automação sejam acessíveis mesmo diante de endereços IP dinâmicos ([ALHAMMADI et al., 2017](#)).

2.1.5 Raspberry Pi

O Raspberry Pi é um microcomputador de baixo custo, alto desempenho e propósito geral, construído utilizando a arquitetura *Arm* e compatível com sistemas operacionais Linux ([NUTTALL, 2021](#)). O objetivo inicial da criação dessa plataforma, em 2012 pela Raspberry Pi Foundation, foi de incentivar a educação e aprendizado sobre ciências da computação e o uso desse computador para pesquisa e desenvolvimento de tecnologias digitais ([RASPBERRYPI, 2012](#)).

Ao longo dos anos, a plataforma do Raspberry Pi tem recebido suporte oficial de *software* e *hardware* pela "Raspberry Pi Foundation" ([FOUNDATION, 2024](#)). Neste trabalho propõe-se a utilização da versão Raspberry Pi 3B em conjunto com o Sistema Operacional Oficial da Raspberry Pi Foundation (Raspbian OS), que por sua vez, é baseado no Linux. Esta versão do Raspberry Pi está equipada com Interface Ethernet, Wi-Fi e Bluetooth; uma entrada para Cartão MicroSD; uma saída HDMI; quatro portas USB 2.0; 1GB de Memória RAM e um processador *Arm* 64-bit de quatro núcleos.

Por ser um mini-computador de baixo custo, o Raspberry Pi é bastante utilizado em projetos que exigem baixo investimento, p.ex., em trabalhos em que se demonstra os benefícios do uso dessa plataforma no ambiente de aprendizado e desenvolvimento de

sistemas no contexto de IoT (Internet of Things) ([KURKOVSKY; WILLIAMS, 2017](#)). Um outro exemplo de uso do Raspberry Pi refere-se ao desenvolvimento de um sistema de monitoramento com reconhecimento de fala e face ([CASON; COSTA, 2021](#)).

2.1.6 ZoneMinder

O ZoneMinder é um conjunto de aplicações de *software* livre projetado para fornecer uma solução completa de vigilância por vídeo. Sua função principal é capturar, analisar, gravar e monitorar fluxos de vídeo provenientes de câmeras de segurança IP ou analógicas conectadas a um sistema operacional baseado em Linux. Essa ferramenta atua como um sistema NVR (Network Video Recorder) ou DVR (Digital Video Recorder) baseado em *software* ([MURÇA; SANTOS, 2020](#)).

Uma de suas características centrais é a compatibilidade com uma vasta gama de câmeras, sem limitar o usuário a modelos proprietários, permitindo o uso de dispositivos com diferentes funcionalidades, como visão noturna, alta resolução ou *Pan-Tilt-Zoom*. Ele funciona como um *broker* que abstrai as especificidades de cada dispositivo, suportando protocolos de transmissão de vídeo como RTSP e HTTP para a integração e gerenciamento centralizado através de uma interface web completa ([ZONEMINDER, 2023](#)). Além disso, oferece APIs para integração com sistemas de terceiros e aplicativos móveis dedicados, como o zmNinja, para acesso e controle remotos ([MELLO; EUZÉBIO, 2013](#)).

O ZoneMinder inclui funcionalidades avançadas como detecção de movimento configurável através de zonas de interesse com sensibilidades distintas por zona, controle PTZ para câmeras suportadas, gravação contínua ou baseada em eventos, e reprodução de gravações com estatísticas de evento. A plataforma permite que o usuário mantenha controle total sobre seus dados, com armazenamento local. A comunidade ativa e as integrações de terceiros permitem estender suas capacidades, incluindo reconhecimento de placas de automóveis e detecção de objetos ou pessoas baseada em Inteligência Artificial. Sua arquitetura modular com processos independentes para captura e análise de vídeo visa alta performance e redundância ([DESHMUKH; MIHOVSKA; PRASAD, 2012](#)).

2.1.7 Docker

O Docker é uma plataforma de código aberto que automatiza a implantação de aplicações dentro de contêineres de *software*, proporcionando portabilidade, isolamento e escalabilidade. Os contêineres são unidades padronizadas que encapsulam aplicações e todas as suas dependências, permitindo que sejam executadas de maneira consistente em diferentes ambientes computacionais ([MERKEL, 2014](#)). Essa abordagem difere das máquinas virtuais tradicionais, uma vez que os contêineres compartilham o *kernel* do sistema hospedeiro, resultando em menor uso de recursos computacionais e inicialização

mais rápida.

O conceito de contêinerização antecede o Docker, remontando a tecnologias como *chroot* (1979), *FreeBSD Jails* (2000) e *Linux Containers* (2008) (COSTA; SANTOS, 2021). No entanto, o lançamento do Docker em 2013 marcou um ponto de avanço significativo ao simplificar o processo de criação, distribuição e execução de contêineres, além de oferecer uma interface de fácil utilização e integração com plataformas de gerenciamento, como o *Kubernetes*. Desde então, a adoção do Docker cresceu exponencialmente tanto no setor empresarial quanto em ambientes acadêmicos, impulsionando práticas de *DevOps* e a modernização de arquiteturas de *software* (EMMANNI, 2023).

Desde a formalização da *Open Container Initiative* (OCI) em 2015, o ecossistema de contêineres Docker tem experimentado avanços significativos em padronização, segurança e orquestração (OCI, 2025). A OCI estabeleceu especificações abertas para formatos de imagem e tempo de execução de contêineres, promovendo interoperabilidade entre diferentes ferramentas e plataformas. No âmbito da segurança, mecanismos como *AppArmor* e *SELinux* foram integrados para reforçar o isolamento de processos e restringir permissões, mitigando riscos associados à execução de aplicações em contêineres (VRABEC, 2020). Com a evolução do Docker e necessidade de padronização nos ambientes de desenvolvimento de *software*, o artigo de Dayo (2021) investiga as vantagens no uso de contêineres em comparação às máquinas virtuais, apresentando resultados favoráveis ao Docker quanto à performance e à simplicidade de implantação.

2.2 Trabalhos Correlatos

Stürmer e Ahlert (2020) apresentam e analisam os resultados do desenvolvimento de um sistema de videomonitoramento CFTV (circuito fechado de televisão) de baixo custo, com gerenciamento centralizado e armazenamento remoto das informações. Diante desses objetivos, os autores conduzem uma pesquisa bibliográfica sobre os procedimentos técnicos para elaborar o sistema de videomonitoramento com as especificações adequadas.

Após o levantamento de especificações do sistema, houveram tentativas de implantação de sistemas com diferentes configurações e dispositivos para atender às demandas e limitações do trabalho, para isso, utilizaram-se os seguintes equipamentos e aplicativos: i) um microcomputador Raspberry Pi 3 para armazenamento e monitoramento de imagens recebidas de uma rede remota; ii) um microcomputador Orange Pi One na rede remota para enviar as imagens da câmera IP à rede de armazenamento; iii) uma Câmera IP Intelbras conectada à rede remota; iv) dois roteadores Mikrotik configurados conectar a rede remota à rede de armazenamento por meio de um serviço VPN; v) o *software* Zoneminder instalado nos dois microcomputadores para análise, gerenciamento e monitoramento das imagens da câmera IP. Como resultado, observou-se que é possível construir um sistema

de videomonitoramento eficiente de pequeno porte e baixo custo. Portanto, este trabalho é relevante por possuir objetivos em comum ao do projeto em relação ao conjunto de *hardware* e *software* que será utilizado durante o desenvolvimento do sistema.

A monografia de [Silva \(2018\)](#) investiga câmeras IP isoladas em uma rede doméstica, fora de um sistema completo de videomonitoramento: descreve o funcionamento do dispositivo, especifica requisitos de segurança da informação, identifica vulnerabilidades e propõe soluções para mitigar riscos. Para isso, o autor realiza uma análise teórica, onde apresenta os riscos de conectar câmeras IP à Internet, com exemplos de ataques em grande escala a câmeras IP isoladas sem controle de segurança adequado, tornando-as vulneráveis a *malwares* como o BASHLITE, que acessa os dispositivos por telnet utilizando força bruta ou credenciais padrões dos fabricantes. Além disso, o autor utiliza uma câmera IP da marca Hikvision, configurada de acordo com as sugestões de segurança propostas em seu próprio trabalho, para realizar testes de ataque de força bruta e ataque com uso da vulnerabilidade *Hikvision Password Reset* para acesso sem conhecimento das credenciais.

Durante o estudo e teste de possíveis ameaças, [Silva \(2018\)](#) demonstra a importância do processo de configuração da câmera IP para minimizar riscos de vulnerabilidades e ataques, assim como as diretrizes de segurança recomendadas, por exemplo: i) desabilitar recursos não utilizados, como FTP, DDNS, P2P e SMTP; ii) limitar o acesso da câmera IP à rede local, ou seja, não abrir portas no roteador para acesso direto via Internet; iii) utilizar senhas fortes e diferentes de outros serviços; iv) manter o *firmware* da câmera IP atualizado. O projeto compartilha objetivos com este trabalho por utilizar os resultados obtidos para aprimorar a segurança de acesso e da informação aplicando a configuração apropriada de câmeras IP no sistema de videomonitoramento que será desenvolvido.

O trabalho de [Liranzo e Hayajneh \(2017\)](#) evidencia as questões de segurança e falta de privacidade relacionadas às câmeras IP baseadas em nuvem instaladas em ambientes residenciais e aponta recomendações sobre a proteção desses dispositivos IoT. Com esse objetivo, o estudo inclui uma revisão da literatura relacionada a câmeras IP, privacidade e segurança de dados em nuvem, além de realizar testes de segurança experimentais utilizando possíveis brechas para vazamento de dados dos usuários do serviço em nuvem, como: i) vulnerabilidades no dispositivo IoT; ii) interceptação da troca de dados "em-trânsito"; iii) senhas fracas; iv) *firmware* desatualizado.

Diante de tais vulnerabilidades, o trabalho destaca as principais falhas de segurança da câmera IP testada: transmissão de credenciais sem criptografia ao se conectar ao dispositivo e a existência de uma conta *backdoor* imposta pelo fabricante, que permite acesso total às informações e ressalta a facilidade com que esses dispositivos podem ser comprometidos. O trabalho de [Liranzo e Hayajneh \(2017\)](#) se relaciona ao projeto proposto por evidenciar a importância do controle e segurança da transmissão e armazenamento de dados em um sistema de videomonitoramento.

3 Desenvolvimento

Neste capítulo, é apresentada uma visão geral do trabalho, as especificações técnicas dos dispositivos utilizados e uma descrição detalhada de cada etapa do desenvolvimento do sistema.

3.1 Descrição Geral do Sistema de Monitoramento

Para o desenvolvimento do sistema, propõe-se a utilização e combinação de diferentes tecnologias de *hardware* e *software*. Dentre os componentes de *hardware*, destacam-se: i) 01 (um) minicomputador Raspberry Pi 3B - componente principal utilizado para instalação e configuração dos programas ([NUTTALL, 2021](#)); ii) 02 (duas) Câmeras IP *Pan Tilt Zoom* (PTZ) para videomonitoramento; e iii) 01 (um) Roteador no qual todos os dispositivos serão conectados e estarão na mesma Rede Local.

Dentre os componentes de *software*, destacam-se: i) Protocolo VPN (*Virtual Private Network*) por meio do WireGuard ([DONENFELD, 2017](#)) para acesso à rede na qual o sistema está inserido; ii) Serviço de DNS Dinâmico [DuckDNS \(2024\)](#) para permitir associar o Endereço IP público atual da Rede Doméstica a um subdomínio fixo; iii) Aplicativo Web para acesso às imagens e vídeos armazenados em tempo real das câmeras e iv) Sistema Operacional "Raspberry Pi OS" para ser utilizado no minicomputador.

A Figura 3 ilustra as interconexões dos componentes principais do sistema. As câmeras e o Raspberry Pi estão conectados ao roteador doméstico via cabo Ethernet, o roteador está conectado à Internet e um cliente em uma rede externa está conectado ao Raspberry Pi utilizando a conexão VPN.

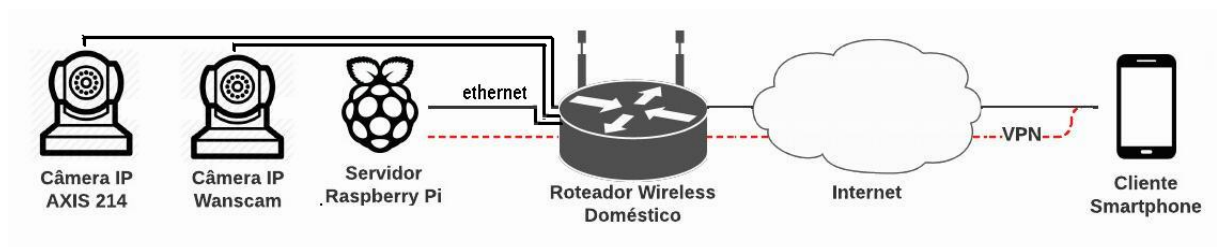


Figura 3 – Diagrama de Rede do Sistema.

3.2 Configuração das Câmeras PTZ IP

Inicialmente, as duas câmeras, de modelos diferentes, estão instaladas em um local que oferece visão ampla da área a ser monitorada, conectadas ao roteador via cabo Ethernet e à energia por uma fonte de alimentação.

O primeiro passo da configuração consiste em restaurar as definições padrão de fábrica, o que pode ser feito pressionando um botão físico em cada uma das câmeras. Isso é necessário para remover configurações anteriores, por exemplo, autenticação e atribuição de IP, permitindo atribuição de uma nova senha de administrador, bem como obtenção de um endereço IP por meio do protocolo DHCP.

As próximas subseções descrevem detalhes específicos de cada câmera e ajustes necessários para integração com o *software* de monitoramento.

3.2.1 Câmera AXIS 214 PTZ

A Câmera AXIS 214 PTZ (*Pan-Tilt-Zoom*) (AXIS, 2021) dispõe de uma Interface Web que permite alterar configurações de segurança, acesso, rede e vídeo, assim como visualizar a imagem em tempo real, bem como controles PTZ (Fig. 4).

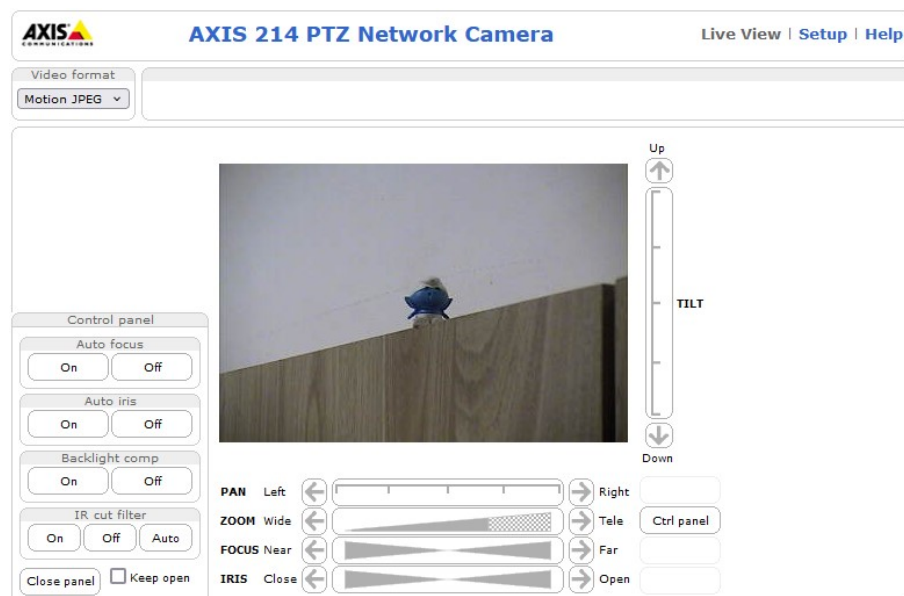


Figura 4 – Controles PTZ e Visualização de Imagens da Câmera AXIS.

Após a configuração inicial de credenciais para login no Aplicativo Web, procede-se com as seguintes alterações: i) diminuição da resolução da imagem para 352x240 para reduzir o custo computacional de processamento de imagem pelo Raspberry Pi; ii) obtenção do IP da câmera, IP do servidor DNS, IP do servidor NTP a partir do DHCP; iii) habilitação do *Real Time Streaming Protocol* (RTSP) para carregamento do vídeo pelo *software* de monitoramento; iv) desabilitação dos protocolos SOCKS (*SOCK*et *Secure*),

SMTP (*Simple Mail Transfer Protocol*), SNMP (*Simple Network Management Protocol*) e UPnP (*Universal Plug and Play*) e v) desabilitação do acesso anônimo (sem credenciais) à interface da câmera.

Durante a configuração da câmera, é essencial que o protocolo selecionado na câmera esteja entre o leque de protocolos contemplados no Zoneminder. Considerando diferentes câmeras no ambiente, com suporte a diferentes protocolos, o Zoneminder funciona como um *broker* para controle das câmeras.

A Tab. 1 detalha algumas características relevantes da Câmera AXIS 214 PTZ, como pode ser percebido há uma série de protocolos nativos da câmera, sendo alguns habilitados e outros desabilitados.

AXIS 214 PTZ	Wanscam FR4020A2
PTZ: $\pm 170^\circ$ Pan, -30° a 90° Tilt, 18x Zoom Resolução 576p (704x576), 30fps Ethernet 100Mb/s Protocolos IPv4/v6, HTTP, HTTPS, FTP, SMTP, UPnP, DNS, DynDNS, RTSP, RTP, TCP, UDP, DHCP	PTZ: 270° Pan, 90° Tilt Resolução 480p (640x480), 25 fps Ethernet 100Mb/s, Wireless LAN Protocolos IPv4, TCP/IP, HTTP, ICMP, DHCP, FTP, UPnP, SMTP

Tabela 1 – Características das Câmeras AXIS 214 PTZ e Wanscam FR4020A2.

3.2.2 Câmera Wanscam FR4020A2

A Câmera Wanscam FR4020A2, assim como a Câmera AXIS, dispõe de uma Interface Web que permite alterar configurações de segurança, acesso, rede, vídeo, visualizar a imagem em tempo real e possui controles PTZ (Fig. 5).

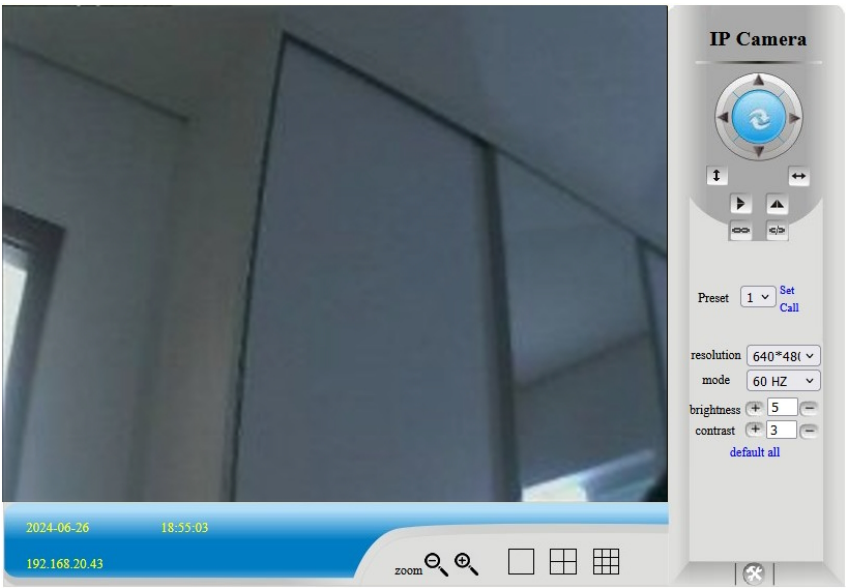


Figura 5 – Controles PTZ e Visualização de Imagens da Câmera Wanscam.

Na Tab. 1 listam-se as especificações da Câmera Wanscam FR4020A2, que possui uma resolução máxima menor e menos protocolos nativos em relação ao outro modelo. Se destaca a ausência do protocolo RTSP, sendo necessário utilizar requisições HTTP como alternativa para carregamento de vídeo pelo *software* de monitoramento Zoneminder, que atua como um *broker* para o controle das câmeras ao contemplar suporte para diferentes protocolos de requisição de imagem.

Com as credenciais de login definidas no primeiro acesso à Interface Web, configura-se a câmera com os seguintes parâmetros: i) resolução de imagem 640x480; ii) DHCP habilitado para obtenção do IP da câmera; iii) desabilitados os protocolos SMTP (*Simple Mail Transfer Protocol*) e UPnP (*Universal Plug and Play*) e iv) conexão Wireless desabilitada, uma vez que a conexão Ethernet é mais estável e viável no ambiente escolhido.

3.3 Configuração do Servidor

A configuração do servidor é centralizada no Raspberry Pi, que atua como *host* para os principais *softwares* do sistema, assumindo as tarefas de um NVR. O servidor é fundamental para o gerenciamento das imagens das câmeras, bem como para garantir o acesso seguro e remoto ao sistema. As próximas subseções descrevem a instalação do *host* e os ajustes de todos os componentes de *software*.

3.3.1 Configuração do Raspberry Pi

O microcomputador Raspberry Pi 3B está instalado próximo ao roteador, conectado à Rede via cabo Ethernet e à energia por uma fonte de alimentação 5V, 2.5A. Um Cartão microSD com 32GB de capacidade de armazenamento é inserido no Raspberry Pi após a instalação do sistema operacional.

A Tab. 2 apresenta as principais características de *hardware* do Raspberry Pi 3B, nota-se o baixo requisito de energia, uma interface Ethernet 100Mb/s que comporta a transmissão das imagens das câmeras e as portas USB caso seja necessário o uso de dispositivos de armazenamento externo.

Raspberry Pi 3B
Processador Quad Core 1.2GHz Broadcom 64-bit
1GB Memória RAM
Ethernet 100Mb/s
Saída de vídeo HDMI
Quatro portas USB
Uma entrada para cartão Micro SD
Fonte de alimentação 5V 2.5A

Tabela 2 – Especificações do microcomputador Raspberry Pi 3B.

Primeiramente, instala-se o sistema operacional Raspberry Pi OS no cartão microSD. Para isso, a Raspberry Pi Foundation fornece a ferramenta multiplataforma ”*Raspberry Pi Imager*” (HOLLINGWORTH, 2020), que por sua vez é instalada em um computador Windows, macOS ou Linux. Ao abrir o programa e conectar o cartão microSD no computador, são selecionadas as seguintes opções: i) dispositivo: Raspberry Pi 3; ii) sistema operacional: Raspberry Pi OS 64-bit (derivado da distribuição Linux *Debian Bookworm*) e iii) armazenamento: cartão microSD 32GB (Fig. 6).



Figura 6 – Ferramenta Raspberry Pi Imager.

Ainda na ferramenta de instalação, configura-se um *hostname*, usuário e senha, fuso horário, além da habilitação do protocolo SSH (*Secure Shell*) (Fig. 7). Essa etapa é necessária para permitir o acesso remoto ao Raspberry Pi na primeira inicialização do sistema operacional, adiantando o processo de configuração inicial.

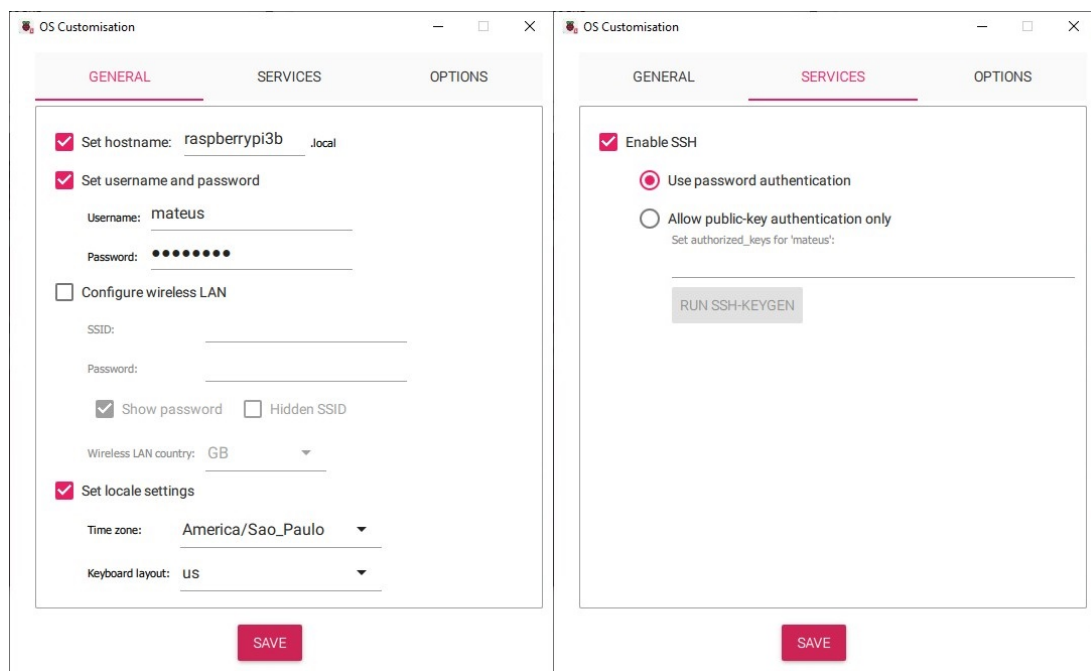


Figura 7 – Personalização da Instalação do Raspberry Pi OS.

Após gravar a imagem do sistema operacional no cartão microSD, insere-se o cartão no Raspberry Pi e liga-se o dispositivo. Depois de inicializado, utiliza-se outro dispositivo

para abrir uma conexão SSH com o servidor para instalação e configuração dos *softwares* no sistema. Para isso, executa-se o comando: "ssh mateus@raspberrypi3b.local".

Para facilitar a gestão das aplicações utilizadas no servidor, instala-se o Docker seguindo as instruções oficiais¹ para um sistema Raspberry Pi OS 64-bit. O Docker é uma ferramenta que simplifica a instalação das aplicações em conjunto com suas dependências, bibliotecas e configurações de forma isolada e consistente em pacotes chamados *containers* (MERKEL, 2014). Nas próximas subseções são apresentados detalhes sobre os dois *containers* utilizados no sistema, Zoneminder e *WireGuard Easy*.

3.3.2 Configuração do Zoneminder

O primeiro passo para instalação do Zoneminder foi a escolha da imagem Docker "nardo86/zoneminder"², por ser compatível com a arquitetura ARM64 do Raspberry Pi 3B e conter uma versão recente do Zoneminder (1.36). Em seguida, cria-se um arquivo "compose.yaml" com as variáveis de ambiente a seguir:

```
1 services:
2   zoneminder:
3     # Imagem Docker utilizada:
4     image: nardo86/zoneminder
5     # Nome do container:
6     container_name: zoneminder
7     ports:
8     # Porta utilizada para acessar a Interface Web:
9     - "443:443"
10    # Variáveis de ambiente:
11    environment:
12    # Fuso horário:
13    - "TZ=America/Sao_Paulo"
14    # Gerar um certificado "selfsigned" para a página web:
15    - "SELF_SIGNED=1"
16    # Nome de domínio para acesso ao Zoneminder:
17    - "FQDN=localhost"
18    # Mapeamento dos arquivos de configuração:
19    volumes:
20    - "./ZoneMinder/config:/config"
21    - "./ZoneMinder/zmcache:/var/cache/zoneminder"
22    - "./Swag/etc/letsencrypt/live:/sslcert/live"
23    - "./Swag/etc/letsencrypt/archive:/sslcert/archive"
24    # Quantidade de RAM alocada para videomonitoramento:
25    shm_size: '256mb'
26    restart: unless-stopped
```

¹ Instruções para instalação do Docker: <<https://docs.docker.com/engine/install/debian/>>

² Documentação para "nardo86/zoneminder":<<https://github.com/Nardo86/zm-docker>>

Destaca-se as seguintes configurações: i) definição da porta 443 para acesso à interface *web*; ii) uso de certificado SSL em um domínio *localhost* (não é definido um nome de domínio público, o tráfego não será exposto à Internet ao utilizar o VPN) e iii) alocação de 256MB de memória RAM para videomonitoramento. Após a criação desse arquivo, utiliza-se o comando "`docker compose up -d`" para inicializar o *container*.

Na Interface *Web* do Zoneminder, disponível no endereço IP local do Raspberry Pi ("`https://192.168.20.26/zm/`"), adiciona-se as configurações das Câmeras IP de acordo com as recomendações disponíveis na Zoneminder Wiki. As Figuras 8 e 9 apresentam os parâmetros para transmissão de vídeo das Câmeras AXIS 214 PTZ e Wanscam FR4020A2.

Figura 8 – Configuração da Câmera AXIS 214 PTZ no Zoneminder.

Figura 9 – Configuração da Câmera Wanscam FR4020A2 no Zoneminder.

Nota-se a diferença no uso do protocolo em cada câmera, o RTSP para a Câmera AXIS 214 PTZ e o HTTP para a Wanscam FR4020A2.

3.3.3 Configuração do WireGuard VPN

O próximo *software* a ser instalado no sistema é o WireGuard VPN. A imagem Docker "`ghcr.io/wg-easy`"³ foi escolhida por simplificar a configuração das interfaces de rede do servidor e permite administrar credenciais de acesso ao servidor através de uma interface web. A imagem conta com a instalação do WireGuard VPN e uma Interface Web para administradores.

Primeiramente, cria-se um diretório "`wg-easy`" para os arquivos de configuração e credenciais de acesso gerados pela aplicação. Em seguida, cria-se um arquivo "`compose.yaml`" com as variáveis de ambiente a seguir:

³ Documentação para "`wg-easy`": <https://wg-easy.github.io/wg-easy/Pre-release/getting-started/>

```
1 services:
2   wg-easy:
3     # Imagem Docker utilizada:
4     image: ghcr.io/wg-easy/wg-easy
5     # Nome do container:
6     container_name: wg-easy
7     # Variáveis de ambiente:
8     environment:
9       - LANG=en
10      # Nome de domínio para acesso ao WireGuard:
11      - WG_HOST=mattcc-ipcam.duckdns.org
12      # Senha para acessar a Interface Web:
13      - PASSWORD=wgadmintcc
14      # Porta para a Interface Web:
15      - PORT=51821
16      # Porta para o WireGuard VPN:
17      - WG_PORT=51820
18      # Mapeamento dos arquivos de configuração do WireGuard:
19      volumes:
20        - ./wg-easy/:/etc/wireguard
21      ports:
22        # Porta UDP utilizada pelo VPN WireGuard:
23        - "51820:51820/udp"
24        # Porta TCP utilizada para acessar a Interface Web:
25        - "51821:51821/tcp"
26      # Recursos necessários para configurar a rede:
27      cap_add:
28        - NET_ADMIN
29        - SYS_MODULE
30      # Parâmetros necessários para o funcionamento do WireGuard:
31      sysctls:
32        - net.ipv4.conf.all.src_valid_mark=1
33        - net.ipv4.ip_forward=1
34      restart: unless-stopped
```

Nesse arquivo, destaca-se as seguintes configurações: i) "WG_HOST" contém o nome de domínio público correspondente ao IP público do servidor registrado no serviço de DNS dinâmico DuckDNS; ii) "PASSWORD" é a senha da Interface *Web* e pode ser trocada por um *hash bcrypt* no parâmetro "PASSWORD_HASH" e iii) *Port Forwarding* é configurado para porta 51820 na configuração do Roteador. Após a criação desse arquivo, utiliza-se o comando "docker compose up -d" para inicializar o *container*.

Na Interface *Web* da configuração do WireGuard, disponível no endereço local "https://192.168.20.26:51821", é possível listar, criar, editar, deletar, habilitar e desabilitar clientes VPN. Ao criar um novo cliente, um arquivo de configuração com credenciais de acesso é gerado para esse usuário.

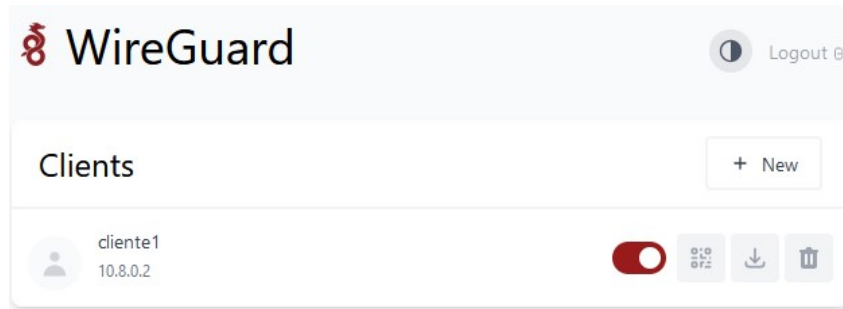


Figura 10 – Interface Web do WireGuard VPN.

A Figura 10 apresenta o "cliente1", criado e habilitado na Interface Web. Em seguida, o arquivo de exemplo "cliente1.conf" gerado pelo WireGuard com os dados necessários para conexão ao servidor (utilizado posteriormente na configuração do cliente):

```
1 [Interface]
2 PrivateKey = KIEwiGcoyKJZ0wvtr+8LBkdXXXXx1iZ6+ez6UbxxxbGk=
3 Address = 10.8.0.2/24
4 DNS = 1.1.1.1
5
6 [Peer]
7 PublicKey = ypaGND4UMihaXXXXXwyJjBSVcuMfTKRxxxUBf6/7RWA=
8 PresharedKey = Bb2XZ09xzNXXXXXf8fg0ZJGUnTAIxxxvP1tKtPefrTs=
9 AllowedIPs = 0.0.0.0/0, ::/0
10 PersistentKeepalive = 0
11 Endpoint = mattcc-ipcam.duckdns.org:51820
```

3.3.4 Configuração do DuckDNS

Ao distribuir endereços IP aos seus clientes, provedores de Internet atribuem endereços IP públicos dinâmicos, que podem mudar periodicamente de acordo com a política de rotatividade de IPs de cada provedor (PADMANABHAN, 2020). Isso é um obstáculo para acessar recursos da Rede Local doméstica remotamente, já que o endereço IP público associado ao roteador doméstico pode variar, dificultando o acesso.

O DuckDNS resolve esse problema ao fornecer um nome de domínio estático. Nesse trabalho, utiliza-se o domínio "mattcc-ipcam.duckdns.org", que está associado ao endereço IP dinâmico da Rede Local doméstica. Quando o IP público do roteador de Internet muda, é possível atualizar automaticamente a associação do domínio ao novo IP. Isso permite o acesso aos serviços hospedados na rede doméstica usando o mesmo registro de domínio, independentemente das mudanças de IP.

A partir disso, configura-se um nome de domínio no serviço DuckDNS para associar o Endereço IP público atual da Rede Doméstica a um subdomínio estático. Primeiramente, cria-se uma conta no *website* "duckdns.org". Após o cadastro e *login*, deve-se configu-

rar um subdomínio na página de gerenciamento de domínios. No campo *sub domain* foi inserido o nome de domínio "mattcc-ipcam" para registrar o URL (Uniform Resource Locator) "mattcc-ipcam.duckdns.org" (Fig. 11).



Figura 11 – Painel de Gerenciamento de Domínios DuckDNS.

Para atualizar periodicamente o endereço IP público associado ao domínio registrado, utiliza-se um *script* que é executado no Raspberry Pi, localizado na rede doméstica, cujo roteador possui o IP público associado ao domínio. Inicialmente, cria-se o arquivo "mattcc-ipcam.sh" no diretório "duckdns" com o seguinte conteúdo:

```
1 echo url="https://www.duckdns.org/update?domains=mattcc-ipcam&token=[token]&ip="
  |
2 curl -k -o ~/duckdns/duck.log -K -
```

Na URL de atualização do DuckDNS, o parâmetro `&token=[token]` requer o *token* de autenticação exclusivo do usuário, obtido na interface de gerenciamento do DuckDNS. Este *token* funciona como mecanismo de autorização, assegurando que apenas o proprietário possa efetuar modificações no registro DNS associado. Por sua vez, o parâmetro `&ip="` determina o endereço IP a ser vinculado ao domínio. Ao omitir deliberadamente um valor para este parâmetro (deixando-o vazio após o sinal de igual), ativa-se a funcionalidade de detecção automática de IP por parte do servidor DuckDNS. Neste cenário, o servidor receptor analisa a conexão de rede da requisição HTTP recebida (originada pela ferramenta *curl*) para identificar o endereço IP público de origem do cliente e, subsequentemente, utiliza este endereço IP detectado para realizar a atualização do registro DNS correspondente ao parâmetro `?domains=mattcc-ipcam`.

Em seguida, para tornar o arquivo executável, executa-se o comando:

```
chmod +x mattcc-ipcam.sh
```

Finalmente, para automatizar a atualização do endereço IP, utiliza-se a ferramenta *crontab*, que permite agendar a execução periódica de *scripts*. Para isso, deve-se adicionar a seguinte linha no arquivo de tarefas *cron*, após executar o comando "crontab -e":

```
0 */1 * * * ~/duckdns/mattcc-ipcam.sh >/dev/null 2>&1
```

3.4 Configuração do Roteador

A configuração do roteador envolve duas etapas: i) alocação de endereços IP estáticos locais para os três dispositivos pertencentes ao sistema: duas Câmeras IP e um Raspberry Pi e ii) configuração do *Port Forwarding* para permitir acesso externo, pela Internet, ao servidor VPN WireGuard, hospedado no Raspberry Pi conectado à Rede Local doméstica. As próximas subseções descrevem essas etapas, além de apresentar os conceitos de NAT e *Port Forwarding* para justificar a necessidade dessa configuração.

3.4.1 Alocação de Endereços IP Estáticos

A atribuição de endereços IP fixos a dispositivos na rede local, seja por meio da reserva de IP via DHCP ou pela configuração manual de um IP estático, é uma prática utilizada quando necessita-se localizar consistentemente determinados dispositivos por meio de seus endereços IP. Essa abordagem assegura que o endereço IP atribuído a um dispositivo permaneça inalterado, mesmo após reinicializações ou após o término do tempo de concessão (*lease*) do DHCP, evitando potenciais conflitos ou falhas de comunicação em serviços que dependam de um mapeamento estável entre endereço IP e dispositivo.

Na Figura 12, observa-se a interface *web* do roteador doméstico onde associa-se os Endereços MAC (Media Access Control) dos três dispositivos aos endereços IP correspondentes: i) "192.168.20.12" para a Câmera AXIS; ii) "192.168.20.43" para a Câmera Wanscam e iii) "192.168.20.26" para o Raspberry Pi.

MAC Address	Static arp	IP Address ▲	IPTraffic	Hostname
00:40:8C:7C:73:9B		192.168.20.12		axis-00408c7c739b
B8:27:EB:5D:38:68		192.168.20.26		raspberrypi3b
00:D6:FB:00:7C:A1		192.168.20.43		ipcam_00D6FB007CA1
00:00:00:00:00:00	☐	192.168.20.10	☐	
00:00:00:00:00:00				

Figura 12 – Interface do roteador para reservar endereço IP estático.

Os endereços IP estáticos locais definidos no roteador são necessários no sistema para diferentes objetivos: i) o Zoneminder, hospedado no Raspberry Pi, encontra as Câmeras IP na Rede Local utilizando o endereço IP privado; ii) o *Port Forwarding* configurado no roteador para o Raspberry Pi necessita que o dispositivo tenha um IP estático e privado e iii) os clientes WireGuard VPN e Zoneminder acessam o servidor dessas aplicações pelo endereço IP do Raspberry Pi.

3.4.2 Conceitos de NAT, NAPT e *Port Forwarding*

O *Network Address Port Translation* (NAPT), é uma extensão do conceito de *Network Address Translation* (NAT) que permite a tradução simultânea de múltiplos endereços IP privados e suas respectivas portas de origem para um único endereço IP público, diferenciando cada conexão por meio do número da porta (EGEVANG; SRISURESH, 2001). Esse método é amplamente empregado em redes corporativas e residenciais para otimizar a utilização de endereços IPv4 públicos, que são recursos escassos, ao possibilitar que diversos dispositivos em uma rede local compartilhem um único endereço IP externo para acessar a Internet (JUNIPER, 2023). O NAPT atua modificando o endereço IP e a porta, mantendo uma tabela dinâmica de associações entre os endereços/portas internos e externos, o que assegura que as respostas dos servidores remotos sejam corretamente direcionadas ao dispositivo de origem na rede privada.

Além de viabilizar a economia de endereços IP públicos, o NAPT também desempenha um papel relevante no reforço da segurança da rede interna, uma vez que dificulta o acesso direto de *hosts* externos aos dispositivos da rede local, já que as conexões devem ser iniciadas de dentro da rede privada (SMITH; HUNT, 2002). No entanto, essa abordagem introduz desafios para aplicações que dependem de conexões de entrada, como servidores hospedados em uma rede interna, exigindo configurações adicionais como o *Port Forwarding* para garantir o encaminhamento das conexões de entrada.

O *Port Forwarding* ou "Encaminhamento de Portas" é uma aplicação do NAT que redireciona requisições de conexão de uma combinação de endereço IP e porta a outro IP e porta, onde os pacotes passam por um dispositivo intermediário na rede (*gateway*): um *firewall* ou um roteador. O uso dessa aplicação permite que dispositivos remotos, externos à rede local, acessem um serviço ou computador presente em uma rede local privada, anteriormente não acessível pela ausência de uma tabela NAPT no roteador de borda da rede local que traduza os endereços IP e portas para estabelecer uma conexão (VERMA; KASHYAP; JHA, 2018).

3.4.3 Configuração do *Port Forwarding*

Na Figura 13, observa-se um diagrama correspondente ao sistema criado para este trabalho, onde está presente o roteador e sua tabela NAPT utilizada para traduzir e mapear a porta e endereço IP público da rede externa para a rede interna privada, onde: i) o IP público está associado ao domínio "`mattcc-ipcam.duckdns.org`"; ii) o IP interno é estático e está associado ao Raspberry Pi e iii) a porta "51820" está aberta tanto para a rede interna privada quanto à rede externa pública e está associada ao VPN WireGuard.

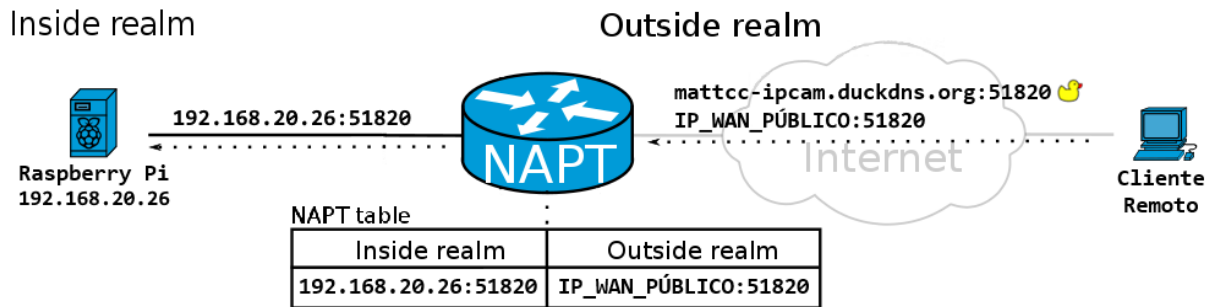


Figura 13 – *Port Forwarding* e tabela NAPT. Fonte: adaptado de Bakni (2020)

A partir da organização dos dispositivos na rede interna e do diagrama (Fig. 13) correspondentes ao sistema de monitoramento, configura-se o *Port Forwarding* para popular a tabela NAPT do roteador. Para isso, cria-se uma regra na interface *web* do roteador (Fig. 14), onde especifica-se o mapeamento para a aplicação VPN WireGuard no servidor: i) a porta WAN "51820" e porta LAN "51820"; ii) o endereço IP interno "192.168.20.26" do Raspberry Pi e iii) o protocolo UDP.

Port Forwarding

On	Protocol	Src Address	Ext Ports	Int Port	Int Address	Description
☒	UDP		51820	51820	192.168.20.26	wireguard

Buttons: Delete, OK, Cancel

Figura 14 – Interface do roteador para configurar o *Port Forwarding*.

3.4.4 Configuração opcional da VLAN (*Virtual Local Area Network*)

A segmentação da rede local através de VLANs representa uma camada adicional de segurança e organização, embora opcional, é recomendável para o sistema de videomonitoramento proposto. A configuração de VLANs permite isolar o tráfego entre as câmeras IP e o NVR, neste caso o Raspberry Pi, do restante da rede local, mitigando riscos de acesso não autorizado e interferências.

A configuração de VLANs no roteador doméstico é tipicamente realizada através de sua interface *web* de gerenciamento. Um exemplo de mapeamento dessas associações a portas físicas Ethernet é ilustrado na Figura 16. Nesta interface, são definidos múltiplos identificadores para as redes virtuais: especifica-se se o tráfego para uma determinada porta deve ser *tagged* (quando o dispositivo é membro de múltiplas VLANs), ou *untagged* (para portas de acesso que pertencem a uma única rede). Permite-se, ainda, associar as VLANs ou portas físicas a diferentes *bridges* de rede. Por exemplo, uma rede virtual específica (nesse caso, a "VLAN 3" na "bridge 1") pode ser criada exclusivamente para as câmeras IP e uma das interfaces virtuais do Raspberry Pi. As portas físicas do roteador

às quais as câmeras estão conectadas seriam, então, configuradas como portas *untagged* para este domínio isolado.

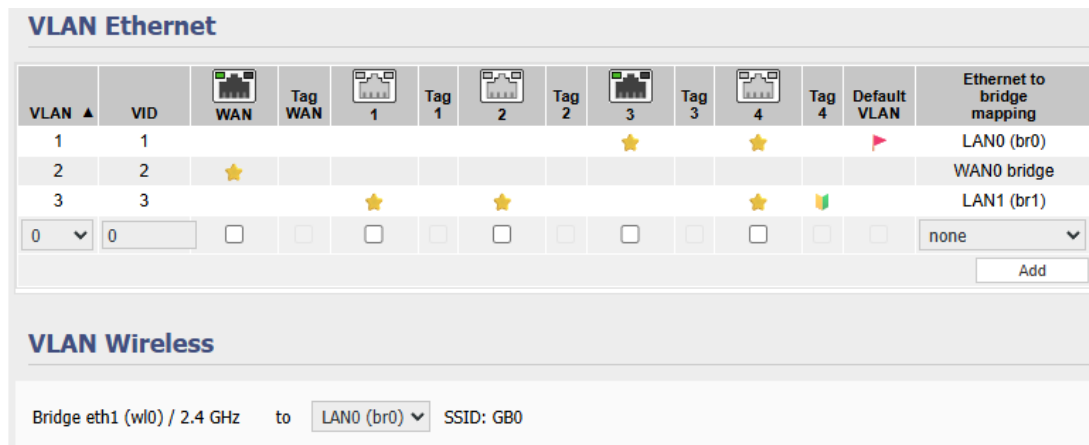


Figura 15 – Interface de configuração de VLAN no roteador.

A principal vantagem de segurança ao utilizar uma rede virtual para as câmeras é o isolamento do tráfego. Dispositivos na outra VLAN, como computadores de usuários ou dispositivos de visitantes conectados à rede Wi-Fi, associados à "VLAN 1" na "bridge 0" (Fig. 15), não conseguem comunicar-se diretamente com as câmeras IP na "VLAN 3", porém conseguem se conectar ao Raspberry Pi, associado às duas VLANs. Isso impede que usuários não autorizados ou dispositivos comprometidos na rede local possam descobrir, acessar ou tentar explorar vulnerabilidades nos equipamentos de videomonitoramento.

Portanto, para o correto funcionamento e gerenciamento do sistema, o servidor Raspberry Pi necessita operar em duas VLANs simultaneamente: i) VLAN das Câmeras IP (VLAN 3), que permite comunicação direta com as câmeras, captura dos fluxos de vídeo e envio de comandos de controle; e ii) VLAN de Gerenciamento/Usuário (VLAN 1), que permite acesso dos usuários à interface do ZoneMinder na rede local.

Essa participação em múltiplas redes virtuais é implementada mediante configuração da porta Ethernet do roteador, à qual o Raspberry Pi se conecta, como uma porta de *trunking*. Essa porta transporta o tráfego encapsulado de ambas as VLANs, devidamente identificadas por *tags*. Em seguida, o sistema operacional do Raspberry Pi é ajustado para reconhecer essas marcações e associá-las a interfaces de rede virtuais distintas para cada VLAN (por exemplo, `eth0.1` e `eth0.3` para uma interface física `eth0`).

É fundamental que, dentro de suas respectivas VLANs, os dispositivos possuam endereços IP configurados adequadamente. A Figura 16 ilustra a criação da interface *bridge* (`br1`), utilizada na VLAN isolada das câmeras IP. Esta *bridge* abrange endereços IP para os dispositivos clientes na faixa de "192.168.30.2" a "192.168.30.10". As etapas de configuração para o endereçamento IP estático no roteador, conforme descritas anteriormente, devem ser realizadas para este novo intervalo de endereços.

Bridge	STP	IP Address	Netmask	DHCP	IP Range (first/last)	Lease Time (mins)
br0	Disabled	192.168.20.1	255.255.255.0	Enabled	192.168.20.2 - 50	1440
1 ▼	☐	192.168.30.1	255.255.255.0	☒	192.168.30.2 192.168.30.10	1440

Figura 16 – Interface de configuração da *bridge* (br1) no roteador.

A configuração de VLANs, embora adicione complexidade à instalação, eleva significativamente a configuração de segurança do sistema de videomonitoramento, alinhando-se com as boas práticas de segmentação de rede.

3.5 Configuração do Cliente

Após a instalação dos dispositivos e serviços, pertencentes ao sistema de videomonitoramento na Rede Doméstica, configura-se os clientes para permitir acesso às imagens fornecidas pelo Zoneminder e conexão remota à Rede Local por meio de uma VPN. Para isso, instala-se dois aplicativos nos dispositivos clientes: i) WireGuard (cliente), que permite adicionar as credenciais de acesso ao VPN WireGuard (servidor) e iniciar a conexão segura ao servidor; ii) zmNinja, que oferece uma interface gráfica para gerenciar e monitorar fontes de imagem do servidor Zoneminder.

Os aplicativos WireGuard⁴ e zmNinja⁵ são compatíveis com diversos sistemas operacionais, como p. ex. Android, iOS e Windows. No desenvolvimento do presente trabalho, utilizou-se um dispositivo móvel Android versão 13 para testes, porém nota-se que outros dispositivos com diferentes sistemas operacionais também seguem etapas de configuração semelhantes às detalhadas nas próximas subseções.

3.5.1 Configuração do Cliente Wireguard (VPN)

O aplicativo WireGuard permite adicionar as credenciais de acesso ao servidor VPN localizado na rede doméstica e iniciar uma conexão segura. Para isso, o aplicativo é instalado por meio da plataforma de distribuição *Google Play Store* e configurado a partir do perfil de cliente criado na etapa de configuração do servidor WireGuard, onde utiliza-se um *QR Code* para preencher todos os campos de forma automática.

Na Figura 17, observa-se as informações necessárias para se conectar ao servidor VPN: i) chave privada do cliente; ii) chave pública do cliente; iii) endereço "10.8.0.2/24" da interface VPN; iv) servidor DNS; v) chave pública do servidor; vi) chave privada do servidor e vii) endereço público do servidor "mattcc-ipcam.duckdns.org:51820".

⁴ Sistemas operacionais compatíveis com o WireGuard: <<https://www.wireguard.com/install/>>

⁵ Sistemas operacionais compatíveis com o zmNinja: <<https://zmninja.zoneminder.com>>

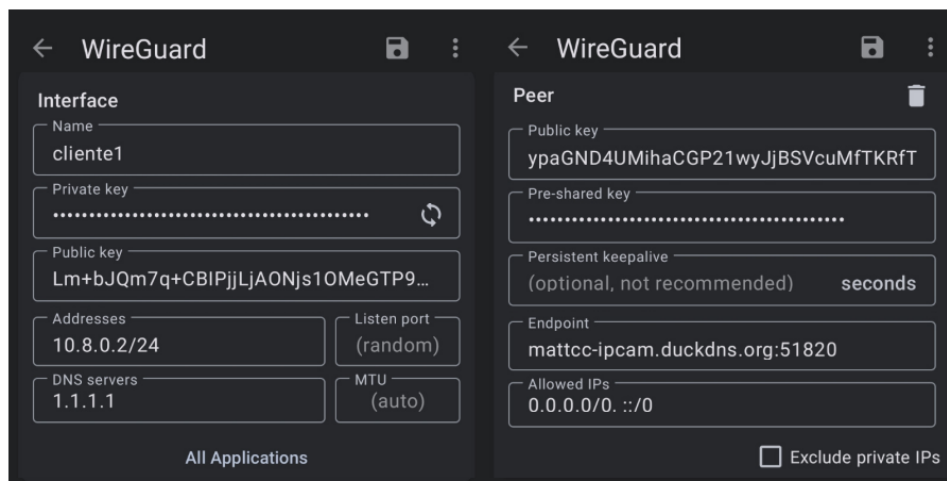


Figura 17 – Configuração de acesso para o cliente WireGuard.

Após realizar essa configuração, utilizando o mesmo aplicativo WireGuard, o cliente conecta-se ao servidor VPN para acessar o serviço ZoneMinder por meio do aplicativo zmNinja. Esse processo de autenticação e estabelecimento do túnel criptografado garante a confidencialidade dos dados trafegados entre o cliente remoto e a rede doméstica, protegendo contra interceptações e ataques de terceiros. Além disso, a utilização de chaves criptográficas exclusivas para cada dispositivo assegura que apenas clientes autorizados possam estabelecer a conexão. Dessa forma, a arquitetura empregada possibilita o acesso remoto seguro a recursos internos da rede, preservando a privacidade e a segurança durante a transferência de dados.

3.5.2 Configuração do Cliente zmNinja (ZoneMinder)

O aplicativo zmNinja oferece uma interface gráfica para gerenciar e monitorar fontes de imagem do servidor Zoneminder, além de suportar o controle PTZ individual das câmeras. Esse aplicativo é disponibilizado gratuitamente por meio do repositório oficial do projeto no *GitHub*⁶ ou adquirido na versão paga na plataforma *Google Play Store*.

Durante o primeiro acesso, na ausência de uma configuração prévia, o usuário deve realizar um procedimento inicial composto por três etapas principais (Fig. 18) para estabelecer a conexão com o servidor: i) fornecer a URL correspondente ao Zoneminder, p. ex., "<https://192.168.20.26/zm>"; ii) inserir credenciais de autenticação; e iii) utilizar o mecanismo de auto-detecção dos endereços vinculados ao Zoneminder, procedimento que valida o acesso e identifica as funcionalidades disponíveis do serviço.

⁶ Disponível em: <<https://github.com/ZoneMinder/zmNinja/releases>>

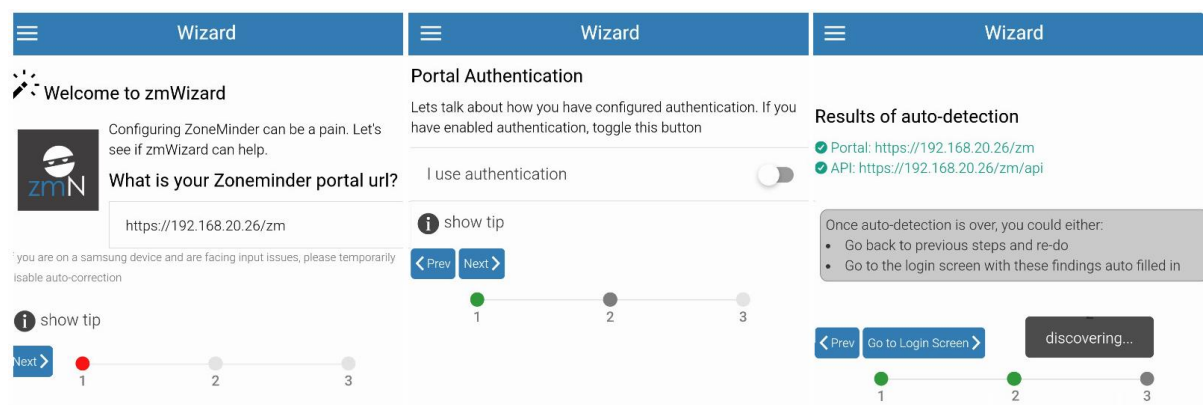


Figura 18 – Etapas de configuração inicial do zmNinja.

Após a conclusão dessas etapas de configuração, o cliente encontra-se devidamente apto para acessar e operar o sistema de videomonitoramento de forma remota, usufruindo de uma solução segura e centralizada para o acompanhamento e controle de ambientes monitorados.

4 Resultados

Neste capítulo, são apresentados e analisados os resultados obtidos a partir da implementação do Sistema de Videomonitoramento. Inicialmente, descreve-se a arquitetura do sistema, destacando a forma de interconexão entre seus componentes e quais funcionalidades específicas de um NVR foram implementadas. Em seguida, discute-se o comportamento do cliente em diferentes cenários de conectividade de rede, bem como realiza-se uma comparação detalhada entre o sistema proposto e uma solução comercial convencional de videomonitoramento, ressaltando vantagens e limitações observadas.

4.1 Operação Padrão do Sistema de Videomonitoramento

Para utilizar o sistema de videomonitoramento descrito e implementado, é necessário um cliente conectado ao servidor, ou seja, cliente na mesma rede do servidor ou na rede externa conectado via VPN ao servidor. Para acessar as imagens de videomonitoramento, utiliza-se o aplicativo zmNinja no cliente.

Para conectar-se ao servidor Zoneminder, o cliente realiza uma conexão direta se estiver na mesma rede local, sem a necessidade de estabelecer a conexão VPN no aplicativo WireGuard. Em uma rede externa, com conexão à Internet, o cliente habilita o perfil VPN correspondente ao servidor para acessar o sistema de videomonitoramento (Fig. 19). Essa conexão à VPN criptografa todo o tráfego de dados e vídeo e garante que somente dispositivos externos autorizados tenham acesso ao servidor.

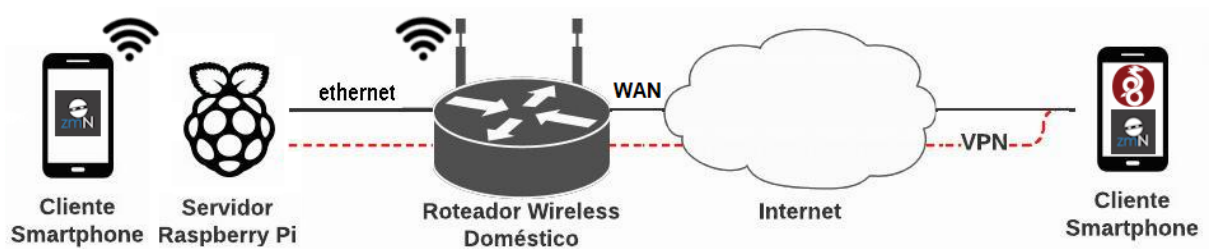


Figura 19 – Diagrama de conexão dos clientes ao servidor.

A Figura 20 ilustra a interface do aplicativo cliente WireGuard em um dispositivo móvel, demonstrando visualmente o momento em que o perfil de conexão VPN configurado para o servidor doméstico foi ativado pelo usuário e o status de "conectado" foi alcançado. Isso confirma o estabelecimento bem-sucedido do túnel seguro, pré-requisito para o acesso remoto ao ZoneMinder a partir de redes externas.

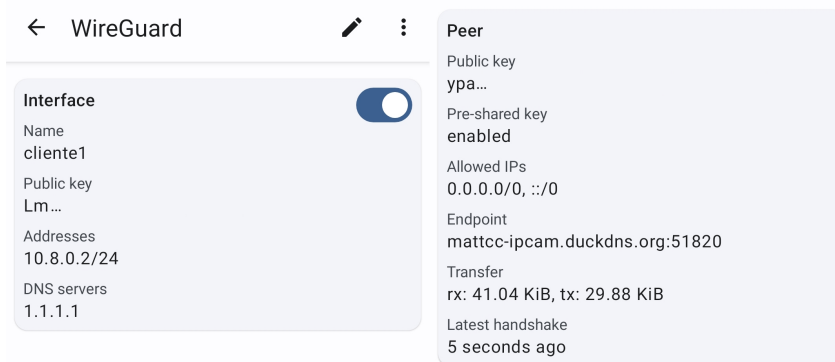


Figura 20 – Perfil VPN WireGuard habilitado e conectado.

Em seguida, para visualizar as imagens e controlar o posicionamento das câmeras, o cliente utiliza o aplicativo zmNinja. Na Figura 21, observa-se a tela principal com a visão geral das imagens de todas as câmeras; é possível reorganizar a ordem das fontes de vídeo, ocultá-las ou abrir o modo de visualização individual ao clicar sobre a imagem da câmera desejada. O layout dinâmico e personalizável proporciona maior flexibilidade na gestão do sistema de videomonitoramento, facilitando a supervisão simultânea de múltiplos ambientes e a rápida identificação de eventos de interesse.

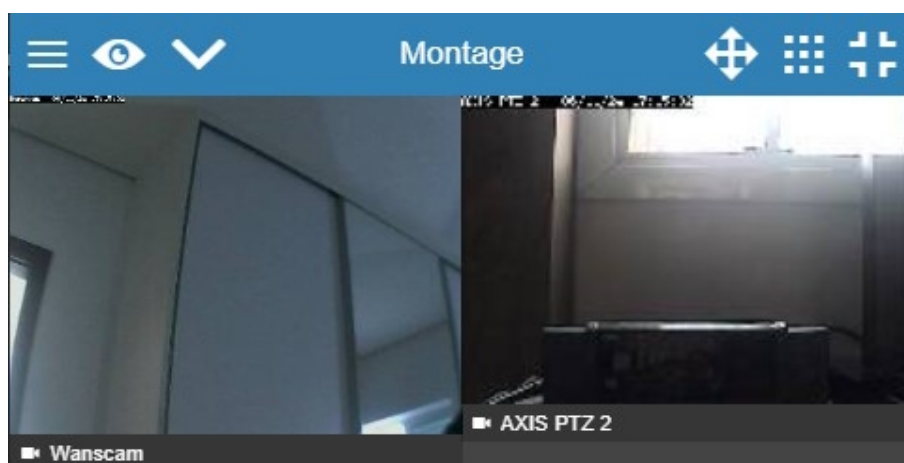


Figura 21 – Visão geral das imagens das câmeras no zmNinja.

Ao selecionar uma câmera para visualização individual, o aplicativo permite o controle PTZ nativo das câmeras: rotação, inclinação e zoom diretamente da interface (Fig. 22). Também oferece funções para capturar a imagem atual e salvar até 16 configurações de posicionamento de câmera. Além dessas funcionalidades, a interface individual possui recursos de navegação rápida entre diferentes câmeras e o acionamento de comandos predefinidos de movimentação que contribuem para uma operação eficiente e personalizada do sistema de monitoramento, atendendo tanto a demandas de supervisão contínua quanto àquelas relacionadas a eventos específicos ou áreas de interesse.

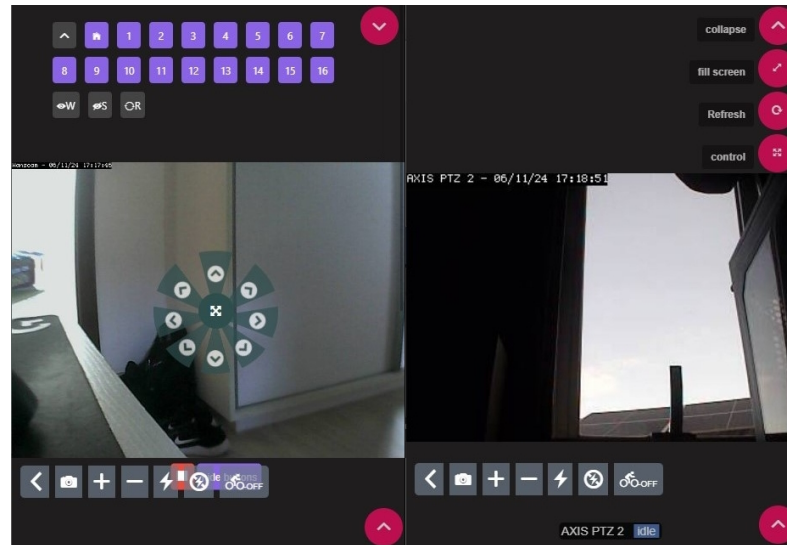


Figura 22 – Visão individual e controle das câmeras no zmNinja.

4.2 Videomonitoramento pela WAN ou pela LAN

Foram analisados quatro cenários com condições diferentes de conectividade entre o cliente e o servidor: i) cliente dentro da rede doméstica; ii) cliente fora da rede doméstica sem VPN; iii) cliente fora da rede doméstica com VPN na rede de dados móveis; iv) cliente fora da rede doméstica com VPN na rede da UFU. Nas próximas subseções, detalha-se cada particularidade dos cenários.

4.2.1 Cliente dentro da Rede Doméstica

No primeiro cenário, o cliente consegue se conectar diretamente ao servidor Zoneminder presente na mesma rede doméstica, uma vez que não existe um *firewall* configurado no servidor para bloquear acesso por dispositivos locais (Fig. 23). Esse ambiente é considerado o mais estável para acesso ao sistema de videomonitoramento por ter menos pontos de falha (não requer uma conexão com a Internet, nem o uso da VPN), além da limitação do uso de banda de rede ser definida pela capacidade dos equipamentos utilizados (portas Ethernet 100MB ou Wi-Fi) e não pelo plano de Internet contratado.

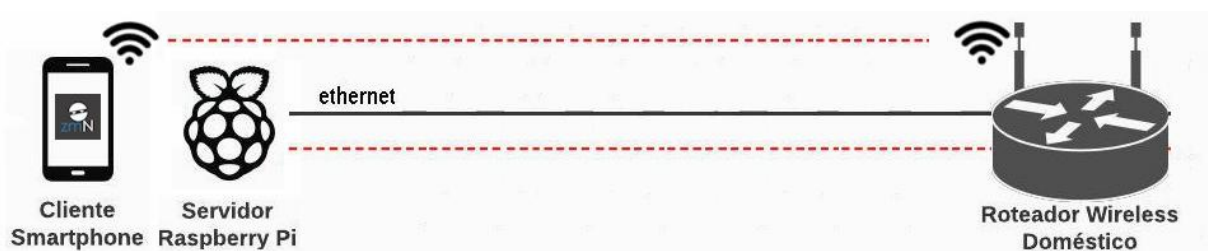


Figura 23 – Diagrama para conexão do cliente dentro da rede doméstica.

Para identificar o endereço IP do servidor na rede local, utiliza-se o aplicativo Fing¹ no dispositivo móvel (cliente), que realiza uma varredura dos dispositivos conectados à mesma rede e lista os endereços IP correspondentes. A utilização de ferramentas de varredura de rede, como o Fing, é fundamental em redes locais nas quais não se dispõe de uma listagem formal dos dispositivos conectados ou quando os endereços IP são atribuídos de forma dinâmica pelo servidor DHCP do roteador. No contexto deste sistema, embora os endereços IP dos dispositivos principais sejam conhecidos, a varredura com o Fing serve como uma etapa adicional de verificação, garantindo que o cliente identifique com precisão o endereço IP do servidor Zoneminder, evitando possíveis equívocos de endereçamento na rede. Na Figura 24, observa-se os endereços IP encontrados na varredura: i) do roteador "192.168.20.1", ii) do cliente "192.168.20.25" e iii) do servidor "192.168.20.26".

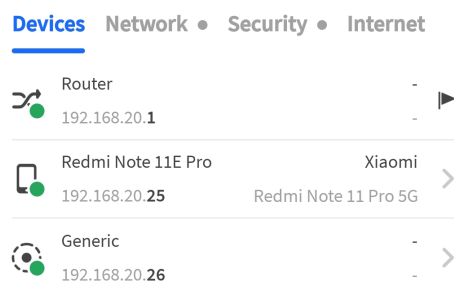


Figura 24 – Lista de dispositivos encontrados na Rede Local.

4.2.2 Cliente fora da Rede Doméstica sem VPN

No segundo cenário, o cliente não consegue se conectar ao servidor Zoneminder, uma vez que o servidor não é visível na rede na qual o cliente está conectado. O aplicativo zmNinja não carrega as imagens das câmeras e, ao tentar acessar o painel do Zoneminder diretamente, não obtém-se resposta. Na Figura 25, o diagrama indica que não há comunicação direta entre o cliente e o servidor.



Figura 25 – Diagrama para conexão do cliente fora da rede doméstica sem VPN.

O endereço "mattcc-ipcam.duckdns.org" refere-se ao endereço do IP público do Roteador da Rede Doméstica que está registrado em um Servidor DNS Dinâmico externo

¹ Ferramenta de varredura de rede, similar ao *nmap*, disponível em: <<https://www.fing.com/app/>>

(Duck DNS). Na tentativa de acesso por este endereço, não há sucesso, uma vez que o roteador doméstico não está configurado para permitir requisições externas (WAN) ao servidor Zoneminder (sem uma conexão VPN). O acesso pelo endereço IP local também não é possível, posto que este endereço é privado e o cliente não está conectado a essa rede nesse cenário (Fig. 26).

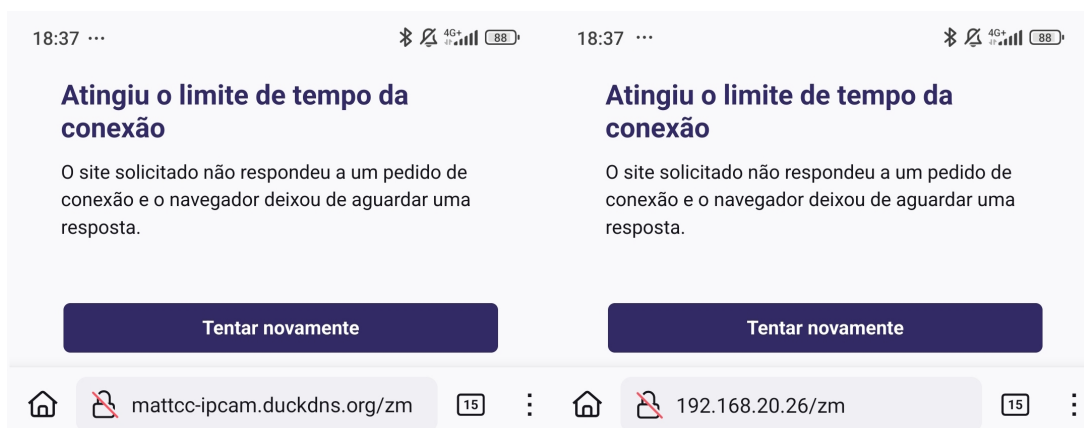


Figura 26 – Tentativa de conexão ao Zoneminder pelo navegador sem sucesso.

Por isso, há a necessidade de estabelecer uma conexão VPN com o servidor Zoneminder na Rede Doméstica. Adicionalmente, faz-se necessário a execução do cliente VPN WireGuard no celular para, em seguida, utilizar o cliente zmNinja. Este cenário é discriminado na Subseção 4.2.3.

4.2.3 Cliente fora da Rede Doméstica com VPN via Rede Celular

No terceiro cenário, o cliente fora da rede doméstica, via rede celular, consegue conectar-se ao Zoneminder utilizando o aplicativo WireGuard para estabelecer uma conexão VPN segura entre o dispositivo e o servidor que hospeda o Zoneminder. Na Figura 27, o diagrama apresenta a conexão VPN entre o cliente e servidor, onde o cliente consegue se comunicar com os dispositivos presentes na rede local a qual o servidor está conectado. Em um cenário semelhante, onde o cliente está fora da rede doméstica porém em outra rede residencial, as etapas de conexão e resultados são os mesmos.

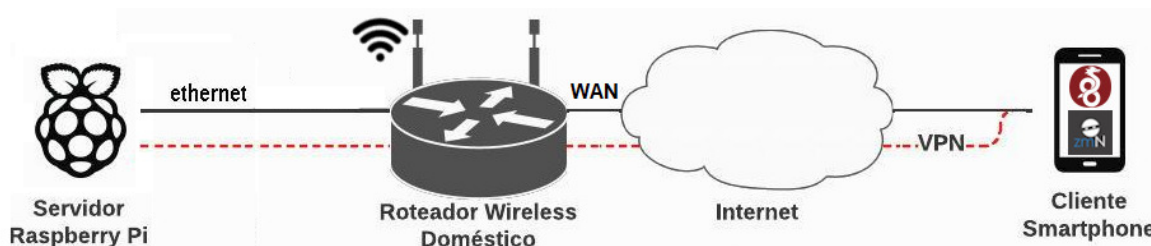


Figura 27 – Diagrama para conexão do cliente fora da rede doméstica com VPN.

O desempenho do sistema nessa condição é aceitável, o tempo de resposta e qualidade de imagem são satisfatórios para videomonitoramento em tempo real das imagens capturadas pelas duas câmeras instaladas.

4.2.4 Cliente fora da Rede Doméstica com VPN via Rede UFU

No quarto cenário, o cliente não consegue se conectar ao Zoneminder com uma VPN. Mesmo que semelhante ao cenário anterior, a rede Wi-Fi Institucional oferecida pela Universidade Federal de Uberlândia (UFU) não permite que clientes estabeleçam conexões VPN a redes externas, possivelmente por restrições de *firewall* definidas pela universidade. Na Figura 28, o diagrama apresenta o impedimento da conexão VPN ao servidor fora da rede institucional.

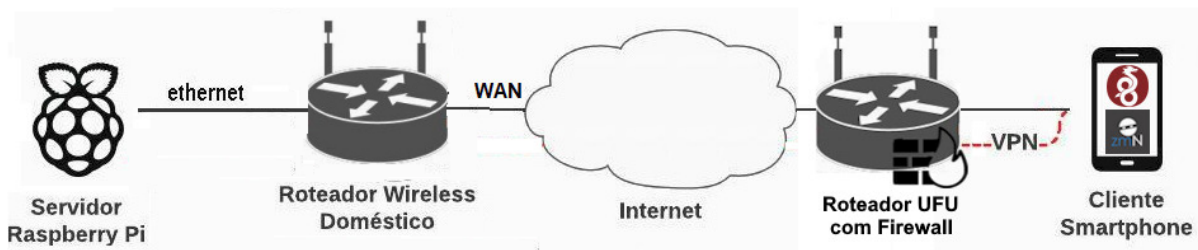


Figura 28 – Diagrama para conexão do cliente na rede da Universidade com VPN.

Na Figura 29, observa-se registros no aplicativo WireGuard da tentativa de conexão ao servidor, não foi possível completar o *handshake* necessário para acessar a rede doméstica externa.

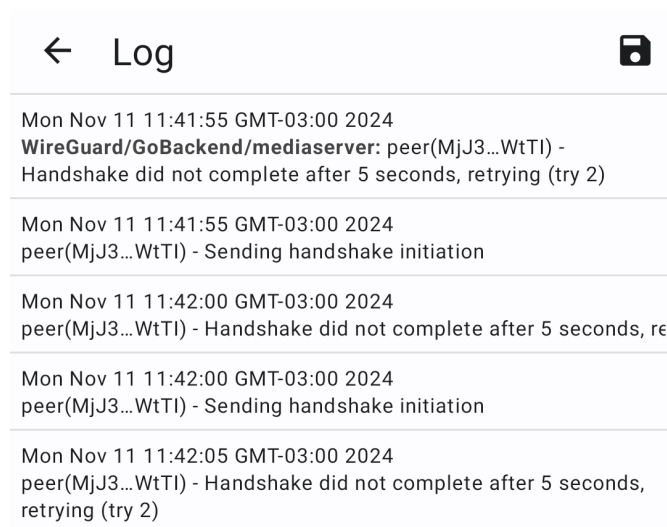


Figura 29 – Registros de tentativa de conexão à VPN sem sucesso.

Por isso, é necessário atenção ao contexto da rede a qual o cliente pertence, podem existir restrições que impeçam o acesso ao servidor Zoneminder.

4.3 Comparação a um Sistema Comercial de Videomonitoramento

Para avaliar as vantagens e limitações do sistema de videomonitoramento baseado em ZoneMinder, WireGuard e Raspberry Pi, compara-se ao sistema comercial da empresa Intelbras. A análise considera a estrutura do sistema, facilidade de uso e segurança. O diagrama na Fig. 30 ilustra a situação inicial das câmeras e do cliente na rede.

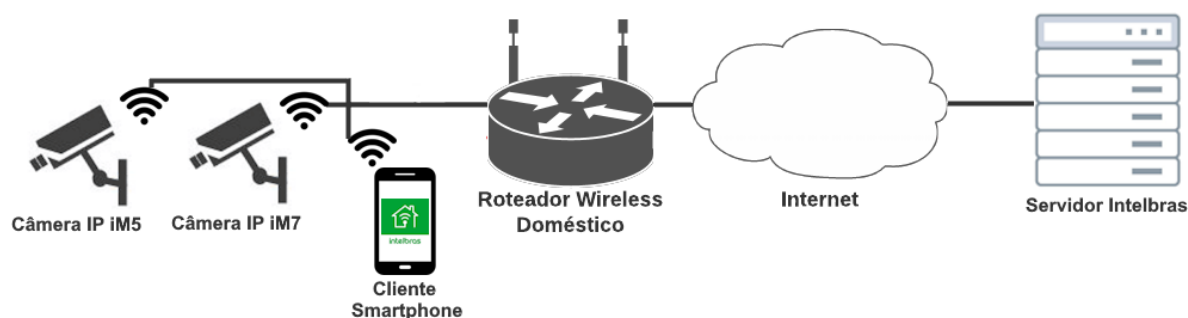


Figura 30 – Diagrama da configuração inicial das Câmeras IP Intelbras.

4.3.1 Configuração e Funcionalidades

Para analisar o sistema comercial, foram instaladas duas câmeras IP modelos iM7² e iM5³ da Intelbras. Seguindo as instruções oficiais⁴, inicialmente instala-se o aplicativo *Mibo Smart*, desenvolvido pela fabricante, em um *smartphone*. Em seguida, é necessário criar uma conta no serviço Intelbras para utilizar o aplicativo. Após criação da conta, é necessário realizar o *login* e adicionar a câmera no *Mibo Smart* selecionando o modelo correspondente e lendo o *QR code* associado à câmera. Na sequência, o cliente conecta-se ao ponto de acesso *wireless* das câmeras para inserir os dados da rede Wi-Fi à qual as câmeras conectam-se para acesso à Internet. Por fim, os dispositivos são associados à conta do usuário e permitem o acesso remoto via Internet pelo aplicativo.

O aplicativo proprietário *Mibo Smart* fornece funções de videomonitoramento e controle PTZ. A Figura 31 apresenta a visualização individual para uma das câmeras configuradas. Essa interface conta com o controle PTZ e opções para: silenciar o áudio da transmissão, ativar o holofote acoplado na câmera, alterar a qualidade de imagem da transmissão, tirar uma foto da visualização atual, travar a orientação da tela do celular e modo grade para visualizar mais de uma câmera.

² Descrição da câmera iM7: <<https://www.intelbras.com/pt-br/camera-externa-inteligente-wi-fi-3mp-360deg-im7-3mp-full-color>>

³ Descrição da câmera iM5: <<https://www.intelbras.com/pt-br/camera-externa-inteligente-wi-fi-full-color-im5-full-color>>

⁴ Manual de instruções *Mibo Smart*: <<https://manual-mibo.intelbras.com.br/pt-br/manuais/manual.html>>



Figura 31 – Visualização individual da câmera no aplicativo *Mibo Smart*.

Na Figura 32, observa-se a tela principal do aplicativo com a lista das câmeras disponíveis e botões para: i) alterar modo de listagem dos dispositivos (lista ou grade); ii) abrir visualização individual da câmera; iii) contratar o serviço de nuvem disponível na plataforma; iv) acessar a configuração individual de cada câmera.

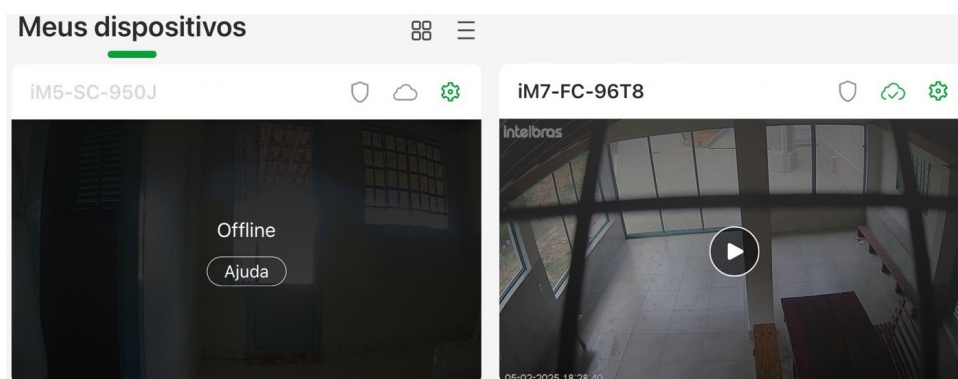


Figura 32 – Visão geral das imagens das câmeras disponíveis no aplicativo *Mibo Smart*.

Na Figura 33, lista-se as funcionalidades adicionais oferecidas pelo serviço pago *Mibo Cloud*, com opções para contratar ou renovar um plano do serviço. Dentre as funcionalidades, destacam-se: i) armazenamento de imagens em nuvem; ii) registro de eventos no "Relatório Diário" e iii) análise por Inteligência Artificial para identificação de objetos.



Figura 33 – Funcionalidades de videomonitoramento oferecidas por serviço pago.

4.3.2 Protocolo Peer-to-Peer para Câmeras IP

Para a efetivação da conexão P2P, a comunicação deve superar os desafios impostos pelo *Network Address Translation* (NAT), que impede conexões diretas de dispositivos externos com dispositivos localizados em redes privadas. Nesse contexto, o roteador doméstico desempenha um papel fundamental ao gerenciar o tráfego de rede entre os dispositivos internos e a Internet, atribuindo endereços IP privados e realizando sua tradução para um único IP público compartilhado. Conforme discutido por [Pattanaik, Sharvadze e Draheim \(2020\)](#), uma solução para esse problema é a aplicação da técnica de *Hole Punching*, um método amplamente utilizado para viabilizar conexões diretas entre dispositivos localizados atrás de roteadores NAT.

Hole Punching baseia-se na premissa de que os roteadores NAT permitem a recepção de pacotes de resposta para conexões previamente iniciadas por dispositivos internos. Assim, para estabelecer a comunicação direta, a câmera IP e o cliente *Mibo Smart* inicialmente se conectam a um servidor intermediário, que coordena o processo de *Hole Punching*. Esse servidor atua como um facilitador, auxiliando na troca dos endereços IP e portas externas utilizadas por cada dispositivo. A câmera IP primeiro envia um pacote de saída para o servidor, criando uma entrada temporária na tabela NAT do roteador, associando a porta utilizada a esse fluxo de comunicação. De maneira semelhante, o cliente *Mibo Smart* também se comunica com o servidor intermediário, registrando sua própria entrada na tabela NAT. Com essas informações, o servidor instrui ambos os dispositivos a tentarem a comunicação direta, enviando pacotes para os endereços e portas previamente descobertos. Como as entradas na tabela NAT permanecem ativas por um curto período, o roteador NAT permite que os pacotes do cliente alcancem a câmera IP, viabilizando

a conexão ponto a ponto (P2P). Embora o servidor intermediário seja necessário para iniciar o processo de *Hole Punching*, a comunicação subsequente ocorre de forma direta entre os dispositivos, eliminando a passagem de dados pelo servidor Intelbras.

A partir das informações reunidas, supõe-se que o aplicativo estabelece uma conexão P2P com as câmeras, utilizando o servidor apenas para a troca inicial de informações, como mapeamento de portas e IPs. A câmera mantém uma conexão constante com a Intelbras. Ao utilizar o serviço pago *Mibo Cloud* (Fig. 33), confirma-se que os arquivos de vídeo são armazenados na nuvem. Embora criptografados, esses arquivos permanecem sob a posse de um terceiro, sem que o usuário tenha controle total sobre o conteúdo.

4.3.3 Arquitetura do Sistema de Videomonitoramento Intelbras

A ausência de documentação pública sobre a infraestrutura do sistema utilizada para as câmeras IP residenciais da Intelbras, aliada à exigência de criação de uma conta no serviço Intelbras durante o fluxo padrão de configuração e acesso por meio do aplicativo *Mibo Smart*, sugere que a arquitetura do sistema seja composta pelos seguintes elementos: i) uma câmera IP conectada à Internet; ii) um servidor intermediário, controlado e mantido pela fabricante; e iii) um aplicativo proprietário (*Mibo Smart*), responsável pela interface de configuração das câmeras e exibição das imagens, mediadas pela comunicação com o servidor intermediário.

Para contornar a necessidade de configurar o *Port Forwarding* no roteador doméstico e registrar um IP público em um servidor de DNS Dinâmico, fabricantes de produtos de videomonitoramento implementam o protocolo *Peer-to-Peer* (P2P) para acesso remoto às imagens das câmeras (REOLINK, 2024). A principal característica de uma conexão P2P é a conexão direta entre um par de dispositivos, sem necessidade de um servidor central para receber e enviar dados, onde o par pode compartilhar e requisitar informações com privacidade. Porém, mesmo se tratando de um protocolo descentralizado, um mediador ainda é necessário para facilitar a conexão inicial entre os dispositivos. O trabalho Patanaik, Sharvadze e Draheim (2020) discute o uso do protocolo *Peer-to-Peer* na *web* e descreve a estrutura de um *framework* para compartilhamento de dados utilizando um servidor sinalizador entre os pares, abordagem semelhante àquela adotada em sistemas de videomonitoramento.

No sistema de videomonitoramento da Intelbras, considera-se que a câmera IP mantém conexão constante com o servidor da fabricante enquanto estiver ligada e com acesso à Internet, permitindo que o servidor tenha conhecimento do IP público da câmera. A partir disso, na Figura 34, o diagrama apresenta uma suposição de como o cliente *Mibo Smart* faz uma requisição ao servidor da Intelbras: i) cliente envia requisição ao servidor contendo o Identificador Único (UID) da câmera IP, registrado no momento inicial de configuração da câmera; ii) servidor envia a requisição de conexão para a câmera com o

endereço público do cliente; iii) câmera IP valida o UID e devolve requisição ao servidor para sinalizar ao cliente que está pronta para estabelecer uma conexão P2P; iv) câmera IP e cliente *Mibo Smart* têm conhecimento do IP público das duas partes e estabelecem uma conexão P2P criptografada.

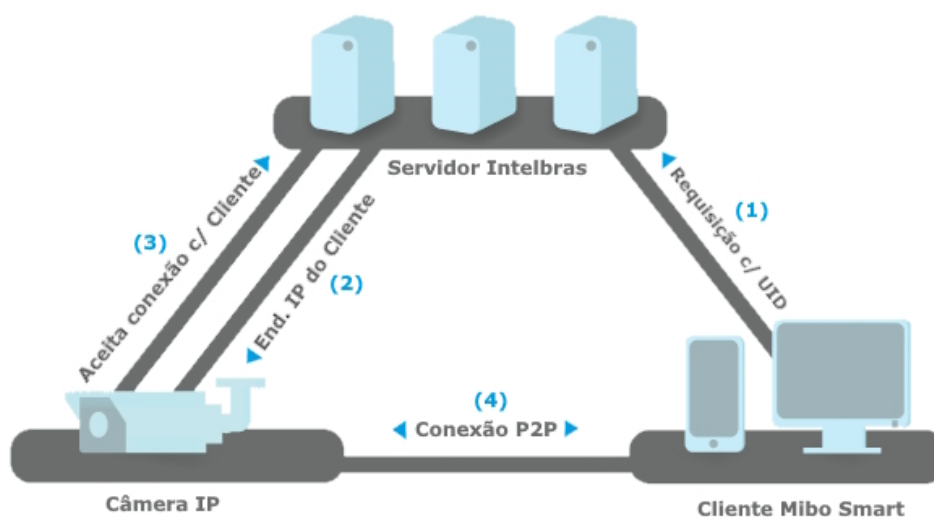


Figura 34 – Diagrama da conexão P2P em câmeras IP. Adaptado de [Reolink \(2024\)](#)

5 Conclusão

Este trabalho detalha a organização de componentes de *hardware* e *software* bem como executa o processo de configuração necessário para a criação do sistema de videomonitoramento de pequeno porte em uma rede doméstica. Utiliza-se câmeras IP de diferentes fabricantes integradas em um único sistema, combinando diferentes tecnologias de *software* livre e garantia de acesso às imagens com segurança.

O sistema apresenta uma solução funcional conforme planejado e viabiliza o acesso remoto seguro às imagens das câmeras com o controle total sobre o sistema e os dados gerados. A comunicação segura é estabelecida com sucesso entre um dispositivo na rede pública e o servidor na rede privada por meio de um DNS público e VPN. O Raspberry Pi mostrou-se suficiente para uma prova de conceito, porém é necessário ter maior poder computacional para um sistema mais robusto se mais câmeras IP forem utilizadas.

O processo de configuração é descrito de forma sequencial, abordando cada componente do sistema e suas respectivas especificidades. Verifica-se que tal procedimento exige conhecimento técnico na área de sistemas operacionais, redes de computadores e tecnologia da informação, o que o torna um projeto de implementação complexo, demandando conhecimentos específicos que podem representar um obstáculo para usuários sem a devida qualificação, sobretudo aqueles que procuram alternativas aos sistemas e serviços comerciais de videomonitoramento disponíveis comercialmente.

Destaca-se o papel fundamental do ZoneMinder na integração e uniformização de câmeras IP de diferentes fabricantes e protocolos, como RTSP e HTTP, evidenciado na configuração das câmeras AXIS e Wanscam. O ZoneMinder atuou como um *broker*, abstraindo as diferenças tecnológicas entre os dispositivos e permitindo seu gerenciamento centralizado, inclusive com suporte a funcionalidades como o controle PTZ. Essa capacidade de adaptação contribuiu para a flexibilidade do sistema, oferecendo uma alternativa viável e segura frente a NVRs tradicionais.

Durante a construção do sistema, é possível aplicar com sucesso os princípios fundamentais da segurança da informação: confidencialidade, integridade, autenticidade e controle de acesso. A confidencialidade e controle de acesso foram asseguradas por meio da implementação do protocolo WireGuard VPN, que criptografa toda a comunicação entre o cliente e o servidor, impedindo acessos não autorizados às imagens. A integridade e a autenticidade dos dados foi garantida ao manter o fluxo de vídeo e armazenamento restrito à rede local e à VPN, eliminando a dependência de servidores intermediários.

Os resultados indicam que essa abordagem oferece vantagens significativas em termos de controle e privacidade, uma vez que todo o fluxo de dados e o armazenamento das

imagens permanece sob controle direto do usuário, diferentemente de soluções comerciais que dependem de serviços em nuvem de terceiros, como exemplificado pela análise do serviço *Mibo Cloud*, onde o fluxo de dados passa por terceiros.

5.1 Trabalhos Futuros

A partir dos resultados obtidos neste trabalho e observações durante a implementação do sistema de videomonitoramento, identificam-se oportunidades de expansão de funcionalidades do sistema para estudos futuros com diferentes objetivos:

- utilizar servidor mais robusto para comportar maior número de câmeras e outras funcionalidades que requerem maior poder computacional;
- habilitar sistema de gravação de vídeo no Zoneminder com uso de armazenamento externo para realizar *backup* das imagens;
- configurar função de detecção de movimento no Zoneminder e combiná-la com outros sistemas para acionar alarmes, enviar notificações e/ou registrar em disco;
- desenvolver um serviço que facilite a implementação de um sistema semelhante ao proposto no trabalho, sem requisitar conhecimento técnico avançado.

Referências

- ALHAMMADI, A.; ALSHAMMARI, A.; ABDULLAH, R.; ABDULRAHMAN, A. Using a Dynamic Domain Name System (DDNS) Technology to Remotely Control a Building Appliances Network. In: IAENG. **Proceedings of the International MultiConference of Engineers and Computer Scientists 2017 (IMECS 2017)**. 2017. v. 1, p. 256–259. Disponível em: <https://www.iaeng.org/publication/IMECS2017/IMECS2017_pp256-259.pdf>. Citado na página 14.
- AXIS. **Suporte ao Produto para AXIS 214 PTZ Network Camera**. 2021. Acesso em: 26 jun. 2024. Disponível em: <<https://www.axis.com/pt-br/products/axis-214-ptz/support-0>>. Citado na página 19.
- BAKNI, M. **How Dynamic Network Address Port Translation (NAPT) Works**. 2020. Acesso em: 11 set. 2024. Disponível em: <<https://commons.wikimedia.org/wiki/File:NAPT-en.svg>>. Citado na página 30.
- BAU, J.; MITCHELL, J. C. **A Security Evaluation of DNSSEC with NSEC3**. 2010. Cryptology ePrint Archive, Paper 2010/115. Disponível em: <<https://eprint.iacr.org/2010/115>>. Citado na página 14.
- BONASTRE, O. M.; VEÀ, A. Origins of the Domain Name System. **IEEE Annals of the History of Computing**, v. 41, n. 2, p. 48–60, 2019. Citado na página 13.
- BRAVO, R. M. Monografia, **Infraestrutura de Monitoramento Remoto por Vídeo Utilizando Câmeras IP e um Dispositivo ARM**. Ponta Grossa, PR: [s.n.], 2015. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/23137/3/PB_ES_PRC_II_2015_07.pdf>. Citado na página 9.
- CASON, P. d. C.; COSTA, P. H. d. A. Trabalho de Conclusão de Curso, **Sistema de Monitoramento com Reconhecimento Facial e Fala Utilizando Raspberry Pi**. São Paulo, SP: [s.n.], 2021. Disponível em: <<https://dspace.mackenzie.br/items/d3f8bc b5-00b9-4400-9ac7-4e52899401dd>>. Citado na página 15.
- COSTA, G. M. F.; SANTOS, C. E. dos. Utilização de Containers: Um Contexto Histórico. **Revista Científica e-Locução**, v. 1, n. 20, p. 23, 2021. Disponível em: <<https://periodicos.faex.edu.br/index.php/e-Locucaao/article/view/388>>. Citado na página 16.
- DAYO, A. O. A Multi-Containerized Application Using Docker Containers and Kubernetes Clusters. **International Journal of Computer Applications**, v. 183, n. 44, p. 55–60, 2021. Citado na página 16.
- DESHMUKH, A.; MIHOVSKA, A.; PRASAD, R. Zoneminder as ‘Software as a Service’ and Load Balancing of Video Surveillance Requests. In: **Networking and Electronic Commerce Research Conference (NAEC) 2012**. [S.l.]: American Telecommunications Systems Management Association, 2012. Networking and Electronic Commerce Research Conference 2012, NAEC 2012 ; Conference date: 11-10-2012 Through 14-10-2012. Citado na página 15.

- DONENFELD, J. A. WireGuard: Next Generation Kernel Network Tunnel. In: INTERNET SOCIETY. **Network and Distributed System Security Symposium (NDSS)**. [S.l.], 2017. Citado 2 vezes nas páginas 13 e 18.
- DUCKDNS. **Duck DNS: Free Dynamic DNS Hosted on AWS**. 2024. Acesso em: 28 fev.2024. Disponível em: <<https://www.duckdns.org/about.jsp>>. Citado na página 18.
- EDEVANG, K. B.; SRISURESH, P. **Traditional IP Network Address Translator (Traditional NAT)**. RFC Editor, 2001. RFC 3022. (Request for Comments, 3022). Disponível em: <<https://www.rfc-editor.org/info/rfc3022>>. Citado na página 29.
- EMMANNI, P. S. The Impact of Docker, DevOps, and GitHub on Collaborative Open Source Software Development. **European Journal of Advances in Engineering and Technology**, v. 10, n. 8, p. 39–46, 2023. Disponível em: <<https://doi.org/10.5281/zenodo.11079634>>. Citado na página 16.
- FOUNDATION, R. P. **Raspberry Pi Documentation**. [S.l.], 2024. Acesso em: 19 mar. 2024. Disponível em: <<https://www.raspberrypi.com/documentation/>>. Citado na página 14.
- HOFFMAN, P. E.; MCMANUS, P. **DNS Queries over HTTPS (DoH)**. RFC Editor, 2018. RFC 8484. (Request for Comments, 8484). Disponível em: <<https://www.rfc-editor.org/info/rfc8484>>. Citado na página 14.
- HOLLINGWORTH, G. **Introducing Raspberry Pi Imager, Our New Imaging Utility**. 2020. Acesso em: 3 jul. 2024. Disponível em: <<https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility>>. Citado na página 22.
- HU, Z.; ZHU, L.; HEIDEMANN, J.; MANKIN, A.; WESSELS, D.; HOFFMAN, P. E. **Specification for DNS Over Transport Layer Security (TLS)**. RFC Editor, 2016. RFC 7858. (Request for Comments, 7858). Disponível em: <<https://www.rfc-editor.org/info/rfc7858>>. Citado na página 14.
- JUNIPER, N. **Network Address Port Translation (NAPT) Overview**. 2023. Acesso em: 22 fev. 2025. Disponível em: <<https://www.juniper.net/documentation/us/en/software/junos/interfaces-next-gen-services/topics/concept/nat-napt-overview-usf.html>>. Citado na página 29.
- KURKOVSKY, S.; WILLIAMS, C. Raspberry Pi as a Platform for the Internet of Things Projects: Experiences and Lessons. In: **Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education**. [S.l.: s.n.], 2017. p. 64–69. Citado na página 15.
- KUROSE, J. F.; ROSS, K. W. **Computer Networking: A Top-down Approach Featuring the Internet, 3/E**. [S.l.]: Pearson Education India, 2005. Citado na página 12.
- LEARNCCTV. **What is NVR (Network Video Recording)**. 2019. Acesso em: 22 mar. 2024. Disponível em: <<https://learncctv.com/what-is-nvr/>>. Citado na página 12.

- LIRANZO, J.; HAYAJNEH, T. Security and Privacy Issues Affecting Cloud-Based IP Camera. In: **2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)**. [S.l.: s.n.], 2017. p. 458–465. Citado 2 vezes nas páginas 9 e 17.
- MELLO, E. R.; EUZÉBIO, M. V. d. M. DROIDMI - Monitoramento de Câmeras de Vigilância Através de um Telefone Celular Android. **Revista Técnico-Científica do IFSC**, v. 2, 2013. Disponível em: <<https://periodicos.ifsc.edu.br/index.php/rtc/article/view/966>>. Citado na página 15.
- MERKEL, D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. **Linux Journal**, v. 2014, n. 239, p. 2, 2014. Disponível em: <<https://dl.acm.org/doi/10.5555/2600239.2600241>>. Citado 2 vezes nas páginas 15 e 23.
- MOCKAPETRIS, P. **Domain Names - Concepts and Facilities**. RFC Editor, 1987. RFC 1034. (Request for Comments, 1034). Disponível em: <<https://www.rfc-editor.org/info/rfc1034>>. Citado na página 13.
- MURÇA, I. C. O.; SANTOS, C. E. Desenvolvimento de um NVR Portátil para Utilização em Câmeras de Vigilância IP. **Revista Científica e-Locução**, v. 1, n. 17, 2020. Disponível em: <<https://periodicos.faex.edu.br/index.php/e-Locucão/article/view/240>>. Citado na página 15.
- NUTTALL, B. **What is a Raspberry Pi?** 2021. Acesso em: 28 fev. 2024. Disponível em: <<https://opensource.com/resources/raspberry-pi>>. Citado 2 vezes nas páginas 14 e 18.
- OCI. **About Open Container Initiative**. 2025. Acesso em: 21 jan. 2025. Disponível em: <<https://opencontainers.org/about/overview/>>. Citado na página 16.
- PADMANABHAN, R. **Reasons Dynamic Addresses Change**. 2020. RIPE Labs. Disponível em: <https://labs.ripe.net/author/ramakrishna_padmanabhan/reasons-dynamic-addresses-change/>. Citado 2 vezes nas páginas 14 e 26.
- PATTANAIK, V.; SHARVADZE, I.; DRAHEIM, D. A Peer-to-Peer Data Sharing Framework for Web Browsers: Analysis and Evaluation. **SN Computer Science**, v. 1, Junho 2020. Citado 2 vezes nas páginas 43 e 44.
- RASPBERRYPI. **About Us**. 2012. Acesso em: 19 mar. 2024. Disponível em: <<https://www.raspberrypi.org/about/>>. Citado na página 14.
- REOLINK. **What Is IP Camera P2P & How Does a P2P Camera Work**. 2024. Acesso em: 20 fev. 2025. Disponível em: <<https://reolink.com/blog/p2p-ip-camera/>>. Citado 2 vezes nas páginas 44 e 45.
- SAMSARA. **How to Choose the Best NVR Camera System for Your Business**. 2021. Acesso em: 21 mar. 2024. Disponível em: <<https://www.samsara.com/guides/nvr-camera-system/>>. Citado na página 12.
- SAUKKONEN, S. Monografia, **Implementing WireGuard to a Home Office Environment**. 2020. Disponível em: <https://www.theseus.fi/bitstream/handle/10024/347871/Saukkonen_Samu.pdf>. Citado na página 13.

- SILVA, A. M. A. Trabalho de Conclusão de Curso, **Segurança para Câmeras IP**. Uberlândia, MG: [s.n.], 2018. Disponível em: <<https://repositorio.ufu.br/handle/123456789/26148>>. Citado 3 vezes nas páginas 9, 11 e 17.
- SMITH, M.; HUNT, R. Network Security Using NAT and NAPT. In: **Proceedings 10th IEEE International Conference on Networks (ICON 2002). Towards Network Superiority (Cat. No.02EX588)**. [S.l.: s.n.], 2002. p. 355–360. Citado na página 29.
- STÜRMER, R.; AHLERT, E. M. Implementação de Sistema de Videomonitoramento de Baixo Custo Utilizando Zoneminder e Câmeras IP. **Revista Destaques Acadêmicos**, v. 12, n. 4, 2020. Citado 3 vezes nas páginas 9, 12 e 16.
- VERMA, N.; KASHYAP, M.; JHA, A. Extending Port Forwarding Concept to IOT. In: **2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)**. [S.l.: s.n.], 2018. p. 37–42. Citado na página 29.
- VIEIRA, D. **O que é uma Câmera de Segurança IP?** 2023. Acesso em: 20 mar. 2024. Disponível em: <<https://www.tecmundo.com.br/produto/263451-camera-seguranca-ip.htm>>. Citado na página 11.
- VIXIE, P. A.; THOMSON, S.; REKHTER, Y.; BOUND, J. **Dynamic Updates in the Domain Name System (DNS UPDATE)**. RFC Editor, 1997. RFC 2136. (Request for Comments, 2136). Disponível em: <<https://www.rfc-editor.org/info/rfc2136>>. Citado na página 13.
- VRABEC, L. **Technologies for Container Isolation: A Comparison of AppArmor and SELinux**. 2020. Acesso em: 21 jan. 2025. Disponível em: <<https://www.redhat.com/pt-br/blog/apparmor-selinux-isolation>>. Citado na página 16.
- ZONEMINDER. **ZoneMinder Documentation**. [S.l.], 2023. Acesso em: 23 mar. 2024. Disponível em: <<https://zoneminder.readthedocs.io/en/stable/index.html>>. Citado na página 15.