

Luiz Gustavo Pfitscher e Feldmann

**Projeto de um Sistema de Comunicação de Voz  
com Controle Remoto de Acesso por meio de  
*Power Line Communication***

Porto Alegre

2023

Luiz Gustavo Pfitscher e Feldmann

**Projeto de um Sistema de Comunicação de Voz com  
Controle Remoto de Acesso por meio de *Power Line  
Communication***

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL – UFRGS

ESCOLA DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Orientador: Prof. Dr. Ivan Müller

Porto Alegre

2023

Luiz Gustavo Pfitscher e Feldmann

Projeto de um Sistema de Comunicação de Voz com Controle Remoto de Acesso por meio de *Power Line Communication*/ Luiz Gustavo Pfitscher e Feldmann. – Porto Alegre, 2023-

72p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Ivan Müller

Trabalho de conclusão de curso (Graduação) –  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL – UFRGS  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA, 2023.

1. Comunicação. 2. Telefonia. 3. VoIP. 4. PLC. 5. Redes.

I. Prof. Dr. Ivan Müller, orientador. II. Universidade Federal do Rio Grande do Sul. III. Departamento de Engenharia Elétrica.

Luiz Gustavo Pfitscher e Feldmann

**Projeto de um Sistema de Comunicação de Voz com  
Controle Remoto de Acesso por meio de *Power Line  
Communication***

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

Banca Examinadora:

---

**Prof. Dr. Ivan Müller**

Universidade Federal do Rio Grande do Sul  
Porto Alegre  
Orientador

---

**Prof. Dr. Raphael Martins Brum**

Universidade Federal do Rio Grande do Sul  
Porto Alegre  
Convidado

---

**Me. Max Feldman**

Universidade Federal do Rio Grande do Sul  
Porto Alegre  
Convidado

Porto Alegre  
Agosto de 2023

*Este trabalho é dedicado à minha família, pelo constante apoio durante essa longa jornada.*

# Agradecimentos

Agradeço à minha família por me proporcionar a possibilidade de realizar esta Graduação.

Agradeço ao Prof. Dr. Ivan Müller pela orientação no presente trabalho.

Agradeço também a todos os amigos e colegas que de alguma forma colaboraram com essa conquista.

*“Tudo tem o seu tempo determinado,  
e todo propósito debaixo do céu tem o seu tempo:  
Há atempo de nascer, e tempo de morrer;  
tempo de plantar, e tempo de colher;  
(Ec 3, 1-2)*

# Resumo

A portaria eletrônica é amplamente utilizada como recurso de segurança em residências e condomínios em resposta à crescente criminalidade que aflige o país. Nesse contexto, é apresentado o projeto de uma solução integrada de interfonia e portaria eletrônica utilizando como canal de comunicação a rede elétrica predial. Essa solução pode ser vantajosa por reutilizar condutores preexistentes de maneira a facilitar a instalação e eventualmente reduzir custos. O sistema proposto implementa um servidor *Voice over Internet Protocol (VoIP)* conectado via *modems Power-Line Communication (PLC)* a *gateways* telefônicos distribuídos para realização de chamadas, e um relé para controle de acesso que é comandado por meio da discagem em um telefone. São discutidos os fundamentos funcionais do *PLC* e os principais protocolos que subsidiam o *VoIP*. É feito um apanhado sobre as normas e padrões de mercado que tratam sobre as diversas camadas dessa pilha. Apresenta-se o projeto de *software* do supracitado servidor *proxy Session Initiation Protocol (SIP)* que executa em um computador *Linux* embarcado e do *firmware* do microcontrolador que comanda o relé. Apresenta-se o projeto a nível de sistema dos protótipos e sua construção, sobretudo por meio da integração de componentes *off-the-shelf*. Ensaios realizados para caracterizar o desempenho do *PLC* atestaram que de fato ele atende a demanda de banda necessária para o *VoIP* empregado. Testes de repetibilidade qualificaram o bom funcionamento e a robustez do *software* desenvolvido.

**Palavras-chave:** Comunicação. Telefonia. VoIP. PLC. Redes.



# Abstract

Electronic gatekeeping is widely employed as a security countermeasure in residences and condominiums in response to the escalating criminality that afflicts the country. In this context, it's presented the project of an integrated solution of inter-phone features using the building's electricity grid as the communications channel. Such solution may be advantageous for reusing preexisting conductors and therefore facilitating the installation and reducing costs. The presented system implements a *Voice over Internet Protocol (VoIP)* server which is accessed via *Power-Line Communication (PLC)* modems by distributed telephone gateways for making calls, and a relay for access control which is operated by dialing on a phone. The working fundamentals of the *PLC* and the network protocols comprising the *VoIP* are discussed. An overview is made regarding the standards about the multiple layers of the employed stack. A software design is presented for the aforementioned *SIP* proxy server, which runs in an embedded *Linux* computer, as well as for the firmware of the microcontroller which commands the relay. It's presented the systems-level project and the construction of the prototypes, which is done foremostly by integration of off-the-shelf components. Tests carried out to assess the performance of the *PLC* have shown it indeed meets the demands of bandwidth required by the *VoIP*. Repeatability tests have certified the adequate working and the robustness of the developed software.

**Keywords:** Communication. Telephony. VoIP. PLC. Networks.

# Lista de ilustrações

Figura 1 – Representação simplificada do conceito do sistema de comunicação de voz via PLC proposto . . . . .	17
Figura 2 – Formas de onda simplificadas de tensão e corrente elétricas na linha telefônica . . . . .	27
Figura 3 – Diagrama de frequências (em Hz) correspondentes às teclas de discagem	28
Figura 4 – Diagrama das camadas que compõe a pilha VoIP . . . . .	30
Figura 5 – Formato do pacote RTP . . . . .	36
Figura 6 – Formato do pacote <i>telephone-event</i> . . . . .	37
Figura 7 – Ilustração do conceito dos protótipos . . . . .	38
Figura 8 – Diagrama de fluxo de dados entre blocos do sistema . . . . .	47
Figura 9 – Diagrama de sequência de uma chamada interceptada por servidor <i>proxy</i>	48
Figura 10 – Capturas de tela das páginas <i>web</i> do servidor do relé comandado por rede	49
Figura 11 – Esquemático do circuito de acionamento do relé . . . . .	50
Figura 12 – Fotografia dos protótipos construídos . . . . .	53
Figura 13 – Taxa de dados medida em função da distância . . . . .	55
Figura 14 – Diagrama de sequência do registro SIP . . . . .	63
Figura 15 – Diagrama de sequência de uma chamada em SIP . . . . .	65
Figura 16 – Diagrama de herança das classes de campos . . . . .	67
Figura 17 – Diagrama de herança das classes de mensagens . . . . .	68
Figura 18 – Diagramas de classes variadas . . . . .	69
Figura 19 – Diagrama de sequência de chamada com interceptação de DTMF . . . .	70
Figura 20 – Diagrama de atividades do algoritmo de interpretação de requisições HTTP . . . . .	71
Figura 21 – Diagrama de atividades da lógica de geração de respostas do servidor HTTP . . . . .	72

# Lista de quadros

Quadro 1 – Bandas da G3-PLC por região . . . . .	20
Quadro 2 – Amostra de produtos PLC no mercado . . . . .	21
Quadro 3 – Amostra de produtos VoIP no mercado . . . . .	22
Quadro 4 – Banda e taxa de dados para diversos perfis do <i>HomePlug AV 2.1</i> . . . . .	27
Quadro 5 – Alguns métodos de requisições SIP . . . . .	32
Quadro 6 – Alguns códigos de resposta SIP . . . . .	32
Quadro 7 – Campos do cabeçalho de mensagens SIP . . . . .	33
Quadro 8 – Parâmetros da descrição SDP . . . . .	35
Quadro 9 – Alguns tipos de carga útil utilizados com RTP . . . . .	36
Quadro 10 – Resumo da lista de materiais . . . . .	44

# Lista de tabelas

Tabela 1 – Ocupação de recursos do microcontrolador . . . . .	53
Tabela 2 – Resultado do teste de tempo de resposta do servidor HTTP . . . . .	54
Tabela 3 – Resultados do ensaio de transferência de dados no PLC . . . . .	54
Tabela 4 – Resultados do ensaio de repetibilidade de chamada VoIP . . . . .	55
Tabela 5 – Resultados do ensaio de repetibilidade de acionamento do relé . . . . .	55

# Lista de abreviaturas e siglas

<b>AES</b>	<i>Advanced Encryption Standard</i>
<b>AVR</b>	Alf & Vegard RISC
<b>CPU</b>	<i>Central Processing Unit</i>
<b>CRLF</b>	<i>Carriage Return + Line Feed</i>
<b>DC</b>	<i>Direct Current</i>
<b>DIN</b>	<i>Deutsches Institut für Normung</i>
<b>DTMF</b>	<i>Dual-Tone Multi-Frequency</i>
<b>EEPROM</b>	<i>Electrically-Erasable Programmable Read-Only Memory</i>
<b>GSM</b>	<i>Global System for Mobile Communications</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>HTTP</b>	<i>HyperText Transfer Protocol</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IETF</b>	<i>Internet Engineering Task Force</i>
<b>IHM</b>	Interface Homem-Máquina
<b>IP</b>	<i>Internet Protocol</i>
<b>ITU</b>	<i>International Telecommunication Union</i>
<b>LAN</b>	<i>Local Area Network</i>
<b>MCU</b>	<i>MicroController Unit</i>
<b>OFDM</b>	<i>Orthogonal Frequency Division Multiplexing</i>
<b>PCM</b>	<i>Pulse-Code Modulation</i>
<b>PLC</b>	<i>Power-Line Communication</i>
<b>PNAD</b>	Pesquisa Nacional por Amostra de Domicílios
<b>POTS</b>	<i>Plain Old Telephone Service</i>
<b>PSK</b>	<i>Phase Shift Keying</i>

<b>PSTN</b>	<i>Public Switched Telephone Network</i>
<b>QAM</b>	<i>Quadrature Amplitude Modulation</i>
<b>QoS</b>	<i>Quality of Service</i>
<b>RAM</b>	<i>Random Access Memory</i>
<b>RTP</b>	<i>Real-Time Transport Protocol</i>
<b>SBC</b>	<i>Single Board Computer</i>
<b>SDP</b>	<i>Session Description Protocol</i>
<b>SIP</b>	<i>Session Initiation Protocol</i>
<b>TCP</b>	<i>Transport Control Protocol</i>
<b>UA</b>	<i>User Agent</i>
<b>UDP</b>	<i>User Datagram Protocol</i>
<b>URI</b>	<i>Uniform Resource Identifier</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>VoIP</b>	<i>Voice over Internet Protocol</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>18</b>
2.1	História	18
2.2	Normas	19
2.3	Produtos Disponíveis no Mercado	20
2.4	Trabalhos Relacionados	23
2.4.1	Trabalhos Relacionados ao PLC	23
2.4.2	Trabalhos Relacionados ao VoIP	24
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>25</b>
3.1	Princípios de PLC	25
3.2	<i>Plain Old Telephone Service (POTS)</i>	27
3.3	<i>Dual-Tone Multi-Frequency (DTMF)</i>	28
3.4	Pilha de Protocolos do VoIP	30
3.5	Protocolos de Transporte	31
3.6	<i>Session Initiation Protocol (SIP)</i>	32
3.7	<i>Session Description Protocol (SDP)</i>	34
3.8	<i>Real-Time Transport Protocol (RTP)</i>	35
<b>4</b>	<b>MATERIAIS E MÉTODOS</b>	<b>38</b>
4.1	Premissas	38
4.2	Hipóteses	39
4.3	Ferramentas de Software	39
4.3.1	Linguagem de Programação	39
4.3.2	Sistema de <i>Build</i>	40
4.3.3	Documentação	41
4.3.4	Versionamento	41
4.3.5	Depuração e Perfilamento	42
4.4	Materiais	43





# 1 Introdução

A criminalidade é hoje um dos principais problemas sociais que afligem o Brasil. Dados da [PNAD: Sensação de segurança \(IBGE, 2021a\)](#) apontam que mais de 10% dos brasileiros não se sentem seguros no local onde vivem, sendo que 4% dos domicílios foi vítima de furto ou roubo residencial. Para aplacar o problema, segundo o [PNAD: Furtos e roubos \(IBGE, 2021b\)](#), cerca de 68% dos domicílios recorreram a algum recurso de segurança.

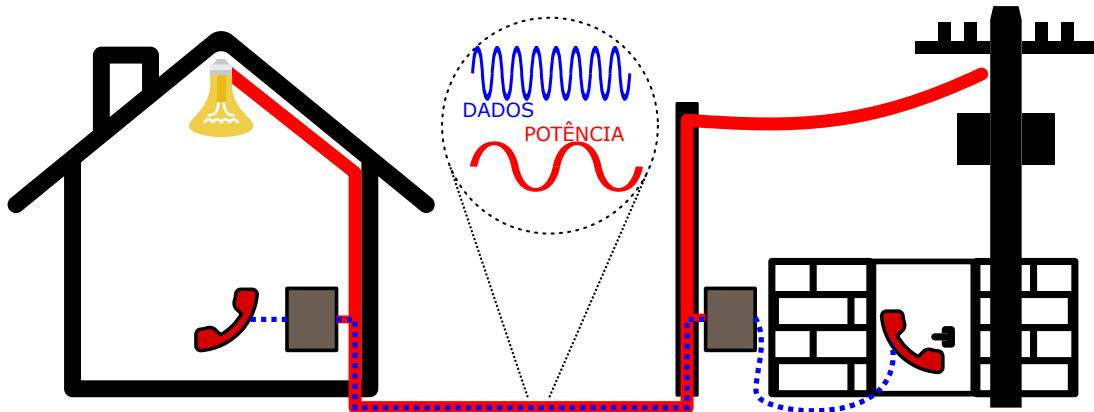
Entretanto, a instalação desse equipamento nem sempre é simples, sobretudo em imóveis mais antigos, pois requer a passagem de fiação telefônica dedicada. Soluções sem fio estão disponíveis para contornar tais impeditivos arquitetônicos, porém com foco no modelo ponto-a-ponto, sem suporte para uma rede condominial. Isso é esperado, dada a dificuldade de promover um enlace de RF confiável entre estações espalhadas em diversos andares ou blocos residenciais sem violar as restrições de potência e uso espectral impostos pelas agências reguladoras. Outras soluções disponíveis se baseiam em *Global System for Mobile Communications (GSM)*, e portanto apresentam a desvantagem de necessitar assinatura de serviços de uma operadora de telefonia.

Nesse contexto, é oportuno explorar meios de comunicação alternativos para atender à aplicação, tais como o [PLC](#): um conjunto de tecnologias que visa formar um canal de comunicação de dados reutilizando a infraestrutura de distribuição de energia elétrica preexistente, eliminando a necessidade de cabos adicionais, simplificando a instalação e reduzindo custos. Similarmente, os sistemas de telefonia *Public Switched Telephone Network (PSTN)* tradicionais, que demandam cabeamento dedicado, deixam de ser a alternativa mais econômica frente à disseminação do [VoIP](#). Essa tecnologia possibilita a transmissão de voz através de redes de dados *Internet Protocol (IP)*, e vem se proliferando, impulsionada pelo reaproveitamento de redes *Local Area Network (LAN)*.

Percebendo a congruência entre essas tendências, o presente trabalho visa apresentar o projeto e construção de um protótipo de sistema que implemente funcionalidades de comunicação de voz e controle de acesso para uso residencial ou condominial, baseado na associação entre o [PLC](#) e o [VoIP](#). Ambas as tecnologias, sozinhas, estão consolidadas,

porém a sua união em uma única solução integrada, autocontida, eficiente, escalável, e de fácil instalação para a aplicação proposta é um problema de engenharia ainda em aberto e que assume a pauta do projeto que se segue, e cujo conceito está ilustrado na [Figura 1](#).

Figura 1 – Representação simplificada do conceito do sistema de comunicação de voz via PLC proposto



Fonte: o Autor, 2023.

O presente trabalho apresenta o desenvolvimento do *software* de um servidor VoIP utilizando o padrão SIP para subsidiar ligações entre telefones interconectados em uma rede local formada por *modems* PLC. Similarmente, é aqui descrito o desenvolvimento de um *firmware* para microcontrolador, administrável por página *web*, que permite o controle remoto de um relé para acionamento de um dispositivo de controle de acesso, por meio da discagem de uma tecla específica do telefone. Os artefatos de *software* desenvolvidos são embarcados em duas unidades protótipo, construídas pela integração de diversos componentes *off-the-shelf*. Tais protótipos constituem uma prova de conceito de mínimo produto viável, e subsidiam os experimentos de verificação de viabilidade técnica da solução proposta.

Os ensaios demonstraram que a taxa de dados obtida no enlace PLC foram suficientes para atender à demanda das codificações empregadas no VoIP. Testes de repetibilidade comprovaram a robustez do *software* criado e o bom funcionamento dos protótipos construídos. Desta forma, o presente conceito poderia futuramente evoluir para um produto comercial. No meio tempo, o código fonte ora produzido foi disponibilizado à comunidade com licença livre.

## 2 Revisão Bibliográfica

### 2.1 História

O conceito de [PLC](#) surgiu logo após o estabelecimento das primeiras redes de distribuição de energia elétrica, com a patente de *Melhorias relacionadas a medidores de eletricidade*, de J. Routin e C. E. Lancelot Brown, na Suíça, em 1898. A primeira aplicação foi justamente para leitura remota de medidores e operação de chaves de controle nas redes de distribuição. Até então, as distribuidoras utilizavam linhas telefônicas para essa finalidade, porém estas não eram consideradas confiáveis durante o inverno ou em regiões montanhosas ([SCHWARTZ, 2009](#)).

Essa tecnologia logo ganhou atratividade para as aplicações de telefonia. Em 1918, no Japão, foi realizado o primeiro teste com esse propósito em uma linha de transmissão de 144 km. O primeiro teste nos Estados Unidos foi realizado em uma linha de 19 km, em 1920, com uma portadora de 5 kHz. Só em 1922 foi implementada uma solução para uso efetivo: comunicação telefônica entre uma usina hidroelétrica, na Califórnia, e uma subestação utilizando como meio as próprias linhas de alta tensão. Em 1923 a *GE* já comercializava modems [PLC](#) de 50 W para atender serviços comerciais de telefonia de até 85 milhas ([SCHWARTZ, 2009](#)).

Esses sistemas comerciais operavam de 15 a 500 kHz em linhas de média e alta tensão. Porém nas redes de distribuição de baixa tensão a filtragem causada pelos transformadores impunha a limitação de até 3 kHz ([LAMPE, 2016](#), p. 3). O [PLC](#) então caiu no esquecimento por várias décadas, tendo readquirido foco em 1991 com a publicação da norma [CENELEC EN 50065](#) que trata desses sistemas em baixa tensão e frequências entre 3 a 148 kHz. Em contínua evolução desde então, as normas atuais como a *HomePlug* já atingem taxas de dados na ordem de centenas de Mbps (vide [seção 2.2](#)).

O conceito do [VoIP](#) foi proposto pela primeira vez em 1973 com a invenção do *Network Voice Protocol*, parte do *ARPANET*, porém a primeira implementação de mercado de *internet phone* surgiu apenas apenas em 1995 com o *Vocaltec* ([FASIKU, 2018](#)). A qualidade era limitada e ambos o chamador e o receptor necessitavam possuir o mesmo

aplicativo.

Em 1996 a [IETF](#) deu início aos primeiros grupos de trabalho visando padronizar a tecnologia, resultando na publicação da RFC do [RTP](#). Em 1999 foi aprovada a normatização do [SIP](#) ([SINNREICH; JOHNSTON, 2006](#), p. 7). Nos anos seguintes as funcionalidades básicas de chamadas foram estendidas pela adição do suporte à mensagens instantâneas. Em 2004, as operadoras telefônicas passaram a ofertar planos de assinatura de serviços [VoIP](#).

Na atualidade, o [VoIP](#) está transparentemente integrado a outras infraestruturas de telecomunicações, como é o caso do *VoLTE* (*Voice over Long Term Evolution*), utilizado para chamadas de voz em aparelhos móveis *4G* ([CHAKRABORTY; MISRA; PRASAD, 2019](#), p. 18).

## 2.2 Normas

No ano de 2000 foi fundada a *HomePlug Powerline Alliance*, uma associação industrial destinada a acelerar o desenvolvimento e a adoção de soluções [PLC](#) ([LATCHMAN, 2013](#), p. 4). O *HomePlug 1.0* foi o primeiro padrão publicado pela entidade, definindo as camadas física e de enlace de uma rede que suportaria uma taxa de dados de até 14 Mbps.

Em 2012 foi lançado o padrão *HomePlug v2.0*, atingindo até 1,5 Gbps. Esse padrão utiliza [OFDM](#) de 917 subportadoras, que podem ser moduladas deste BPSK até [QAM-1024](#), na faixa de frequências de 1,8 - 30 MHz. Os quadros possuem 520 bytes, incorporam código de correção de erro turbo convolucional e são criptografados com [AES](#) de 128 bits ([LATCHMAN, 2013](#), p. 20). O acesso múltiplo se dá por TDMA e CSMA, gerenciados por um nodo central da rede. Em um mesmo circuito elétrico podem coexistir diversas redes de dados segregadas virtualmente através do mecanismo de sub-redes lógicas virtuais (AVLN) baseado em criptografia. Nodos podem ingressar na rede virtual por meio de uma chave criptográfica pré-compartilhada. Esse é o mesmo mecanismo que impede o potencial *crosstalk* com redes [PLC](#) em imóveis vizinhos ([LATCHMAN, 2013](#), p. 15).

Em 2010 foi publicada a norma [IEEE 1901](#) ([IEEE... , 2021](#)), desenvolvida com base no padrão *HomePlug*, visando redes com taxas de dados até 400 Mbps. A banda foi ampliada para até 50 MHz, e o número de subportadoras [OFDM](#) cresceu para 1974 ([LATCHMAN, 2013](#), p. 269). Também foi dado suporte a uma segunda camada física,

opcional, constituída de 512 subportadoras **OFDM** na faixa **DC** - 31,25 MHz, moduladas M-PAM, até  $M = 32$  (LATCHMAN, 2013, p. 274).

Em 2009 a Maxim publicou a especificação aberta *G3-PLC* visando atender a aplicações de leitura automática de medidores de energia, automação residencial, e redes locais. Posteriormente, o padrão foi adotado por outras doze empresas, dentre elas *STMicroelectronics*, *Texas Instruments*, *Cisco*, etc. Visando manter e desenvolver a norma, passaram a constituir a *G3-PLC Alliance*. Em 2012 a norma foi republicada como **ITU-T G.9903** (LAMPE, 2016, p. 520). As bandas utilizadas variam conforme a região geográfica (Quadro 1). A modulação empregada pode variar de **B-PSK** a **8-PSK**, nesse último caso atingindo até 300 kbps. Também é utilizada criptografia **AES-128** e correção de erros *Reed-Solomon*.

Quadro 1 – Bandas da G3-PLC por região

Região	Banda	Início	Fim
Europa	CENELEC-A	35,938 kHz	90,625 kHz
Estados Unidos	FCC-1a	154,687 kHz	262,5 kHz
Estados Unidos	FCC-1b	304,687 kHz	487,5 kHz
Ásia	ARIB	154,7 kHz	403,1 kHz

A *PRIME Alliance* é ainda outra entidade, fundada em 2009, visando padronizar o uso do **PLC** via normas não-proprietárias, com foco específico em aplicações de medição remota de consumo de eletricidade e gás. Foi responsável por desenvolver a **ITU-T G.9904**, publicada em 2012. Nesse padrão são usadas 96 subportadoras **OFDM** na banda de 42–89 kHz, modulação **8-PSK**, atingindo uma taxa de dados de até 128 kbps (LAMPE, 2016, p. 527).

As principais normas que estabelecem padrões para o **VoIP** são a **ITU-T H.323** proposta pela **ITU**, com arquitetura centralizada e já em desuso, e a mais comum **SIP**, de arquitetura distribuída, mantida pela **IETF** (FASIKU, 2018). Esta última é abordada em detalhe na seção 3.6.

## 2.3 Produtos Disponíveis no Mercado

O Quadro 2 apresenta um apanhado de produtos disponíveis comercialmente que utilizam do **PLC**. Predominam aplicações de distribuição e medição inteligentes de energia elétrica e de **LAN** residencial.

Quadro 2 – Amostra de produtos PLC no mercado




Imagem	Aplicação	Marca/Modelo	Taxa Dados
	LAN residencial	D-Link DHP-W221AV	200 Mbps
	LAN residencial	TP-Link TL-PA7017P	1000 Mbps
	Monitor de geração solar Agregador de dados Rede inteligente	AP Systems ECU-C	—
	Medição de energia Rede inteligente	Itron OpenWay Riva	—
	Interfone interno (sem controlador de acesso)	Intercom Central FM-246	—

Fonte: o Autor, 2023.

O **Quadro 3** apresenta uma amostra de dispositivos **VoIP** comercialmente disponíveis. Nota-se que outros tipos de produtos, não apenas telefones, vêm se valendo dessa tecnologia e seus padrões para propósitos diversos.

O mercado também dispõe de uma série de ferramentas de *software* de código aberto voltadas a implantação de sistemas baseados em **VoIP**, dentre as quais a mais

Quadro 3 – Amostra de produtos VoIP no mercado

Imagem	Marca/Modelo	Descrição
	Intelbras TIP 635 G	Telefone nativo VoIP para até 20 contas SIP e com suporte a chamada de vídeo.
	Khomp Speaker Wall 130 E	Alto falante para plataformas de anúncio com conexão por SIP.
	Khomp Botão SIP	Botão de emergência ou alarme com integração via chamada VoIP.

Fonte: o Autor, 2023.

notável é o *Asterisk*. De acordo com Muggelen, Bryant e Madsen (2019, p. 10), o Asterisk se trata de uma plataforma modular para construção de PABXs VoIP ou PSTN com suporte integrado a diversos recursos como: ramais, interconexão entre ramos, fila de espera, menus de atendimento automatizado, gravação de chamadas, caixa postal, conversão de mídia, conferências, etc. As funcionalidades, e por consequência também a complexidade, são tamanhas que surgiu uma interface de usuário dedicada a administrar esse sistema: o *FreePBX*.

Internamente, o *Asterisk* utiliza a biblioteca de código aberto *pjSIP* como canal para suas terminações SIP. Além desse, essa biblioteca suporta os protocolos SDP, RTP, STUN, TURN e ICE. Também estão implementadas as lógicas de servidor e UA, bem como recursos de conversão de mídia e de atravessamento de redes separadas por tradução de endereços (MUGGELEN; BRYANT; MADSEN, 2019, p. 67).

## 2.4 Trabalhos Relacionados

Corcoran (2005) patenteou um sistema de comunicação de longa distância que utiliza VoIP para transmitir voz por meio de linhas de transmissão de média tensão utilizando o PLC. É proposto o uso de filas de prioridade de pacotes e outras formas de *traffic shaping* para favorecer a qualidade do serviço VoIP em detrimento de outros serviços que compartilhem tal rede de dados PLC. Essa solução sugere ainda a instalação de dispositivos de desvio em paralelo com transformadores BT/MT, visando permitir ao sinal de alta frequência do PLC contornar tais transformadores para minimizar a atenuação.

Zaqout, Nasr e Abu-Nasser (2017) empregou um servidor *Asterisk* conectado por intermédio de *modems PLC a gateways VoIP* para caracterizar chamadas de voz e vídeo. Foram ensaiadas ligações combinando telefones analógicos, o *softphone Zoiper*, e telefones celulares (pela conexão do *gateway* com a rede PSTN) e também utilizando múltiplos *codecs*.

Nas subseções a seguir são explorados outros trabalhos que empregam as tecnologias PLC e VoIP isoladamente.

### 2.4.1 Trabalhos Relacionados ao PLC

Ginot et al. (2010) propôs o PLC aplicado à instrumentação industrial. Foi concebido um sistema de controle de velocidade de um motor, em malha fechada, onde o tacômetro instalado no eixo do motor e também um sensor de temperatura se comunicam com o inversor de frequência variável utilizando as próprias linhas de potência. Esse experimento inova ao tratar de um canal especialmente poluído devido ao conteúdo harmônico oriundo do chaveamento PWM do inversor. É validado o uso de modems de padrão *HomePlug* na aplicação, porém com taxa de dados inferior à nominal e decrescente com a potência entregue pelo motor.

Barmada et al. (2010) investigou a viabilidade do uso de PLC em navios de cruzeiro. Foi desenvolvido um modelo teórico da rede elétrica da embarcação com base na teoria de linhas de transmissão. Posteriormente foi realizada uma campanha de medições para validá-lo.

Similarmente, Degardin et al. (2013) investigou o emprego de PLC em aeronaves, visando simplificar seus cabeamentos, que vêm crescendo em complexidade devido a



introdução de novos sistemas eletrônicos aviônicos. Foi caracterizada a rede elétrica de iluminação da cabine, determinando a sua função de transferência em uma faixa de frequência de 1,8 - 30 MHz. O modelo produzido foi em seguida introduzido em uma simulação computacional para estimar a taxa de dados possível, que foi de 98 Mbit/s.

Em face às tendências de cidades conectadas, [Farkas et al. \(2018\)](#) realizou um experimento onde foi empregado o PLC para controle proporcional, individual, da intensidade das lâmpadas de iluminação pública. O ciclo de potência de cada lâmpada foi customizado baseado em fatores como a localização geográfica e a época do ano, sendo ao final obtida uma economia de energia de 9,6%.

## 2.4.2 Trabalhos Relacionados ao VoIP

[Bai e Du \(2013\)](#) propôs o uso do VoIP para comunicação entre navios pesqueiros interligados por uma rede de dados *mesh*. Foi demonstrado compartilhamento da conexão via satélite de uma embarcação, atuando como *gateway* para rotear chamadas de outros barcos desprovidos de tal comunicação via satélite.

[Hendrawan e Aditya \(2019\)](#) explorou a associação do VoIP com o *RoIP* (*Radio over Internet Protocol*) para comunicação entre aeronaves e estações de controle de tráfego aéreo.

[Khan et al. \(2022\)](#) estudou o desempenho de um *Single Board Computer* (SBC) *Raspberry Pi 3* como hospedeiro de um servidor *Asterisk*. Foram levantados os perfis de utilização de CPU, da memória, e de tráfego de rede em função do número de chamadas simultâneas para diversos *codecs* de áudio. Dependendo do caso, entre 70–100 chamadas simultâneas foram possíveis.

## 3 Fundamentação Teórica

### 3.1 Princípios de PLC

Pela modulação da amplitude (3.1), fase (3.2) ou frequência (3.3) de uma onda portadora utilizando como entrada um sinal mensagem em banda base, é possível deslocar tal mensagem no espectro para frequências mais altas (LATHI; DING, 2019, p. 187). Esse é o princípio fundamental de comunicações em banda passante, como é o caso do PLC. Em especial, esse deslocamento permite que o sinal de dados coexista com a corrente alternada em 50 ou 60 Hz que é utilizada para a transmissão de energia elétrica.

$$\phi_{AM} = Am(t)\cos(\omega_C t) \quad (3.1)$$

$$\phi_{PM} = A\cos(\omega_C t + K_P m(t)) \quad (3.2)$$

$$\phi_{FM} = A\cos(\omega_C t + K_F \int_{-\infty}^t m(\alpha) d\alpha) \quad (3.3)$$

$\phi(t)$	–	Sinal modulado
$m(t)$	–	Mensagem em banda base
$A$	–	Amplitude
$\omega_C$	–	Frequência angular da portadora
$K_F, K_P$	–	Constante de modulação

A reutilização da rede de distribuição elétrica é atraente por não necessitar de investimentos para instalação de uma rede de dados dedicada à aplicação sendo implantada. Todavia, essa rede não foi projetada para a finalidade específica de servir como canal de comunicação, o que traz uma série de fatores dificultantes para o bem sucedido uso do PLC. Alguns deles são:

- As redes elétricas residenciais costumam apresentar ramificações, o que dificulta o uso de modelos mais convencionais de linha de transmissão como ferramenta de análise. Outros modelos próprios ao PLC devem ser empregados, como o de *Phillips* ou de *Zimmermann-Dostert*. (ANATORY; THEETHAYI, 2010, p. 23)
- A impedância do canal é desconhecida, podendo variar largamente com a conexão ou desconexão de cargas na rede elétrica. Isso dificulta o casamento de impedância e

pode gerar problemas de reflexão ou baixa entrega de potência do sinal (ANATORY; THEETHAYI, 2010, p. 68).

- O canal é ruidoso em razão da variedade de cargas que podem estar conectadas gerando interferência em um amplo espectro. Esse problema é crescente com a popularização de fontes chaveadas. (ANATORY; THEETHAYI, 2010, p. 118).

Para a presente aplicação, é importante notar que a banda da voz humana está limitada em 3500 Hz, de acordo com Lathi e Ding (2019, p. 187). Para que um sinal digitalizado possa ser reconstruído sem distorções, deve ser obedecido o critério de Nyquist (3.4), que estabelece o limite mínimo da frequência de amostragem como o dobro da banda da mensagem (LATHI; DING, 2019, p. 286).

$$f_s \geq 2B \quad (3.4)$$

$f_s$  – Frequência de amostragem [Hz]  
 $B$  – Largura de banda da mensagem [Hz]

Se o sinal de voz for amostrado com uma resolução de 8 bits, então o limite mínimo para a capacidade do canal de comunicação é de 56 kbit/s. Valendo-se do teorema de Shannon-Hartley, é possível estimar a banda necessária do canal de comunicação em função da sua relação sinal-ruído (LATHI; DING, 2019, p. 860).

$$C = B \log_2 \left( 1 + \frac{S}{N} \right) \quad (3.5)$$

$C$  – Capacidade do canal [bit/s]  
 $B$  – Largura de banda do canal [Hz]  
 $\frac{S}{N}$  – Razão (linear) entre as potências do sinal e do ruído

A potência do sinal pode ser estimada pelas especificações de máscara espectral apresentadas na ITU-T G.9964. Já a característica do ruído na rede elétrica residencial depende de diversos fatores, como as cargas nela conectadas e da resistência elétrica do cabeamento. Nesse contexto Vines et al. (1984) apresenta uma análise da densidade espectral de ruído para diversos aparelhos domésticos.

Para uma análise preliminar, é mais direto consultar as taxas de dados estipuladas para padrões de indústria do PLC como o HomePlug. Comparando-se a estimativa anterior com os dados do Quadro 4, a presente proposta aparenta ser tecnicamente viável.

Quadro 4 – Banda e taxa de dados para diversos perfis do *HomePlug AV 2.1*

Perfil	Banda [MHz]	Taxa de Dados [Mbps]	Modulação
SISO750	1.8 – 86,13	758	917-OFDM 4096-QAM
MIMO500	1.8 – 30	576	
MIMO1000	1.8 – 65	1162	

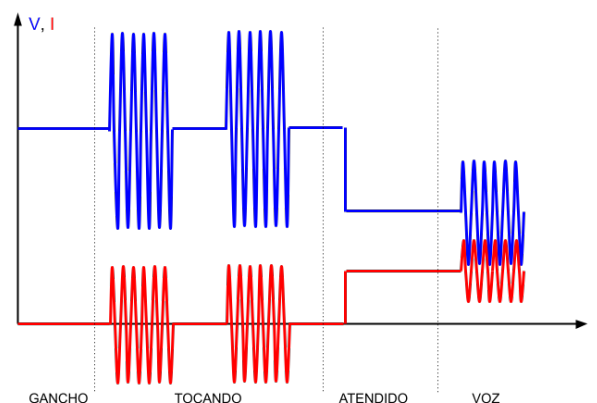
Fonte: adaptado de [HomePlug...](#) (2014).

### 3.2 Plain Old Telephone Service (POTS)

Os aparelhos de telefone implementam um protocolo chamado **POTS** (*Plain Old Telephone Service*, Serviço Telefônico Antigo Simples em inglês) para interagir com a operadora da linha. O presente projeto visa utilizar um aparelho de telefone convencional como interface de usuário, por isso deve implementar o mesmo protocolo, ou utilizar um *gateway VoIP* – um dispositivo que faz a ponte entre o presente protocolo analógico e o domínio digital, convertendo sinais **POTS** em mensagens **SIP** e **RTP**.

A linha telefônica fornece constantemente uma tensão contínua em torno de 45 VDC. O aparelho telefônico, no gancho, se comporta como um circuito aberto em corrente contínua. O circuito que emite o alarme sonoro quando há uma ligação (*ringer*), está sempre conectado, porém é desacoplado em DC. Para tocar o telefone, um sinal alternado na faixa de 40 - 130 Vrms a 20 Hz é sobreposto ao nível DC preexistente. Quando o telefone é retirado do gancho, passa a circular uma pequena corrente, cujo valor depende da resistência do aparelho em si ( $\sim 400 \Omega$ ) da linha ( $\sim 0,03 \Omega/m$ ) e da central telefônica ( $350 \Omega - 800 \Omega$ ). É pela presença dessa corrente que a central detecta essa condição ([BIGELOW; CARR; WINDER, 2001](#), p. 42-67). A [Figura 2](#) esquematiza a forma de onda resultante do protocolo descrito.

Figura 2 – Formas de onda simplificadas de tensão e corrente elétricas na linha telefônica

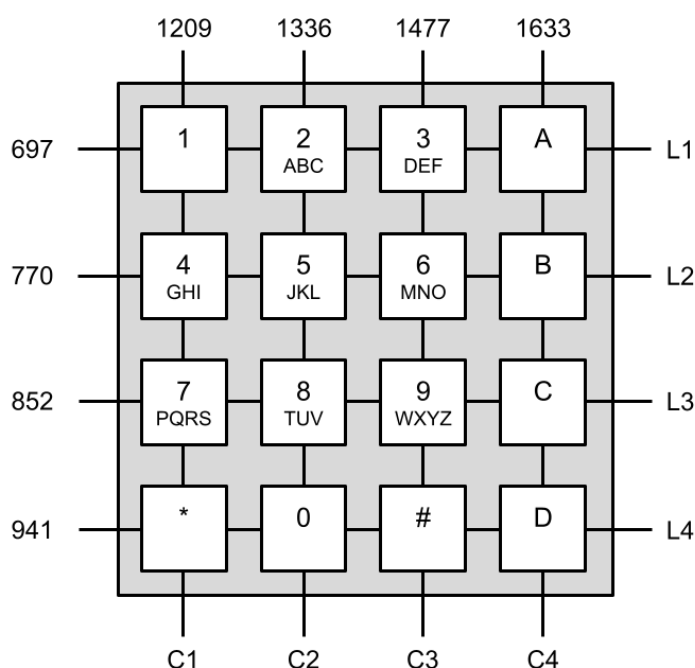


Fonte: o Autor, 2023.

### 3.3 Dual-Tone Multi-Frequency (DTMF)

O **DTMF** (*Dual-Tone Multi-Frequency*, Dois Tons de Múltiplas Frequências em inglês) é um método de codificação empregado pelos terminais telefônicos, que consiste em representar cada dígito do teclado por meio de dois tons com frequências padronizadas sobrepostos, conforme ilustrado na [Figura 3](#). Essas frequências estão compreendidas na banda da voz humana, permitindo que esses sinais sejam transmitidos pela linha telefônica ([BIGELOW; CARR; WINDER, 2001](#), p. 50).

Figura 3 – Diagrama de frequências (em Hz) correspondentes às teclas de discagem



Fonte: o Autor, 2023.

A central telefônica registra o dígito quando o par de tons é detectado por um período mínimo de 50 milissegundos. Para realizar a detecção, há algumas abordagens possíveis. Uma delas é pelo uso de filtros de banda estreita sintonizados nas frequências esperadas, com suas saídas conectadas a comparadores. Outro método, mais adequado a sistemas digitais, é utilizar análise espectral.

O espectro do sinal recebido na linha telefônica pode ser encontrado aplicando-se a transformada discreta de Fourier (DFT). Utilizando-se (3.6) transforma-se um vetor  $x[n]$  de  $N$  amostras igualmente espaçadas no tempo em um vetor  $X[k]$  de coeficientes de Fourier de mesmo tamanho ([OPPENHEIM; SCHAFER, 2010](#), p. 626).

A potência do sinal na frequência  $f_k$ , dada por (3.7), está associada a  $|X[k]|^2$  de

acordo com o teorema de Parseval ([OPPENHEIM; SCHAFER, 2010](#), p. 60).

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}nk} \quad (3.6)$$

$$f_k = f_s \frac{k}{N} \quad (3.7)$$

- $x[n]$  – Vetor de amostras do sinal
- $X[k]$  – Vetor de coeficientes de Fourier
- $N$  – Número de amostras
- $f_k$  – Frequência correspondente ao  $k$ -ésimo coeficiente
- $f_s$  – Frequência de amostragem

Nota-se que para se obter todas as  $k$  raias do espectro são necessárias  $nk$  operações de exponenciação (ou trigonométricas), de soma e de multiplicação. Como  $n = k$ , um algoritmo para esse cômputo tem complexidade  $O(n^2)$ , além de exigir que todas as amostras sejam mantidas na memória.

Um algoritmo alternativo mais eficiente é o de [GOERTZEL](#), que calcula  $X[k]$  como a saída de um sistema linear excitado pelas amostras  $x[n]$ . Por ser de segunda ordem, apenas duas amostras devem ser guardadas entre as iterações ([OPPENHEIM; SCHAFER, 2010](#), p. 721). O sistema de equações de diferenças que descreve o algoritmo é (3.8 - 3.9).

$$v_k[n] = 2\cos(2\pi k/N)v_k[n-1] - v_k[n-2] + x[n] \quad (3.8)$$

$$X[k] = v_k[N] - v_k[N-1]e^{-j\frac{2\pi}{N}k} \quad (3.9)$$

Há circuitos integrados que implementam esse algoritmo em hardware, facilitando o uso do DTMF. Um desses CIs é o MT8870, que é dotado de uma entrada analógica diferencial para o sinal telefônico e provê uma saída digital de 4-bits codificando a teclagem detectada.

Naturalmente, a principal finalidade do [DTMF](#) é a discagem do número chamado, porém também pode servir como [IHM](#) para: navegação pelos menus de sistemas de atendimento automatizado; entrada de dados (senha) em sistemas de controle de acesso baseados em telefonia, e acionamento de porteiro eletrônico.

Aparelhos telefônicos que operam por [VoIP](#) (e também os *gateways* que interconectam telefones [POTS](#) convencionais à rede [VoIP](#)) produzem, além do sinal sonoro, um pacote [RTP](#) suplementar contendo a notificação do mesmo evento, já em domínio digital.

Isso poupa os sistemas remotos da tarefa de analisar o fluxo de áudio e detectar eventos **DTMF** via processamento de sinais. Basta interpretar a mensagem **RTP** correspondente. Esse processo está normatizado na **RFC 4733** (SCHULZRINNE; TAYLOR, 2006).

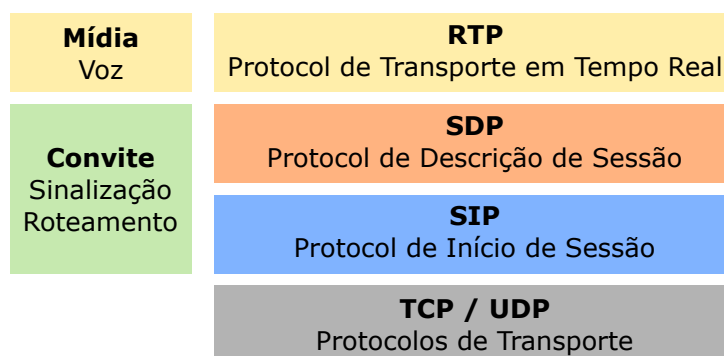
### 3.4 Pilha de Protocolos do VoIP

Com a crescente ubiquidade de redes **IP**, que podem utilizar os mais diversos meios para comunicação, como *Wi-Fi*, *Ethernet*, **GSM** ou mesmo **PLC**, e que são agnósticas ao teor dos dados que nelas trafega, é natural que as mais diversas aplicações abandonem o uso de infraestruturas dedicadas e passem compor o ecossistema de protocolos baseados em **IP**. O mesmo é válido para a telefonia, que transiciona do **PSTN** para o **VoIP**.

As principais normas que estabelecem padrões para o **VoIP** são a **ITU-T H.323** proposta pela **ITU**, já em desuso, e a mais comum **SIP**, mantida pela **IETF**. Esse último padrão é, na verdade, uma pilha de protocolos (vide **Figura 4**) que se complementam para formar uma solução integrada para telefonia **VoIP**:

Na camada mais baixa, os tradicionais **TCP** e **UDP** ficam a cargo do transporte de mensagens no protocolo **SIP** que, por sua vez, é responsável por permitir que contatos se anunciem e se encontrem na rede, convidem e aceitem uns aos outros para chamadas. Localizados os pares e acordado o intuito da sessão, cabe ao **SDP** propor e negociar formas de mídia suportadas pelos dispositivos para concretizar a comunicação. Finalmente, o **RTP** é utilizado para efetivamente trocar os pacotes de mídia.

Figura 4 – Diagrama das camadas que compõe a pilha **VoIP**



Fonte: o Autor, 2023.

## 3.5 Protocolos de Transporte

Sistemas de comunicação são arquitetados de maneira a estratificar diferentes níveis de responsabilidade em sucessivas camadas empilhadas, cada qual abstrai detalhes de baixo nível da implementação, oferecendo serviços às camadas superiores. Consoante ao modelo OSI, a camada de transporte é responsável pela entrega de dados independentes de aplicação ponta-a-ponta. Para isso, estão disponíveis dois protocolos distintos: O [TCP](#) e o [UDP](#).

O [TCP](#), normatizado na [RFC 9293](#) ([EDDY, 2022](#)) é orientado à conexão e confiável, isto é, ocorre um *handshake* entre as partes, e os pacotes são trocados com ordenamento rígido, confirmação de recebimento, e reenvio em caso de perda ou corrupção. Como resultado, a camada superior é provida de um fluxo de dados bidirecional e com garantia de integridade, o que torna o presente protocolo o mais adequado para aplicações onde deva haver incorruptibilidade na troca de informações, como transferência de arquivos, navegação na *web* e envio de e-mails. Em contrapartida, os mecanismos necessários para viabilizar tais garantias tornam o protocolo complexo, e por vezes ineficiente, acarretando assim o seu desuso em casos para os quais o fator tempo seja crítico.

Em oposição, o protocolo [UDP](#) – normatizado na [RFC 768](#) ([POSTEL, 1980](#)) – é orientado a datagramas: unidades básicas de informação trocadas de maneira independente umas das outras. Os datagramas podem ser perdidos, corrompidos, ou reordenados no percurso, tornando-os pouco confiáveis. A camada superior da pilha deve implementar mecanismos suplementares de [QoS](#) para viabilizar uma comunicação coerente. Por exemplo, o [RTP](#) o faz pela inclusão, na mensagem, de um número de sequência e carimbo de tempo. Apesar disso, a simplicidade, eficiência e flexibilidade do protocolo [UDP](#) o tornam ideal para aplicações em tempo real e que sejam naturalmente robustas à algum grau de incompletude. É o caso do [VoIP](#), onde a perda (ou descarte) esporádica de alguns datagramas, cada um correspondendo à poucos milissegundos de áudio, é geralmente imperceptível. Em casos mais extremos, a qualidade da chamada pode deteriorar, porém ainda assim ela não é inviabilizada, tendo em vista as estratégias de ocultação de erros nativas às codificações de mídia empregadas e demais recursos de [QoS](#) como renegociação da taxa de dados.



## 3.6 Session Initiation Protocol (SIP)

O SIP é um protocolo de comunicação de camada de sessão amplamente utilizado em sistemas VoIP para gerenciar o ciclo de vida das chamadas telefônicas e sessões de conferência. É um padrão IETF descrito na RFC 3261 (SCHOOLER et al., 2002).

O protocolo segue uma abordagem cliente-servidor, com troca de mensagens codificadas em texto ASCII, estruturadas em um cabeçalho padrão seguido (ou não) de um corpo de conteúdo variado. A primeira linha do cabeçalho das requisições contém um método e um endereço URI associado, conforme Quadro 5. Já a primeira linha das respostas indica um código de status, assim como exemplificado em Quadro 6. A lista completa dos métodos e códigos previstos na norma é extensa, e esses quadros apresentam um subconjunto apenas.

Quadro 5 – Alguns métodos de requisições SIP

Método	Descrição
REGISTER	Registra em um servidor de localização uma associação entre dados de contato e endereço de rede do UA
INVITE	Convida um participante para uma chamada (nova ou existente)
ACK	Notifica que a chamada foi aceita e o canal de mídia está pronto para uso
CANCEL	Cancela um convite em progresso antes do seu aceite
BYE	Finaliza uma sessão em andamento

Fonte: adaptado de RFC 3261 (SCHOOLER et al., 2002).

Quadro 6 – Alguns códigos de resposta SIP

Código	Descrição
180 Ringing	O telefone do contato remoto está chamando
200 OK	Uma solicitação de chamada foi atendida
401 Unauthorized	Esse pedido requer autenticação e o cliente deve fornecer credenciais
404 Not Found	O contato chamado (número de telefone) não existe
486 Busy Here	Ocupado

Fonte: adaptado de RFC 3261 (SCHOOLER et al., 2002).

As linhas subsequentes do cabeçalho das mensagens SIP contém pares *nome: valor* (Quadro 7), indicando o tipo de tamanho do conteúdo do corpo, identificação da origem e destino, bem com dados de roteamento para direcionar a mensagem.

Essas mensagens podem ser transportadas tanto por **UDP** quando por **TCP**, ou mesmo por ambos os protocolos em diferentes trechos da rota entre dois pares de **UAs**. Isso é possível pois tais mensagens contém dados próprios de roteamento e são autossuficientes nesse aspecto. À medida em que uma mensagem é propagada pela rede, cada nó *proxy* estima o próximo salto – um palpite sobre qual o próximo nó que levará a mensagem para mais perto do destinatário final – e anexa à mensagem um campo **Via** relativo a sí próprio. Quando a mensagem finalmente chega ao seu destino final, e uma resposta deve ser devolvida ao remetente, essa coleção de **Vias** que descreve exatamente o caminho traçado é então utilizada, em reverso, para encaminhá-la à origem.

Quadro 7 – Campos do cabeçalho de mensagens **SIP**

Campo	Descrição
Via	Identifica o caminho percorrido pela mensagem utilizando endereços <b>IP</b> e/ou nomes de domínio dos nodos envolvidos no roteamento
From	Especifica o remetente da mensagem: nome de exibição e endereço único ( <b>URI</b> )
To	Indica o destinatário da mensagem
Contact	Informa como o remetente da mensagem pode ser contatado, contendo um endereço <b>SIP</b> e informações de transporte.
Authorization	Contém a resposta de um desafio criptográfico utilizado para autenticar um cliente
Call-ID	Identifica de forma única a chamada a qual pertence essa mensagem
CSeq	Número de sequência para cada mensagem em uma chamada, permitindo ordenamento e detecção de perda
Content-Type	Indica o tipo de mídia do corpo da mensagem, como por exemplo uma descrição <b>SDP</b>
Content-Length	O tamanho em <i>bytes</i> do corpo da mensagem.

Fonte: adaptado de **RFC 3261 (SCHOOLER et al., 2002)**.

O protocolo oferece mecanismos para que usuários autenticuem-se em servidores de registro de maneira a expôr uma associação entre informações de contato, como nome de usuário, número de telefone, ou e-mail, e a localização atual onde o cliente pode ser contatado (endereço de rede, porta). Uma infraestrutura de servidores de registro e de *proxy* é utilizada para encontrar e rotear mensagens **SIP** entre **UAs**. Para elucidar a troca de mensagens **SIP** envolvida no registro de um cliente, é apresentado no **Apêndice A** um diagrama de sequência do registro na **Figura 14**.

Naturalmente, o objetivo final do protocolo é estabelecer e gerenciar sessões de

chamadas. Isso é realizado, simplificadaamente, através do envio, pelo chamador, de uma requisição **INVITE**, a qual é respondida por uma confirmação **200 OK**. Ambas estas mensagens **SIP** portam em seus corpos sub-mensagens **SDP** descrevendo formatos e codificações de mídia suportados, bem como protocolos de transporte a serem empregados para concretizar a chamada. Esse fluxo é explicado em maior detalhe no **Apêndice A** em um diagrama de sequência de transação de chamada na **Figura 15**.

A **RFC 3665** (**DONOVAN et al., 2004**) contém uma coleção extensa de exemplos similares.

### 3.7 *Session Description Protocol (SDP)*

O **SDP** é um protocolo utilizado no contexto do **VoIP** para estabelecer e negociar as características do canal de dados de uma sessão de comunicação. As mensagens **SDP** são transmitidas pelo chamador no corpo das requisições **SIP INVITE** e pelo receptor da chamada no corpo da resposta **200 OK**. Uma vez que a sessão seja iniciada pelo **SIP**, as informações trocadas e acordadas utilizando o **SDP** são utilizadas pelos dispositivos para estabelecer uma conexão direta e transmitir o áudio da chamada. Ela descreve os parâmetros e opções para que as partes troquem pacotes de voz, tais como *codecs* de áudio, endereços **IP**, portas e outras informações relevantes: capacidades e requisitos dos dispositivos, taxa de amostragem, tamanho do pacote de áudio, etc.

Conforme normatizado na **RFC 4566** (**PERKINS; HANDLEY; JACOBSON, 2006**), a descrição **SDP** é formatada como uma série de campos no formato *nome=valor*, um por linha, onde o nome é composto por um único caractere minúsculo. Para facilitar a interpretação, os campos mantêm uma ordenação fixa.

Quadro 8 – Parâmetros da descrição SDP

Sigla	Parâmetro	Descrição
v=	Versão	Versão do protocolo
o=	Origem	Nome de usuário, ID da sessão, tipo de rede e endereço
s=	Nome	Nome de exibição da sessão
i=	Info	Descrição textual da sessão
u=	URI	URL para algum recurso relevante da sessão
e=	E-Mail	Endereço de e-mail do anfitrião
p=	Fone	Número de telefone do anfitrião
c=	Conexão	Dados de conexão, como protocolo e endereço
t=	Tempo	Horário de início e término de sessões agendadas
m=	Mídia	Tipo de mídia (áudio, vídeo...), número da porta, protocolo de transporte, formato (codificação) da mídia
a=	Atributo	Informações adicionais no formato nome:valor

Fonte: adaptado de RFC 4566 (PERKINS; HANDLEY; JACOBSON, 2006).

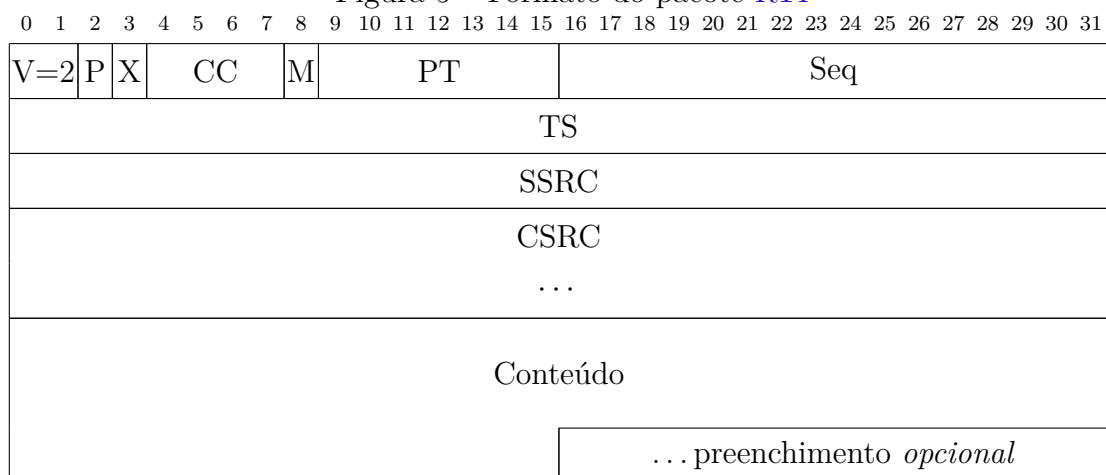
### 3.8 Real-Time Transport Protocol (RTP)

O RTP é um protocolo de transporte voltado à transmissão de mídia em tempo real, que opera sobre o UDP e complementa-o com metadados adicionais sobre a mídia sendo transportada. Sobretudo, contém informações de ordenamento e temporização que permitem o correto escalonamento das amostras na fila de reprodução. Opera por meio de pacotes em binário, formatados como apresentado em Figura 5. O protocolo RTCP é usado paralelamente ao RTP, na porta de número subsequente à do fluxo principal, para sinalização e QoS *out-of-band*. Ambos estão normatizados na RFC 3550 (SCHULZRINNE et al., 2003).

Nota-se que o cabeçalho do RTP é agnóstico ao tipo de mídia sendo transportado. Apenas é definido um código identificador para o tipo de carga útil, que deve ser interpretado pela aplicação que irá reproduzir o *stream*. O SDP utiliza o campo de atributo *rtptime* para realizar essa associação entre PT e o *codec*. Para o VoIP, entretanto, a RFC 3551 (CASNER; SCHULZRINNE, 2003) define uma tabela padrão de códigos para *codecs* de voz e vídeo comuns (Quadro 9).

O item 101 é de especial importância para o presente projeto. Pelo recebimento e interpretação de pacotes RTP contendo o referido PT, o *software* desenvolvido poderá interceptar eventos DTMF, de modo a utilizá-los como mecanismo de interação com os usuários (IHM). O formato de tal carga útil é normatizado pela RFC 4733 (SCHULZRINNE; TAYLOR, 2006) e está representado na Figura 6. Valores na faixa 0...15 do campo EVT

Figura 5 – Formato do pacote RTP



Fonte: adaptado de RFC 3550 (SCHULZRINNE et al., 2003).

- V* – Versão do protocolo
- P* – Indica se há *bytes* de preenchimento no final do pacote
- X* – Indica se há um cabeçalho de extensão antes da carga útil
- CC* – Número de identificadores CSRC presentes
- M* – Marcador: sua interpretação depende do tipo de conteúdo transportado
- PT* – Tipo do conteúdo (por exemplo, qual o *codec* de áudio)
- Seq* – Número de sequência do pacote, para ordenamento
- TS* – Carimbo de data/hora, em unidade que depende do tipo de mídia
- SSRC* – Número aleatório que identifica a fonte da mensagem
- CSRC* – Lista de identificadores de fontes contribuintes (para o caso de mixagem)

correspondem, respectivamente, aos dígitos 0...9, e caracteres \*, # e A...D no teclado telefônico.

Quadro 9 – Alguns tipos de carga útil utilizados com RTP

PT	Codec	Amostragem [Hz]	Dados [kbps]	Descrição
0	G.711 U	8000	64	PCM $\mu$ -Law
1	FS-1016	8000	4,8	CELP
2	G.726	8000	32	ADPCM
3	GSM-FR	8000	13	LPC
4	G.723	8000	5,3 – 6,3	ACELP
8	G.711 A	8000	64	PCM A-Law
101	DTMF	8000	-	<i>telephone-event</i>

Fonte: adaptado de RFC 3551 (CASNER; SCHULZRINNE, 2003).

Figura 6 – Formato do pacote *telephone-event*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
EVT								E	R	VOL								DUR													

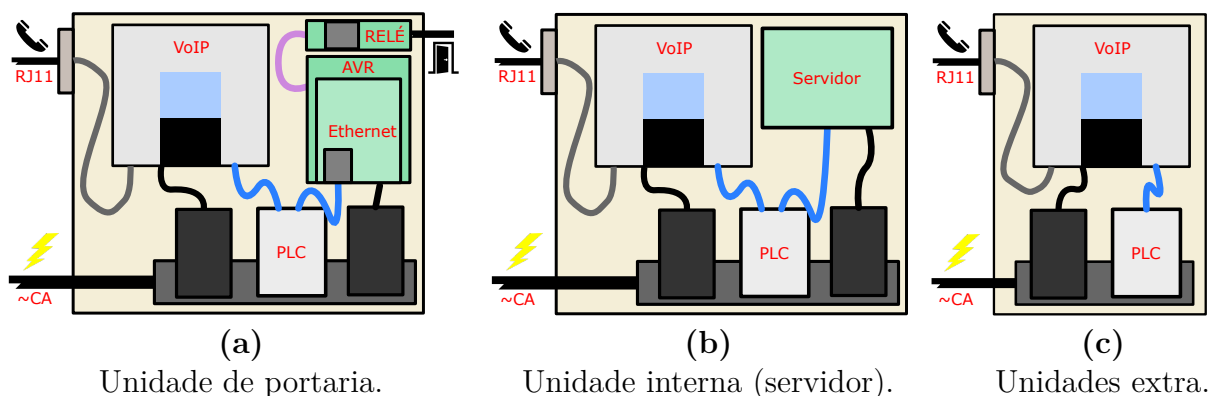
Fonte: adaptado de [RFC 3550](#) (SCHULZRINNE et al., 2003).

- EVT* – Código do evento
- E* – Indica que esse pacote corresponde ao término de um evento
- R* – Reservado
- VOL* – Volume [dBm, módulo]
- DUR* – Duração do evento [unidade de [TS](#)]

## 4 Materiais e Métodos

A especificação dos materiais e métodos que segue é balizada por uma arquitetura de alto nível pretendida para o sistema, tal como representado na [Figura 7](#). O protótipo consiste de duas unidades (a) e (b), ambas contendo um *modem* PLC, para comunicação entre elas, e um *gateway* VoIP para interface com o telefone convencional. A unidade utilizada na portaria se diferencia por possuir um relé, dedicado ao acionamento do controle de acesso, e um microcontrolador AVR para comandá-lo remotamente. Já a unidade para uso interno ao condomínio possui, ao invés disso, um computador embarcado atuando como servidor. Apenas essas duas primeiras unidades são obrigatórias para um sistema mínimo, porém este poderia ser expandido pela adição de unidades internas extras mais simples (c), desprovidas do servidor, uma vez que apenas um servidor é necessário na rede.

Figura 7 – Ilustração do conceito dos protótipos



Fonte: o Autor, 2023.

### 4.1 Premissas

Partindo do pressuposto de que o presente projeto não visa inicialmente ao desenvolvimento de um produto de consumo, mas trata-se de uma solução *ad hoc* para casos onde seria oneroso o emprego da tecnologia tradicional para a instalação de um circuito de interfonia condominial: não se justifica o investimento em P&D de um circuito eletrônico específico; ao invés disso, a abordagem metodológica a ser seguida é a de integração de subsistemas, sobretudo pelo desenvolvimento do *software* de ligação.

Tal *software* deve, entretanto, ser suficientemente portátil e flexível para permitir seu reaproveitamento na hipótese de evolução do presente conceito ao patamar de produto. Indiferente desse desfecho, o mesmo está disponível à ampla comunidade como código aberto, sob licença permissiva, de modo a colaborar com a iniciativa extensionista.

## 4.2 Hipóteses

Pela construção do protótipo, deseja-se avaliar a viabilidade técnica do emprego da tecnologia PLC como canal de comunicação para aplicações VoIP e de controle de acesso em um circuito residencial ou condominial. Pretende-se determinar a robustez e capacidade de banda do PLC nesse cenário. O desenvolvimento de *software* realizado procurará evidenciar a adequação e flexibilidade dos protocolos da pilha VoIP para tais finalidades. No intuito de responder aos questionamentos levantados, serão conduzidos os experimentos descritos na seção 4.6.

## 4.3 Ferramentas de Software

O presente projeto é intensivo em desenvolvimento de *software*. Essa seção apresenta as ferramentas computacionais preconizadas para subsidiar esse processo.

### 4.3.1 Linguagem de Programação

O ponto de partida é a escolha das linguagens de programação utilizadas: o C/C++. Essas linguagens são conhecidas por seu alto desempenho, portabilidade e capacidade de uso em plataformas embarcadas. Por gerar código de máquina nativo, ou seja, compilado diretamente para instruções específicas da arquitetura do processador de destino, é obtido desempenho superior em comparação com linguagens interpretadas ou com *bytecode* intermediário. Na verdade, essas últimas alternativas estariam prontamente descartadas em plataformas sem sistema operacional (*bare-metal*).

Além disso, a possibilidade de manipulação direta de memória permite a escrita de um *software* otimizado também em relação ao consumo de memória e armazenamento, recursos esses que podem ser escassos. Outrossim, o ambiente C/C++ oferece uma ampla variedade de bibliotecas *open source* de alto nível de maturidade que podem ser facilmente



integrados ao projeto, de maneira a aproveitar funcionalidades preexistentes e previamente testadas pela comunidade, assim acelerando o desenvolvimento e melhorando a qualidade do produto final. Por exemplo, visando fundamentar-se em uma base robusta para comunicação [UDP](#), e assim poder focar em desenvolver as camadas superiores da solução, o presente projeto empregou a biblioteca [BOOST](#): um tradicional complemento à biblioteca padrão do *C++*, e que oferece, entre muitos outros recursos, APIs multiplataforma para uso de *sockets*.

### 4.3.2 Sistema de *Build*

Apesar das inúmeras vantagens da linguagem, a compilação de projetos em *C++* pode ser um processo complexo. O código geralmente é dividido em vários arquivos fontes que precisam ser compilados individualmente de maneira correta e posteriormente vinculados entre si. Quando se trata de projetos maiores, gerenciamento de dependências de bibliotecas agrava ainda mais esse desafio.

Tradicionalmente, os projetos *C++* eram compilados usando arquivos de *script* personalizados – os *makefiles* – ou por meio de arquivos de projetos da [IDE](#). Tais sistemas de *build* apresentam forte dependência do compilador e sistema operacional utilizados, muitas vezes exigindo uma configuração manual detalhada para cada projeto, plataforma alvo e ambiente hospedeiro. Isso acaba resultando em um processo trabalhoso de executar, difícil de manter e penaliza a portabilidade.

Nesse contexto, o *CMake* vem despontando como uma ferramenta capaz de sanar tais percalços. Esse *software* de código aberto utiliza uma descrição declarativa da estrutura do projeto, suas dependências, e algumas poucas propriedades indicando como ele deve ser compilado. Essa forma de descrição abstrai detalhes do compilador, do sistema operacional e ambiente de desenvolvimento, promovendo maior portabilidade e proporcionando flexibilidade ao desenvolvedor para escolher e combinar esses três aspectos livremente ([SWIDZINSKI, 2022](#)). Essas descrições são entradas para o *CMake*, que as traduz levando em conta as especificidades do ambiente escolhido e então gera, como artefatos de saída, diversos arquivos de projeto compatíveis com várias [IDEs](#) populares, como o *Visual Studio*, *Xcode* e *Eclipse* ou mesmo sistemas de *build* puros tais como *makefiles* ou *Ninja*. Isso permite que os desenvolvedores trabalhem com as ferramentas de sua preferência, facilitando a colaboração e a manutenção do projeto. O *CMake* lida com a complexidade da

compilação do *C++* de forma transparente, minorando a sobrecarga nos desenvolvedores e outorgando-lhes pleno foco em tarefas efetivamente agregadoras de valor.

### 4.3.3 Documentação

Um código-fonte é escrito uma única vez, alterado algumas vezes, e lido muitas vezes mais. É essencial documentá-lo para ajudar a entender sua estrutura, funcionalidades e propósito visando ao longo prazo. A documentação do código-fonte oferece uma visão geral do sistema, explicando suas interfaces bem como o seu funcionamento interno: dependências, fluxo de dados e lógica subjacente. Ela permite que os desenvolvedores entendam rapidamente o propósito de cada componente, classe ou função, e ajuda assim a evitar extensos trabalhos de investigação na tentativa de redescobrir o funcionamento de códigos legados. Além disso, a documentação adequada pode facilitar a depuração e o teste, fornecendo informações detalhadas sobre as entradas, saídas e comportamento esperado de cada parte do sistema.

Uma ferramenta popular para a esse tipo de documentação é o *Doxygen*. Trata-se de um extrator de anotações especiais distribuídas pelo código. Os desenvolvedores inserem no código-fonte comentários seguindo uma sintaxe específica, para descrever classes, métodos, variáveis e outros elementos. Esses comentários são encontrados, analisados e agregados pelo *Doxygen*, culminando em uma compilação da documentação em uma gama de formatos, como  $\text{\LaTeX}$ , PDF, ou mesmo uma página [HTML](#) navegável com referências cruzadas entre partes relacionadas do sistema. Além da documentação textual, são gerados diagramas de herança de classes no padrão UML, de colaboração, gráficos de chamadas e outros recursos úteis para visualizar a estrutura do código.

### 4.3.4 Versionamento

O versionamento do código-fonte é uma prática essencial no desenvolvimento de software, que consiste no controle e gerenciamento de alterações ao longo do tempo. Esse histórico facilita encontrar alterações indesejadas e revertê-las, por exemplo, para encontrar em que momento fora introduzido um erro latente que se observou apenas mais tarde. Outro recurso importante dos sistemas de versionamento é o mecanismo de ramificações, que permite que desenvolvedores trabalhem de forma colaborativa e simultânea, submetendo contribuições incrementalmente em ramos concorrentes do repositório. As versões paralelas

podem ser então comparadas e re combinadas no tronco, com gerenciamento e resolução de conflitos. O mesmo vale para desenvolvedores trabalhando individualmente, porém hora em um ramo, hora em outro.

Alguns sistemas de versionamento oferecem ainda fluxos de trabalho para revisão e aceitação de contribuições, por exemplo, exigindo que outros desenvolvedores revisem as alterações submetidas em um ramo antes que ele seja mesclado ao tronco. Esse tipo de gestão majora a maturidade da governança de projetos em equipes.

O *Git* é um dos sistemas de controle de versão mais populares e amplamente utilizados atualmente. Adotando uma abordagem de versionamento distribuído, cada desenvolvedor possui uma cópia completa do repositório (com histórico) em seu sistema de arquivos local. Isso permite que os desenvolvedores trabalhem desconectados e façam alterações em suas versões locais livremente e sem afetar o trabalho dos demais. Por isso, também há uma maior robustez contra indisponibilidades de servidores. As alterações locais podem ser sincronizadas com um servidor posteriormente, momento este em que são aplicadas as políticas de governança do repositório, por exemplo, a abertura de um fluxo de trabalho *pull request* para aprovação da mesclagem (PONUTHORAI, 2022). Além disso, o *Git* possui um ecossistema rico de ferramentas e serviços que o complementam. Existem plataformas populares e gratuitas, por exemplo, [GitHub](#), [GitLab](#) e [Bitbucket](#), que fornecem recursos adicionais, como hospedagem de repositórios, *pipelines* de CI/CD, fóruns integrados para discussão, rastreamento de problemas e revisão de código. É um dos pilares da comunidade de desenvolvedores de *software* de código aberto.

### 4.3.5 Depuração e Perfilamento

No desenvolvimento de aplicações que se valham de comunicação por rede, é imprescindível dispor de uma ferramenta de análise e depuração de tráfego. Um capturador de pacotes fornece *insights* valiosos sobre a origem e o destino das mensagens, seus tamanhos, os tempos de transmissão, latência, conteúdo da carga útil, etc. Essas informações podem ser utilizadas para investigar problemas de comunicação, identificar gargalos de desempenho, verificar a conformidade dos protocolos utilizados e melhorar a segurança do sistema.

O *Wireshark* é um exemplo de ferramenta amplamente utilizada nesse contexto. Trata-se de um *software* de código aberto que realiza a captura, armazenamento, interpre-




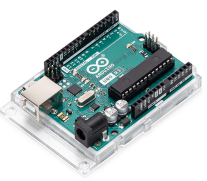


tação e análise em tempo real de pacotes trafegando pela rede. Os pacotes recebidos pela NIC são exibidos em uma linha do tempo. Para uma variedade de protocolos suportados, incluindo [HTTP](#), [SIP](#), [SDP](#) e [RTP](#), a estrutura interna das mensagens é interpretada e os dados são exibidos discriminados por campo ([JAIN, 2022](#)).

Outra ferramenta de análise é o [iPerf](#). Projetado para perfilar comunicações [TCP](#) ou [UDP](#) entre dois pontos, utiliza uma abordagem cliente-servidor, onde uma instância do aplicativo deve estar executando em cada um desses. Pode ser utilizado para medir perda de pacotes, taxa de dados, latência etc.

## 4.4 Materiais

Os protótipos utilizam um par de *modems* [HomePlug TP-Link AV600](#) para o enlace [PLC](#). Telefones convencionais serão conectados por intermédio de dois *gateways* [VoIP LinkSys PAP2T](#). Essa decisão de projeto visa a maximizar a facilidade de migração a partir de um sistema analógico, dado que os telefones antigos poderiam ser reutilizados. Na unidade interna, o *software* do *servidor proxy* [VoIP](#) executará em um ambiente *Linux ARM* (Ubuntu Server) em um [SBC Raspberry Pi 3 B](#). Na unidade de portaria, um [relé 12V](#), montado em trilho [DIN](#), será comandado pelo um microcontrolador [ATmega328p](#) convenientemente disponível na placa de prototipagem [Arduino Uno](#). Nele será hospedado o *firmware* de controle via [HTTP](#). Um módulo *ethernet* [W5500](#) permitirá sua conexão com tal rede de dados.

Quadro 10 – Resumo da lista de materiais

Imagem	Modelo	Descrição
 <p>Fonte: <a href="http://tp-link.com">tp-link.com</a>.</p>	TP-Link AV600	<p>Modem adaptador de PLC para Ethernet e Wi-Fi:</p> <p>PLC: HomePlug AV/AV2 600Mbps</p> <p>Alcance: 300 m</p> <p>LAN: 2xRJ45 10/100M</p> <p>Wi-Fi: IEEE802.11b/g/n</p>
 <p>Fonte: <a href="http://cisco.com">cisco.com</a>.</p>	LinkSys PAP2T	<p>Gateway VoIP padrão SIP:</p> <p>Codecs: G.711 A, G.711 U G.726, G.723 G.729</p> <p>Ringler: 40-55 VRMS 10-40 Hz</p> <p>Impedância: 600Ω</p> <p>POTS: 2xRJ11</p> <p>LAN: 1xRJ45</p>
 <p>Fonte: <a href="http://raspberrypi.com">raspberrypi.com</a>.</p>	Raspberry Pi 3 B	<p>Computador SBC com:</p> <p>CPU: ARM Cortex-A53 Quad-Core 1.2GHz</p> <p>GPU: VideoCore IV 400MHz</p> <p>RAM: 1GB DDR3</p> <p>LAN: 1xRJ45 10/100M</p>
 <p>Fonte: <a href="http://arduino.cc">arduino.cc</a>.</p>	Arduino Uno	<p>Placa de prototipagem com:</p> <p>MCU: AVR ATmega328P</p> <p>Clock: 16 MHz</p> <p>SDRAM: 2 KB</p> <p>Flash: 32 KB</p> <p>EEPROM: 1 KB</p>
 <p>Fonte: <a href="http://wiznet.io">wiznet.io</a>.</p>	Shield W5500	<p>Módulo Ethernet para Arduino:</p> <p>N. Conexões: 8 simultâneas</p> <p>Buffer: 32 KB</p> <p>Interface: SPI</p> <p>LAN: 1xRJ45 10/100M</p>
 <p>Fonte: <a href="http://3pinus.com.br">3pinus.com.br</a>.</p>	Relé AA1R	<p>Relé de interface entre MCU e atuador do controle de acesso:</p> <p>Tensão de bobina: 12 VDC</p> <p>Corrente de bobina: 25 mA</p> <p>Tensão nominal: 220 V</p> <p>Corrente nominal: 15 A</p> <p>Fixação: DIN 35 mm</p>

Fonte: compilado pelo Autor, 2023.

## 4.5 Atendimento de Normas

A [Resolução nº 527, de 8 de abril de 2009](#) da ANATEL estabelece as condições de uso de radiofrequências para o sistema de “banda larga por meio de redes de energia elétrica” (BPL), em especial quanto às radiações indesejadas causadas por estes sistemas, e que somente poderá ocorrer na faixa de 1,705 MHz a 50 MHz. A [Resolução Normativa nº 375, de 25 de agosto de 2009](#) da ANEEL regulamenta o uso das redes de distribuição de energia elétrica pelas concessionárias de distribuição de energia e provedores de acesso. Nota-se que a legislação vigente versa sobre uso do PLC para provimento de acesso à internet banda larga por operadoras, e não se aplica ao caso de uso do presente projeto, pois este opera em banda estreita e confinado às instalações residenciais e condominiais. O *modem PLC* empregado utiliza o padrão [HomePlug](#) está homologado na ANATEL com o certificado de número [03274-14-03177](#).

## 4.6 Experimentos

Para caracterizar o sistema prototipado, alguns ensaios são conduzidos:

**Experimento 1** [Análise da ocupação](#) de recursos do microcontrolador, de modo a validar a viabilidade de uso do *hardware* escolhido para a aplicação. Esses dados são disponibilizados pelo compilador após a construção do executável.

**Experimento 2** [Medição do tempo de resposta do servidor HTTP](#) embarcado no microcontrolador, que é responsável por acionar o controle de acesso mediante o comando [DTMF](#), e prover uma interface de administração. Será utilizado o [Wireshark](#) para medir o intervalo entre requisição e resposta, e assim aferir a performance do *software* desenvolvido.

**Experimento 3** Utilizando o [iPerf](#), será avaliada a [transferência de dados](#) por [UDP](#) via [PLC](#) em várias distâncias. A taxa de dados obtida será comparada com os requisitos dos *codecs* na [Quadro 9](#) de modo a certificar a adequação de enlace [PLC](#) para esse uso.

**Experimento 4** Para avaliar a robustez e a confiabilidade do servidor [VoIP](#) desenvolvido, será conduzido um [ensaio de repetibilidade de chamadas](#). Ligações serão

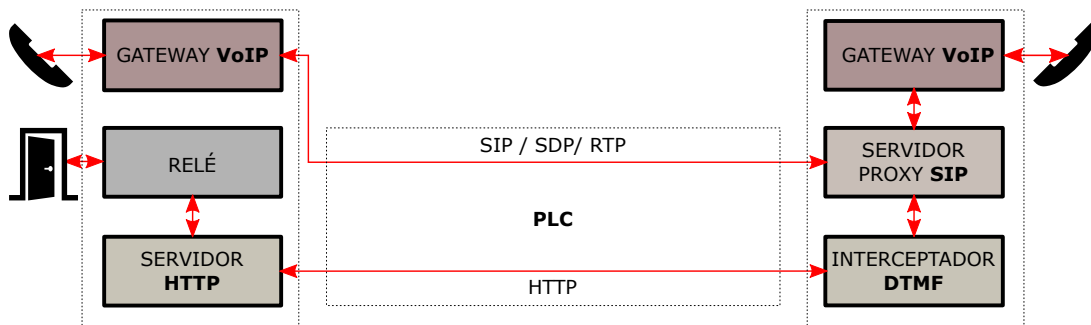
realizadas diversas vezes, em ambas as direções, e será registrada a taxa de sucesso.

**Experimento 5** Similarmente, pretendendo avaliar a robustez do acionamento do controle de acesso, será ensaiada a [repetibilidade de comando do relé](#). Repetidos comandos serão emitidos, e será registrada a taxa de sucesso.

## 5 Desenvolvimento do Projeto

O produto divide-se em blocos funcionais que se comunicam conforme ilustrado na [Figura 8](#). A peça fundamental do sistema é um servidor *proxy* SIP que permite aos telefones (por intermédio dos *gateways*) chamarem uns aos outros. Esse mesmo servidor atua como interceptador da comunicação, e analisa os pacotes RTP em busca de eventos DTMF. Ao detectar o pressionar de uma tecla específica, é emitida uma requisição HTTP que comanda o acionamento do relé de controle de acesso na unidade remota. Todo esse fluxo de dados ocorre pelo canal PLC.

Figura 8 – Diagrama de fluxo de dados entre blocos do sistema



Fonte: o Autor, 2023.

### 5.1 Desenvolvimento do Servidor VoIP

O servidor VoIP padrão SIP ora desenvolvido desempenha as seguintes funções:

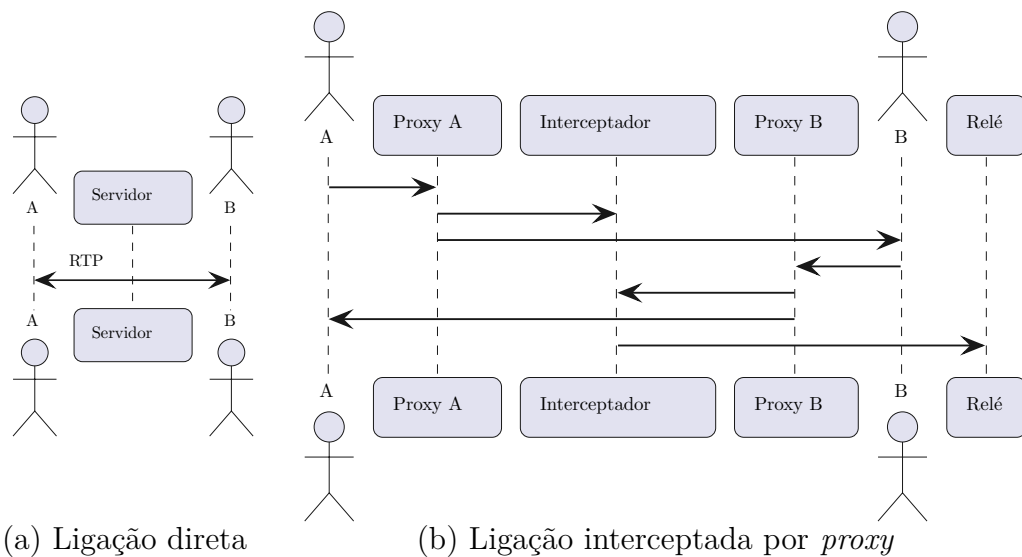
1. Tratar os pedidos de registro dos *gateways* telefônicos, gravando uma associação entre cada número telefônico e o respectivo endereço IP e porta para onde ligações devem ser roteadas de modo a chamar aquele telefone.
2. Receber mensagens SIP, interpretá-las, localizar no registro do item (1) o endereço IP e porta do destinatário da mensagem, e encaminhá-la ao destino.
3. Interceptar eventos DTMF no fluxo RTP, e tratar a discagem do # realizando uma requisição HTTP para o relé de controle remoto descrito na [seção 5.2](#) de modo a acioná-lo.



As primeiras duas funcionalidades não são especiais, e são encontradas nos servidores SIP amplamente utilizados no mercado, tais como o Asterisk ou outros citados em seção 2.3. Entretanto, o ponto (3) não é usual, e é o que justifica a necessidade de criação de um servidor dedicado à presente aplicação. O servidor SIP costuma estar envolvido apenas na etapa de negociação da sessão (chamada), após a qual o fluxo RTP é estabelecido par-a-par. Como os pacotes de mídia não transitam pelo servidor central, isso inicialmente impede que os eventos DTMF sejam interceptados e utilizados para qualquer lógica. A descrição de sessão SDP compartilhada na negociação contém os endereços IP e portas do chamador e do chamado, e nenhuma informação relativa ao servidor, que geralmente limita-se apenas a encaminhar tais mensagens de maneira transparente.

Para possibilitar a interceptação do RTP o presente servidor, diferente dos demais, realiza a reescrita do SDP contido nas requisições INVITE e respostas 200 OK. Os dados de conexão dos UAs são substituídos pelo endereço IP do servidor e uma porta alocada no ato da chamada antes do encaminhamento da descrição SDP ao destinatário. No mesmo ato, o servidor instancia dois proxies UDP configurados para escutar no endereço e porta inseridos na mensagem SDP, e em seguida encaminhar os datagramas recebidos para a terminação que originalmente constava na mensagem que foi reescrita.

Figura 9 – Diagrama de sequência de uma chamada interceptada por servidor proxy



Fonte: o Autor, 2023.

Como resultado, a chamada passa a ser intermediada por esse servidor, com todo o fluxo de pacotes de mídia acessível para tratamento e uso da lógica de IHM por meio de DTMF. Efetivamente, uma chamada  $A \leftrightarrow B$  é dividida em duas:  $A \leftrightarrow S$  e  $S \leftrightarrow B$ , conforme

exemplificado na [Figura 9](#).

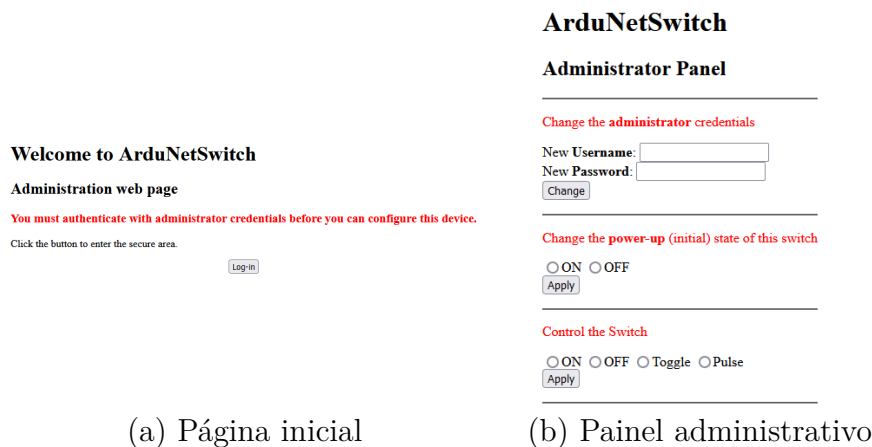
Para implementar o *software* descrito foi utilizada uma arquitetura de classes conforme consta no [Apêndice B](#). A [Figura 19](#) traz o diagrama de sequência relativo à lógica supracitada, mencionando as classes envolvidas. Os detalhes da implementação são demasiado extensos para aprofundamento no presente documento, porém estão documentados por [Doxygen](#), disponível no repositório público [Git](#) da biblioteca, cujo *link* consta no [Capítulo 7](#).

## 5.2 Desenvolvimento do Relé Controlado via Rede

Para acionamento do controle de acesso, é utilizado um relé comandado por um microcontrolador que hospeda um servidor *web*. O acionamento se dá por meio de uma requisição [HTTP](#) em um endereço específico. O cabeçalho dessa requisição deve conter um campo de autenticação com usuário e senha, que são validados comparando-os com as credenciais armazenadas na memória [EEPROM](#).

Uma interface de administração está disponível para acesso pelo navegador (vide [Figura 10](#)), também mediante autenticação com usuário e senha. Por esse painel administrativo é possível redefinir tais credenciais, configurar o estado padrão do relé (normalmente aberto/normalmente fechado) e também emitir comandos de acionamento.

Figura 10 – Capturas de tela das páginas *web* do servidor do relé comandado por rede



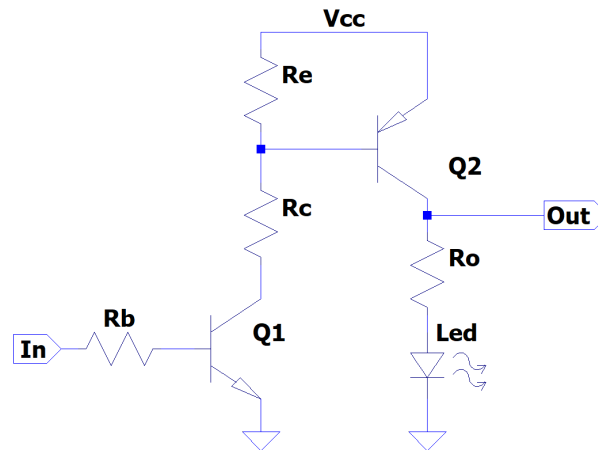
(a) Página inicial

(b) Painel administrativo

Fonte: o Autor, 2023.

Um pequeno circuito, conforme esquematizado na [Figura 11](#), faz a mudança de nível lógico (de 5V para 12V) e provê ganho de corrente entre o pino de saída digital do microcontrolador e a bobina do relé.

Figura 11 – Esquemático do circuito de acionamento do relé



Fonte: o Autor, 2023.

### 5.2.1 Algoritmo do Servidor *Web*

O adaptador *ethernet W5500* acompanha uma biblioteca própria que implementa funções básicas da camada **TCP**, como a escuta, abertura e fechamento de conexões, e a leitura/escrita de dados no fluxo. Já o *software* do servidor **HTTP** foi desenvolvido especialmente para a presente aplicação.

O **Apêndice C** apresenta os diagramas de atividade relativos ao seu funcionamento, que é explicado de maneira simplificada na sequência:

Inicialmente, a linha de execução fica presa em um ciclo que espera a obtenção do endereço **IP** por meio de DHCP e, em seguida, adentra o ciclo principal. Nesse, aguarda-se uma nova conexão de um cliente, e então a respectiva requisição é lida da FIFO *byte* por *byte*, cada um salvo em um *buffer*. A mensagem é interpretada e tratada durante sua leitura, evitando a necessidade de armazená-la por completo na memória. Esse processamento se dá por meio de uma máquina de estados. Quando um estado completa a leitura e interpretação de um fragmento lógico da mensagem, o *buffer* atual é descartado e passa a auxiliar na interpretação do próximo fragmento. O primeiro estado lê e interpreta o método (*get*, *post*, etc...) da requisição e a **URL**. O endereço é comparado com as páginas conhecidas e salvo em um enumerado. O texto é descartado. Quando a sequência **CRLF** é lida, ocorre a transição para o próximo estado: leitura dos campos. Nesse estado, os octetos são lidos e *bufferizados* até que seja encontrada a sequência **CRLF**, que delimita o final da linha. Então, a cadeia de caracteres é iterada até encontrar o delimitador dois-pontos, de maneira a separar o nome e o valor do campo. Alguns campos têm tratamento especial e têm seus

valores salvos, como é o caso do campo que especifica o tamanho do corpo da mensagem que segue após o cabeçalho. Outro exemplo é o campo que contém usuário/senha de autenticação. Nesse último caso, as credenciais fornecidas são prontamente comparadas com aquelas que constam na [EEPROM](#), e o resultado da comparação é salvo. O valor textual do campo é descartado.

Ao encontrar um campo vazio, que sinaliza o final do cabeçalho, ocorre a transição para o estado de leitura do corpo. Passa-se a consumir um número de octetos igual ao tamanho especificado no campo apropriado, anteriormente lido. Ao final da leitura do corpo, é gerada uma resposta, e as variáveis de estado são reiniciadas. A máquina de estados volta ao estado inicial, para processar uma possível nova requisição na mesma conexão *keep-alive*. Caso a conexão seja fechada, retorna-se ao laço principal que aguarda um novo cliente. A resposta enviada contém um código de erros apropriado em caso de autenticação inválida, ou então o [HTML](#) da página *web* solicitada. Alguns endereços não correspondem a páginas, como é o caso do endereço que requisita o acionamento do relé, ou a troca de usuário/senha. Esses casos são tratados respectivamente, e é gerada uma resposta com código de redirecionamento para a página anterior. O acionamento do relé se dá por um período de 4s, com um pulso gerado pelo *hardware* temporizador *Timer 1* do [AVR](#).

## 5.2.2 Otimizações

O recurso mais escasso do microcontrolador empregado é a memória [SDRAM](#), conforme constatado no [Quadro 10](#), e portanto foi enfatizada sua otimização. Assim como já mencionado em [subseção 5.2.1](#), uma das estratégias adotadas para tal foi o processamento e interpretação imediatos das requisições, durante seu recebimento, de maneira a evitar o armazenamento das mensagens completas para tratamento subsequente. Os campos pertinentes são tratados e salvos em inteiros ou enumerados, estruturas de dados estas mais enxutas do que as respectivas representações textuais.

Entretanto, a mais significativa otimização está na maneira como são gravadas as páginas [HTML](#). Por exemplo, a página que corresponde ao painel administrativo, sozinha, ocuparia mais da três quartos da memória [RAM](#) do [MCU](#), caso estivesse alocada em uma variável *string* comum dentro do código-fonte do programa. Para contornar essa limitação, lançou-se mão do armazenamento das páginas na memória de programa (*flash*). As páginas

são convertidas em arquivos de objeto do compilador, e vinculadas na seção *PROGMEM* do artefato executável, por meio de comandos como:

```
avr-objcopy -I binary -O elf32-avr --rename-section .data=.progmem,alloc
,load,readonly,data,contents "index.html" "index.html.o"
```

O vinculador então disponibiliza um ponteiro para acesso aos dados binários dentro do código-fonte:

```
extern const char _binary_index_html_start[] PROGMEM;
extern const char _binary_index_html_end[] PROGMEM;
```

Devido à arquitetura do [AVR](#) ser do tipo *Harvard*, tais ponteiros não podem ser lidos diretamente, uma vez que a memória de programa está em um barramento distinto daquele utilizado para memória de dados. Instruções próprias são utilizadas para copiar os dados para a memória [RAM](#) logo antes do seu uso e envio pela conexão [HTTP](#). Isso é realizado iterativamente, em pequenos blocos, já que a cópia completa exauriria a memória semelhante ao problema original que se está tentando contornar.

## 6 Resultados

Os protótipos construídos são apresentados na [Figura 12](#). O presente capítulo discorre sobre os resultados dos testes de verificação de funcionalidade neles realizados.

Figura 12 – Fotografia dos protótipos construídos



Fonte: o Autor, 2023.

A [Tabela 1](#) apresenta os resultados do [Experimento 1](#). Fica constatada uma margem de folga na utilização dos recursos do microcontrolador escolhido. Esse sucesso é atribuído às otimizações descritas na [subseção 5.2.2](#). O espaço remanescente poderia ser, futuramente, aproveitado para adição de novos recursos, como por exemplo: múltiplos usuários ([Trabalho Futuro 3](#)) ou criptografia ([Trabalho Futuro 4](#)).

Tabela 1 – Ocupação de recursos do microcontrolador

Recurso	Ocupação [bytes]
Memória de programa	24824 (76 %)
Memória RAM	926 (45 %)
Memória EEPROM	65 (6 %)

Fonte: o Autor, 2023.

Os resultados do [Experimento 2](#) estão expostos na [Tabela 2](#). Os tempos que constam são a média aritmética de 10 amostras. Nota-se que os tempos de resposta são humanamente imperceptíveis, atestando à boa performance do *software* desenvolvido, apesar das inerentes limitações de desempenho do microcontrolador.

Tabela 2 – Resultado do teste de tempo de resposta do servidor **HTTP**

<b>Página</b>	<b>Tamanho do HTML</b> [bytes]	<b>Tempo de resposta</b> [ms]
Acionamento do relé	0	26
Página de Início	752	28
Painel de Administração	1803	38

Fonte: o Autor, 2023.

O **Experimento 3** produziu os resultados discriminados na **Tabela 3**. Nota-se uma relação inversa das taxas de dados em relação à distância, no intervalo estudado. Isso era esperado, porém o desempenho geral se mostrou muito aquém das especificações do *modem* utilizado, que propagandaia taxas de dados mais elevadas em enlaces muito mais distantes, conforme **Quadro 10**. Ainda assim, as taxas de dados obtidas pelo **PLC** nesse experimento estão várias ordens de grandeza acima daquelas demandadas pelos principais *codecs* de voz utilizados no **VoIP**, conforme **Quadro 9**. Extrapolando-se os resultados por meio de uma curva de tendência exponencial como exibida em **Figura 13**, pode-se estimar que a taxa de dados degrade até os  $64Kbps$ , correspondentes ao *codec* G.711 A, a uma distância de  $100m$ .

Ainda maior variabilidade no desempenho do **PLC** poderia ser introduzida pela injeção de poluição da rede elétrica (vide **Trabalho Futuro 1**), entretanto, essa abordagem experimental foi descartada devido à ausência de recursos práticos à disposição para tal ensaio. Além disso, o enfoque desse estudo é majoritariamente voltado à aplicação em mais alto nível, em detrimento ao detalhamento de cada subsistema individualmente.

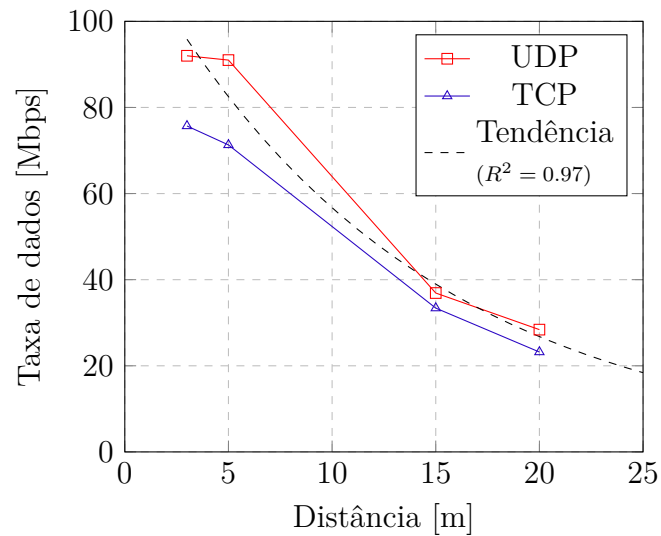
Tabela 3 – Resultados do ensaio de transferência de dados no **PLC**

<b>Indicador</b>	<b>3 m</b>	<b>5 m</b>	<b>15 m</b>	<b>20 m</b>
Taxa de dados <b>TCP</b> [Mbps]	75,7	71,3	33,4	23,2
Taxa de dados <b>UDP</b> [Mbps]	92,0	91,0	36,9	28,4
<i>Jitter</i> da latência [ms]	0,249	2,054	1,605	3,705
Perda de pacotes [%]	0,000	0,036	0,230	0,000

Fonte: o Autor, 2023.

Após a realização do **Experimento 4**, cujos resultados são exibidos na **Tabela 4**, fica evidenciada a confiabilidade do *software* **VoIP** desenvolvido mediante o cenário proposto. Situações de maior demanda, por exemplo com chamadas simultâneas, poderiam acarretar a piora dos indicadores, entretanto ensaios práticos dessa natureza demandariam maior investimento da construção de protótipos adicionais, ficando relegados ao **Trabalho**

Figura 13 – Taxa de dados medida em função da distância



Fonte: o Autor, 2023.

## Futuro 2.

Tabela 4 – Resultados do ensaio de repetibilidade de chamada VoIP

Indicador	Resultado	
	A → B	B → A
Número de chamadas	50	50
Sucessos	50	50
Falhas	0	0
Taxa de sucesso	100%	100%

Fonte: o Autor, 2023.

Similarmente, o **Experimento 5** demonstrou (vide [Tabela 5](#)) a plena capacidade do sistema de comunicação em comandar o relé do controlador de acesso sempre que solicitado. Esse ensaio é especialmente elucidativo da integração realizada, uma vez que o fluxo de informação que produz o fenômeno observado percorre desde a interface do usuário (a tecla do telefone), passando pelos inúmeros protocolos VoIP, implementados pelo servidor ora desenvolvido, transportados pelo PLC, e terminando no servidor HTTP embarcado no microcontrolador.

Tabela 5 – Resultados do ensaio de repetibilidade de acionamento do relé

Indicador	Resultado
Número de acionamentos	50
Sucessos	50
Falhas	0
Taxa de sucesso	100%

Fonte: o Autor, 2023.



---

Pela análise geral dos resultados de todos os experimentos realizados, pode-se atestar a confiabilidade do sistema prototipado. Todos os subsistemas se comportaram como esperado, e atuaram sinergicamente para o bom atendimento da aplicação. Fica comprovada a viabilidade da associação entre as tecnologias VoIP e PLC para o presente caso de uso, nas condições experimentadas, valendo-se do *software* desenvolvido.

## 7 Conclusões

O presente trabalho de diplomação demonstrou o projeto de um sistema integrado para comunicação de voz e comando remoto de controlador de acesso que utiliza o PLC como canal de dados, permitindo o reuso das redes elétricas residenciais ou condominiais, facilitando assim a implantação do serviço de interfonia ou portaria eletrônica. Para tal, foi desenvolvida uma biblioteca de *software*<sup>1</sup> livre que implementa os protocolos SIP, SDP e RTP, amplamente utilizados em aplicações VoIP. Subsidiado por essa biblioteca, foi construído um servidor VoIP que permite chamadas entre terminais telefônicos interconectados em uma rede local por meio do PLC, e que comanda o acionamento de um relé quando o usuário disca a tecla corresponde do telefone. Similarmente, foi desenvolvido e disponibilizado um *firmware*<sup>2</sup> de código aberto que comanda o relé remoto através de requisições HTTP que trafegam pela mesma rede PLC.

Como prova de conceito foram construídos os protótipos de duas unidades, utilizando modelos de *modems* PLC e *gateways* VoIP disponíveis no mercado. O *software* do servidor desenvolvido foi embarcado em um dos protótipos em um Raspberry Pi, ao passo que a unidade complementar recebeu uma placa de prototipagem Arduino programado com o *firmware* desenvolvido para o relé de controle remoto.

Conduziu-se um ensaio de desempenho do enlace PLC em função da distância, que evidenciou uma rápida degradação dos indicadores com o aumento da separação física. Apesar disso, as taxas de dados obtidas, mesmo no pior dos casos, ainda foram várias ordens de grandeza, superiores àquelas demandadas pelos principais *codecs* comumente utilizados no VoIP, portanto, atestando mesmo assim a viabilidade da conjunção entre ambas as tecnologias no cenário estudado. Foi estimado que os requisitos de banda continuariam a ser atendidos para um enlace de até 100m.

Como verificação de funcionalidade do sistema proposto, uma série de ensaios de repetibilidade foram conduzidos, e demonstraram que a confiabilidade e a robustez do trabalho de integração realizado estão satisfatórios.

<sup>1</sup> Disponível em: <https://github.com/luizfeldmann/ardunetswitch>

<sup>2</sup> Disponível em: <https://github.com/luizfeldmann/proxsip>

## 7.1 Trabalhos futuros

Do ponto de vista de caracterização do presente protótipo, os ensaios poderiam ser futuramente aprofundados, por exemplo:

**Trabalho Futuro 1** Cobrir um intervalo maior de distâncias de enlace PLC, variando também a incidência de cargas poluidoras na rede elétrica. Possíveis estratégias para tal incluem a realização dos ensaios em horários diferentes (de ponta e fora de ponta) e em residências com diferentes padrões de consumo de energia.

**Trabalho Futuro 2** Avaliar o comportamento do sistema na presença de chamadas simultâneas, e a interação entre mais do que dois *modems* PLC. O *software* desenvolvido prevê suporte a esse cenário, porém a comprovação prática demandaria a construção de protótipos adicionais.

Em termos de margem para melhoria nos recursos oferecidos, as funcionalidades adicionais poderiam ser implementados:

**Trabalho Futuro 3** Suporte a mais de um usuário ou níveis de privilégio para administração do relé de controle por protocolo de rede.

**Trabalho Futuro 4** Adoção de criptografia de camada de transporte (TLS/SSL) nas transações com o relé de controle de acesso, possivelmente, acompanhado da substituição do microcontrolador por outro mais capaz.

**Trabalho Futuro 5** Aumento de escopo do projeto para suportar também chamadas de vídeo. Novos estudos de capacidade de banda e readequação da lista de materiais necessitariam ser conduzidos.

Finalmente, o sistema como um todo poderia ser miniaturizado, pela integração do PLC e o VoIP em um único circuito, viabilizando a produção seriada como um produto. Poder-se-ia, então, reavaliar a proposta de valor do equipamento: de uma solução nichada para casos de dificuldade de instalação, para uma alternativa conveniente e amplamente útil ao público em geral. Seria necessária uma análise de viabilidade de mercado e parceria com alguma empresa do ramo, ou então a incubação de uma *startup*.

## Referências

- ANATEL. Resolução nº 527, de 8 de abril de 2009: Aprova o regulamento sobre condições de uso de radiofrequências por sistemas de banda larga por meio de redes de energia elétrica. *Diário Oficial da União*, Brasília, DF, p. 49, Apr 2009. ISSN 1677-7042. Disponível em: <<https://informacoes.anatel.gov.br/legislacao/resolucoes/2009/101-resolucao-527>>. Acesso em: 21 nov. 2021. Citado na página 45.
- ANATORY, J.; THEETHAYI, N. *Broadband power-line communication systems: theory & applications*. Southampton ; Boston: WIT, 2010. OCLC: ocn462893751. ISBN 9781845644161. Citado 2 vezes nas páginas 25 e 26.
- ANEEL. Resolução normativa nº 375, de 25 de agosto de 2009: Regulamenta a utilização das instalações de distribuição de energia elétrica como meio de transporte para a comunicação digital ou analógica de sinais. *Diário Oficial da União*, Brasília, DF, p. 110, Ago 2009. Disponível em: <<http://www2.aneel.gov.br/cedoc/ren2009375.pdf>>. Acesso em: 21 nov. 2021. Citado na página 45.
- BAI, Y.; DU, W. Voip services for ocean fishery vessels over integrated wireless and wireline networks. In: *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. [S.l.: s.n.], 2013. p. 3461–3465. Citado na página 24.
- BARMADA, S. et al. Analysis of power-line communication channels in ships. *IEEE Transactions on Vehicular Technology*, v. 59, n. 7, p. 3161–3170, Sep 2010. ISSN 0018-9545. Citado na página 23.
- BIGELOW, S. J.; CARR, J. J.; WINDER, S. *Understanding telephone electronics*. 4th ed. ed. Boston: Newnes, 2001. ISBN 9780750671750. Citado 2 vezes nas páginas 27 e 28.
- BOOST. *Boost C++ Libraries*. 2023. Disponível em: <<http://www.boost.org/>>. Citado na página 40.
- CASNER, S. L.; SCHULZRINNE, H. *RTP Profile for Audio and Video Conferences with Minimal Control*. RFC Editor, 2003. RFC 3551. (Request for Comments, 3551). Disponível em: <<https://www.rfc-editor.org/info/rfc3551>>. Citado 2 vezes nas páginas 35 e 36.
- CENELEC. *EN 50065-1: Signalling on low-voltage electrical installations in the frequency range 3 khz to 148.5 khz - part 1: General requirements, frequency bands and electromagnetic disturbances*. [S.l.], 2011. Disponível em: <<https://standards.globalspec.com/std/1379982/EN%2050065-1>>. Acesso em: 21 nov. 2021. Citado na página 18.
- CHAKRABORTY, T.; MISRA, I. S.; PRASAD, R. *VoIP Technology: Applications and Challenges*. Springer International Publishing, 2019. Disponível em: <<http://dx.doi.org/10.1007/978-3-319-95594-0>>. Citado na página 19.
- Kevin Corcoran. *Power line communication voice over IP system and method*. 2005. US7856007B2. Depósito: 21 Out. 2005. Concessão: 21 Dez. 2010. Disponível em: <<https://patents.google.com/patent/US7856007B2>>. Citado na página 23.

- DEGARDIN, V. et al. Theoretical approach to the feasibility of power-line communication in aircrafts. *IEEE Transactions on Vehicular Technology*, v. 62, n. 3, p. 1362–1366, Mar 2013. ISSN 0018-9545, 1939-9359. Citado na página 23.
- DONOVAN, S. et al. *Session Initiation Protocol (SIP) Basic Call Flow Examples*. RFC Editor, 2004. RFC 3665. (Request for Comments, 3665). Disponível em: <<https://www.rfc-editor.org/info/rfc3665>>. Citado na página 34.
- EDDY, W. *Transmission Control Protocol (TCP)*. RFC Editor, 2022. RFC 9293. (Request for Comments, 9293). Disponível em: <<https://www.rfc-editor.org/info/rfc9293>>. Citado na página 31.
- FARKAS, T. D. et al. Application of power line communication technology in street lighting control. *International Journal of Design and Nature and Ecodynamics*, v. 13, n. 2, p. 176–186, Jun 2018. ISSN 1755-7437, 1755-7445. Citado na página 24.
- FASIKU, A. A review of voice over internet protocol. *The Journal of Scientific and Engineering Research*, v. 5, p. 96–106, 01 2018. Citado 2 vezes nas páginas 18 e 20.
- GINOT, N. et al. Application of power line communication for data transmission over pwm network. *IEEE Transactions on Smart Grid*, v. 1, n. 2, p. 178–185, Sep 2010. ISSN 1949-3053, 1949-3061. Citado na página 23.
- GOERTZEL, G. *An Algorithm for the Evaluation of Finite Trigonometric Series*. JSTOR, 1958. 34 p. Disponível em: <<http://dx.doi.org/10.2307/2310304>>. Citado na página 29.
- HENDRAWAN, H.; ADITYA, B. *Asterisk and Radio Over IP Integration at Voice Communication System Air Traffic Control*. IEEE, 2019. Disponível em: <<http://dx.doi.org/10.1109/TSSA48701.2019.8985483>>. Citado na página 24.
- HOMEPLUG AV Specification Version 2.1. Home Plug Alliance, 2014. Disponível em: <[https://docbox.etsi.org/Reference/homeplug\\_av21/homeplug\\_av21\\_specification\\_final\\_public.pdf](https://docbox.etsi.org/Reference/homeplug_av21/homeplug_av21_specification_final_public.pdf)>. Citado na página 27.
- IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua: Vitimização: Sensação de segurança*. Rio de Janeiro: IBGE, 2021. ISBN 978-85-240-4555-4. Disponível em: <[https://biblioteca.ibge.gov.br/visualizacao/livros/liv101984\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv101984_informativo.pdf)>. Citado na página 16.
- IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua: Vitimização: Furto e roubo*. Rio de Janeiro: IBGE, 2021. ISBN 978-85-240-4554-7. Disponível em: <[https://biblioteca.ibge.gov.br/visualizacao/livros/liv101983\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv101983_informativo.pdf)>. Citado na página 16.
- IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications. *IEEE Std 1901-2020 (Revision of IEEE Std 1901-2010)*, p. 1–1622, 2021. Citado na página 19.
- ITU-T. *G.9964: Unified high-speed wireline-based home networking transceivers - power spectral density specification*. [S.l.], 2011. Disponível em: <<https://www.itu.int/rec/T-REC-G.9964-201112-I/en>>. Acesso em: 21 nov. 2021. Citado na página 26.

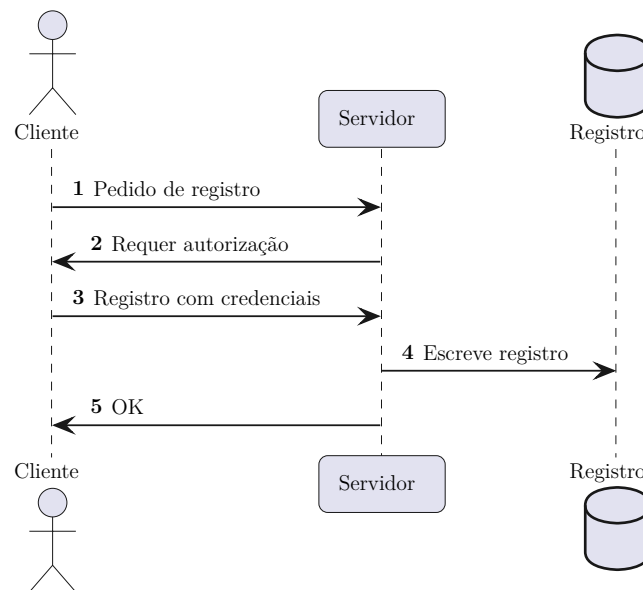
- ITU-T. *G.9904*: Narrowband orthogonal frequency division multiplexing power line communication transceivers for prime networks. [S.l.], 2012. Disponível em: <<https://www.itu.int/rec/T-REC-G.9904-201210-I/en>>. Acesso em: 21 nov. 2021. Citado na página 20.
- ITU-T. *G.9903*: Narrowband orthogonal frequency division multiplexing power line communication transceivers for g3-plc networks. [S.l.], 2017. Disponível em: <<https://www.itu.int/rec/T-REC-G.9903-201708-I/en>>. Acesso em: 21 nov. 2021. Citado na página 20.
- ITU-T. *H.323*: Packet-based multimedia communications systems. [S.l.], 2022. Disponível em: <<https://www.itu.int/rec/T-REC-H.323/en>>. Acesso em: 29 mai. 2023. Citado 2 vezes nas páginas 20 e 30.
- JAIN, V. *Wireshark fundamentals*. 1. ed. Berlim: APress, 2022. Citado na página 43.
- KHAN, B. M. et al. *Performance Analysis of Raspberry Pi 3 IP PBX Based on Asterisk*. MDPI AG, 2022. 3313 p. Disponível em: <<http://dx.doi.org/10.3390/electronics11203313>>. Citado na página 24.
- LAMPE, L. *Power line communications: principles, standards and applications from multimedia to smart grid*. Second edition. [S.l.]: Wiley, 2016. ISBN 9781118676714. Citado 2 vezes nas páginas 18 e 20.
- LATCHMAN, H. A. *Homeplug AV and IEEE 1901: a handbook for PLC designers and users*. [S.l.]: Wiley, 2013. ISBN 9780470410738. Citado 2 vezes nas páginas 19 e 20.
- LATHI, B. P.; DING, Z. *Modern digital and analog communication systems*. Fifth edition. New York: Oxford University Press, 2019. (Oxford series in electrical and computer engineering). ISBN 9780190686840. Citado 2 vezes nas páginas 25 e 26.
- MUGGELEN, J. V.; BRYANT, R.; MADSEN, L. *Asterisk: The definitive guide*. 5. ed. Sebastopol, CA: O'Reilly Media, 2019. Citado na página 22.
- OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-time signal processing*. 3rd ed. ed. [S.l.]: Pearson, 2010. ISBN 9780131988422. Citado 2 vezes nas páginas 28 e 29.
- PERKINS, C.; HANDLEY, M. J.; JACOBSON, V. *SDP: Session Description Protocol*. RFC Editor, 2006. RFC 4566. (Request for Comments, 4566). Disponível em: <<https://www.rfc-editor.org/info/rfc4566>>. Citado 2 vezes nas páginas 34 e 35.
- PONUTHORAI, P. *Version control with git*. 3. ed. Sebastopol, CA: O'Reilly Media, 2022. Citado na página 42.
- POSTEL, J. *User Datagram Protocol*. RFC Editor, 1980. RFC 768. (Request for Comments, 768). Disponível em: <<https://www.rfc-editor.org/info/rfc768>>. Citado na página 31.
- SCHOOLER, E. et al. *SIP: Session Initiation Protocol*. RFC Editor, 2002. RFC 3261. (Request for Comments, 3261). Disponível em: <<https://www.rfc-editor.org/info/rfc3261>>. Citado 2 vezes nas páginas 32 e 33.

- SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC Editor, 2003. RFC 3550. (Request for Comments, 3550). Disponível em: <<https://www.rfc-editor.org/info/rfc3550>>. Citado 3 vezes nas páginas 35, 36 e 37.
- SCHULZRINNE, H.; TAYLOR, T. *RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals*. RFC Editor, 2006. RFC 4733. (Request for Comments, 4733). Disponível em: <<https://www.rfc-editor.org/info/rfc4733>>. Citado 2 vezes nas páginas 30 e 35.
- SCHWARTZ, M. Carrier-wave telephony over power lines: Early history [history of communications]. *IEEE Communications Magazine*, v. 47, n. 1, p. 14–18, Jan 2009. ISSN 0163-6804, 1558-1896. Citado na página 18.
- SINNREICH, H.; JOHNSTON, A. B. *Internet communications using SIP*. 2. ed. Nashville, TN: John Wiley & Sons, 2006. (Networking Council). Citado na página 19.
- SWIDZINSKI, R. *Modern CMake for C++*. Birmingham, England: Packt Publishing, 2022. Citado na página 40.
- VINES, R. et al. Noise on Residential Power Distribution Circuits. *IEEE Transactions on Electromagnetic Compatibility*, EMC-26, n. 4, p. 161–168, nov. 1984. ISSN 0018-9375. Disponível em: <<http://ieeexplore.ieee.org/document/4091748/>>. Citado na página 26.
- ZAQOUT, I.; NASR, M. A.; ABU-NASSER, B. Voip using power line communications. *Journal of Natural Sciences of Al-Azhar University*, v. 19, p. 57–81, dez. 2017. Citado na página 23.

# APÊNDICE A – Fluxos de Mensagens SIP

## A.1 Transação de Registro

Figura 14 – Diagrama de sequência do registro SIP



Fonte: o Autor, 2023.

1. O cliente envia uma requisição **REGISTER** para o servidor. A mensagem contém um campo **Contact** indicando endereço IP, porta e protocolo de transporte onde poderá receber chamadas.

```

REGISTER sip:servidor.com SIP/2.0
Via: SIP/2.0/UDP 192.168.0.100:5061;branch=z9hG4bK-16c9a52
From: Fulano <sip:Fulano@servidor.com>;tag=e038c4e3122afd40o0
To: Fulano <sip:Fulano@servidor.com>
Contact: Fulano <sip:Fulano@192.168.0.100:5061 >
Call-ID: 7138eb5c-69d3df68@192.168.0.100
CSeq: 1 REGISTER
  
```

2. O servidor responde com código de erro **401** indicando que o usuário deve fornecer credenciais. A resposta também contém uma proposta de esquema de autenticação e um desafio criptográfico.

```

SIP/2.0 401 Unauthorized
WWW-Authenticate: Digest realm="servidor.com", nonce="xyz123"
...
  
```



3. O cliente repete a solicitação, mas agora incluindo suas credenciais criptografadas de acordo com o esquema proposto pelo servidor.

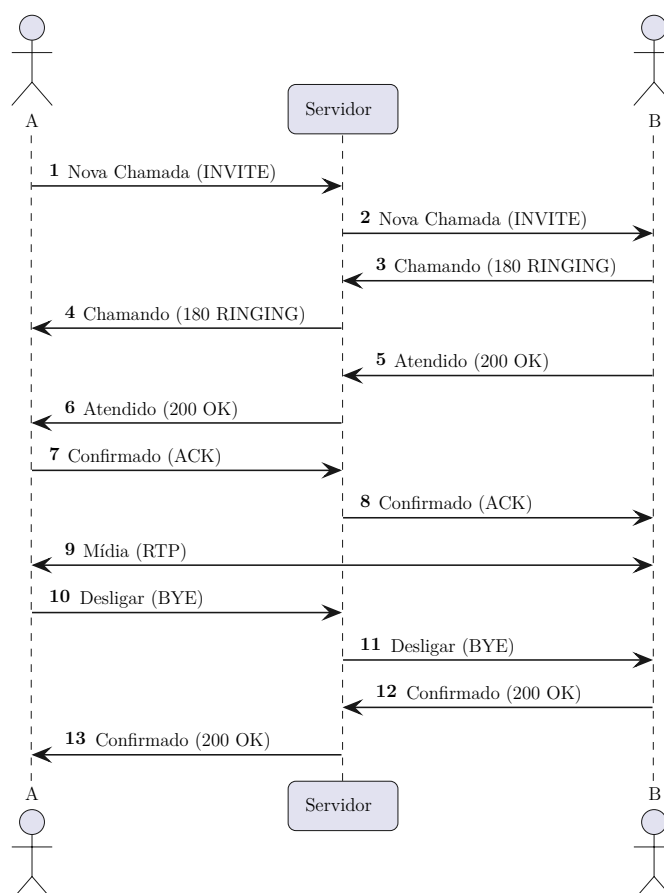
```
REGISTER sip:servidor.com SIP/2.0
Authorization: Digest username="Fulano", realm="servidor.com",
               nonce="xyz123", uri="sip:servidor.com",
               response="abcdef"
...
```

4. O servidor grava a associação do contato e seus dados de localização em um banco de dados.
5. O servidor responde com o código de sucesso.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.0.100:5061;branch=z9hG4bK-16c9a52
From: Fulano <sip:Fulano@servidor.com>;tag=e038c4e3122afd40o0
To: Fulano <sip:Fulano@servidor.com>
Contact: Fulano <sip:Fulano@192.168.0.100:5061>
Call-ID: 7138eb5c-69d3df68@192.168.0.100
CSeq: 2 REGISTER
```

## A.2 Transação de Chamada

Figura 15 – Diagrama de sequência de uma chamada em SIP



Fonte: o Autor, 2023.

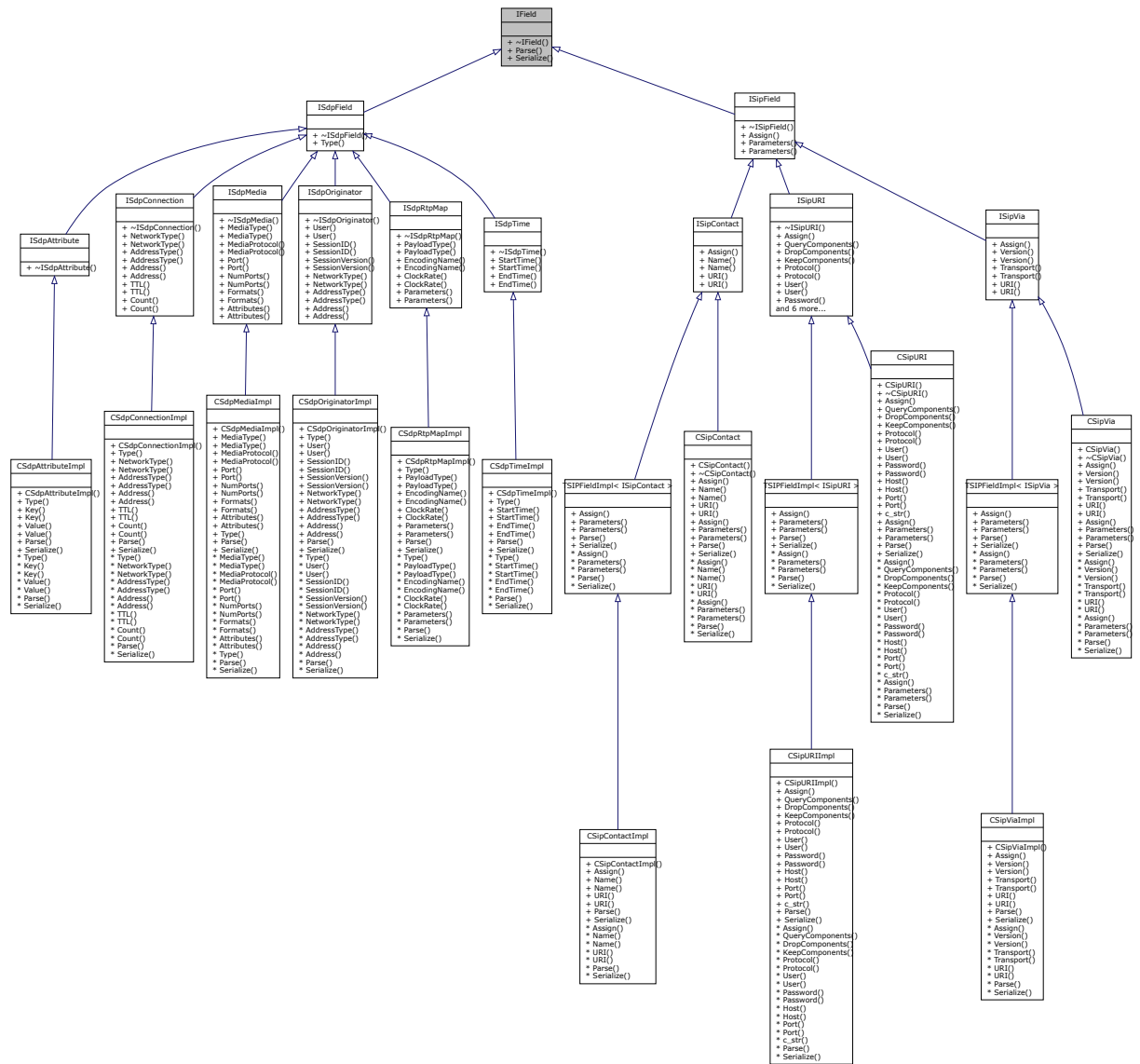
1. O chamador envia ao servidor um pedido **INVITE** contendo o **URI** do contato para quem deseja telefonar. O corpo da mensagem é um **SDP** contendo uma lista de canais e mídias suportadas para concretizar a chamada.
2. O servidor encaminha a mensagem ao destinatário final (número discado). Para isso, o servidor consulta os dados de localização desse **UA** no **registro** preenchido anteriormente, conforme **Figura 14**. O servidor também acrescenta ao pedido um campo **Via** referenciado a sí.
3. O contato sendo chamado responde com um status **180 Ringing** indicando que o telefone está tocando.
4. O servidor encaminha a resposta ao chamador.

5. O contato chamado aceita a ligação, e um status **200 OK** é enviado ao servidor. Similarmente, o corpo da mensagem é um **SDP** descrevendo os formatos de mídia suportados pelo **UA**.
6. O servidor encaminha a resposta ao chamador.
7. O chamador envia um **ACK** indicando que o canal de voz está pronto e a chamada pode começar.
8. O servidor encaminha a mensagem de confirmação.
9. É estabelecida uma comunicação duplex no protocolo **RTP**, com os formatos de mídia especificados nos **SDPs** transacionados conforme acima.
10. Um dos pares na chamada encerra a ligação. Uma mensagem **BYE** é enviada ao servidor.
11. O servidor encaminha a mensagem de encerramento ao contato oposto.
12. Uma resposta de confirmação de término é gerada.
13. O servidor repassa a confirmação ao outro cliente e a sessão termina.

# APÊNDICE B – Design do Servidor Proxy

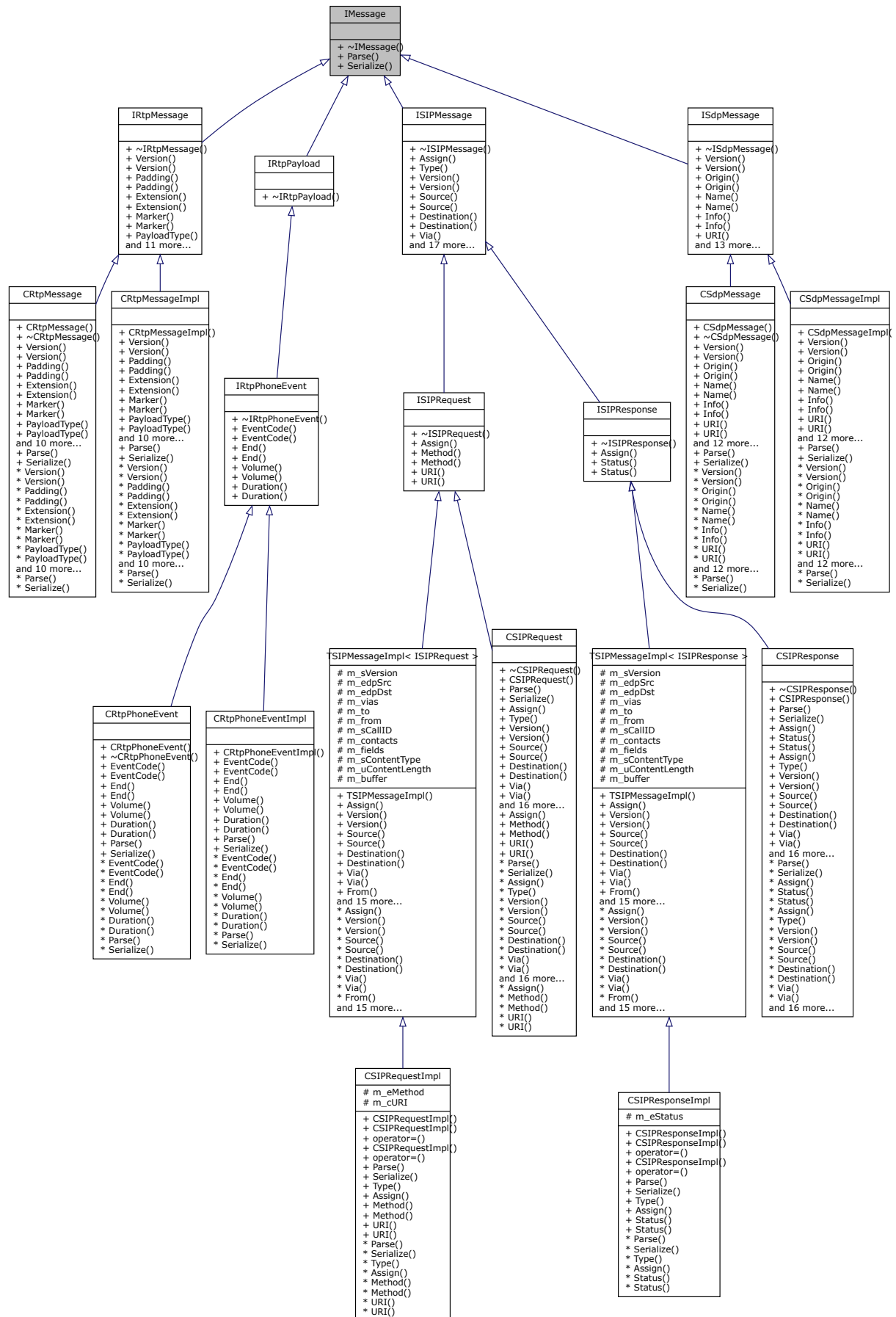
## B.1 Diagramas de Classes

Figura 16 – Diagrama de herança das classes de campos



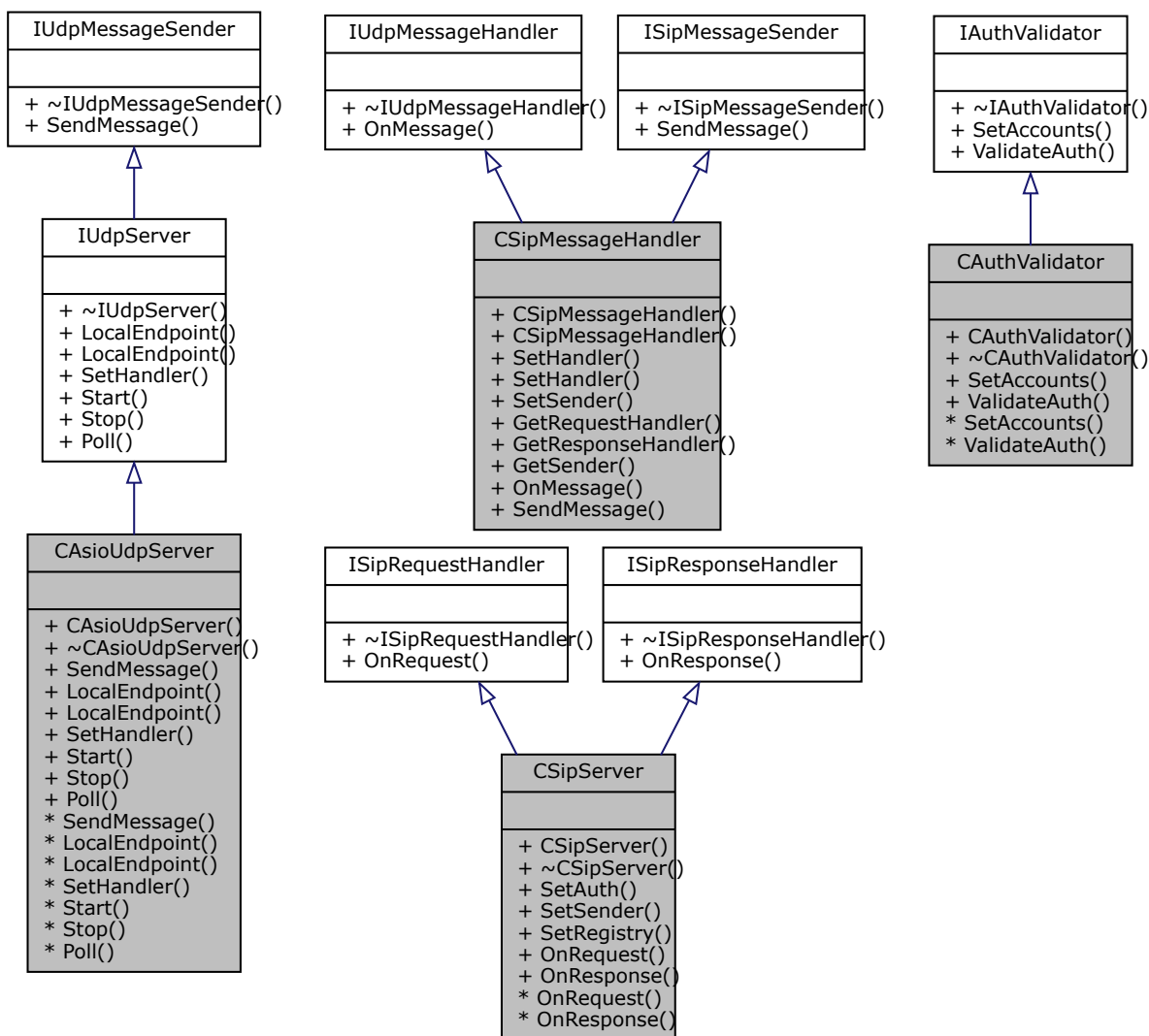
Fonte: o Autor, 2023.

Figura 17 – Diagrama de herança das classes de mensagens



Fonte: o Autor, 2023.

Figura 18 – Diagramas de classes variadas



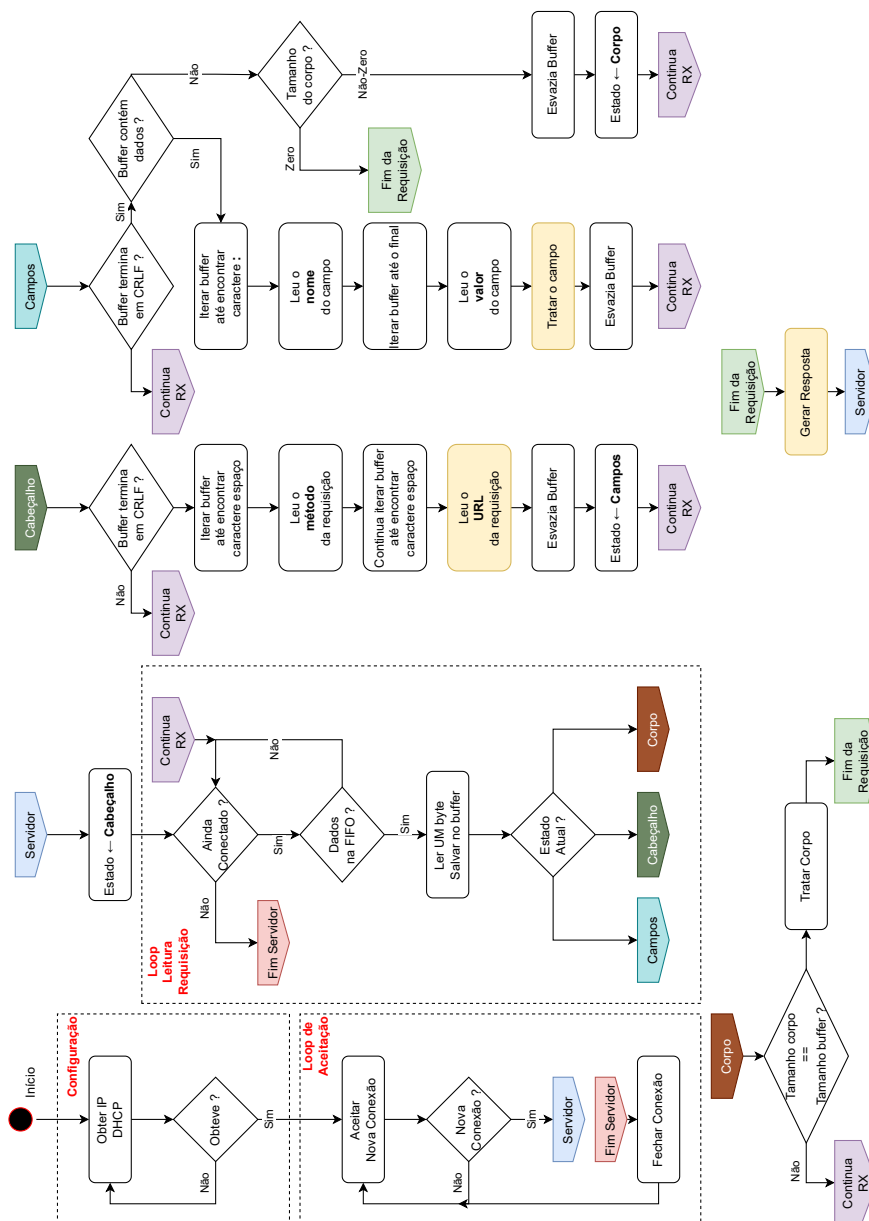
Fonte: o Autor, 2023.



# APÊNDICE C – Funcionamento do *firmware* do Relé Remoto

## C.1 Interpretação da Requisição HTTP

Figura 20 – Diagrama de atividades do algoritmo de interpretação de requisições HTTP

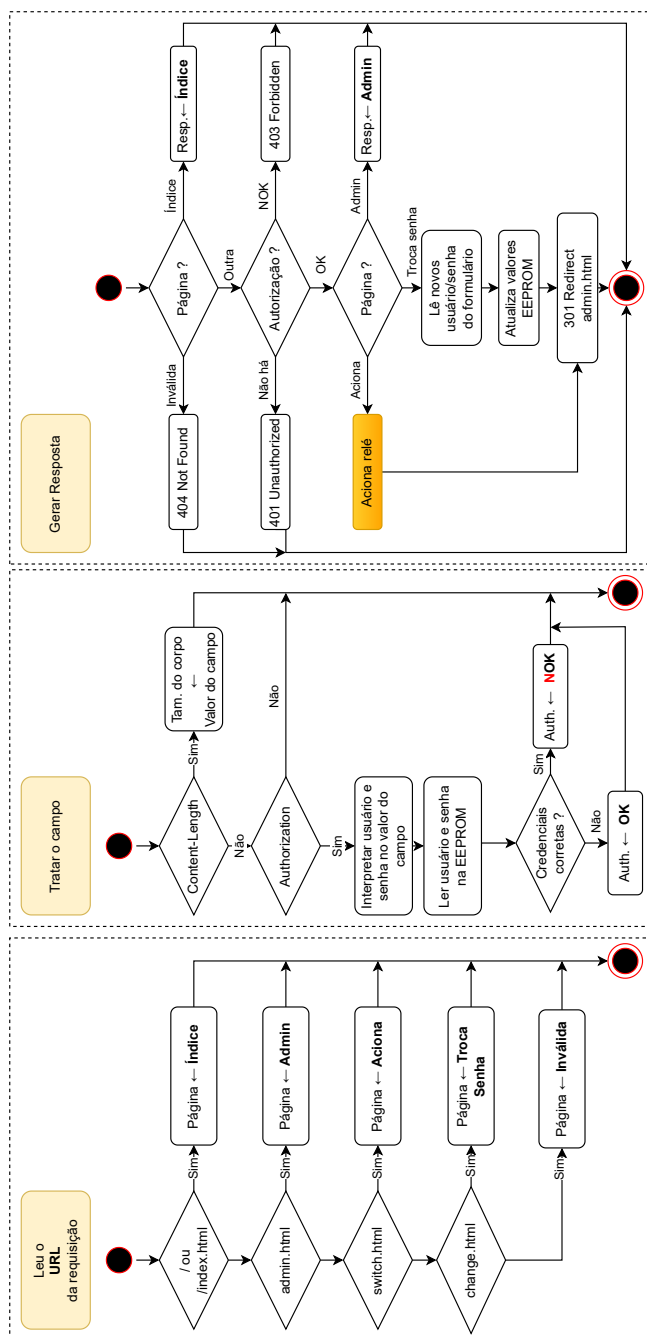


Fonte: o Autor, 2023.



## C.2 Lógica de Geração da Resposta do Servidor

Figura 21 – Diagrama de atividades da lógica de geração de respostas do servidor HTTP



Fonte: o Autor, 2023.