



**INSTITUTO LATINO-AMERICANO
DE CIÊNCIAS DA VIDA E DA NATUREZA**

BACHARELADO EM ENGENHARIA FÍSICA

PROPOSTA DE UM POTENCIOSTATO DE FONTE ABERTA PARA ENSINO

Kaique de Sotti Silva

Foz do Iguaçu, Paraná
2025



**INSTITUTO LATINO-AMERICANO
DE CIÊNCIAS DA VIDA E DA NATUREZA**

BACHARELADO EM ENGENHARIA FÍSICA

PROPOSTA DE UM POTENCIOSTATO DE FONTE ABERTA PARA ENSINO

Kaique de Sotti Silva

Trabalho de Conclusão de Curso apresentado ao Instituto Latino-Americano de Ciência da Vida e da Natureza (ILACVN), como requisito parcial à obtenção do título de Bacharel em Engenharia física.

Orientador: Prof Dr. Edson Massayuki Kakuno

Coorientador: Prof Dr. Johan Alexander Cortes Suárez

Foz do Iguaçu, Paraná
2025

Catálogo elaborado pelo Setor de Tratamento da Informação
Catálogo de Publicação na Fonte. UNILA - BIBLIOTECA LATINO-AMERICANA - CENTRAL

S586

Silva, Kaique de Sotti.

Proposta de um Potenciostato de Fonte Aberta para Ensino / Kaique de Sotti Silva. - Foz do Iguaçu, 2026.
102 f.: il. color.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal da Integração Latino-Americana, Instituto Latino-Americano de Ciência da Vida e da Natureza, Bacharelado em Engenharia Física. Foz do Iguaçu - PR, 2026.

Orientador: Edson Massayuki Kakuno.

Coorientador: Johan Alexander Cortes Suárez.

1. Instrumentos - Potenciostato. 2. Voltametria. 3. Eletroquímica - Instrumentação. 4. Arduino (Controlador programável). 5. Eletrodos (Eletrodo de Referência). 6. Automação - *Open Hardware*. I. Kakuno, Edson Massayuki. II. Suárez, Johan Alexander Cortes. III. Título.

CDU 621.7

PROPOSTA DE UM POTENCIOSTATO DE FONTE ABERTA PARA ENSINO

Trabalho de Conclusão de Curso apresentado ao Instituto Latino-Americano de Ciência da vida e da Natureza (ILACVN), como requisito parcial à obtenção do título de Bacharel em Engenharia física.

BANCA EXAMINADORA

Orientador: Prof. Dr. Edson Massayuki Kakuno, UNILA

Coorientador: Prof. Dr. Johan Alexander Cortes Suárez, UNILA

Prof. Dr. Márcio de Sousa Góes, UNILA

Dr. Fabio Plut Fernandes, TECCER/ITAI

Foz do Iguaçu, 17 de Dezembro de 2025.

*Dedico este trabalho a todos aqueles a quem
essa trabalho possa ajudar de alguma forma.*

AGRADECIMENTOS

Em primeiro lugar, agradeço à minha família, cujo apoio foi fundamental. Agradeço em especial aos meus pais por toda a dedicação, paciência, compreensão e ensinamentos que tornaram possível a realização deste curso.

Agradeço meu orientador, professor Edson Massayuki Kakuno, não apenas pela orientação e dedicação durante a elaboração deste trabalho, mas sobretudo pela confiança, paciência, incentivo e oportunidades ao longo desta jornada. Sua contribuição foi essencial para que este projeto se tornasse realidade e para o meu desenvolvimento pessoal.

Agradeço o meu coorientador, professor Johan Alexander Cortes Suárez pela confiança e bom humor, que sempre tornou o laboratório um lugar confortável de se estar. Expresso minha gratidão pela orientação.

Agradeço ao pesquisador, Fabio Plut pela oportunidade e pelas diversas lições que levo comigo diariamente.

Agradeço à Teccer Ceramicas pelo o apoio e o financiamento nesse trabalho.

Agradeço à minha namorada, Isabel Rodrigues da Silva, por toda a paciência e parceria; sem o seu apoio e compreensão não seria possível concluir este trabalho. Serei eternamente grato.

Agradeço em especial, aos meus amigos, Marielle Bastos, Michelle Guerrero de Oliveira e Yan Costa Carneosso, a companhia de vocês tornou esta caminhada possível e consideravelmente mais leve. Sem vocês, ela seria impossível.

Aos colegas de curso, meu sincero agradecimento pelo apoio, pela troca de conhecimentos e pelos momentos compartilhados.

Por fim, agradeço à UNILA, instituição que me proporcionou a oportunidade de cursar Engenharia Física, com acesso a uma infraestrutura de qualidade e diversas oportunidades.

SILVA, S.; Kaique. Proposta de um Potenciostato de Fonte Aberta para Ensino. 17 de dezembro de 2025. 102 Páginas. Trabalho de conclusão de curso (TCC) - Bacharelado em Engenharia Física - Universidade Federal da Integração Latino-Americana, Foz do Iguaçu, Paraná.

RESUMO

Este trabalho apresenta o projeto, o desenvolvimento e a validação de um protótipo de potenciostato modular de fonte aberta (*open hardware*), destinado à realização de ensaios de voltametria cíclica em contextos didáticos e de pesquisa. O objetivo central consistiu na criação de um instrumento de baixo custo com alta capacidade de corrente, superando limitações de equipamentos educacionais convencionais. A arquitetura eletrônica baseia-se na integração do microcontrolador Arduino Nano com uma interface de controle em Python e um hardware dedicado, composto por um conversor digital-analógico de 16 bits (DAC714) para a geração do sinal do equipamento, um amplificador operacional para o controle do potencial (OP07 ou TL1028), um driver de potência (OPA541) capaz de fornecer até $\pm 2,5$ A nas condições atuais, e um sensor de corrente modular (INA282). O sistema inclui ainda um buffer de alta impedância (OPA128) para o eletrodo de referência e um Eletrodo de Referência de Hidrogênio (CHE) desenvolvido localmente. Os resultados da caracterização elétrica demonstraram excelente linearidade, com um tempo de acomodação de $33 - 50 \mu s$ na versão original, reduzido para aproximadamente $4 \mu s$ após a otimização da malha de controle. A validação funcional com uma célula simulada (*dummy cell*) frente a um potenciostato comercial Autolab PGSTAT302N revelou uma sobreposição quase perfeita das curvas voltamétricas, confirmando a precisão da topologia adotada. O eletrodo de referência construído manteve estabilidade sob alta corrente por mais de duas horas. Conclui-se que o equipamento é uma solução robusta e modular, consolidando-se como uma ferramenta eficiente para a instrumentação científica aberta e o ensino interdisciplinar em Engenharia Física.

Palavras-chave: Potenciostato; Voltametria Cíclica; Instrumentação Eletroquímica; Arduino; Eletrodo de Referência; Open Hardware.

SILVA, S.; Kaique. Propuesta de un Potenciostato de Fuente Abierta para Enseñanza. 17 de diciembre de 2025. 102 Páginas. Trabajo de conclusión de curso (TCC) - Grado en Ingeniería Física - Universidad Federal de la Integración Latinoamericana, Foz de Iguazú, Paraná.

RESUMEN

Este trabajo presenta el diseño, el desarrollo y la validación de un prototipo de potenciostato modular de código abierto (*open hardware*), destinado a la realización de ensayos de voltamperometría cíclica en contextos didácticos y de investigación. El objetivo central consistió en la creación de un instrumento de bajo costo con alta capacidad de corriente, superando las limitaciones de los equipos educativos convencionales. La arquitectura electrónica se basa en la integración del microcontrolador Arduino Nano con una interfaz de control en Python y un hardware dedicado, compuesto por un convertidor digital-analógico de 16 bits (DAC714) para la generación de la señal del equipo, un amplificador operacional para el control del potencial (OP07 o TL1028), un driver de potencia (OPA541) capaz de suministrar hasta $\pm 2,5$ A en las condiciones actuales, y un sensor de corriente modular (INA282). El sistema incluye además un buffer de alta impedancia (OPA128) para el electrodo de referencia y un Electrodo de Referencia de Hidrógeno (CHE) desarrollado localmente. Los resultados de la caracterización eléctrica demostraron una excelente linealidad, con un tiempo de asentamiento de $33 - 50 \mu s$ en la versión original, reducido a aproximadamente $4 \mu s$ tras la optimización del lazo de control. La validación funcional con una celda simulada (*dummy cell*) frente a un potenciostato comercial Autolab PGSTAT302N reveló una superposición casi perfecta de las curvas voltamperométricas, confirmando la precisión de la topología adoptada. El electrodo de referencia construido mantuvo la estabilidad bajo alta corriente durante más de dos horas. Se concluye que el equipo es una solución robusta y modular, consolidándose como una herramienta eficiente para la instrumentación científica abierta y la enseñanza interdisciplinaria en Ingeniería Física.

Palabras clave: Potenciostato; Voltamperometría Cíclica; Instrumentación Electroquímica; Arduino; Electrodo de Referencia; Open Hardware.

SILVA, S.; Kaique. Proposal of an Open-Source Potentiostat for Teaching. December 17, 2025. 102 Pages. Final Paper (TCC) - Bachelor in Physical Engineering - Federal University of Latin American Integration, Foz do Iguaçu, Paraná.

ABSTRACT

This work presents the design, development, and validation of an open-source modular potentiostat prototype aimed at performing cyclic voltammetry tests in didactic and research contexts. The central objective was the creation of a low-cost instrument with high current capacity, overcoming the limitations of conventional educational equipment. The electronic architecture is based on the integration of the Arduino Nano microcontroller with a Python control interface and dedicated hardware, consisting of a 16-bit digital-to-analog converter (DAC714) for signal generation, an operational amplifier for potential control (OP07 or TL1028), a power driver (OPA541) capable of supplying up to ± 2.5 A under current conditions, and a modular current sensor (INA282). The system also includes a high-impedance buffer (OPA128) for the reference electrode and a locally developed Hydrogen Reference Electrode (CHE). Electrical characterization results demonstrated excellent linearity, with a settling time of 33–50 μs in the original version, reduced to approximately 4 μs after control loop optimization. Functional validation with a dummy cell against a commercial Autolab PGSTAT302N potentiostat revealed a near-perfect overlap of the voltammetric curves, confirming the accuracy of the adopted topology. The constructed reference electrode maintained stability under high current for more than two hours. It is concluded that the equipment is a robust and modular solution, establishing itself as an efficient tool for open scientific instrumentation and interdisciplinary teaching in Physical Engineering.

Keywords: Potentiostat; Cyclic Voltammetry; Electrochemical Instrumentation; Arduino; Reference Electrode; Open Hardware.

LISTA DE FIGURAS

Figura 1	Diagrama esquemático de amplificadores operacionais	17
Figura 2	Diagrama esquemático de amplificadores operacionais com pinos de ajuste de <i>offset</i>	18
Figura 3	Circuito conversor corrente-tensão	20
Figura 4	Amplificador inversor (<i>scaler</i>)	21
Figura 5	Somador inversor	21
Figura 6	Seguidor de tensão	22
Figura 7	Circuito para controle de potencial no ponto A, independente das variações em R_1 e R_2	23
Figura 8	Conversor digital-analógico de 4 bits com resistores ponderados em binário	23
Figura 9	Conversor digital-analógico tipo escada R/2R	24
Figura 10	Análise de funcionamento de um DAC com rede R/2R	25
Figura 11	Representação da operação de amostragem e retenção	26
Figura 12	Funcionamento básico de um conversor analógico-digital	26
Figura 13	Representação simplificada da conversão sigma-delta	27
Figura 14	Diagrama funcional simplificado de um ADC sigma-delta	27
Figura 15	Exemplo de implementação de um conversor sigma-delta	28
Figura 16	Circuitos comuns de medição de corrente e as respectivas funções de transferência: (a) Um conversor i/V simples usando um resistor de realimentação. (b) Um amplificador de instrumentação com fator de ganho G amplifica a queda de tensão em um resistor relativamente pequeno. (c) Um circuito integrador acumula a carga correspondente à corrente de entrada por um período de tempo fixo.	28
Figura 17	Pinout do Arduino Nano.	31
Figura 18	Esquemático de uma Célula Eletroquímica de Três Eletrodos	33
Figura 19	Visões de uma célula eletroquímica como rede de impedâncias conectada aos três eletrodos	34
Figura 20	Representações da Célula Eletroquímica: (a) Símbolo da Célula Eletroquímica; (b) Esquemático do Circuito Elétrico aproximado da Célula (valores arbitrários)	34
Figura 21	Exemplo de Eletrodo de Hidrogênio Conveniente (CHE). Fonte: Adaptado de (Smith; Stevenson, 2007).	37
Figura 22	Sinais de excitação de tensão versus tempos empregados na Voltametria	38
Figura 23	Sinal de excitação triangular e voltamograma cíclico típico	40
Figura 24	Um potenciostato simples	41
Figura 25	Um potenciostato somador básico	42
Figura 26	Um sistema potenciostático completo	43
Figura 27	Diagrama de blocos de um instrumento eletroquímico típico controlado por computador	44
Figura 28	As três configurações básicas de polarização: (a) Eletrodo de Trabalho Aterrado (GW); (b) Eletrodo de Referência Aterrado (GR); (c) Eletrodo Auxiliar Aterrado (GA)	45
Figura 29	Configurações de medição de corrente: (a) CVC referido ao terra no eletrodo de trabalho; (b) CVC flutuante no eletrodo auxiliar; (c) Medição no retorno da fonte de alimentação	46
Figura 30	Medição de corrente ideal usando a configuração GA (Auxiliar Aterrado)	47
Figura 31	Diagrama de blocos do potenciostato	48

Figura 32	Todos os parâmetros de configuração do potenciostato	51
Figura 33	Interface de controle desenvolvida em Python (tela principal)	51
Figura 34	Diagrama de blocos da interface de controle e aquisição de dados	53
Figura 35	Diagrama de blocos da lógica de geração de sinais do DAC	53
Figura 36	Eletrodo de referência de hidrogênio desenvolvido	54
Figura 37	Esquema do eletrodo de referência de hidrogênio desenvolvido	55
Figura 38	Potenciostato montado para teste de <i>dummy cell</i> . 1- Fonte de alimentação simétrica de $\pm 24 V$. 2- Buffer de potência. 3- Transformador $\pm 12 V$, com 3 A. 4- Regulador de tensão de $\pm 15 V$, DAC714 e amplificador operacional (OP07). 5- Arduino Nano. 6- Buffer de alta impedância. 7- Sensor de corrente INA282 modificado. 8- Dummy cell	57
Figura 39	Regulador de tensão de $\pm 15 V$, DAC714 e amplificador operacional (OP07)	58
Figura 40	Buffer de potência em caixa de alumínio contendo o OPA541, com ganho ajustável entre 1 e 11, configurado para ganho unitário. O conjunto inclui chaves e potenciômetro para ajuste do limite de corrente em até 1 A. A caixa também abriga o estágio de alimentação, composto por um retificador de onda completa (limitado a 3 A) e capacitor de filtro de $10\,000\ \mu F$. O transformador é conectado externamente à caixa	59
Figura 41	Sensor de corrente (INA282) com duas escalas de corrente	60
Figura 42	Buffer de alta impedância (OPA128LM) e amplificador do eletrodo de referência	61
Figura 43	Configuração da <i>Dummy Cell</i> utilizada para testes de carga na escala de $\pm 1 A$, valores dos resistores dos ohms	62
Figura 44	Configuração da <i>Dummy Cell</i> utilizada para testes na escala de $\pm 10 mA$, valores dos resistores em ohms	63
Figura 45	Resposta ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Superior): Tensão de saída estabilizada. CH1 (Inferior): Sinal de controle (descida) enviado pelo DAC	65
Figura 46	Detalhe da borda de descida do sinal do DAC (CH1), apresentando transição rápida e limpa	65
Figura 47	Resposta dinâmica completa: CH2 (Saída) seguindo o comando do CH1 (Entrada) com atraso temporal característico da malha de controle	66
Figura 48	Detalhe do tempo de estabilização. No CH1 observa-se a estabilização final da tensão na carga	67
Figura 49	Resposta dinâmica na escala de 10mA. CH2: Resposta do sistema; CH1: Sinal de descida do DAC	67
Figura 50	Detalhe da descida do sinal de controle (DAC) na configuração de baixa corrente	67
Figura 51	Tempo de subida do DAC na escala de 10mA	68
Figura 52	Estabilização final do sinal na escala de 10 mA	68
Figura 53	Sistema otimizado: resposta rápida ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Superior): Tensão de saída estabilizada. CH1 (Inferior): Sinal de controle (descida) enviado pelo DAC	68
Figura 54	Sistema otimizado: resposta rápida ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Saída) seguindo o comando do CH1 (Entrada) com atraso temporal característico da malha de controle	68
Figura 55	Esquemático da <i>Dummy Cell</i> utilizada no cenário de validação	69
Figura 56	Voltametria cíclica para a <i>Dummy Cell</i> do Cenário 1. Linha laranja: dados brutos do protótipo; Linha preta: dados suavizados (médias móveis)	70

Figura 57 Comparativo de validação (Cenário 1). A linha preta (UNILA corrigida) sobrepõe-se à linha vermelha (AutoLab), indicando alta correlação	70
Figura 58 Esquemático da <i>Dummy Cell</i> utilizada no cenário de validação	71
Figura 59 Voltametria verificando o comportamento da escala de 10mA	71
Figura 60 Montagem da célula eletroquímica para o teste de estabilidade	72
Figura 61 Teste de longa duração do Eletrodo de Referência sob potencial fixo de -3 V. Estabilidade mantida por aproximadamente 2 horas (7000 s)	73
Figura 62 Deterioração do contra eletrodo após 5 horas de teste	74

LISTA DE TABELAS

Tabela 1 Lista de componentes e preço	56
Tabela 2 Dados de calibração elétrica e análise de erro na escala de 1 A	63
Tabela 3 Dados de calibração elétrica e análise de erro na escala de 20 mA	64
Tabela 4 Métricas de resposta dinâmica do sistema	69

SUMÁRIO

LISTA DE FIGURAS	12
LISTA DE TABELAS	13
1 INTRODUÇÃO	15
2 OBJETIVOS	16
3 FUNDAMENTAÇÃO TEÓRICA	17
3.1 ELETRÔNICA APLICADA À INSTRUMENTAÇÃO	17
3.1.1 Amplificadores Operacionais	17
3.1.1.1 Realimentação de Corrente	19
3.1.1.2 Amplificador Inversor	20
3.1.1.3 Somador Inversor	21
3.1.1.4 Realimentação de Tensão	22
3.1.1.5 Controle de Potencial	22
3.1.2 Conversores Digital-Analógico (DAC)	23
3.1.3 Conversores Analógico-Digitais (ADC)	25
3.1.4 Medição de corrente	28
3.1.4.1 Conversor transimpedância (current-to-voltage)	29
3.1.4.2 Amplificação da queda de tensão em resistor de derivação (shunt) com ganho	29
3.1.4.3 Integrador de corrente	29
3.1.5 Arduino	30
3.1.6 Linguagem de Programação Python	32
3.2 ELETROQUÍMICA	32
3.2.1 Célula eletroquímica e Dummy Cell	32
3.2.2 O Eletrodo de Referência de Hidrogênio	35
3.2.2.1 Componentes Estruturais de Eletrodos de Referência	35
3.2.2.2 O Eletrodo Padrão de Hidrogênio (EPH)	36
3.2.2.3 Eletrodo de Hidrogênio Conveniente (CHE)	36
3.2.3 Voltametria Cíclica	38
3.3 POTENCIOSTATO	40
3.3.1 Estrutura	40
3.3.2 Topologias	45
4 METODOLOGIA	48
4.1 ARQUITETURA ELETRÔNICA	49
4.2 SOFTWARE DE CONTROLE E INTERFACE	50
4.3 ELETRODO DE REFERÊNCIA E CONJUNTO ELETROQUÍMICO	54
4.4 VALIDAÇÃO E ANÁLISE DE DESEMPENHO	55
5 RESULTADOS E DISCUSSÃO	56
5.1 LISTA DE COMPONENTES	56
5.2 CONSTRUÇÃO DO POTENCIOSTATO	57
5.3 CARACTERIZAÇÃO ELÉTRICA	61
5.4 ANÁLISE DA RESPOSTA DINÂMICA (VELOCIDADE E ESTABILIZAÇÃO)	64

5.5	VALIDAÇÃO COMPARATIVA: POTENCIOSTATO COMERCIAL VS. PROTÓTIPO (ESCALA DE ± 1 A)	69
5.6	VALIDAÇÃO POTENCIOSTATO: PROTÓTIPO (ESCALA DE ± 10 mA)	71
5.7	ESTABILIDADE DO ELETRODO DE REFERÊNCIA	72
6	CONSIDERAÇÕES FINAIS	75
	REFERÊNCIAS	76
	ANEXOS	78

1 INTRODUÇÃO

Os fenômenos eletroquímicos vêm sendo estudados há mais de dois séculos, desde as primeiras investigações sobre a relação entre eletricidade e reações químicas. Esses estudos formaram a base da eletroquímica moderna, um campo que evoluiu significativamente e abrange desde o armazenamento de energia até processos industriais (BARD; FAULKNER, 2001). Contudo, o avanço desta ciência está intrinsecamente ligado à capacidade de controle e medição precisa de grandezas elétricas em interfaces metal-eletrólito.

Nesse cenário, o potenciostato surge como o instrumento indispensável, sendo responsável por controlar o potencial entre eletrodos e registrar a corrente resultante (BARD; FAULKNER, 2001). Como destacado por Mattos e Silva (Mattos; Silva, 1983), a identificação dos processos que ocorrem em uma interface depende da versatilidade e rapidez desses equipamentos, que tradicionalmente apresentam alto custo de aquisição, muitas vezes exigindo importação a valores de milhares de dólares, dificultando o acesso em diversos ambientes de pesquisa e ensino.

Somado ao elevado investimento inicial, a manutenção desses sistemas comerciais proprietários impõe barreiras adicionais. O custo de reposição de peças específicas e a dependência de assistência técnica especializada podem tornar o equipamento ocioso por longos períodos em caso de falha. Portanto, a existência de opções baseadas em hardware aberto e componentes de fácil obtenção é estratégica; ela não apenas reduz o custo operacional, mas assegura a sustentabilidade da pesquisa através da facilidade de reparo e da autonomia na customização do instrumento.

A caracterização de processos eletroquímicos, como a eletrodeposição e as técnicas voltamétricas, exige uma instrumentação robusta. Embora o entendimento químico seja o objetivo final de muitos usuários, o desenvolvimento do hardware, que envolve o controle do potencial aplicado a partir da tensão do sistema proveniente do eletrodo de referência para garantir estabilidade e a implementação de topologias de circuitos precisas (Smith; Stevenson, 2007), representa um desafio de engenharia e eletrônica aplicada.

Com o advento de plataformas de hardware e software abertos, como Arduino e Python, tornou-se viável o desenvolvimento de instrumentos científicos acessíveis e adaptáveis (CAMBURN et al., 2017). Esta abordagem permite que o foco da pesquisa se desloque para a arquitetura do instrumento, possibilitando a criação de dispositivos de baixo custo sem comprometer a funcionalidade necessária para aplicações educacionais e laboratoriais.

Cabe ressaltar que o foco central deste trabalho é o desenvolvimento, a implementação e a validação da instrumentação eletrônica. Portanto, não será abordado a análise química e física dos resultados gerados no processo de voltametria com a utilização do instrumento desenvolvido. O objetivo é fornecer uma ferramenta funcional e didática, priorizando os aspectos de projeto de hardware e integração de sistemas.

Ao longo deste trabalho, são apresentados os fundamentos teóricos necessários para

a compreensão da carga eletrônica, os princípios de funcionamento da voltametria sob a ótica da instrumentação e as principais topologias de circuitos utilizadas em potenciostatos.

2 OBJETIVOS

Esse trabalho se propõe projeto, desenvolvimento e validação de um protótipo de potenciostato para realização de ensaios de voltametria cíclica.

- **Objetivo Geral:**

- Projetar, desenvolver e validar um protótipo de um potenciostato para realização de ensaios de voltametria cíclica.

- **Objetivos Específicos:**

- Estudar as topologias utilizadas em potenciostatos;
- Projetar e desenvolver um circuito eletrônico para controle de potencial e medição de corrente elétrica em uma célula eletroquímica;
- Desenvolver um programa em Python para interface com o circuito eletrônico;
- Efetuar a calibração do instrumento desenvolvido com padrões resistivos;
- Avaliar o protótipo desenvolvido em ensaios de voltametria cíclica usando uma dummy cell, comparando seus resultados com um potenciostato comercial;
- Caracterizar a vida útil do eletrodo de referência de hidrogênio construído localmente.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 ELETRÔNICA APLICADA À INSTRUMENTAÇÃO

3.1.1 Amplificadores Operacionais

Os amplificadores operacionais são dispositivos eletrônicos amplamente empregados em instrumentação científica e sistemas de aquisição de dados, devido às suas propriedades que permitem amplificação precisa, controle de sinais e elevada estabilidade. Esses circuitos são projetados para operar de forma previsível e linear dentro de uma ampla faixa de aplicações analógicas, proporcionando uma opção de fácil acesso para: conversão, filtragem e condicionamento de sinais elétricos em sistemas de medição. Nesta seção, são descritos seus princípios de funcionamento, propriedades ideais e reais, bem como suas configurações mais comuns, com base em (BARD; FAULKNER, 2001).

Na Figura 1(a), observa-se a representação esquemática básica de um amplificador operacional, destacando suas conexões principais: terminais de entrada, saída e linhas de alimentação de tensão. Já a Figura 2 apresenta uma segunda representação que inclui os pinos de ajuste de *offset*, utilizados para corrigir tensões ou correntes residuais indesejadas que surgem devido a imperfeições de fabricação. Essas imperfeições podem gerar uma tensão de saída diferente de zero mesmo quando não há diferença de potencial entre as entradas. A compensação pode ser realizada por meio de circuitos de ajuste ou trimpots externos (BARD; FAULKNER, 2001). Ressalta-se que existem outros circuitos para a correção do off-set de entrada,

Em geral, os amplificadores operacionais são construídos de modo que nenhum dos terminais de entrada precise estar conectado diretamente ao terra, ou seja, ambos podem estar flutuando em relação a ele. O parâmetro mais importante é a diferença de potencial entre os terminais de entrada, denominada tensão diferencial. Conforme ilustrado na Figura 1(b), o terminal superior é denominado entrada inversora (–) e o inferior, entrada não inversora (+) (BARD; FAULKNER, 2001).

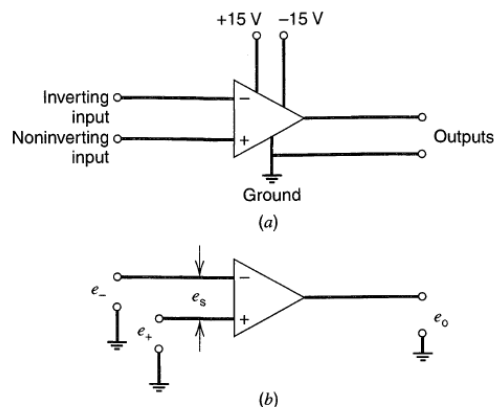


Figura 1: Diagrama esquemático de amplificadores operacionais

Fonte: (BARD; FAULKNER, 2001)

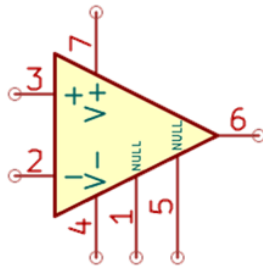


Figura 2: Diagrama esquemático de amplificadores operacionais com pinos de ajuste de *offset*

Fonte: (BARD; FAULKNER, 2001)

A propriedade fundamental de um amplificador operacional é que a tensão de saída (e_o) é proporcional à diferença entre as tensões aplicadas nas entradas (e_s), amplificada por um fator A , denominado ganho em malha aberta. Matematicamente, tem-se (BARD; FAULKNER, 2001):

$$e_o = Ae_s \quad (1)$$

ou, de forma expandida:

$$e_o = A(e_+ - e_-) \quad (2)$$

Portanto:

$$e_o = Ae_+ - Ae_- \quad (3)$$

Assim, a saída representa a combinação dos sinais aplicados às entradas, sendo o sinal da entrada inversora invertido e amplificado, enquanto o da entrada não inversora é amplificado mantendo sua fase original. No caso ideal, o ganho A é considerado infinito, o que implica que mesmo pequenas diferenças de potencial entre as entradas levam a grandes variações na saída. Consequentemente, em um amplificador operacional ideal, os terminais de entrada tendem a permanecer no mesmo potencial elétrico sempre que a saída estiver dentro dos limites de saturação.

Além do ganho infinito, o amplificador operacional ideal possui impedância de entrada infinita (não consome corrente das fontes de entrada), impedância de saída nula (fornece qualquer corrente necessária à carga) e largura de banda infinita (responde instantaneamente a qualquer frequência de sinal). Essas propriedades o tornam ideal para medições precisas, pois permitem amplificar sinais sem interferir em sua fonte. Contudo, dispositivos reais apresentam limitações que devem ser consideradas em aplicações práticas (BARD; FAULKNER, 2001).

As principais não idealidades que influenciam o comportamento dos amplificadores reais são descritas a seguir (BARD; FAULKNER, 2001):

- **Ganho de Malha Aberta:** em amplificadores reais, o ganho A varia tipicamente entre 10^4 e 10^8 para sinais em corrente contínua. Esse ganho diminui conforme a frequência aumenta, o que limita a faixa útil de operação.
- **Largura de Banda:** a largura de banda de ganho unitário é a frequência em que o ganho se reduz a um. Para amplificadores de uso geral, essa faixa varia entre 5 e 20 MHz. A resposta em alta frequência também é caracterizada pela taxa de variação (*slew rate*), que define a velocidade máxima de resposta da saída a variações rápidas de entrada, e pelo tempo de acomodação (*settling time*), que indica o tempo necessário para a saída estabilizar-se dentro de uma margem de erro especificada.
- **Impedância de Entrada:** varia de 10^5 a $10^{13} \Omega$, sendo elevada o suficiente para evitar que o amplificador interfira significativamente na fonte de sinal.
- **Limites de Saída:** dependem da tensão de alimentação do circuito. Em geral, o amplificador pode fornecer correntes de até ± 100 mA, com tensões limitadas a aproximadamente ± 13 a ± 14 V.
- **Tensão de Offset:** mesmo com entradas nulas, pode haver uma pequena tensão de saída devido a desequilíbrios internos. Essa diferença pode ser compensada externamente por meio de resistores ajustáveis.

Para que o amplificador opere de forma estável e previsível, é comum utilizar técnicas de realimentação (*feedback*), que consistem em retornar parte do sinal de saída à entrada do circuito. O tipo de realimentação empregado determina as propriedades e a função do amplificador dentro do sistema (BARD; FAULKNER, 2001). A realimentação negativa diminui (ou limita) ou limita o ganho do Amp. Op e evita oscilações. A realimentação positiva é utilizada quando se deseja projetar osciladores.

3.1.1.1 Realimentação de Corrente

A Figura 3 mostra um circuito básico de realimentação de corrente. O resistor R_f estabelece o caminho de retorno do sinal, e a corrente de realimentação (i_f) se relaciona com a corrente de entrada (i_{in}) pela conservação de carga (BARD; FAULKNER, 2001):

$$i_f = -i_{in} \quad (4)$$

Aplicando a Lei de Ohm, temos:

$$\frac{e_o - e_s}{R_f} = -i_{in} \quad (5)$$

Sabendo que $e_s = e_o / -A$, resulta em:

$$e_o \left(1 + \frac{1}{A} \right) = -i_{in} R_f \quad (6)$$

Como A vale ∞ , o termo entre parênteses fica 1:

$$e_o \simeq -i_{in} R_f \quad (7)$$

Esse circuito (Figura 3) é conhecido como *seguidor de corrente* ou *conversor corrente-tensão* (i/V), pois converte correntes em tensões proporcionais. O ponto somador (S) atua como um *terra virtual*, pois encontra-se sempre no potencial zero e não conectado ao terra, condição que permite realizar a conversão mantendo a fonte de corrente em potencial de referência. Essa característica é amplamente explorada em instrumentação eletroquímica, como sensor de corrente em potenciostatos (BARD; FAULKNER, 2001).

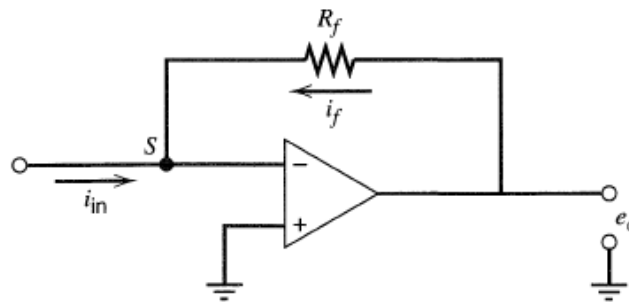


Figura 3: Circuito conversor corrente-tensão

Fonte: (BARD; FAULKNER, 2001)

3.1.1.2 Amplificador Inversor

Na Figura 4, observa-se o circuito de amplificação inversora, em que a entrada é aplicada através de um resistor R_i . Assim, $i_{in} = e_i/R_i$, e a relação de saída torna-se (BARD; FAULKNER, 2001):

$$e_o = -e_i \left(\frac{R_f}{R_i} \right) \quad (8)$$

Esse circuito realiza a amplificação com inversão de fase, representada pelo sinal “-”, sendo o ganho determinado pela razão R_f/R_i . Quando $R_f = R_i$, a saída é uma réplica invertida da entrada. A impedância de entrada é aproximadamente igual a R_i , com valores típicos entre 1 e 100 k Ω (BARD; FAULKNER, 2001).

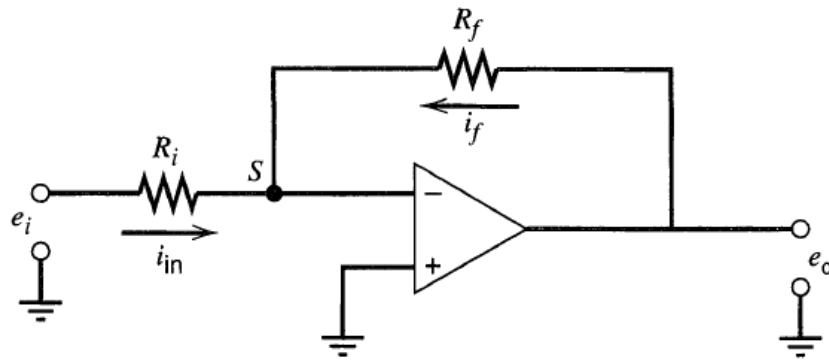


Figura 4: Amplificador inversor (*scaler*)

Fonte: (BARD; FAULKNER, 2001)

3.1.1.3 Somador Inversor

A configuração da Figura 5 permite somar diferentes tensões de entrada, cada uma associada a um resistor específico (R_1 , R_2 , R_3). Pela aplicação da Lei de Kirchhoff, obtém-se (BARD; FAULKNER, 2001):

$$e_o = - \left[e_1 \left(\frac{R_f}{R_1} \right) + e_2 \left(\frac{R_f}{R_2} \right) + e_3 \left(\frac{R_f}{R_3} \right) \right] \quad (9)$$

Quando todos os resistores são iguais ($R_f = R_1 = R_2 = R_3$), o circuito atua como um somador simples:

$$e_o = -(e_1 + e_2 + e_3) \quad (10)$$

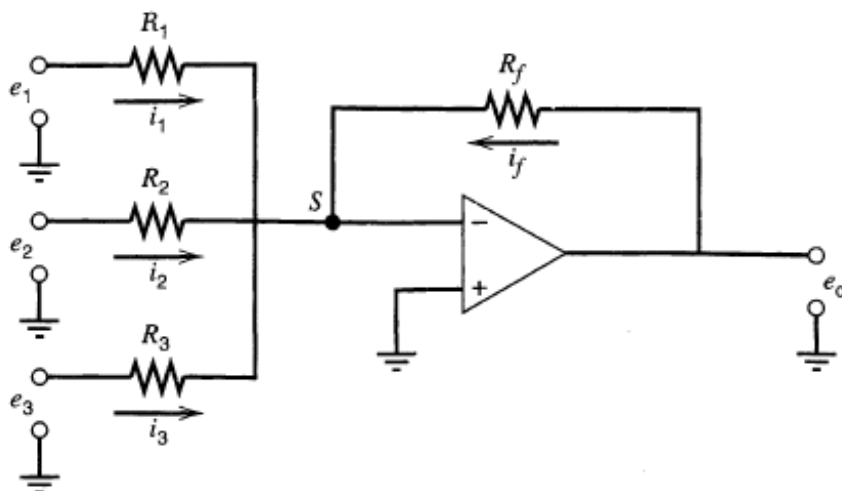


Figura 5: Somador inversor

Fonte: (BARD; FAULKNER, 2001)

3.1.1.4 Realimentação de Tensão

Além da realimentação de corrente, é possível estabilizar o circuito retornando parte da tensão de saída para a entrada inversora. Esse arranjo funciona como um buffer de ganho unitário e alta impedância de entrada. Um exemplo é o circuito seguidor de tensão, mostrado na Figura 6, cuja saída é igual à entrada (BARD; FAULKNER, 2001):

$$e_o \approx e_i \quad (11)$$

Esse tipo de circuito é usado como *buffer*, permitindo a leitura de tensões sem afetar a fonte do sinal (BARD; FAULKNER, 2001).

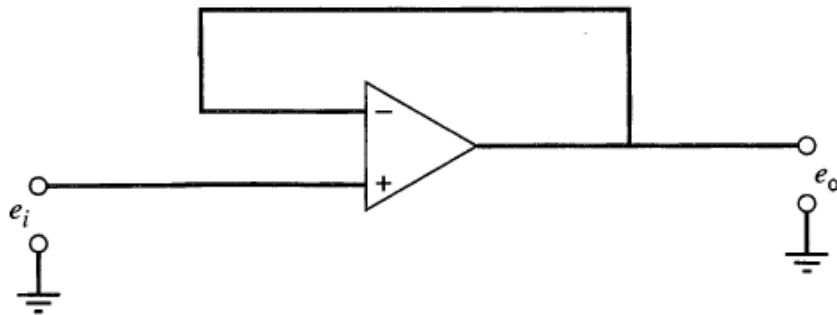


Figura 6: Seguidor de tensão

Fonte: (BARD; FAULKNER, 2001)

3.1.1.5 Controle de Potencial

Por fim, a Figura 7 apresenta um circuito de controle de potencial, no qual o amplificador ajusta sua saída para manter a tensão em um ponto fixo (A) da rede resistiva, mesmo que as impedâncias variem. A relação é dada por (BARD; FAULKNER, 2001):

$$e_o = -e_i \left(\frac{R_1 + R_2}{R_2} \right) \quad \text{ou} \quad e_o = -e_i \left(\frac{R_1}{R_2} + 1 \right) \quad (12)$$

Esse princípio de controle é fundamental em circuitos eletroquímicos, permitindo o ajuste e a manutenção de potenciais precisos em células de medição (BARD; FAULKNER, 2001).

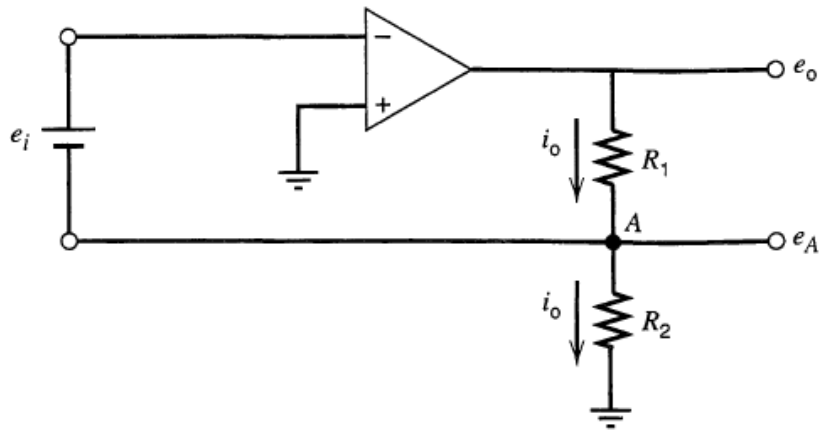


Figura 7: Circuito para controle de potencial no ponto A, independente das variações em R_1 e R_2

Fonte: (BARD; FAULKNER, 2001)

3.1.2 Conversores Digital-Analógico (DAC)

Os conversores digital-analógicos (DAC) são circuitos responsáveis por transformar sinais digitais, representados por combinações binárias de 0 e 1, em sinais analógicos contínuos. Esses dispositivos são fundamentais em sistemas eletrônicos modernos, especialmente em instrumentação e controle, pois permitem que processadores digitais interajam com dispositivos analógicos, como alto-falantes, atuadores e equipamentos de medição (Floyd, 2007).

Um dos métodos clássicos de conversão é o conversor com resistores ponderados, no qual os valores das resistências são proporcionais aos pesos binários dos bits de entrada. Cada bit de entrada, quando em nível lógico alto, permite a passagem de uma corrente proporcional ao seu peso binário, e a soma dessas correntes, aplicada ao amplificador operacional, gera uma tensão de saída proporcional ao valor digital aplicado. A Figura 8 exemplifica um conversor de 4 bits com resistores ponderados em potências de dois.

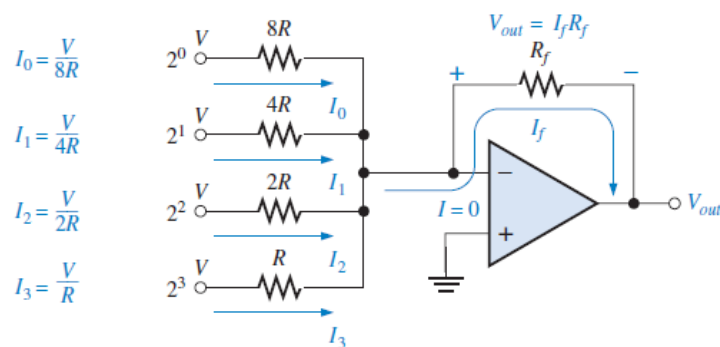


Figura 8: Conversor digital-analógico de 4 bits com resistores ponderados em binário

Fonte: (Floyd, 2007)

Apesar da simplicidade conceitual, esse tipo de DAC apresenta limitações práticas, pois exige resistores com valores muito precisos e tolerâncias extremamente baixas. Em conversores com mais bits, como de 8 ou 12 bits, essa exigência torna-se inviável economicamente, uma vez que o menor erro de resistência causa distorções perceptíveis no sinal de saída (Floyd, 2007).

Para contornar essas dificuldades, utiliza-se frequentemente a topologia de escada $R/2R$, ilustrada na Figura 9. Nesse arranjo, apenas dois valores de resistores são necessários, o que simplifica o processo de fabricação e garante maior estabilidade e precisão na conversão. A estrutura $R/2R$ distribui as correntes de modo que cada bit contribua para a tensão de saída em proporção ao seu peso binário, mantendo a linearidade do conversor.

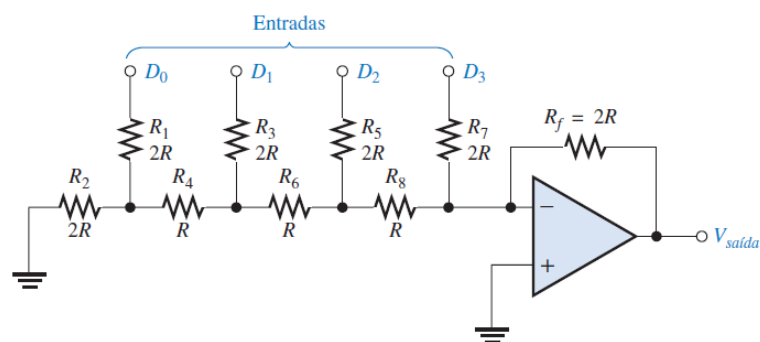


Figura 9: Conversor digital-analógico tipo escada $R/2R$

Fonte: (Floyd, 2007)

A operação do DAC tipo escada baseia-se na soma de correntes binariamente ponderadas que passam por um amplificador operacional configurado como somador inversor. Cada entrada digital ativa resulta em uma corrente específica através da rede resistiva, sendo a tensão de saída proporcional à soma dessas correntes. A precisão do circuito depende da correspondência exata entre os valores de R e $2R$, o que é relativamente simples de se obter na fabricação de circuitos integrados (Floyd, 2007).

A Figura 10 mostra de forma simplificada o funcionamento do DAC no modo $R/2R$.

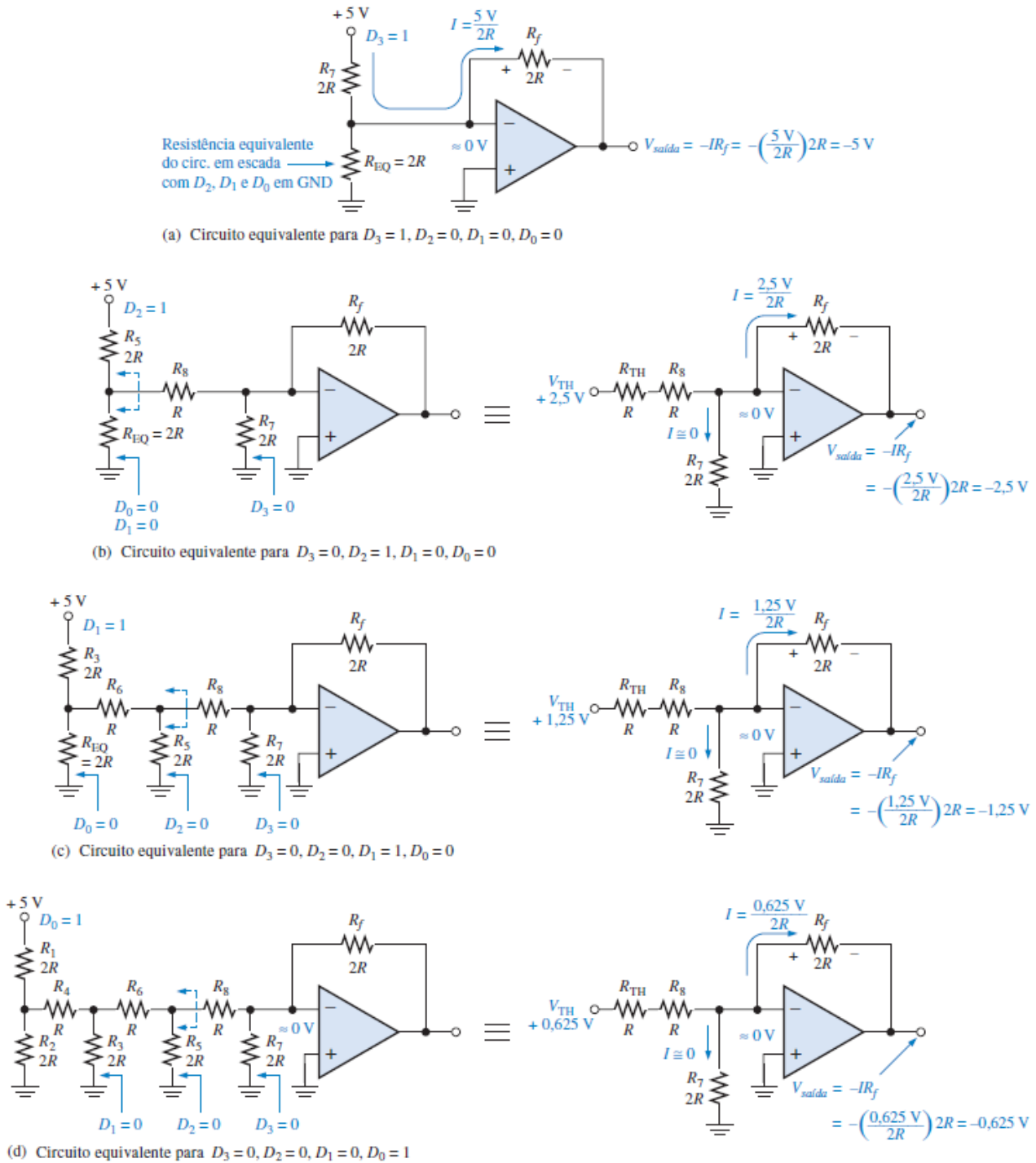


Figura 10: Análise de funcionamento de um DAC com rede R/2R

Fonte: (Floyd, 2007)

3.1.3 Conversores Analógico-Digitais (ADC)

Enquanto o DAC converte dados digitais em sinais analógicos, o conversor analógico-digital (ADC) realiza o processo inverso, permitindo que grandezas físicas contínuas — como tensão, corrente, temperatura e pressão — sejam representadas numericamente para processamento em sistemas digitais (Floyd, 2007).

O sinal analógico, caracterizado por sua variação contínua no tempo e amplitude, precisa ser amostrado e quantizado para que possa ser interpretado por um sistema digital. Esse processo é dividido em etapas principais: condicionamento do sinal, amostragem, retenção e conversão propriamente dita. Cada etapa desempenha um papel crucial na preservação da informação contida no sinal original e na minimização de distorções durante a digitalização.

Inicialmente, o sinal analógico passa por filtros que limitam sua banda de frequência, prevenindo o fenômeno do *aliasing*, que ocorre quando a taxa de amostragem é inferior ao mínimo exigido pelo teorema de Nyquist, que afirma que um sinal analógico pode ser reconstruído perfeitamente a partir de suas amostras se a frequência de amostragem for pelo menos o dobro da frequência mais alta do sinal (Corp., 2025). Em seguida, o circuito de amostragem e retenção (*Sample and Hold* (S/H)) captura e mantém o valor instantâneo do sinal por um breve período, permitindo que o conversor realize a digitalização de forma estável e precisa. Esse processo é ilustrado na Figura 11.

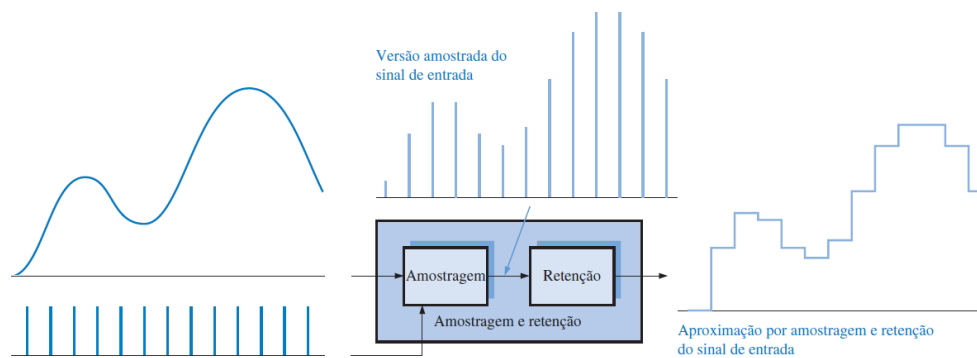


Figura 11: Representação da operação de amostragem e retenção

Fonte: (Floyd, 2007)

Após a etapa de amostragem, o ADC converte o valor mantido em um código binário proporcional à sua amplitude. Cada nível do sinal analógico corresponde a um código digital distinto, conforme demonstrado na Figura 12. Essa quantização permite que sistemas microprocessados realizem medições, cálculos e controles com base em dados discretizados.

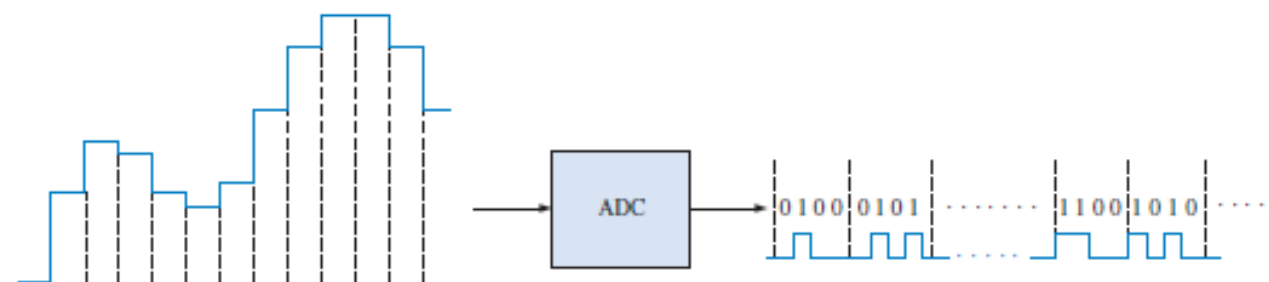


Figura 12: Funcionamento básico de um conversor analógico-digital

Fonte: (Floyd, 2007)

Entre os diversos métodos de conversão, destaca-se o conversor *sigma-delta*, ampla-

mente empregado em aplicações de áudio e instrumentação de precisão. Esse tipo de conversor utiliza o princípio da modulação delta, onde a diferença entre amostras sucessivas é quantizada, resultando em uma sequência de bits de um único nível lógico. A densidade de bits "1" ao longo do tempo representa a amplitude do sinal de entrada. Essa técnica permite alcançar alta resolução através de sobreamostragem (*oversampling*) e filtragem digital, compensando a baixa resolução do quantizador interno. A Figura 13 mostra uma representação simplificada desse processo.

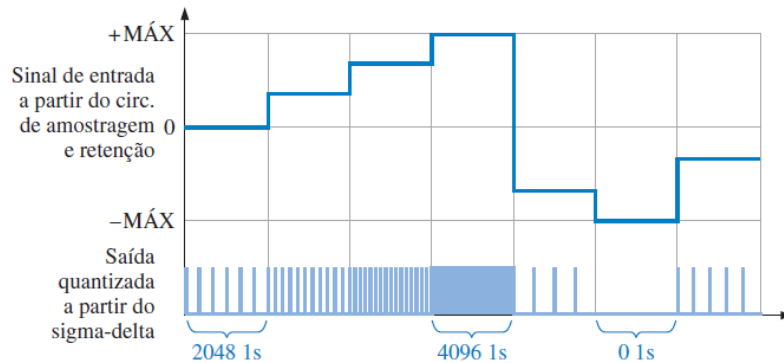


Figura 13: Representação simplificada da conversão sigma-delta

Fonte: (Floyd, 2007)

O funcionamento interno de um conversor sigma-delta é mostrado na Figura 14. O sinal analógico é comparado com o sinal realimentado proveniente de um DAC interno, e a diferença (erro) é integrada ao longo do tempo. Um quantizador de 1 bit — essencialmente um comparador com memória — gera a sequência binária que, ao ser filtrada digitalmente por um filtro passa-baixas, resulta em um código proporcional à amplitude do sinal original. O processo de integração e realimentação garante que o erro de quantização seja distribuído ao longo de múltiplas amostras, melhorando a relação sinal-ruído do conversor.

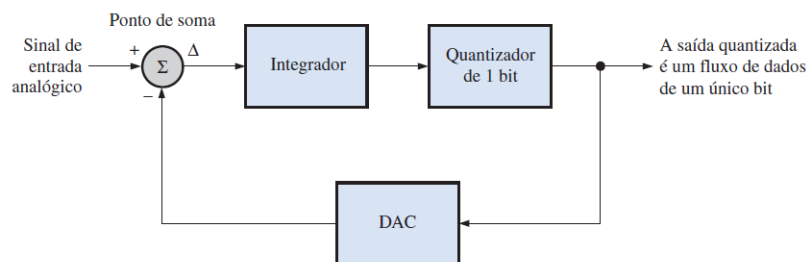


Figura 14: Diagrama funcional simplificado de um ADC sigma-delta

Fonte: (Floyd, 2007)

Por fim, um contador digital ou um decimador contabiliza os bits "1" do fluxo quantizado e gera os códigos binários correspondentes, como apresentado na Figura 15. Essa abordagem proporciona alta resolução e excelente imunidade a ruídos, características que tornam os conversores sigma-delta ideais para medições de sinais analógicos de baixa amplitude. A resolução

efetiva pode atingir 16 bits ou mais, dependendo da taxa de amostragem e da ordem do modulador (Floyd, 2007).

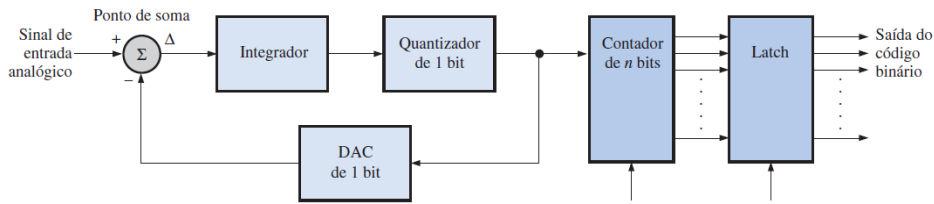


Figura 15: Exemplo de implementação de um conversor sigma-delta

Fonte: (Floyd, 2007)

3.1.4 Medição de corrente

A medição de correntes muito pequenas em sistemas eletrônicos e eletroquímicos requer circuitos de alta sensibilidade e baixo ruído capazes de converter ou integrar a corrente de entrada de forma precisa. Em instrumentação eletrônica, as três topologias mais comuns empregadas para essa finalidade são o conversor corrente–tensão (i – V), também conhecido como transimpedância, o amplificador com resistor de derivação (shunt) e o integrador de corrente. Cada uma dessas abordagens apresenta vantagens e limitações específicas, sendo a escolha dependente da faixa de corrente, da velocidade de resposta requerida e do nível de ruído aceitável. A Figura 16 ilustra as configurações típicas de cada método, amplamente utilizados em medições de correntes, tanto em sensores ópticos e fotodiodos quanto em sistemas potenciostáticos (Koutilellis; Economou; Efstathiou, 2020).

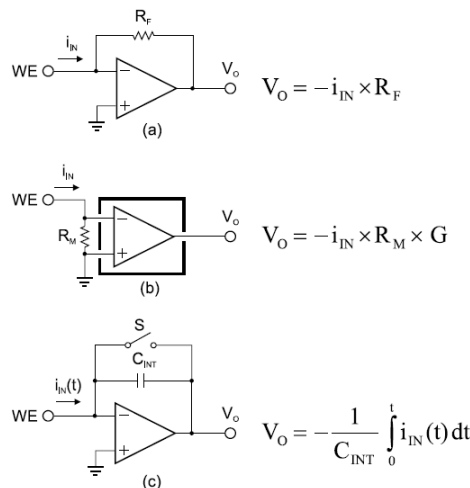


Figura 16: Circuitos comuns de medição de corrente e as respectivas funções de transferência: (a) Um conversor i/V simples usando um resistor de realimentação. (b) Um amplificador de instrumentação com fator de ganho G amplifica a queda de tensão em um resistor relativamente pequeno. (c) Um circuito integrador acumula a carga correspondente à corrente de entrada por um período de tempo fixo.

Fonte: Adaptado de (Koutilellis; Economou; Efstathiou, 2020).

3.1.4.1 Conversor transimpedância (current-to-voltage)

O conversor transimpedância é um dos circuitos mais utilizados para converter correntes em tensões mensuráveis. Ele consiste em um amplificador operacional em configuração inversora com um resistor de realimentação R_F , que gera uma tensão de saída proporcional à corrente de entrada, V_O/R_F deve "anular" I_{in} , considerando que I_- seja zero, Amp Op. ideal.

$$V_O = -I_{IN} \cdot R_F \quad (13)$$

Essa topologia apresenta ampla faixa dinâmica e resposta linear, sendo especialmente útil em sensores de corrente muito baixa, como fotodiodos e eletrodos de trabalho em medições amperométricas. Entretanto, o uso de resistores de realimentação muito altos ($> 100 \text{ M}\Omega$) eleva o ruído térmico ((Koutilellis; Economou; Efstathiou, 2020)) e torna o circuito suscetível a correntes de polarização e instabilidades térmicas. O projeto adequado deve equilibrar ganho, largura de banda e ruído para otimizar o desempenho em medições de corrente contínua (DC) e pulsada.

3.1.4.2 Amplificação da queda de tensão em resistor de derivação (shunt) com ganho

Outra abordagem consiste em medir a corrente indiretamente através da queda de tensão gerada em um resistor de derivação R_M e amplificar esse sinal usando um estágio com ganho G , resultando em:

$$V_O = -I_{IN} \cdot R_M \cdot G. \quad (14)$$

Esse método apresenta menor ruído térmico devido ao baixo valor de R_M , sendo apropriado para correntes mais elevadas. A precisão depende da estabilidade térmica do resistor e do desempenho do amplificador de instrumentação, que deve apresentar alta rejeição de modo comum (CMRR) e baixo ruído de entrada. Essa configuração é amplamente usada em instrumentação geral e medições de potência, além de circuitos de controle de corrente em fontes de bancada e potenciostatos compactos.

3.1.4.3 Integrador de corrente

O integrador de corrente ou amplificador de carga é uma alternativa de alta sensibilidade baseada em um amplificador operacional com um capacitor de integração C_{INT} na realimentação. A saída do circuito é proporcional à carga acumulada durante o intervalo de integração Δt :

$$V_O = -\frac{1}{C_{INT}} \int_{t_0}^{t_0+\Delta t} I_{IN}(t) dt \approx -\frac{\Delta t}{C_{INT}} I_{IN}. \quad (15)$$

Esse tipo de circuito é vantajoso para a detecção de correntes muito baixas ou detecção de carga, pois reduz o ruído por integração temporal e permite ampliar a faixa dinâmica variando Δt e C_{INT} . A implementação requer controle preciso da chave de reset e do tempo de integração, frequentemente realizado por microcontroladores ou conversores analógico-digitais sincronizados. Segundo Koutilellis

et al. (Koutilellis; Economou; Efstathiou, 2020), essa arquitetura foi empregada em um potenciostato experimental com amplificador transimpedância integrador, demonstrando excelente desempenho na medição de correntes ultra-baixas em microcanais e voltametria de baixo ruído.

Em resumo:

- **Transimpedância (a)**: simples, linear e de resposta rápida; limitações de ruído térmico e correntes de fuga em altas resistências.
- **Shunt + ganho (b)**: robusta e de fácil implementação; adequada para correntes moderadas, mas sensível à variação térmica do resistor e à precisão do ganho.
- **Integrador (c)**: ideal para correntes extremamente baixas, com ruído reduzido e ampla faixa dinâmica; requer controle digital e calibração periódica.

A escolha entre as três topologias depende da faixa de corrente de interesse, do nível de ruído aceitável e da arquitetura eletrônica disponível. Em sistemas de instrumentação científica, é comum combinar técnicas — como integração seguida de amplificação — para alcançar máxima precisão e estabilidade de longo prazo em medições na faixa de femtoampères a nanoampères (Koutilellis; Economou; Efstathiou, 2020).

3.1.5 Arduino

O Arduino é uma plataforma *open-source* de eletrônicos desenvolvida com o objetivo de democratizar o acesso à eletrônica e à programação. Além de abranger o conceito de código aberto, o Arduino também se enquadra nos princípios de *Open Hardware*, uma vertente do movimento *open source* voltada a dispositivos físicos. Sob essa filosofia, suas especificações, esquemáticos e projetos são disponibilizados publicamente, permitindo que qualquer pessoa estude, modifique, fabrique e distribua versões derivadas de forma livre, desde que respeitadas as licenças de uso (Arduino Foundation, 2023; Red Hat, Inc., 2023).

De acordo com a Red Hat (Red Hat, Inc., 2023), o *Open Hardware* é caracterizado pela abertura de projetos de engenharia, esquemas elétricos e layouts de placas de circuito impresso (PCB), incentivando modificações, melhorias e distribuições de forma colaborativa. No caso do Arduino, a licença empregada é a CC-BY-SA 3.0 (*Creative Commons Attribution-ShareAlike 3.0*), que permite copiar, distribuir e modificar o conteúdo, inclusive para fins comerciais, desde que o autor original seja devidamente creditado, as alterações sejam indicadas, e as versões derivadas sejam compartilhadas sob a mesma licença (Creative Commons, 2013).

Essa abordagem colaborativa tornou o Arduino uma das plataformas mais utilizadas em ensino, pesquisa e desenvolvimento de protótipos eletrônicos, especialmente em áreas como automação, instrumentação científica e sistemas embarcados. A comunidade global criada em torno da plataforma contribui ativamente para o desenvolvimento de novas bibliotecas, módulos e versões de hardware, entre elas o Arduino Nano, que é o modelo utilizado neste trabalho.

O Arduino Nano é uma placa de prototipagem baseada no microcontrolador ATmega328P da fabricante Microchip (antiga Atmel), caracterizando-se por seu tamanho compacto e compatibilidade elétrica com o Arduino Uno (Arduino Documentation, 2024; Atmel Corporation, 2015). Essa placa possui 14 pinos digitais de entrada/saída, sendo 6 deles capazes de gerar sinais PWM, e 8 entradas analógicas, além de conexões para comunicação serial (UART), I²C e SPI. Sua alimentação pode ser feita via conector USB ou por fonte externa, suportando tensões de 6 V a 12 V, com tensão operacional de 5 V.

O principal diferencial do Nano é sua dimensão reduzida, o que facilita sua integração em projetos de instrumentação e controle, mantendo a funcionalidade completa da plataforma Arduino. Sua frequência de operação é de 16 MHz, e o microcontrolador dispõe de 32 kB de memória *Flash*, 2 kB de SRAM e 1 kB de EEPROM não volátil (Atmel Corporation, 2015). Assim como outras placas da linha, o Nano é programado por meio da IDE oficial do Arduino, utilizando uma linguagem baseada em C/C++, derivada do framework Wiring (Wiring Project, 2003). Além disso, sua compatibilidade com bibliotecas como `pySerial` e firmwares como o `PyFirmata` permite comunicação direta com a linguagem Python, o que amplia suas possibilidades de uso em sistemas de aquisição e automação laboratorial.

A Figura 17 apresenta o diagrama de pinos (*pinout*) do Arduino Nano, destacando as principais funções de cada terminal de conexão.

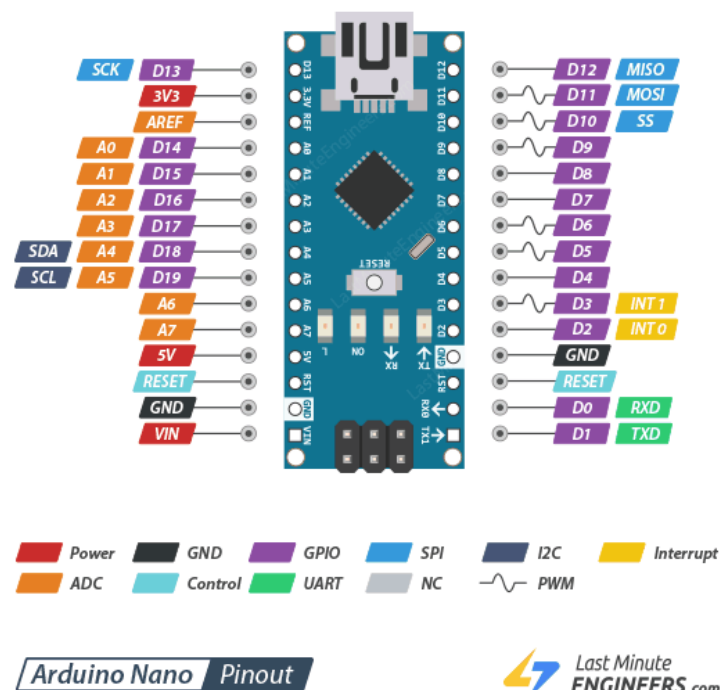


Figura 17: Pinout do Arduino Nano.

Fonte: (Last Minute Engineers, 2024).

3.1.6 Linguagem de Programação Python

A linguagem de programação Python é uma linguagem de alto nível, interpretada e de propósito geral, amplamente utilizada em aplicações científicas, automação, análise de dados e controle de instrumentação. Criada por Guido van Rossum e mantida pela Python Software Foundation, destaca-se por sua sintaxe simples e legibilidade, permitindo rápido desenvolvimento e integração com diversas bibliotecas científicas e de controle (Python Software Foundation, 2024).

No contexto de instrumentação e automação, Python oferece uma ampla gama de bibliotecas, como `pySerial` para comunicação serial com microcontroladores, `matplotlib` e `pyqtgraph` para visualização de dados em tempo real, e `numpy` para cálculos numéricos. Essas características tornam a linguagem especialmente adequada para aplicações que envolvem o controle e aquisição de dados de sensores conectados ao Arduino Nano, facilitando a criação de interfaces gráficas e a análise computacional dos resultados experimentais.

Devido à sua versatilidade e suporte multiplataforma, Python tem se consolidado como uma das linguagens mais utilizadas na área de Engenharia Física e desenvolvimento de sistemas de medição inteligentes, permitindo a integração entre hardware e software de forma acessível e eficiente (Python Software Foundation, 2024).

3.2 ELETROQUÍMICA

3.2.1 Célula eletroquímica e Dummy Cell

A célula eletroquímica também conhecida como sensor eletroquímico utilizada no projeto é constituída por três condutores elétricos denominados eletrodos, imersos em uma solução adequada de um eletrólito que contém a amostra química em análise, chamada de analito (Skoog; Holler; Nieman, 1992). Os três terminais condutores são denominados como denominados: Eletrodo de Trabalho (WE), Eletrodo Auxiliar ou Contra-Eletrodo (CE) e Eletrodo de Referência (RE) (BARD; FAULKNER, 2001). Na Figura 18 é apresentado o esquemático de uma célula de três eletrodos.

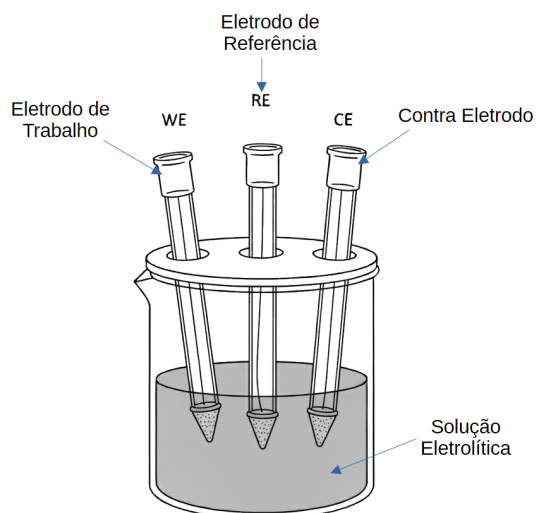


Figura 18: Esquemático de uma Célula Eletroquímica de Três Eletrodos

Fonte: O autor.

Cada eletrodo da célula eletroquímica possui uma função específica. No eletrodo de trabalho, ocorrem as reações de oxidação e redução. O eletrodo de referência tem por função fornecer um potencial estável e bem definido, servindo como base para a medição do potencial do eletrodo de trabalho em relação à solução, sem participar das reações eletroquímicas do sistema, enquanto o contra-eletrodo atua como um condutor, responsável por fornecer a corrente necessária à célula eletroquímica, fluindo até o WE (Skoog; Holler; Nieman, 1992).

A utilização do eletrodo de referência é essencial porque ele estabelece um potencial fixo e reprodutível, permitindo que o potenciostato controle com precisão o potencial aplicado ao eletrodo de trabalho. Sem um eletrodo com potencial estável, não seria possível determinar com exatidão a diferença de potencial que impulsiona as reações eletroquímicas, comprometendo a interpretação dos dados (BARD; FAULKNER, 2001). Além disso, o eletrodo de referência precisa ser quimicamente inerte na solução para evitar qualquer participação em processos de oxirredução ou alterações estruturais que modifiquem seu potencial. Essa inércia garante que o potencial do RE permaneça constante ao longo de toda a medição, atendendo ao requisito fundamental de estabilidade e reprodutibilidade discutido na literatura clássica (Skoog; Holler; Nieman, 1992; BARD; FAULKNER, 2001).

Do ponto de vista eletrônico, uma célula eletroquímica pode ser modelada como uma rede de impedâncias, conforme mostrado no circuito equivalente da Figura 19(a), onde Z_c e Z_{wk} representam as impedâncias interfaciais no CE e de WE, respectivamente. A resistência da solução é dividida em duas parcelas, R_Ω e R_u , de acordo com a posição do eletrodo de referência em relação ao caminho da corrente. A célula apresentada na Figura 19(b) é uma representação simplificada da Figura 19(a), considerando apenas as impedâncias equivalentes finais (BARD; FAULKNER, 2001).

$$Z_1 = Z_C + R_\Omega \quad \text{e} \quad Z_2 = R_u + Z_{wk} \quad (16)$$

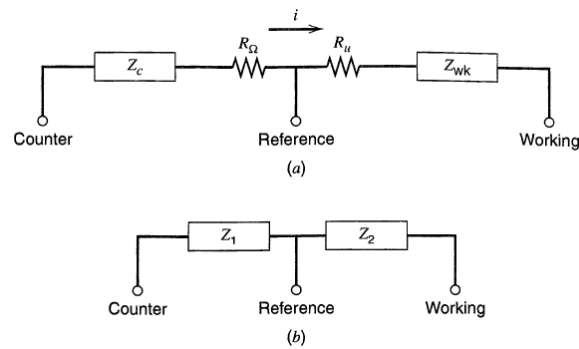


Figura 19: Visões de uma célula eletroquímica como rede de impedâncias conectada aos três eletrodos

Fonte: (BARD; FAULKNER, 2001).

A representação simbólica e o circuito elétrico aproximado da célula eletroquímica são exibidos na Figura 20.

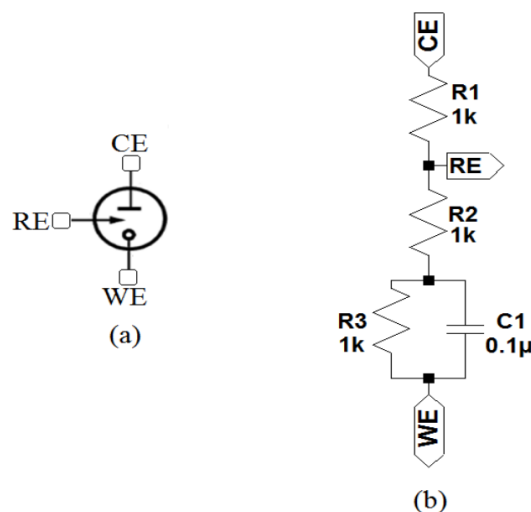


Figura 20: Representações da Célula Eletroquímica: (a) Símbolo da Célula Eletroquímica; (b) Esquemático do Circuito Elétrico aproximado da Célula (valores arbitrários)

Fonte: (Yarnitzky, 2000).

No circuito equivalente (Figura 20b), observa-se a utilização de resistores e de um capacitor, que juntos simulam o comportamento eletroquímico da célula real, formando o que se denomina *dummy cell*. Essa célula permite testar o funcionamento de potenciostatos e sistemas de medição sem a necessidade de uma solução eletrolítica real.

As resistências R_1 , R_2 e R_3 modelam, respectivamente, diferentes componentes da resistência total da célula eletroquímica:

- R_1 : representa a resistência associada ao contra-eletrodo e ao caminho da corrente na solução;
- R_2 : está relacionada à resistência da solução e ao potencial medido pelo eletrodo de referência;

- R_3 : modela a resistência de transferência de carga no eletrodo de trabalho, que influencia a cinética das reações eletroquímicas.

Sendo que:

$$R_1 = Z_C + R_\Omega \quad (17)$$

$$R_2 = R_u \quad (18)$$

$$Z_{wk} = R_3 // C_1 \quad (19)$$

O capacitor C_1 , por sua vez, representa a capacitância da dupla camada elétrica formada na interface entre o eletrodo de trabalho e a solução eletrolítica. Essa camada comporta-se como um condensador devido à separação de cargas entre a superfície do eletrodo e os íons da solução. Conforme discutido por Bard e Faulkner (BARD; FAULKNER, 2001), essa capacitância é um dos elementos mais importantes na resposta de um sistema eletroquímico, pois influencia diretamente a forma e o atraso temporal da corrente medida em experimentos de voltametria e técnicas pulsadas.

Dessa forma, o conjunto resistivo-capacitivo da Figura 20(b) reproduz com boa precisão o comportamento elétrico dinâmico de uma célula real, permitindo validar e calibrar circuitos de controle e aquisição utilizados em instrumentação eletroquímica.

3.2.2 O Eletrodo de Referência de Hidrogênio

Idealmente, RE deve ser não polarizável, significando que seu potencial não se altera com a passagem de corrente. Conforme descrito por Smith e Stevenson (Smith; Stevenson, 2007), a reação no RE deve ser reversível, permitindo que seu potencial seja calculado pela equação de Nernst e que ele recupere seu potencial rapidamente após qualquer sobrecarga de corrente. Esta capacidade permite ao eletrodo atuar como fonte ou dreno para as pequenas correntes de fuga inerentes ao potenciostato, sem exibir desvios de potencial.

3.2.2.1 Componentes Estruturais de Eletrodos de Referência

Eletrodos de referência são tipicamente construídos a partir de quatro componentes essenciais: o corpo, a vedação superior, a junção e o componente ativo. Embora os três primeiros possam ser adaptados para diferentes condições experimentais, é o componente ativo que define o potencial de referência do eletrodo (Smith; Stevenson, 2007).

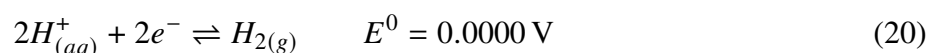
- **Material do Corpo:** A escolha do material para o corpo do RE é ditada pela sua compatibilidade com as condições experimentais e o processo de fabricação. (Smith; Stevenson, 2007).
- **Vedação Superior:** A vedação superior influencia diretamente o desempenho do RE. Uma vedação hermética é crucial em alguns eletrodos de hidrogênio para reter o gás H_2 . Além disso, ela reduz a evaporação da solução de enchimento e seu fluxo através da junção. Vedações não

herméticas (com vazamento) são usadas apenas com soluções de enchimento saturadas (Smith; Stevenson, 2007).

- **Junção:** A junção é a interface que separa a solução de enchimento interna do ER do eletrólito externo da célula. Em certas aplicações, pode-se omitir a junção se as soluções forem compatíveis. Em outros casos, uma junção dupla é empregada para prevenir a contaminação mútua (Smith; Stevenson, 2007).
- **Componente Ativo:** Este é o elemento que define o potencial de base do RE.

3.2.2.2 O Eletrodo Padrão de Hidrogênio (EPH)

O Eletrodo Padrão de Hidrogênio (EPH), é o ponto de referência primário para todos os potenciais de redução padrão. Por convenção internacional, seu potencial (E^0) é definido como exatamente 0.0000 V em todas as temperaturas (Smith; Stevenson, 2007). A reação que define este potencial é:



A definição do EPH exige condições ideais: atividade de íons hidrônio unitária ($a_{H^+} = 1$) e pressão parcial do gás hidrogênio de 1 atm. Na prática, atingir essas condições é complexo (Smith; Stevenson, 2007).

Frequentemente, utilizam-se concentrações de ácido mais baixas e pressões de hidrogênio corrigidas. O potencial real do eletrodo (E) desvia-se do valor padrão, mas pode ser calculado com precisão usando a equação de Nernst:

$$E = E^0 + \frac{RT}{2F} \ln \frac{(a_{H^+})^2}{P_{H_2}} = \frac{RT}{F} \ln \frac{a_{H^+}}{\sqrt{P_{H_2}}} \quad (21)$$

onde R é a constante dos gases, T a temperatura absoluta, F a constante de Faraday, a_{H^+} a atividade do íon hidrônio e P_{H_2} a pressão parcial do hidrogênio. A P_{H_2} é corrigida pela pressão barométrica ($P_{\text{barométrica}}$), a pressão de vapor da água (P_{H_2O}) e a profundidade de imersão (h) da entrada de gás (Smith; Stevenson, 2007):

$$P_{H_2} = P_{\text{barométrica}} - P_{H_2O} + (4.2 \times 10^{-5})h \quad (22)$$

3.2.2.3 Eletrodo de Hidrogênio Conveniente (CHE)

O Eletrodo de Hidrogênio Conveniente (CHE), representa uma alternativa prática ao EPH, pois dispensa o fluxo contínuo e o equipamento complexo de purificação de gás H_2 (Smith; Stevenson, 2007).

Seu design (Figura 21) é otimizado para conter uma **bolha interna de hidrogênio**. A função desta bolha é crucial: ela atua como um reservatório de H_2 . A superfície de platina, localizada acima do nível principal da solução de enchimento, "puxa" uma fina camada de eletrólito por ação capilar. Esta fina camada de solução entra em equilíbrio rápido e direto com a bolha de H_2 contida no espaço fechado.

O propósito desta bolha é, portanto, saturar continuamente a superfície catalítica ativa do eletrodo com H_2 , mimetizando o efeito do borbulhamento constante de gás em um EPH tradicional. Este arranjo permite que o eletrodo mantenha um potencial estável e pode ser facilmente recarregado (seja por geração eletrolítica interna de H_2 ou por preenchimento externo) (Smith; Stevenson, 2007).

Uma limitação é que o design não exclui ativamente o oxigênio atmosférico que pode se difundir. Isso causa um pequeno desvio (alguns milivolts) do potencial teórico, exigindo que o CHE seja calibrado contra outro ER. No entanto, sua robustez e simplicidade o tornam ideal para estudos em ácidos, especialmente em pesquisa de células a combustível, onde a contaminação por íons como cloreto (presente em eletrodos de Ag/AgCl) pode envenenar os catalisadores sob estudo (Smith; Stevenson, 2007).

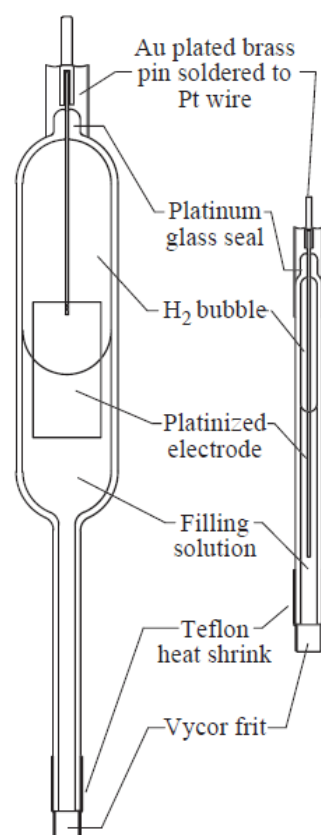


Figura 21: Exemplo de Eletrodo de Hidrogênio Conveniente (CHE). Fonte: Adaptado de (Smith; Stevenson, 2007).

3.2.3 Voltametria Cíclica

Os métodos eletroanalíticos voltamétricos baseiam-se na medição de grandezas elétricas associadas a processos de oxidação e redução que ocorrem na interface eletrodo–solução. Nessas técnicas, espécies eletroativas interagem física e/ou quimicamente com a superfície do eletrodo, produzindo respostas elétricas mensuráveis que podem ser correlacionadas à natureza, concentração e aos mecanismos das reações envolvidas. Devido a essas características, os métodos voltamétricos são amplamente empregados em áreas como monitoramento ambiental, controle de processos industriais e análises biomédicas (Souza, 2016; Skoog; West et al., 2017).

De modo geral, a voltametria fundamenta-se nos fenômenos que ocorrem na região interfacial entre o eletrodo de trabalho e a solução adjacente, onde efetivamente se estabelecem as reações eletroquímicas. A informação analítica é obtida a partir da corrente elétrica que flui entre o eletrodo de trabalho e o eletrodo auxiliar, em resposta a um potencial imposto externamente entre o eletrodo de trabalho e o eletrodo de referência. Assim, nas técnicas voltamétricas, o potencial constitui a variável independente do experimento, enquanto a corrente representa a resposta do sistema eletroquímico (Souza, 2016). Diversas modalidades de voltametria foram desenvolvidas, incluindo a Polarografia, Voltametria por Redissolução, Voltametria Cíclica, Amperometria e técnicas de pulso, cuja escolha depende da forma de aplicação do potencial e dos objetivos analíticos do experimento (Souza, 2016; Skoog; West et al., 2017). Conforme apresentado na Figura 22.

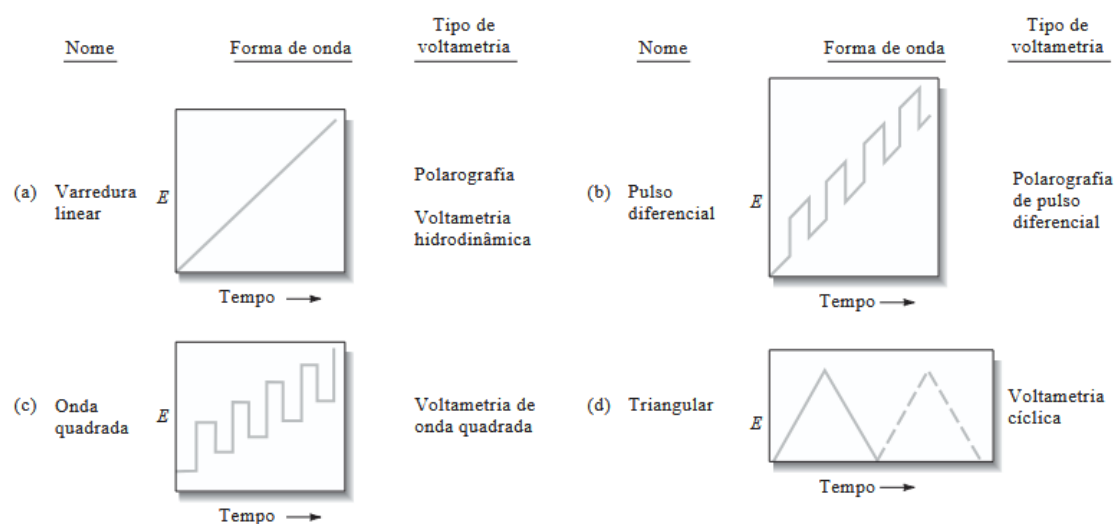


Figura 22: Sinais de excitação de tensão versus tempos empregados na Voltametria

Fonte: (Skoog; West et al., 2017).

Dentre essas técnicas, destaca-se a Voltametria Cíclica (VC), amplamente utilizada na investigação de mecanismos eletroquímicos devido à sua simplicidade experimental e ao elevado conteúdo de informação qualitativa fornecido. Na VC, o potencial do eletrodo de trabalho é forçado pelo potenciostato a variar de maneira controlada no tempo, enquanto a corrente resultante, proveniente das reações redox na interface eletrodo–solução, é continuamente monitorada.

A Figura 23 ilustra esquematicamente os dois elementos fundamentais de um experimento de voltametria cíclica. Na Figura 23(a) é apresentado o sinal de excitação de potencial aplicado ao eletrodo de trabalho em função do tempo. Observa-se que o potencial varia linearmente, segundo uma forma de onda triangular, partindo de um potencial inicial, atingindo um valor máximo — denominado primeiro vértice (1º vertex) — e, em seguida, invertendo o sentido da varredura até alcançar um potencial mínimo — denominado segundo vértice (2º vertex), retornando posteriormente ao valor inicial. A inclinação das rampas corresponde à velocidade de varredura do potencial ($v = dE/dt$), parâmetro experimental de grande influência sobre a resposta do sistema.

É importante ressaltar que o gráfico de potencial versus tempo não representa, por si só, o resultado do experimento, mas sim o sinal imposto ao sistema eletroquímico. Essa característica evidencia que a voltametria cíclica é uma técnica de potencial controlado, na qual o potencial não é determinado pela reação eletroquímica, mas previamente definido pelo operador por meio do potenciostato.

Na Figura 23(b) é apresentado o voltamograma cíclico, correspondente à corrente medida em função do potencial aplicado. Esse gráfico constitui a resposta eletroquímica do sistema ao sinal de excitação mostrado na Figura 23(a). Durante a varredura direta, denominada anódica, o aumento progressivo do potencial favorece processos de oxidação, resultando no surgimento de um pico de corrente anódica, cujo máximo define o potencial de pico anódico ($E_{p,a}$). Após a inversão do sentido da varredura no primeiro vértice, inicia-se a varredura inversa, ou catódica, na qual ocorre a redução das espécies previamente oxidadas, originando o pico catódico associado ao potencial $E_{p,c}$.

Os potenciais $E_{p,a}$ e $E_{p,c}$ fornecem informações fundamentais sobre a reversibilidade e a cinética do processo redox. Em sistemas eletroquimicamente reversíveis, a diferença entre esses potenciais, $\Delta E_p = E_{p,a} - E_{p,c}$, assume valor próximo de $59/n$ mV a 25 °C, sendo n o número de elétrons envolvidos na reação. Desvios significativos desse valor indicam processos quase reversíveis ou irreversíveis, nos quais a cinética de transferência de carga ou o transporte de massa limita a resposta eletroquímica (Skoog; West et al., 2017).

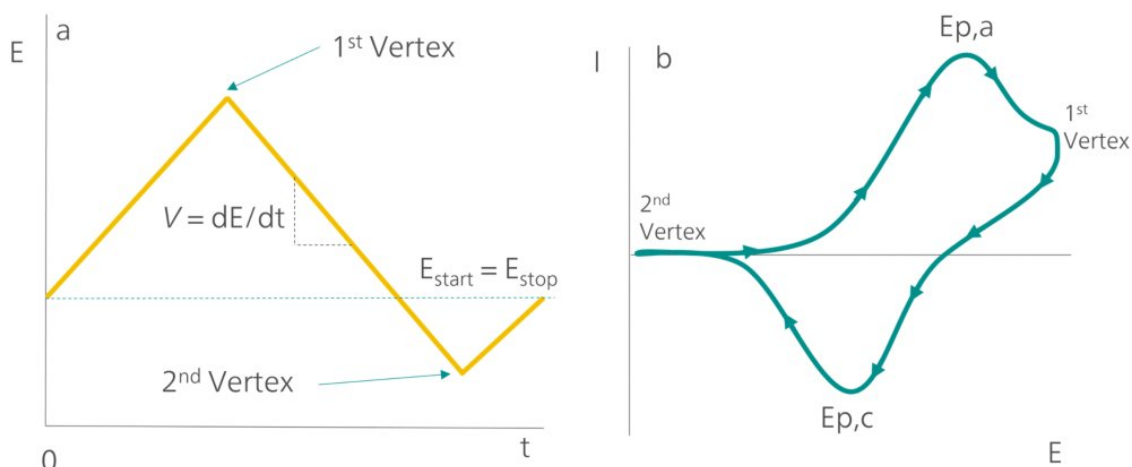


Figura 23: Sinal de excitação triangular e voltamograma cíclico típico

Fonte: (Metrohm, 2025)

3.3 POTENCIOSTATO

3.3.1 Estrutura

O potenciostato é o elemento central de qualquer sistema eletroquímico controlado, sendo responsável por regular o potencial aplicado entre os eletrodos de trabalho e de referência, ao mesmo tempo em que mede a corrente que flui entre os eletrodos de trabalho e auxiliar. Sua função básica é manter constante o potencial do eletrodo de trabalho em relação ao eletrodo de referência, independentemente das variações de impedância da célula ou das correntes envolvidas. Essa capacidade de controle é essencial para a realização de experimentos eletroquímicos reprodutíveis e confiáveis, permitindo o estudo detalhado de mecanismos reacionais e a quantificação de espécies eletroativas.

Na Figura 24 observa-se o diagrama de um potenciostato básico, o qual ilustra os princípios fundamentais do controle de potencial em um sistema de três eletrodos. Nesse circuito, o sinal aplicado ao sistema tem como referência o potencial do eletrodo de referência, e a corrente que circula pela célula é ajustada automaticamente pelo amplificador operacional, de modo que o eletrodo de trabalho mantenha o potencial especificado em relação ao eletrodo de referência. Considerando que o eletrodo de trabalho está aterrado, tem-se:

$$e_{wk} \text{ (vs. ref)} = e_i \quad (23)$$

Sendo que:

$$e_i \equiv \text{Potencial de controle}$$

Esse arranjo garante que a diferença de potencial entre o eletrodo de trabalho e o

de referência seja controlada de forma precisa, mesmo diante de variações na resistência de solução ou de polarização dos eletrodos, conforme discutido anteriormente na seção de célula eletroquímica (BARD; FAULKNER, 2001).

A configuração ilustrada representa o princípio de funcionamento de um potenciostato, mas impõe desafios práticos quanto à implementação de controle. Um dos principais problemas é que nenhum dos terminais da fonte de potencial está conectado a um verdadeiro terra; conseqüentemente, o gerador de funções que fornece a forma de onda de controle de potencial precisa dispor de uma saída diferencial flutuante — característica incomum na maioria dos equipamentos comerciais (BARD; FAULKNER, 2001).

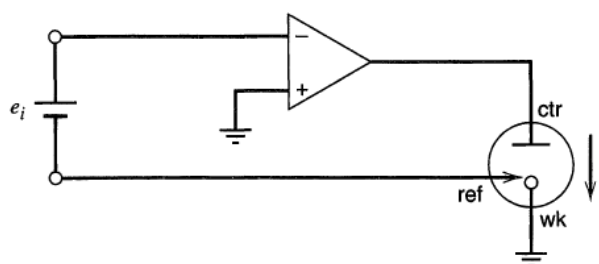


Figura 24: Um potenciostato simples

Fonte: (BARD; FAULKNER, 2001)

Uma modificação que supera essa limitação é o chamado *potenciostato somador*, mostrado na Figura 25. Essa arquitetura permite que múltiplos sinais de entrada sejam combinados e referenciados ao terra, eliminando a necessidade de uma fonte de saída flutuante. No ponto de soma S , as correntes que chegam e saem devem satisfazer:

$$-i_{\text{ref}} = i_1 + i_2 + i_3 \quad (24)$$

e, como S constitui um terra virtual, pois os pontos se encontram no potencial "zero", vale:

$$-e_{\text{ref}} = e_1 \left(\frac{R_{\text{ref}}}{R_1} \right) + e_2 \left(\frac{R_{\text{ref}}}{R_2} \right) + e_3 \left(\frac{R_{\text{ref}}}{R_3} \right) \quad (25)$$

Portanto, o potencial do eletrodo de trabalho, em relação ao eletrodo de referência, é igual à soma ponderada das tensões de entrada. Quando os resistores são iguais, obtém-se:

$$e_{\text{wk}}(\text{vs. ref}) = e_1 + e_2 + e_3 \quad (26)$$

Essa configuração possui a vantagem de permitir a síntese direta de formas de onda complexas — como degraus, pulsos ou rampas — simplesmente combinando sinais de entrada

analógicos. Cada fonte de sinal é referenciada ao terra do circuito, e novos canais podem ser adicionados pela inclusão de resistores adicionais no ponto de soma.

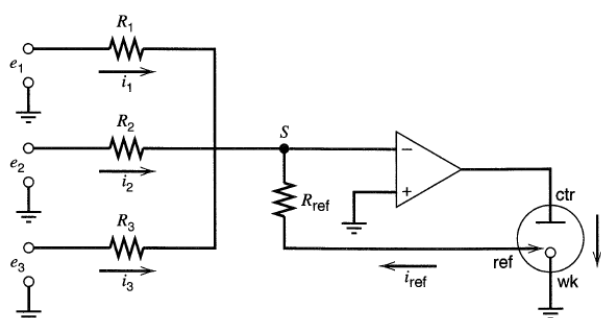


Figura 25: Um potenciostato somador básico

Fonte: (BARD; FAULKNER, 2001)

Apesar de ser funcional, o circuito somador apresenta limitações importantes que restringem sua aplicação prática:

- o eletrodo de referência precisa fornecer corrente significativa (i_{ref}) ao ponto de soma, o que pode causar polarização indesejada e comprometer a estabilidade do potencial de referência;
- o circuito não possui uma forma direta de medir a corrente que flui pela célula, limitando sua utilidade em experimentos que requerem monitoramento preciso da corrente;
- a potência entregue à célula é limitada àquela que o amplificador operacional pode fornecer, restringindo a aplicação a células de alta impedância ou correntes reduzidas.

Para superar essas deficiências, um circuito mais elaborado é apresentado na Figura 26. Nesse arranjo, introduz-se um *seguidor de tensão* (F) no laço de realimentação, isolando o eletrodo de referência da corrente de carga do ponto de soma. Essa configuração garante que o eletrodo de referência não seja polarizado, mantendo sua estabilidade e precisão. A saída desse seguidor, e_F , representa continuamente o potencial $-e_{wk}$ (vs. ref) e pode ser utilizada para registro e monitoramento externo. O eletrodo de trabalho, por sua vez, alimenta um *seguidor de corrente*, cuja saída é proporcional à corrente de célula. Esse estágio mantém o eletrodo de trabalho em terra virtual, condição essencial para a estabilidade do sistema e para a medição precisa da corrente (BARD; FAULKNER, 2001).

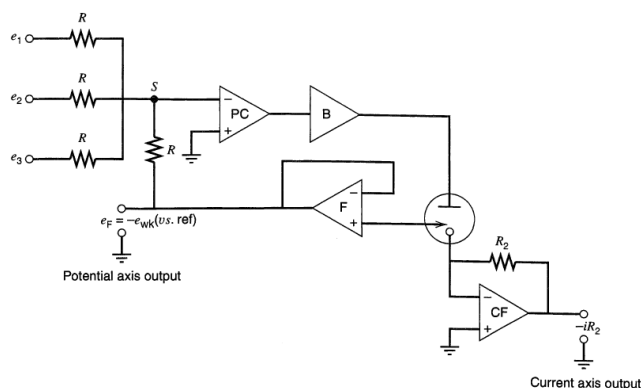


Figura 26: Um sistema potenciostático completo

Fonte: (BARD; FAULKNER, 2001)

Para aumentar a capacidade de corrente do circuito, é comum incluir um *amplificador auxiliar (booster amplifier)* no laço de saída. Esse estágio atua como um amplificador não inversor de baixo ganho, mas com alta capacidade de corrente ou tensão, de modo a fornecer potência suficiente à célula sem comprometer o controle de potencial. O amplificador auxiliar (B) permite que o potenciostato opere com células de menor impedância ou que requeiram correntes mais elevadas, expandindo significativamente sua faixa de aplicação. O ganho total do sistema é dado por:

$$A = A_{OA}A_B$$

em que A_{OA} representa o ganho do amplificador operacional principal (PC) e A_B o ganho do amplificador auxiliar (B).

Com o avanço da eletrônica, os potenciostatos modernos incorporaram sistemas digitais para geração de sinais, aquisição de dados e controle de experimentos. O uso de computadores em experimentos eletroquímicos remonta à década de 1960, quando microprocessadores eram aplicados em tarefas de geração de formas de onda e registro de correntes (BARD; FAULKNER, 2001). Contudo, na época, o alto custo e a baixa velocidade dos sistemas dificultavam sua adoção em larga escala em laboratórios, principalmente em laboratórios de ensino.

Atualmente, a combinação de microcontroladores, conversores digital-analógicos e conversores analógico-digitais permite o controle total dos experimentos por software. O computador é responsável por gerar a forma de onda desejada, aplicá-la ao potenciostato e registrar as respostas eletroquímicas com alta precisão temporal.

No processo de geração de sinais, a forma de onda é criada numericamente na memória do sistema e convertida em tensão analógica via DAC. Essa abordagem permite a produção de formas de onda mais precisas e complexas do que aquelas geradas por circuitos analógicos, além de facilitar a programação de sequências complexas de potenciais. Exemplos incluem ondas triangulares para voltametria cíclica, rampas lineares para voltametria linear, pulsos complexos para técnicas de

polarografia de pulso diferencial e formas de onda personalizadas para experimentos específicos.

Durante a aquisição de dados, o sinal analógico proveniente do seguidor de corrente é convertido em forma digital por meio de um ADC. A precisão da conversão depende tanto da resolução do conversor (número de bits) quanto da taxa de amostragem. Por exemplo, um conversor de 8 bits oferece uma precisão de 1 parte em 255 (aproximadamente 0,4% da escala completa), enquanto um de 12 bits atinge 1 parte em 4095 (aproximadamente 0,024% da escala completa), com tempos de conversão típicos da ordem de 1 μ s por ponto. A escolha da resolução e da taxa de amostragem deve ser adequada à velocidade dos processos eletroquímicos em estudo e à faixa dinâmica da corrente medida.

Equipamentos de alta performance, como registradores transitórios e osciloscópios digitais, alcançam resoluções de até 14 bits e taxas de aquisição da ordem de nanossegundos por ponto, permitindo a captura de eventos eletroquímicos extremamente rápidos. Entretanto, quando o sinal analisado ocupa apenas uma fração da faixa do ADC, ocorre uma perda efetiva de resolução — fenômeno conhecido como *efeito de quantização*. Essa limitação pode ser mitigada realizando múltiplas amostragens sucessivas e calculando a média dos valores obtidos.

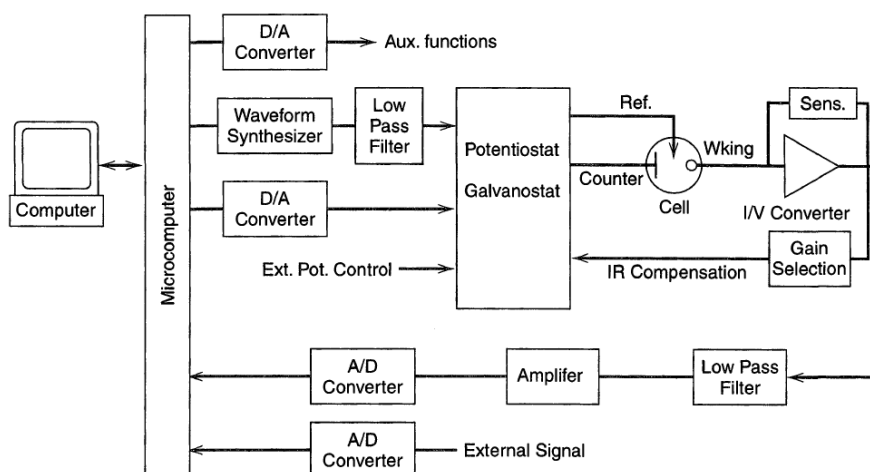


Figura 27: Diagrama de blocos de um instrumento eletroquímico típico controlado por computador

Fonte: (BARD; FAULKNER, 2001)

A Figura 27 apresenta um exemplo moderno de sistema potenciostático computadorizado, no qual o sintetizador de forma de onda e o conversor D/A geram o sinal de varredura desejado, enquanto as funções auxiliares são gerenciadas por software. Essas funções incluem o controle de agitação, desaeração, ou rotação de eletrodos, tornando o sistema altamente automatizado e repetível (BARD; FAULKNER, 2001).

Apesar de sua praticidade, é essencial que o operador compreenda o funcionamento interno do instrumento. A filtragem eletrônica, o tempo de resposta e os métodos de processamento digital aplicados aos dados (como médias móveis ou filtragem algorítmica) podem introduzir distor-

ções nos resultados, especialmente em experimentos de alta velocidade ou que requerem alta precisão temporal. Por isso, recomenda-se a calibração periódica do equipamento utilizando resistores padrão ou células fictícias (*dummy cells*), a fim de garantir a precisão das medições e verificar a resposta dinâmica do sistema (BARD; FAULKNER, 2001).

Por fim, o avanço das placas de aquisição de dados (DAQ) e de linguagens gráficas, como o *LabVIEW*, possibilita a construção de instrumentos eletroquímicos personalizados, controlados diretamente por computador. Essas plataformas permitem que pesquisadores desenvolvam interfaces próprias para geração e leitura de sinais, tornando os potenciostatos mais acessíveis e adequados a aplicações didáticas e de pesquisa.

3.3.2 Topologias

O potenciostato é o instrumento central na eletroquímica, projetado para controlar o potencial elétrico entre o eletrodo de trabalho e o eletrodo de referência, fornecendo a corrente necessária através do eletrodo auxiliar, conforme apresentado anteriormente. A configuração do circuito, especificamente qual dos três eletrodos é mantido no potencial de referência do sistema (terra), define a topologia do potenciostato.

Existem três configurações básicas para um potenciostato, como demonstrado por (Busoni; Carlà; Lanzi, 2002) e ilustrado na Figura 28.

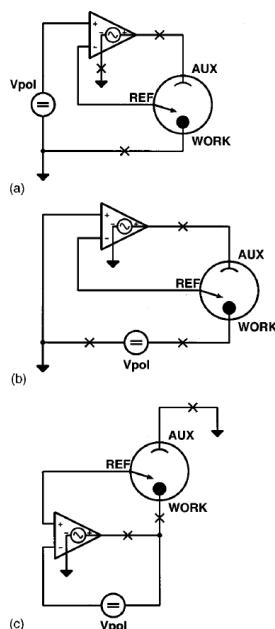


Figura 28: As três configurações básicas de polarização: (a) Eletrodo de Trabalho Aterrado (GW); (b) Eletrodo de Referência Aterrado (GR); (c) Eletrodo Auxiliar Aterrado (GA)

Fonte: (Busoni; Carlà; Lanzi, 2002)

- **Trabalho Aterrado (GW - Grounded Work):** Esta é a configuração mais popular e frequentemente utilizada (Figura 28(a)). Nela, o eletrodo de trabalho é mantido diretamente no potencial

de terra.

- **Referência Aterrado (GR - Grounded Reference):** Esta configuração (Figura 28(b)) é menos comum, sendo usada principalmente em sistemas multieletrodos.
- **Auxiliar Aterrado (GA - Grounded Auxiliary):** A topologia com o auxiliar aterrado (Figura 28(c)) recebeu historicamente pouca atenção, mas, possui vantagens para medição de impedância.

Além da topologia de aterramento, um ponto importante do projeto é *onde* medir a corrente que flui pela célula. A corrente pode ser medida no caminho do eletrodo de trabalho (pontos de medição sinalizados com "x" na figura 28) ou no caminho do eletrodo auxiliar. Esta escolha pode impactar a precisão das medições, pois introduz a impedância do dispositivo de medição (geralmente um conversor corrente-tensão, (CVC)) em série com o eletrodo escolhido. A impedância do (CVC), mesmo que baixa em corrente contínua, pode afetar a resposta em frequência do sistema e introduzir erros em medições de impedância ou em experimentos de alta velocidade.

A abordagem mais intuitiva na popular topologia GW (Figura 28a) é inserir o CVC em série com o eletrodo de trabalho, como mostrado na Figura 29a. No entanto, esta abordagem apresenta duas desvantagens graves:

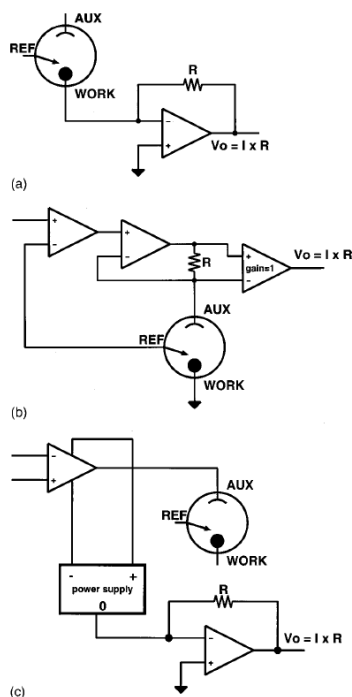


Figura 29: Configurações de medição de corrente: (a) CVC referido ao terra no eletrodo de trabalho; (b) CVC flutuante no eletrodo auxiliar; (c) Medição no retorno da fonte de alimentação

Fonte: (Busoni; Carlà; Lanzi, 2002)

1. **Captação de Ruído:** O eletrodo de trabalho é a interface de medição e, frequentemente, é sensível a interferências eletromagnéticas externas. Se ele não estiver solidamente aterrado (pois o CVC o "levanta" do terra), ele atua como uma antena, captando ruídos que degradam a

medição (Busoni; Carlà; Lanzi, 2002).

2. **Erro de Impedância:** O CVC adiciona uma impedância de entrada em série com o eletrodo de trabalho. Embora essa impedância seja baixa em corrente contínua, ela aumenta rapidamente com a frequência. Pior ainda, o CVC se comporta como uma reatância indutiva em frequências mais altas, que se soma em série com a capacitância interfacial que se deseja medir. Isso pode introduzir erros significativos, especialmente em medições de capacidade ou Espectroscopia de Impedância Eletroquímica (Busoni; Carlà; Lanzi, 2002).

Para contornar isso na topologia GW, seria necessário medir no lado do auxiliar, mas isso exige circuitos complexos, como um "amperímetro flutuante" (Figura 29b) ou medição no retorno da fonte (Figura 29c).

A topologia GA (Auxiliar Aterrado), ilustrada em detalhe na Figura 30, resolve ambos os problemas de forma elegante.

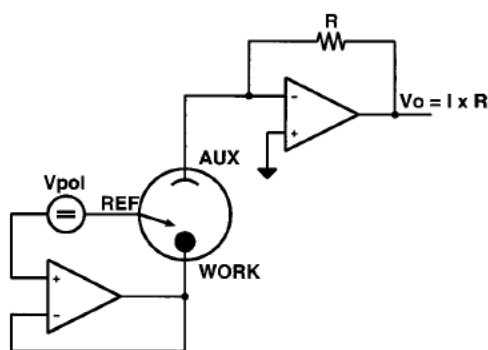


Figura 30: Medição de corrente ideal usando a configuração GA (Auxiliar Aterrado)

Fonte: (Busoni; Carlà; Lanzi, 2002)

Nesta configuração:

1. **Proteção contra Ruído:** O eletrodo de trabalho (WORK) é conectado diretamente à saída de um seguidor de tensão (voltage follower). Este amplificador operacional fornece um caminho de impedância muito baixa para o terra, "encurtando" efetivamente qualquer interferência eletromagnética captada pelo eletrodo.
2. **Medição de Corrente Precisa:** O CVC é implementado no lado do eletrodo auxiliar (AUX). Como o auxiliar já está aterrado por definição nesta topologia, um CVC simples, referido ao terra, pode ser usado.

A vantagem é que qualquer impedância do CVC está agora em série com o eletrodo auxiliar, e não com o eletrodo de trabalho. Essa impedância indesejada fica fora do loop de medição principal (entre WK e RE), não interferindo na medição da impedância interfacial do sistema. Dessa forma, a topologia GA permite medições mais precisas da impedância da célula eletroquímica, especialmente em técnicas como a Espectroscopia de Impedância Eletroquímica, onde a precisão da medição de impedância é importante (Busoni; Carlà; Lanzi, 2002).

4 METODOLOGIA

A Figura 31 apresenta o diagrama de blocos do potenciostato desenvolvido, evidenciando a comunicação entre os módulos de controle, conversão digital-analógica, amplificação e aquisição de dados. O sistema foi projetado para permitir o controle da tensão aplicada à célula eletroquímica e a medição da corrente resultante, viabilizando experimentos de técnicas voltamétricas e potenciostáticas com boa resolução e elevada capacidade de corrente.

O controle central do sistema é realizado por um Arduino Nano, que atua como interface entre o computador e os dispositivos analógicos. O computador, executando um software desenvolvido em Python, é responsável por enviar os parâmetros experimentais e coletar os dados medidos em tempo real via comunicação serial. Essa interface permite ajustar as variáveis do potenciostato, sendo elas:

- Limites de tensão;
- N° de passos para um volt;
- Número de ciclos;
- Velocidade da rampa de tensão em volts por segundo (V/s);
- Número de pontos no gráfico;
- Seleção do tipo de rampa.

Além disso, o software exibe graficamente a relação entre corrente e tensão aplicada à célula eletroquímica durante o experimento.

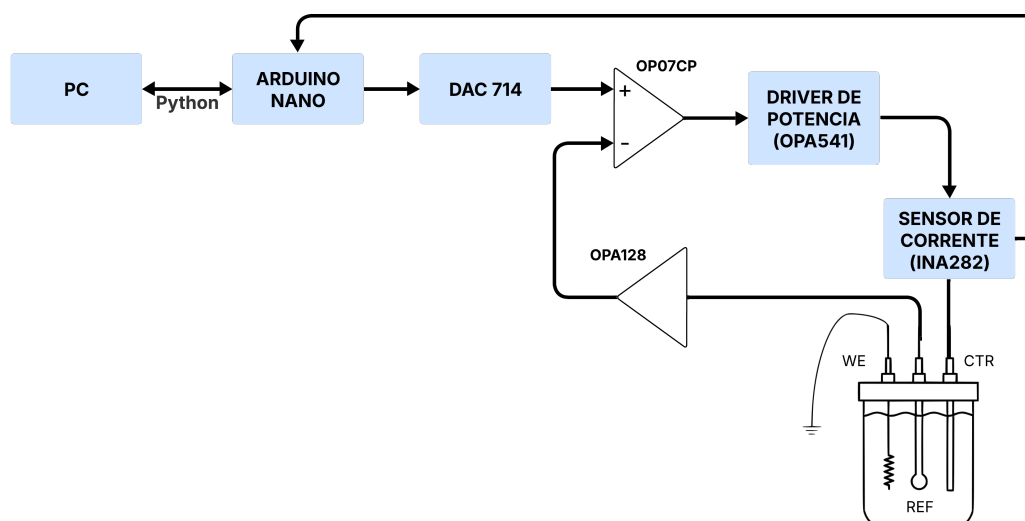


Figura 31: Diagrama de blocos do potenciostato

Fonte: O autor

4.1 ARQUITETURA ELETRÔNICA

O sinal de controle (*set-point*) é gerado digitalmente pelo Arduino e enviado a um Conversor Digital-Analógico (DAC) do modelo DAC714, responsável por converter o sinal digital em uma tensão analógica. Essa tensão representa o potencial de referência a ser aplicado à célula eletroquímica. O DAC714 apresenta resolução de 16 bits, de $-10V$ à $+10V$ com $0,305\text{ mV}/\text{passo}$ ou $305\text{ }\mu\text{V}/\text{passo}$, o que permite um controle fino da tensão aplicada, garantindo alta linearidade e estabilidade na saída.

A tensão de saída do DAC714 é aplicada à entrada não inversora de um amplificador operacional de precisão que atua como driver de potência, implementado com o OPA541. Esse estágio tem como função fornecer a corrente necessária para polarizar a célula eletroquímica, mantendo o potencial de trabalho definido mesmo sob cargas variáveis. O amplificador opera em configuração de seguidor de tensão, garantindo que a tensão no eletrodo de trabalho acompanhe o valor determinado pelo DAC. O sistema implementado possui proteção por meio da associação de resistores, cujo acionamento é definido pela posição de dois *switches* montados na parte frontal do equipamento. A fonte é capaz de fornecer uma tensão de $\pm 35\text{ V}$ com corrente de até $\pm 5\text{ A}$ e pico de $\pm 8\text{ A}$. Observa-se que o amplificador operacional apresenta uma queda aproximada de 5 V quando fornece cerca de 1 A , o que exige alimentação mínima de 40 V para entregar uma saída de $\pm 30\text{ V}$, considerando margens de segurança para evitar operação próxima ao limite de alimentação. O transformador utilizado fornece $\pm 12\text{ V}$, o que resulta em uma tensão máxima aplicada de aproximadamente $\pm 6\text{ V}$ com essa carga de $\pm 1\text{ A}$.

A corrente que circula pela célula é monitorada por um sensor de corrente INA282, conectado em série com a carga eletroquímica. O INA282 fornece um sinal de tensão proporcional à corrente medida. O sensor foi configurado com duas escalas diferentes: uma utilizando um resistor *shunt* de $5\text{ }\Omega$, resultando em uma escala de $\pm 10\text{ mA}$; e outra utilizando um resistor de $0,05\text{ }\Omega$, resultando em uma escala de $\pm 1\text{ A}$. Foi utilizado um TL431 para gerar um sinal de referência de $+2,5\text{ V}$ para o sistema, evitando variações causadas pela alimentação do Arduino. Também foi implementada uma alimentação de $+15\text{ V}$ no sensor para evitar saturação das medidas. As características operacionais relevantes são:

- Resistor: $5\text{ }\Omega$; Escala: $\pm 10\text{ mA}$; Sensibilidade: $39\text{ }\mu\text{A}$
 - Tensão de saída em 0 A : $\approx 2,5\text{ V}$;
 - Tensão de saída em $+0,10\text{ mA}$: $\approx 5,0\text{ V}$;
 - Tensão de saída em $-0,10\text{ mA}$: $\approx 0,0\text{ V}$.
- Resistor: $0,05\text{ }\Omega$; Escala: $\pm 1\text{ A}$. Sensibilidade: 2 mA
 - Tensão de saída em 0 A : $\approx 2,5\text{ V}$;
 - Tensão de saída em $+1\text{ A}$: $\approx 5,0\text{ V}$;

- Tensão de saída em -1 A: $\approx 0,0$ V.

O sinal analógico do INA282 é enviado ao Arduino Nano para conversão A/D utilizando o ADC nativo de 10 bits. Foi implementado um divisor resistivo na entrada analógica para reduzir a tensão de 0–5 V para 0–1,1 V, aumentando a precisão das medidas, especialmente na menor escala. Esse divisor também permite o uso da referência interna de 1,1 V do Arduino, reduzindo oscilações na leitura. Posteriormente, ocorre a transmissão dos dados para o computador, fechando o ciclo de aquisição.

O eletrodo de referência está conectado a um *buffer* de alta impedância implementado com um OPA128. Esse circuito tem como objetivo permitir a medição da tensão entre o eletrodo de referência e o eletrodo de trabalho sem a circulação de corrente, evitando a degradação do eletrodo ao longo do tempo por mais que a degradação do eletrodo também dependa do meio que ele está colocado.

Por fim, utiliza-se o amplificador operacional OP07CP, um amp-op de alta precisão, conectado em modo de retroalimentação com a função principal de fechar o circuito de controle, garantindo que o potencial enviado pelo DAC seja efetivamente aplicado à célula eletroquímica. Esse amplificador considera a tensão medida entre o eletrodo de referência e o eletrodo de trabalho (ligado ao GND do circuito), ajustando a saída de acordo com o valor real aplicado.

Os circuitos esquemáticos completos e detalhados encontram-se nos anexos deste documento.

4.2 SOFTWARE DE CONTROLE E INTERFACE

O software desenvolvido em Python entrega:

- Interface gráfica para configuração de parâmetros (limites, passos, velocidade, número de pontos e ciclos);
- Implementação da rampa de potencial em diferentes formatos (constante, triangular);
- Comunicação serial com checagem de integridade (handshake e ACK);
- Plotagem em tempo real (I vs. E) e salvamento dos dados em formato CSV.

A Figura 32 apresenta o conjunto das telas auxiliares da interface: seleção da rampa, configuração de passos, número de pontos e ajuste da velocidade da rampa.

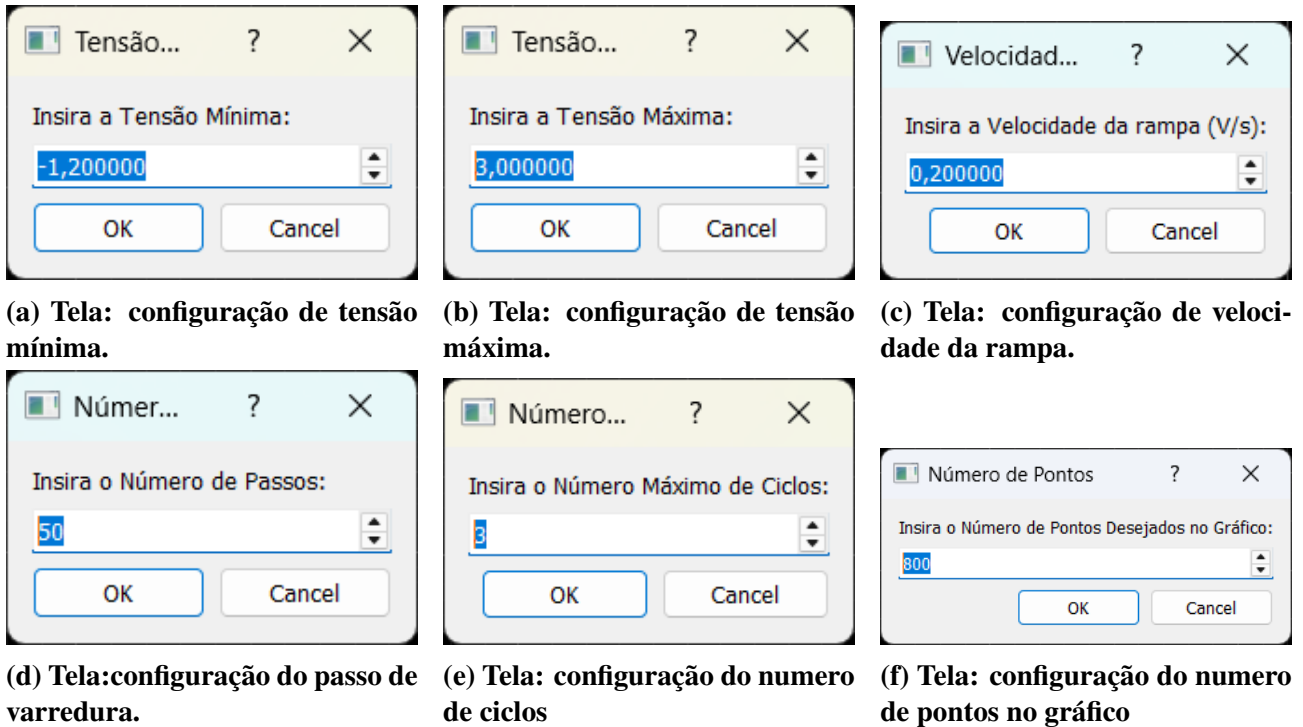


Figura 32: Todos os parâmetros de configuração do potenciostato

Fonte: O autor

A Figura 33 apresenta a interface principal do software de controle, onde podem ser observadas as funcionalidades descritas.

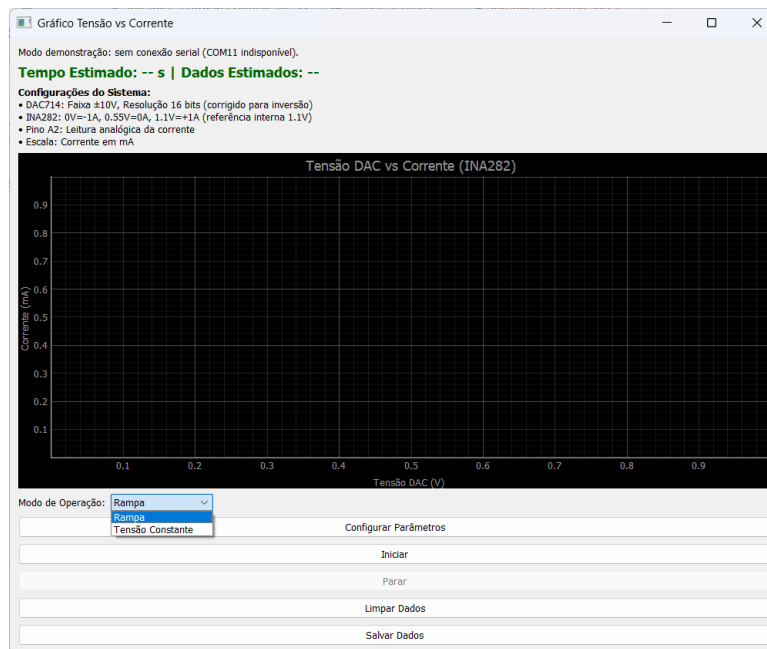


Figura 33: Interface de controle desenvolvida em Python (tela principal)

Fonte: O autor

A arquitetura do sistema foi organizada em dois fluxos principais:

- A interface de controle escrita em Python, representada pelo diagrama de blocos do sistema de controle e aquisição de dados mostrado na Figura 34;
- O firmware do Arduino, responsável pela lógica de geração de sinais do DAC, representado pelo diagrama de blocos da Figura 35

A interface gráfica inicia no estado “INÍCIO”, tenta estabelecer comunicação serial e, em caso de sucesso, passa ao estado “AGUARDANDO”. Nesse estado, o usuário pode configurar parâmetros, iniciar ou parar medições, limpar o gráfico e salvar dados. Quando o usuário solicita uma medição e o Arduino está conectado, a aplicação limpa o histórico, envia o comando de início e passa ao estado “MEDINDO”, no qual executa um ciclo periódico de leitura, processamento, armazenamento e atualização gráfica. O fluxo retorna ao estado de espera tanto por ação do usuário (botão *Parar*) quanto pelo encerramento automático da rampa (mensagem “ACK:DONE”).

No firmware do Arduino, após a fase de inicialização e entrada em “PRONTO”, o microcontrolador aguarda comandos textuais enviados pela interface. Ao receber “MODE:RAMP”, armazena os parâmetros da varredura; ao receber “MODE:CONSTANT”, guarda a tensão fixa de referência; e ao receber “START”, executa o fluxo correspondente ao modo selecionado. No modo rampa, os estados “SUBINDO” e “DESCENDO” regulam o aumento ou diminuição da tensão. Quando o limite superior é alcançado, inicia-se a etapa de descida; ao chegar ao limite inferior, o sistema incrementa o contador de ciclos e decide se inicia novo ciclo ou entra no estado “VOLTANDO A ZERO”. O retorno a zero é sempre gradual, prevenindo estresses no circuito. No modo de tensão fixa, o estado “APLICANDO” mantém a tensão configurada até que o comando “RESET” seja recebido. Durante todo o processo, o Arduino envia periodicamente dados de telemetria (tempo, tensão e corrente).

Os cálculos da rampa são derivados diretamente das grandezas fornecidas pelo usuário. Dado um intervalo de tensão $[V_{\min}, V_{\max}]$, uma velocidade de varredura v (V/s) e um número de passos por segundo N_p , o incremento de tensão por passo é dado por:

$$\Delta V = \frac{v}{N_p}, \quad \Delta t = \frac{1}{N_p}.$$

O tempo de subida é:

$$T_{\text{subida}} = \frac{V_{\max} - V_{\min}}{v},$$

o que implica um ciclo completo de duração $2T_{\text{subida}}$. A duração total da varredura é:

$$T_{\text{total}} = 2T_{\text{subida}} \times N_{\text{ciclos}}.$$

A interface ajusta o intervalo mínimo de transmissão para produzir aproximadamente o número desejado de pontos:

$$\Delta t_{\text{TX}} = \max \left(10 \text{ ms}, \frac{T_{\text{total}}}{N_{\text{pontos}}} \right).$$

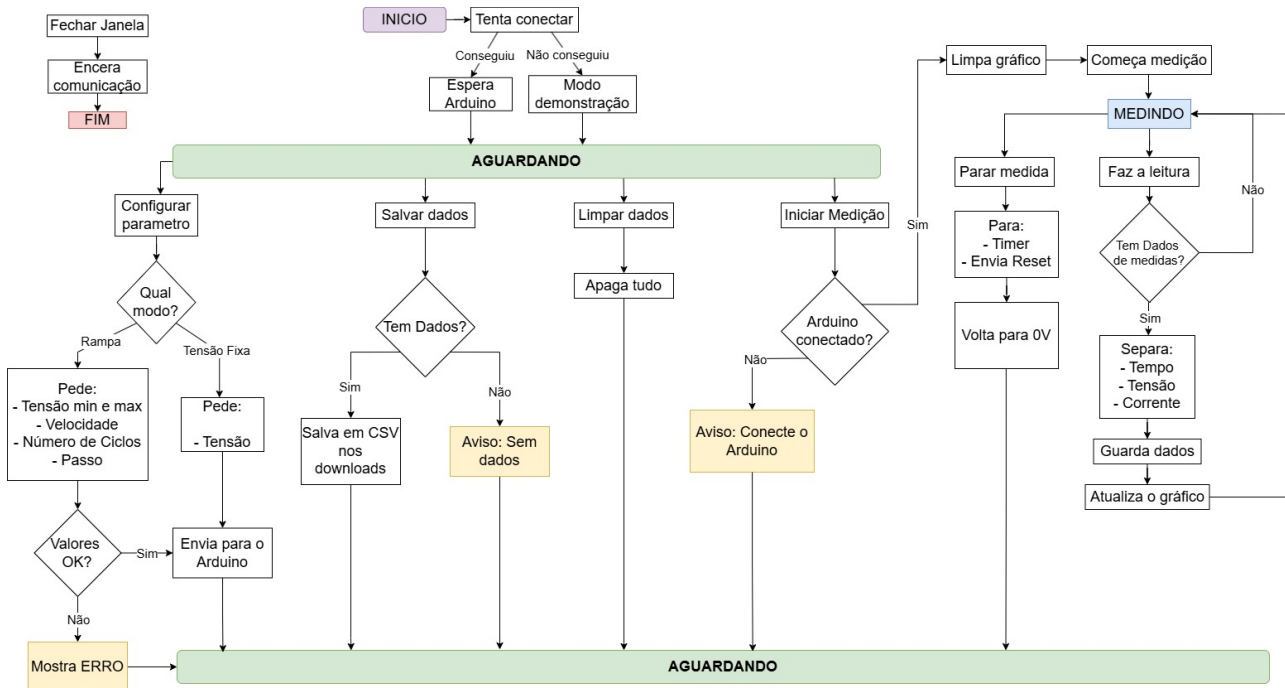


Figura 34: Diagrama de blocos da interface de controle e aquisição de dados

Fonte: O autor

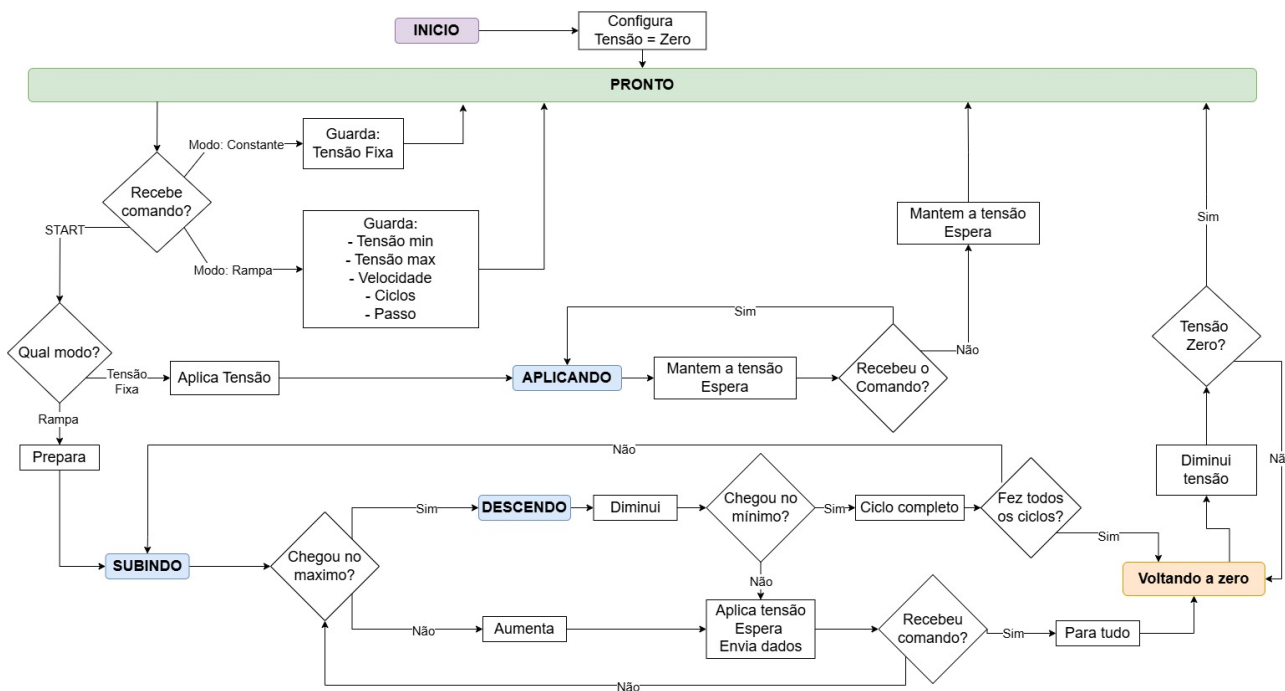


Figura 35: Diagrama de blocos da lógica de geração de sinais do DAC

Fonte: O autor

O código em python e o código do Arduino se encontra nos anexos deste documento.

4.3 ELETRODO DE REFERÊNCIA E CONJUNTO ELETROQUÍMICO

O eletrodo de referência de hidrogênio desenvolvido para este trabalho está apresentado esquematicamente na Figura 36 e na figura 37. A construção seguiu os princípios de vedação e junção descritos na fundamentação teórica, priorizando estabilidade do potencial e segurança no manuseio. A montagem prática incluiu:

- Corpo em material inerte;
- Junção do tipo capilar de Luggin adequada à solução utilizada;

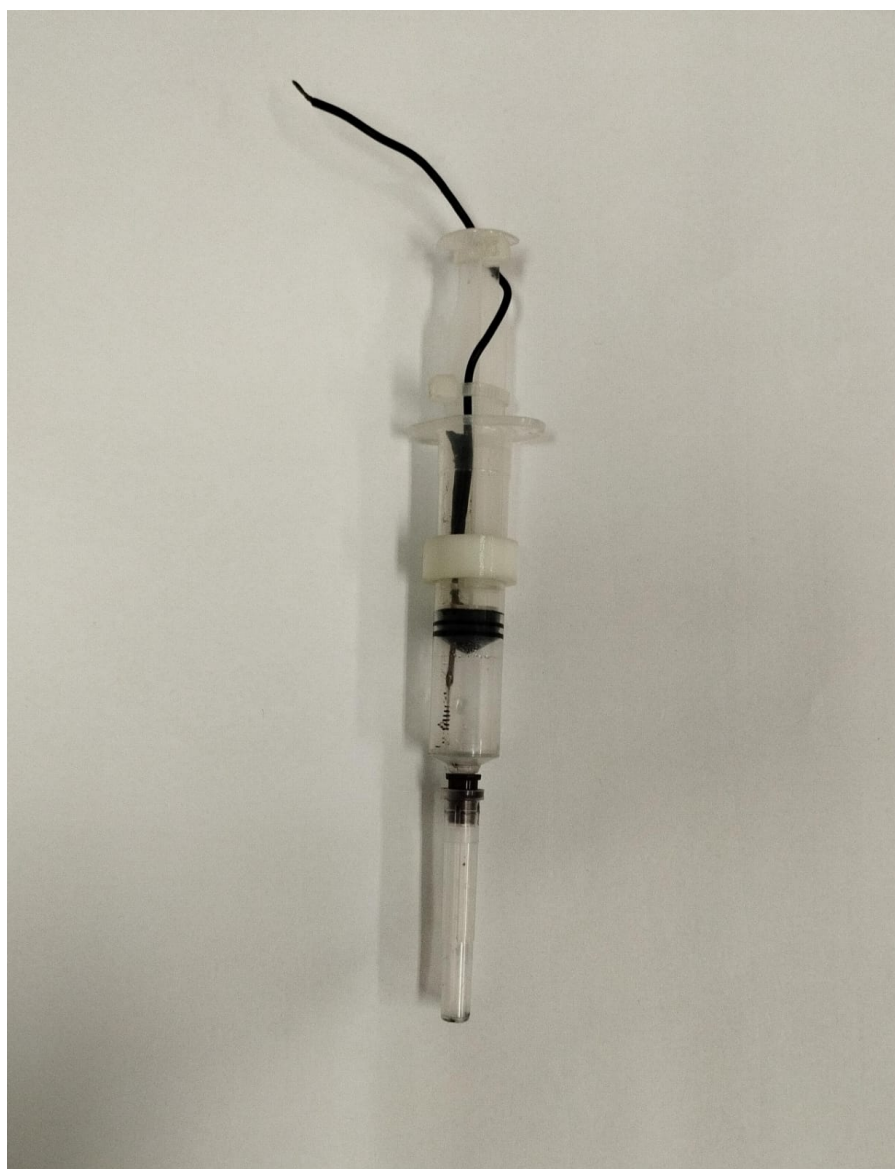


Figura 36: Eletrodo de referência de hidrogênio desenvolvido

Fonte: O autor

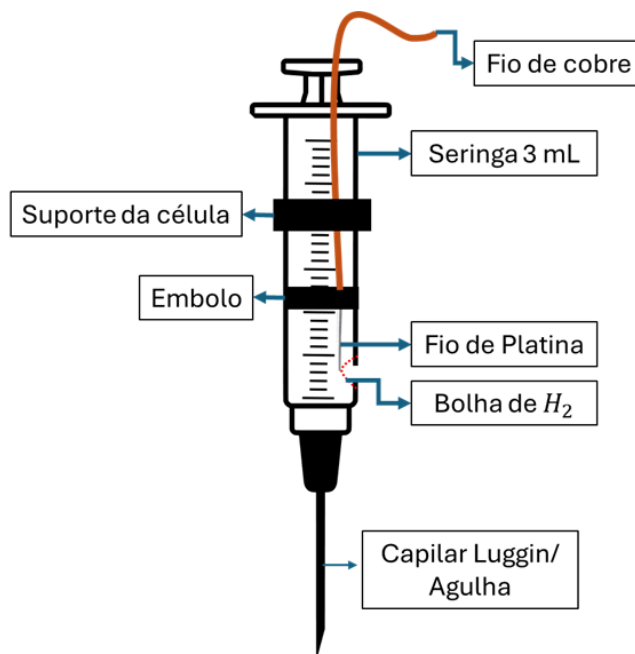


Figura 37: Esquema do eletrodo de referência de hidrogênio desenvolvido

Fonte: O autor

A conexão é realizada por meio de fio tipo jacaré-banana até o *buffer* de alta impedância.

A calibração foi realizada em duas etapas:

- **Calibração de tensão (DAC → saída):** valores conhecidos foram aplicados ao DAC e medidos com multímetro e/ou osciloscópio para verificar linearidade e ganho. Ajustes de *offset* e ganho foram feitos via hardware quando necessário.
- **Calibração de corrente (INA282):** utilizando fontes de corrente e resistores de precisão e amperímetro externo, foram obtidos pontos de referência para 0 A, $\pm 0,1$ A, $\pm 0,5$ A e $\pm 1,0$ A, ajustando a equação digital → corrente no software de aquisição. A calibração incluiu a verificação da repetibilidade das medidas.

4.4 VALIDAÇÃO E ANÁLISE DE DESEMPENHO

Para validar o potenciostato, foram avaliados os seguintes parâmetros:

- Tempo de resposta do sistema (*settling time*) em transições de degraus de tensão;
- Faixa dinâmica e resolução efetiva da medição de corrente;
- Comparação de voltamogramas obtidos com *dummy cell* e potenciostato comercial (qualitativa e quantitativa);
- Tempo de vida útil do eletrodo de referência.

5 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados e discutidos os resultados obtidos durante a caracterização elétrica e a validação funcional do potenciostato desenvolvido. A análise divide-se em três etapas principais: a caracterização dos subsistemas de tensão e corrente, a análise da resposta dinâmica do circuito de controle e, por fim, a validação eletroquímica comparativa utilizando uma *dummy cell*, bem como a análise de estabilidade do eletrodo de referência de hidrogênio construído. Também são apresentados a lista completa dos materiais utilizados e demais informações relevantes ao desenvolvimento do sistema.

5.1 LISTA DE COMPONENTES

Os componentes utilizados na montagem do potenciostato encontram-se relacionados na Tabela 1, juntamente com sua cotação no mês de janeiro de 2026 e, quando pertinente, seus respectivos *datasheets*.

Tabela 1: Lista de componentes e preço

Quantidade	Componente	Loja	Preço por unidade (R\$)	Datasheet
1	INA282	AliExpress	45,00	Clique aqui
1	Arduino Nano	Mercado Livre	34,00	Clique aqui
1	OPA541AP	AliExpress	75,00	Clique aqui
1	OP07	Shopee	15,00	Clique aqui
1	Potenciômetro 50 k Ω	Saravati	2,00	-
8	Capacitor 0,1 μ F	Shopee	15,00	-
2	Capacitor 1 μ F	Shopee	13,00	-
1	OPA128LM	AliExpress	64,00	Clique aqui
1	DAC714	AliExpress	29,00	Clique aqui
2	Capacitor 10000 μ F	Mercado Livre	32,00	-
1	Ponte retificadora	Mercado Livre	19,00	-
1	LED	Mercado Livre	1,00	-
1	LT1028	AliExpress	10,00	Clique aqui
1	TL431	Mercado Livre	2,50	Clique aqui
9	Resistores (0,115 Ω , 1 k Ω , 2 k Ω , 10 k Ω , 36 k Ω)	Mercado Livre	34,00	-
1	Placa fenolite perfurada	Eletrogate	7,50	-
Total Geral			R\$ 430,00	

O amplificador operacional OP07 pode ser substituído pelo LT1028, visando à melhoria do tempo de resposta do sistema. O circuito original se encontra nos anexos. A versão

com a alteração do amplificador, se encontra na adição de um capacitor de $0,1 \mu F$ entre a saída e a entrada inversora do componente.

5.2 CONSTRUÇÃO DO POTENCIOSTATO

A seguir, são apresentadas fotografias do equipamento em sua configuração final.

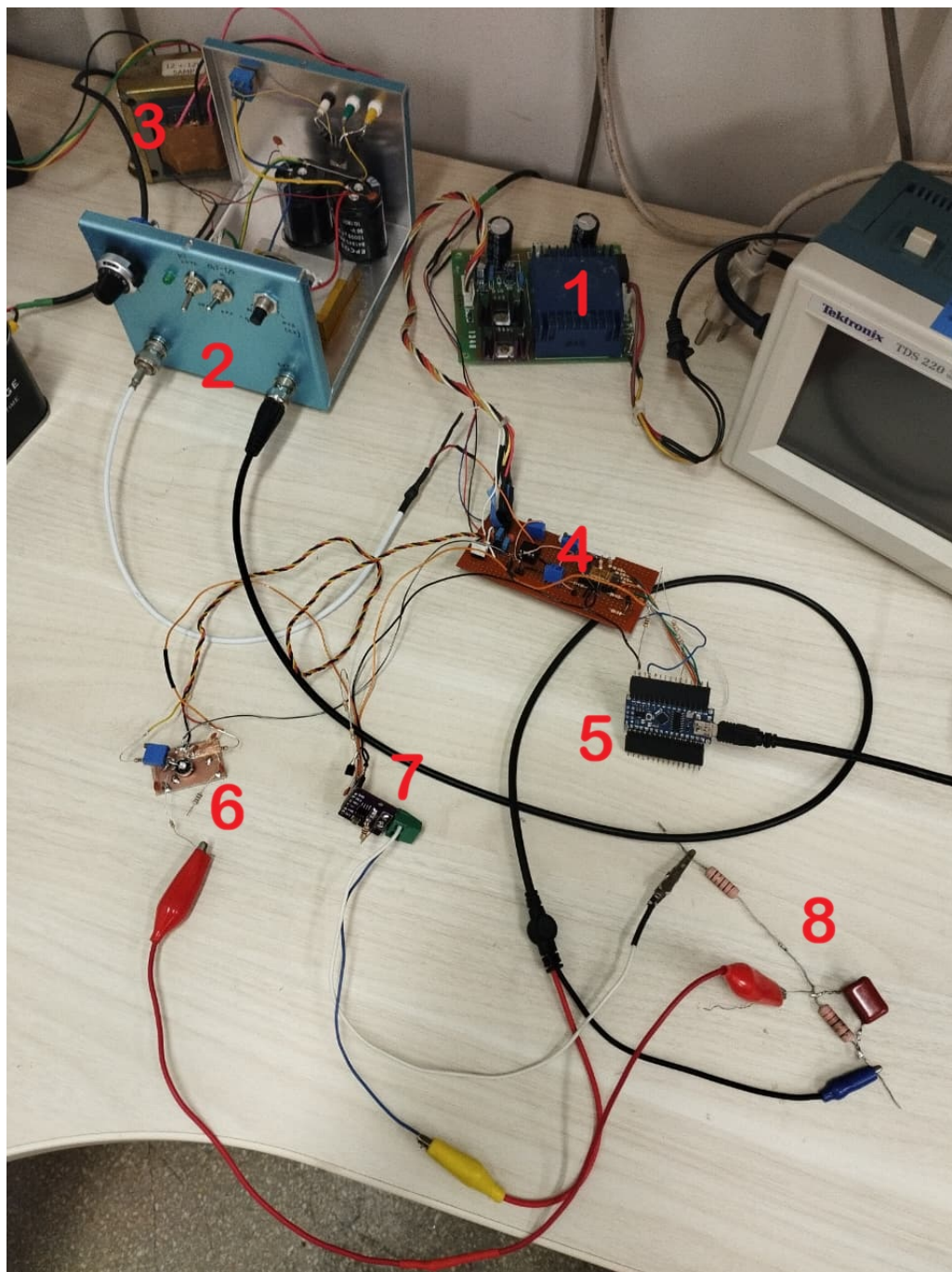


Figura 38: Potenciostato montado para teste de *dummy cell*. 1- Fonte de alimentação simétrica de $\pm 24 V$. 2- Buffer de potência. 3- Transformador $\pm 12 V$, com $3 A$. 4- Regulador de tensão de $\pm 15 V$, DAC714 e amplificador operacional (OP07). 5- Arduino Nano. 6- Buffer de alta impedância. 7- Sensor de corrente INA282 modificado. 8- Dummy cell

Fonte: O autor

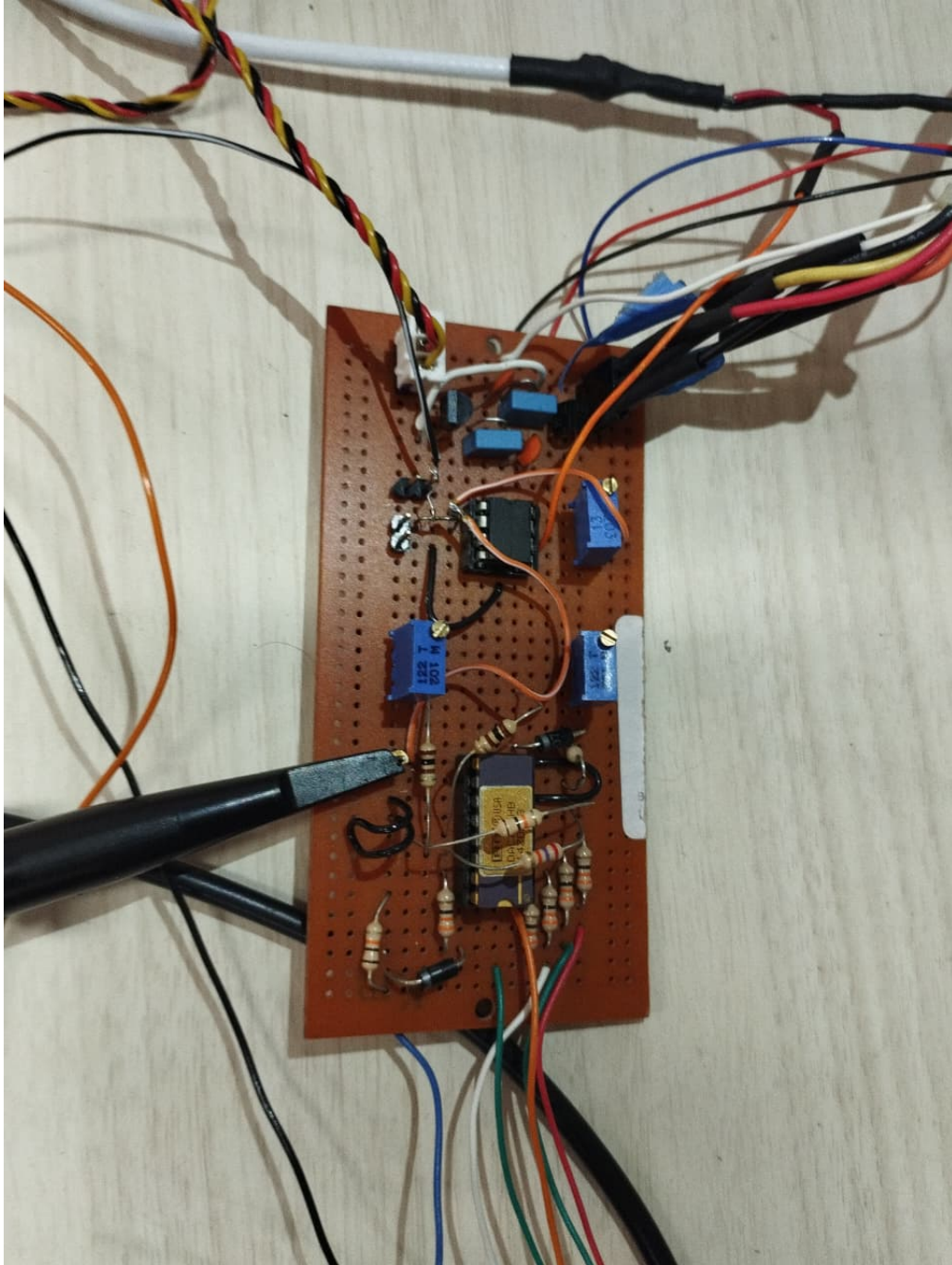


Figura 39: Regulador de tensão de ± 15 V, DAC714 e amplificador operacional (OP07)

Fonte: O autor



Figura 40: Buffer de potência em caixa de alumínio contendo o OPA541, com ganho ajustável entre 1 e 11, configurado para ganho unitário. O conjunto inclui chaves e potenciômetro para ajuste do limite de corrente em até 1 A. A caixa também abriga o estágio de alimentação, composto por um retificador de onda completa (limitado a 3 A) e capacitor de filtro de 10 000 μF . O transformador é conectado externamente à caixa

Fonte: O autor

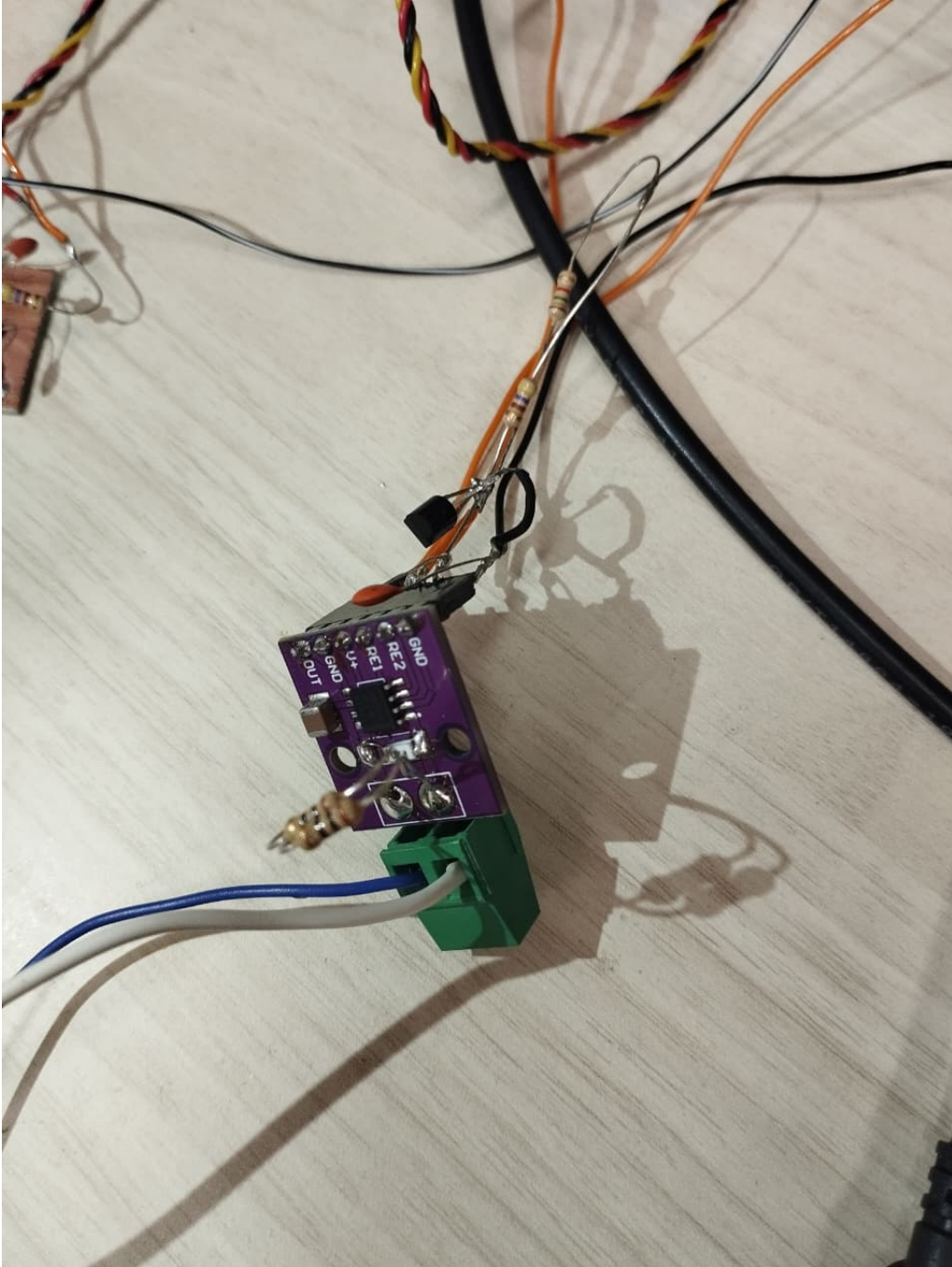


Figura 41: Sensor de corrente (INA282) com duas escalas de corrente

Fonte: O autor

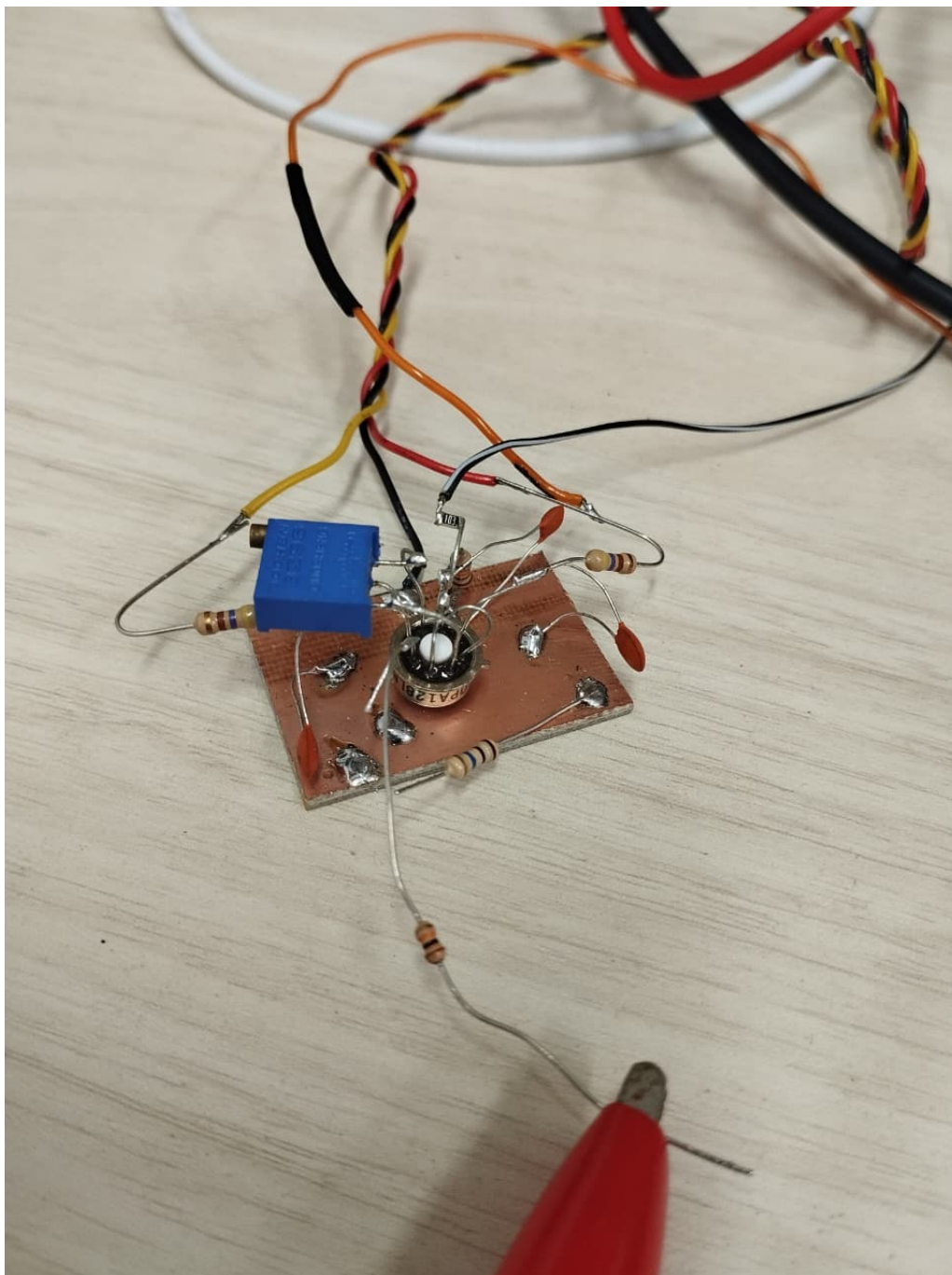


Figura 42: Buffer de alta impedância (OPA128LM) e amplificador do eletrodo de referência

Fonte: O autor

5.3 CARACTERIZAÇÃO ELÉTRICA

A primeira etapa de validação consistiu na verificação da linearidade e da acurácia do Conversor Digital-Analógico (DAC714) e do estágio de potência. O sistema foi configurado para tensões de controle, enquanto as saídas reais (V_{out}) e as correntes circulantes foram monitoradas com multímetros de precisão (referência) e comparadas com os valores adquiridos pelo sensor INA282 (instrumento desenvolvido), com uma calibração padrão.

A Figura 43 apresenta a configuração da carga utilizada para os testes na escala de alta corrente (± 1 A). A entrada da carga/ célula é representada pelo símbolo VCC. A saída da carga/ célula é representada pelo símbolo GND.

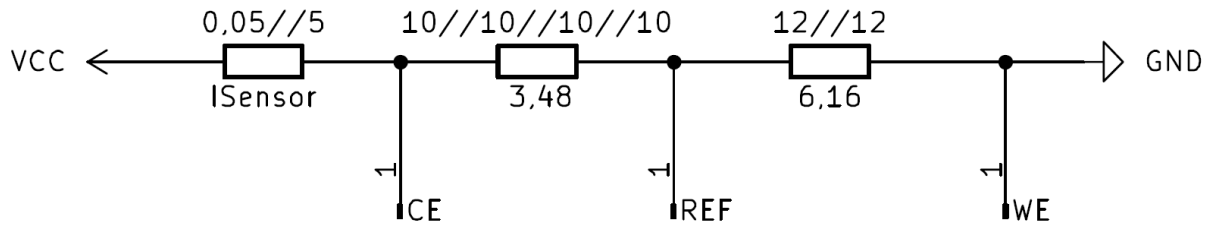


Figura 43: Configuração da *Dummy Cell* utilizada para testes de carga na escala de ± 1 A, valores dos resistores dos ohms

Fonte: O autor

A Tabela 2 detalha os valores medidos e apresenta o erro relativo calculado entre a corrente medida pelo multímetro (I_{ref}) e a corrente lida pelo sensor do protótipo (I_{sensor}). O erro relativo foi calculado conforme a Equação 27:

$$E(\%) = \frac{I_{sensor} - I_{ref}}{I_{ref}} \times 100 \quad (27)$$

Observa-se que o sistema apresenta uma resposta linear consistente. Nota-se, contudo, um erro percentual médio que varia entre 6% e 8% nesta escala. Esse desvio pode ser atribuído à tolerância dos resistores de *shunt* e ao ganho natural do amplificador do INA282, por ser tratar de fontes sistemáticas, ou seja com o tempo não alteram o erro produzido, é possível fazer a correção via software reduzindo a zero. Os erros inferiores a 2% é devido a saturação do sensor de corrente.

Tabela 2: Dados de calibração elétrica e análise de erro na escala de 1 A

V_{ajuste} (V)	V_{out} (V)	V_{ref} (V)	I_{mult} (A)	I_{sensor} (A)	$V_{I,sens}$ (V)	Erro (%)
6,5	10,11	6,400	1,038	0,980	5,217	-5,58
6,0	9,41	5,928	0,961	0,980	5,017	+1,97
5,5	8,50	5,431	0,881	0,943	4,806	+7,03
2,0	3,06	1,968	0,319	0,344	3,338	+7,83
-6,5	-9,93	-6,35	-1,030	-1,020	0,015	-0,97
-6,0	-9,27	-5,917	-0,960	-1,020	0,014	+6,25
-5,5	-8,54	-5,432	-0,881	-0,944	0,199	+7,15

Para aplicações que exigem maior sensibilidade, o sistema foi comutado para a escala de ± 10 mA. A Figura 44 apresenta a célula usada e a Tabela 3 demonstra o comportamento nesta configuração.

Verifica-se que, nesta escala, o erro relativo é significativamente menor (em torno de 2%) para correntes próximas aos fundos de escala. No entanto, para correntes muito próximas de zero, o ruído de fundo, a tensão de *offset* do INA282 e o ganho do mesmo tornam-se percentualmente mais representativos, justificando implementação de rotinas de suavização de dados (médias móveis) nesses casos.

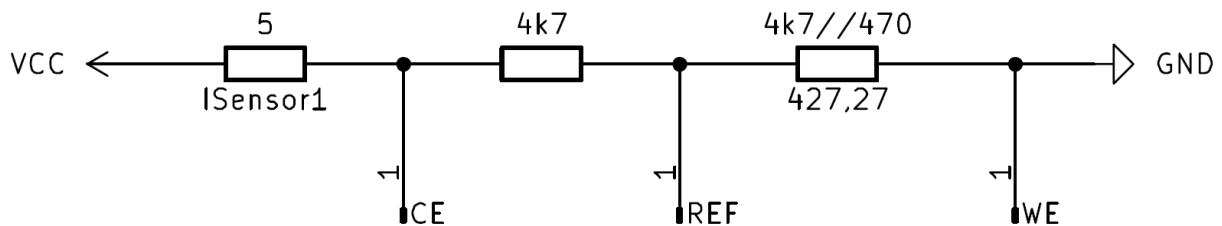


Figura 44: Configuração da *Dummy Cell* utilizada para testes na escala de ± 10 mA, valores dos resistores em ohms

Fonte: O autor

Tabela 3: Dados de calibração elétrica e análise de erro na escala de 20 mA

V_{ajuste} (V)	V_{out} (V)	V_{ref} (V)	I_{mult} (mA)	I_{sensor} (mA)	$V_{I,sens}$ (V)	Erro (%)
10	10,97	10,014	2,24	2,27	3,046	+1,34
5	5,483	5,005	1,15	1,13	2,778	-1,74
0	-0,001	-0,0005	-0,029	-	2,501	-
-5	-5,48	-5,005	-1,12	1,14	2,233	+1,78
-10	-10,85	-9,911	-2,19	2,12	1,970	-3,19

5.4 ANÁLISE DA RESPOSTA DINÂMICA (VELOCIDADE E ESTABILIZAÇÃO)

Para um potenciostato, a precisão não depende apenas do valor da tensão aplicada, mas também da velocidade com que o sistema reage a variações no sinal de controle. Caso a resposta seja lenta, o equipamento não consegue acompanhar reações eletroquímicas rápidas. Dessa forma, foram realizados testes de resposta ao degrau (mudanças abruptas de tensão) com o objetivo de caracterizar o *Slew Rate* (velocidade máxima de variação da tensão) e o *Settling Time* (tempo necessário para a estabilização no valor final).

O teste foi realizado usando o osciloscópio Tektronix TDS220.

O circuito foi projetado em uma configuração inversora com o objetivo de garantir maior estabilidade dinâmica. Conforme observado na Figura 45, quando o sinal de controle proveniente do DAC (Canal 1 – inferior) sofre uma transição descendente, a resposta medida na célula eletroquímica (Canal 2 – superior) apresenta uma transição ascendente.

Observa-se um comportamento característico de um sistema superamortecido, no qual a tensão de saída atinge o valor desejado de forma monotônica, sem apresentar ultrapassagem (*overshoot*) ou oscilações. Em aplicações eletroquímicas, esse comportamento é fundamental, pois oscilações abruptas podem resultar na aplicação de sobretensão, causando degradação da amostra ou dos eletrodos.

A Figura 46 apresenta o sinal de controle proveniente do DAC. Observa-se que a transição ocorre em um intervalo na ordem $1 \mu s$, evidenciando que o atraso observado na resposta global do sistema não é causado pelo computador ou pelo conversor digital-analógico, mas sim pelo estágio de potência responsável por fornecer corrente à carga e pelo amplificador de controle.

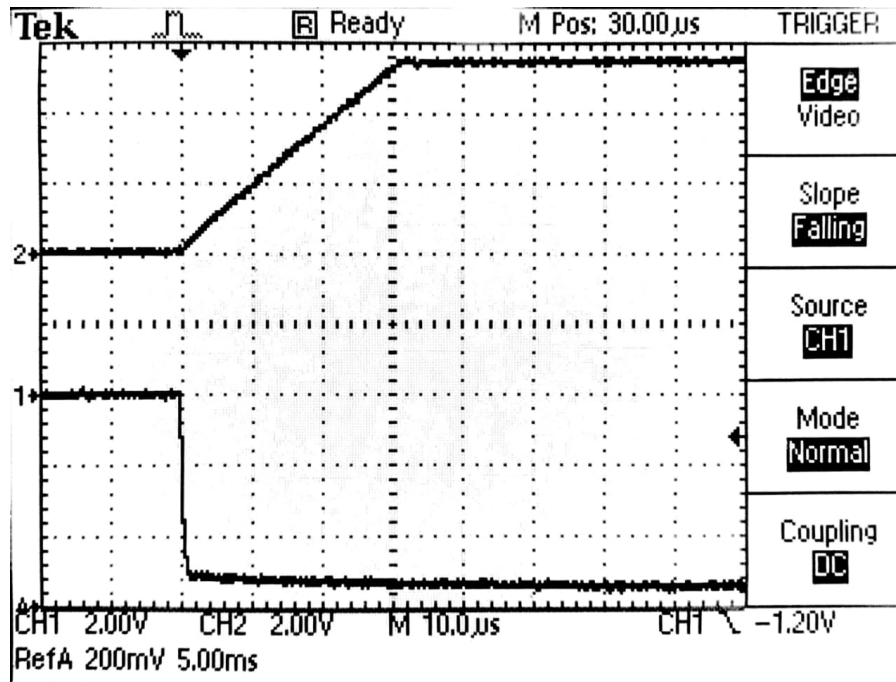


Figura 45: Resposta ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Superior): Tensão de saída estabilizada. CH1 (Inferior): Sinal de controle (descida) enviado pelo DAC

Fonte: O autor

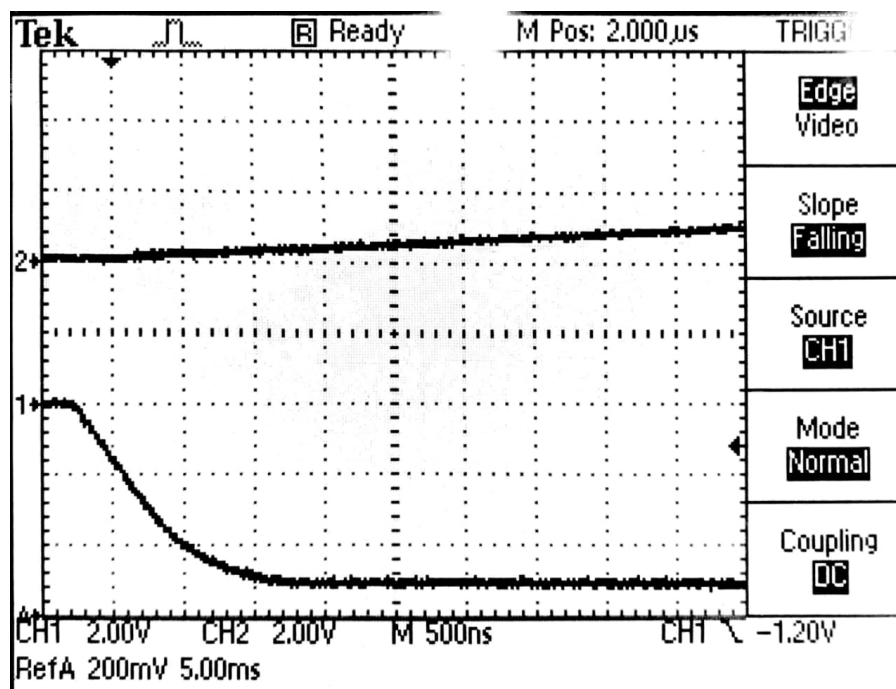


Figura 46: Detalhe da borda de descida do sinal do DAC (CH1), apresentando transição rápida e limpa

Fonte: O autor

A Figura 46 apresenta a resposta em tempo de decida do circuito, e a resposta ao tempo de subida do DAC.

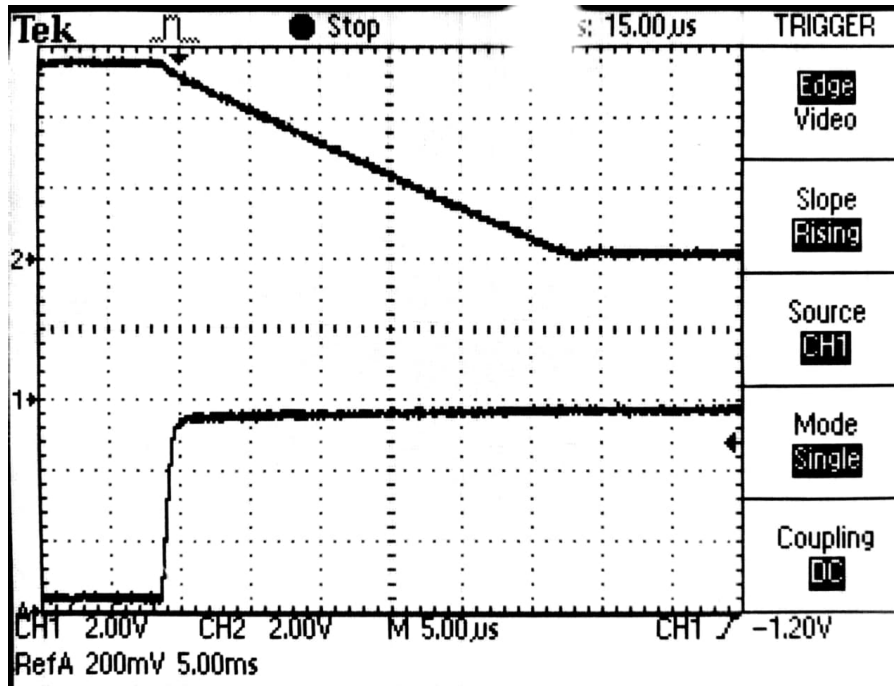


Figura 47: Resposta dinâmica completa: CH2 (Saída) seguindo o comando do CH1 (Entrada) com atraso temporal característico da malha de controle

Fonte: O autor

Conforme ilustrado na Figura 48, ao se aproximar do valor final de tensão, a resposta do sistema deixa de ser linear e passa a apresentar uma curvatura suave até a completa estabilização. Esse comportamento é resultado da atuação dos circuitos de compensação em frequência (filtros e capacitores), que funcionam como mecanismos de amortecimento, reduzindo a velocidade de aproximação do valor final para evitar oscilações e ruídos na célula eletroquímica. Comportamento também observado na escala de 10mA .

Embora o sistema atinja valores próximos ao alvo ($\pm 5\%$) em aproximadamente $25\ \mu\text{s}$, observa-se que são necessários cerca de $50\ \mu\text{s}$ para que a tensão permaneça estabilizada dentro de uma faixa de erro de $\pm 1\%$.

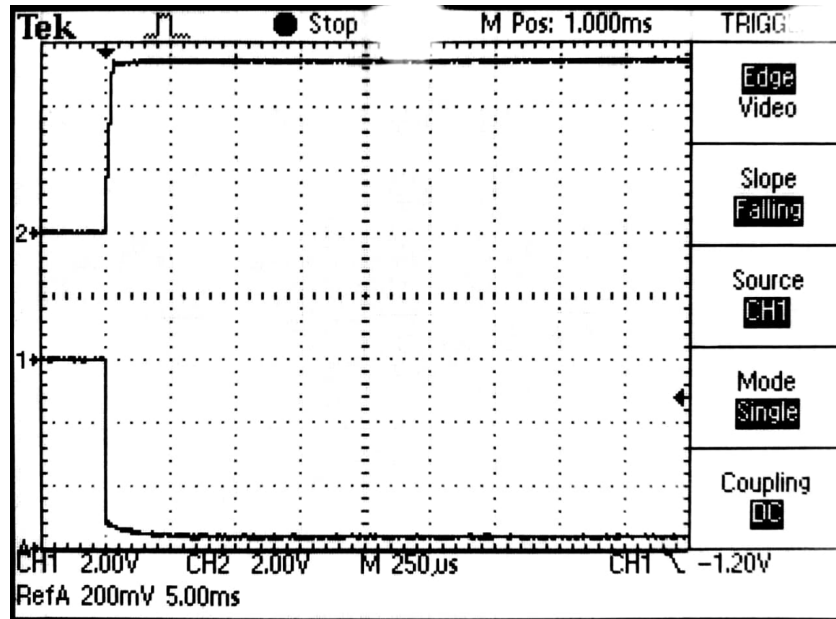


Figura 48: Detalhe do tempo de estabilização. No CH1 observa-se a estabilização final da tensão na carga

Fonte: O autor

Os testes foram repetidos para a escala de baixa corrente ($\pm 10\text{mA}$), conforme apresentado na Figura 49 a 52. Nessa escala, o sistema pode operar com uma tensão de maior valor, o que resulta em uma resposta dinâmica ligeiramente mais lenta.

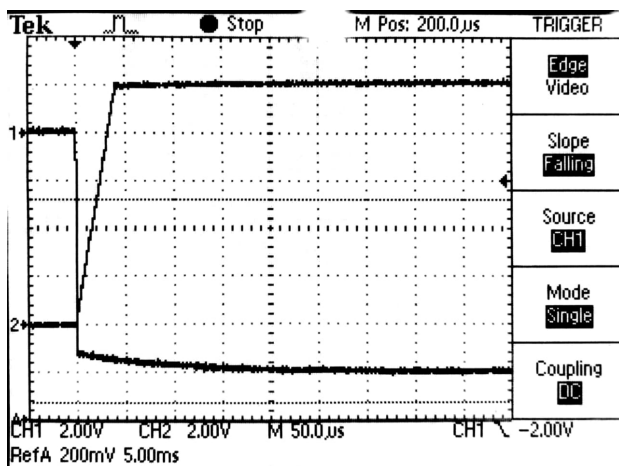


Figura 49: Resposta dinâmica na escala de 10mA. CH2: Resposta do sistema; CH1: Sinal de descida do DAC

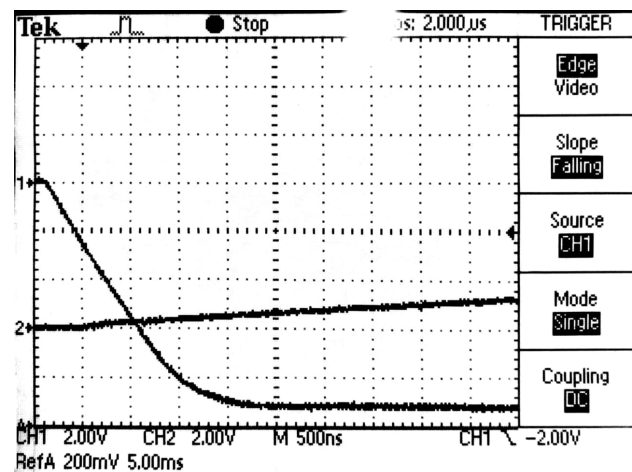


Figura 50: Detalhe da descida do sinal de controle (DAC) na configuração de baixa corrente

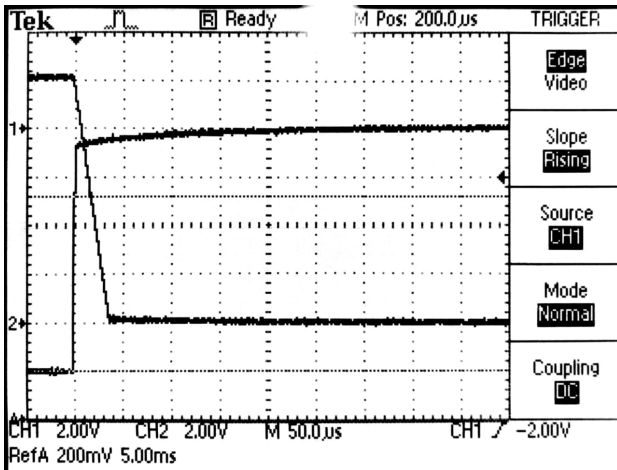


Figura 51: Tempo de subida do DAC na escala de 10mA

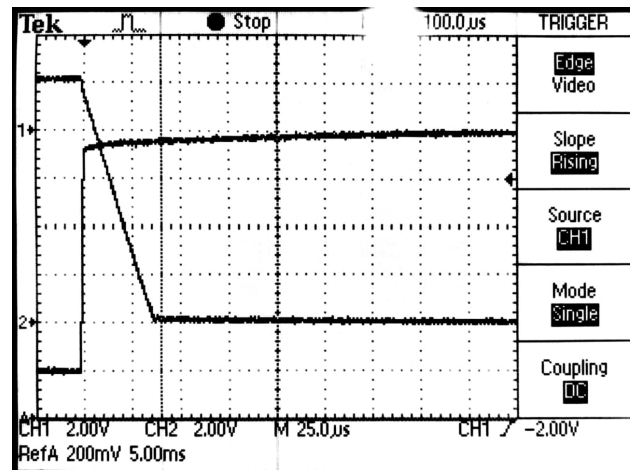


Figura 52: Estabilização final do sinal na escala de 10 mA

Fonte: O autor

As Figuras 53 e 54 apresentam os resultados obtidos após a mudança do hardware. Por meio do ajuste da malha de compensação, foi possível reduzir significativamente o tempo de resposta para aproximadamente $4 \mu s$, mantendo a estabilidade satisfatória do sistema. Essa melhoria amplia o potencial de aplicação do potenciostato em técnicas eletroquímicas de alta velocidade.

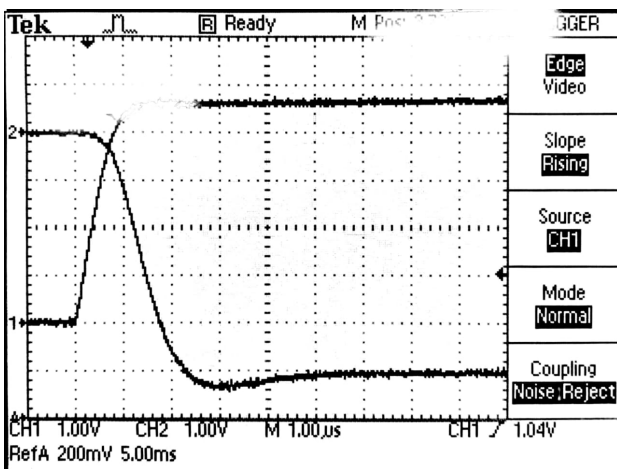


Figura 53: Sistema otimizado: resposta rápida ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Superior): Tensão de saída estabilizada. CH1 (Inferior): Sinal de controle (descida) enviado pelo DAC

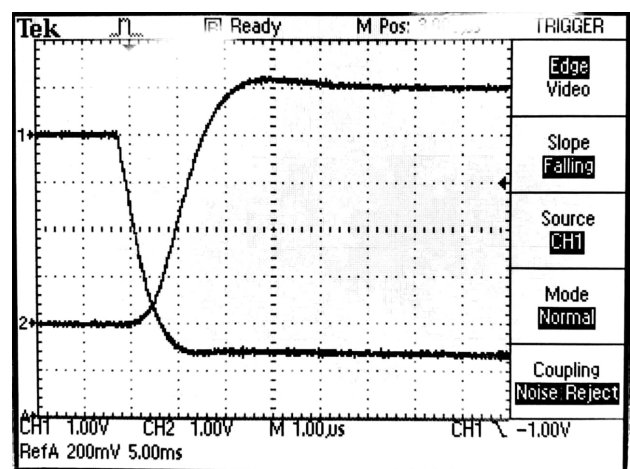


Figura 54: Sistema otimizado: resposta rápida ao degrau do potenciostato (Escala de Alta Corrente). CH2 (Saída) seguindo o comando do CH1 (Entrada) com atraso temporal característico da malha de controle

Fonte: O autor

A Tabela 4 consolida as principais métricas de resposta dinâmica obtidas nos ensaios realizados.

Tabela 4: Métricas de resposta dinâmica do sistema

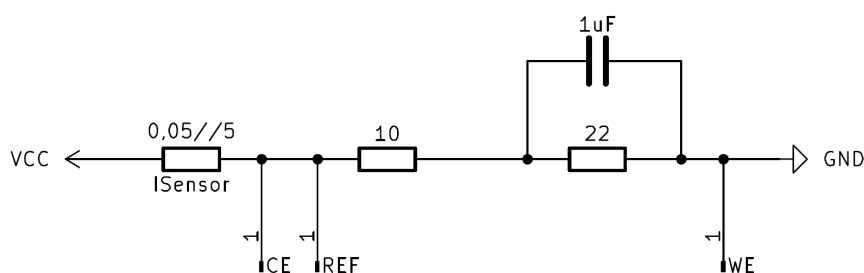
Componente	Escala	Tensão	t_{subida} (10–90%)	$t_{descida}$ (90–10%)	Resposta ($\pm 1\%$)
DAC 714	1 A	$\pm 5,5$ V	750 ns	$1,02 \mu s$	$2 \mu s$
DAC 714	10 mA	± 10 V	$1,20 \mu s$	$1,38 \mu s$	$5 \mu s$
Sistema Original	1 A	$\pm 5,5$ V	$26,2 \mu s$	$24,0 \mu s$	$33 \mu s$
Sistema Original	10 mA	± 10 V	$33,0 \mu s$	$31,0 \mu s$	$50 \mu s$
Sistema Novo	1 A	± 5 V	$1,4 \mu s$	$1,2 \mu s$	$4 \mu s$

Os tempos de resposta observados, variando entre $33 \mu s$ e $50 \mu s$ no sistema original, mostram-se adequados para aplicações eletroquímicas convencionais, permitindo a realização de varreduras de potencial estáveis sem prejuízo à interpretação dos dados experimentais.

5.5 VALIDAÇÃO COMPARATIVA: POTENCIOSTATO COMERCIAL VS. PROTÓTIPO (ESCALA DE ± 1 A)

Uma validação complementar do instrumento foi realizada através de um experimento de Voltametria Linear utilizando uma *Dummy Cell* (circuito resistivo equivalente). O resultado obtido com o protótipo desenvolvido (UNILA) foi comparado diretamente com os dados gerados por um potenciostato comercial de referência Autolab PGSTAT302N (AutoLab).

O cenário de teste (Figura 55) consiste na aplicação de uma diferença de potencial de ± 5 V, com velocidade de varredura de $0,1$ V/s e passo de 5 mV ($1V/200$).

**Figura 55: Esquemático da *Dummy Cell* utilizada no cenário de validação**

Fonte: O autor

A Figura 56 exibe a curva I vs E obtida. Nota-se a presença de ruído no sinal da "Curva Original" do protótipo. Este ruído é inerente à alta sensibilidade do sensor INA282 e à suscetibilidade da montagem experimental à capacitâncias parasitas. A amplitude da oscilação observada é de aproximadamente ± 6 mA, o que corresponde a cerca de 3 unidades de resolução do

ADC na escala de 1 A (onde a resolução mínima é de 2 mA/bit). A aplicação de um filtro de média móvel (Curva Suavizada) mitigou este efeito sem distorcer a resposta do sistema.

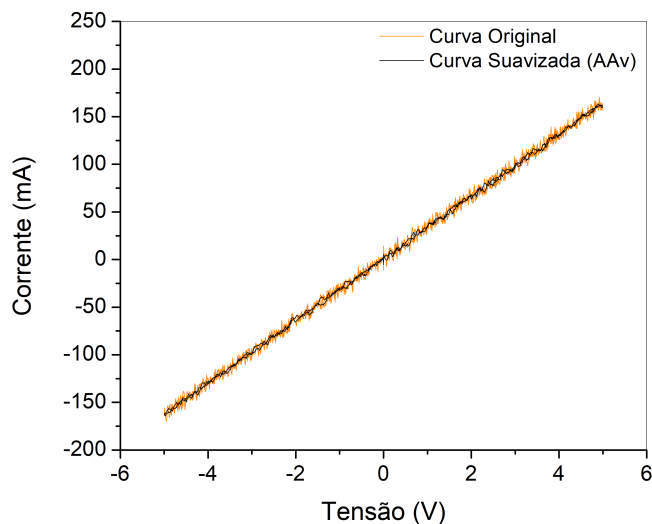


Figura 56: Voltametria cíclica para a *Dummy Cell* do Cenário 1. Linha laranja: dados brutos do protótipo; Linha preta: dados suavizados (médias móveis)

Fonte: O autor

Ao comparar com o equipamento comercial (Figura 57), observa-se uma sobreposição quase perfeita das curvas, validando a precisão do protótipo na aplicação de tensão e leitura de corrente para cargas resistivas lineares.

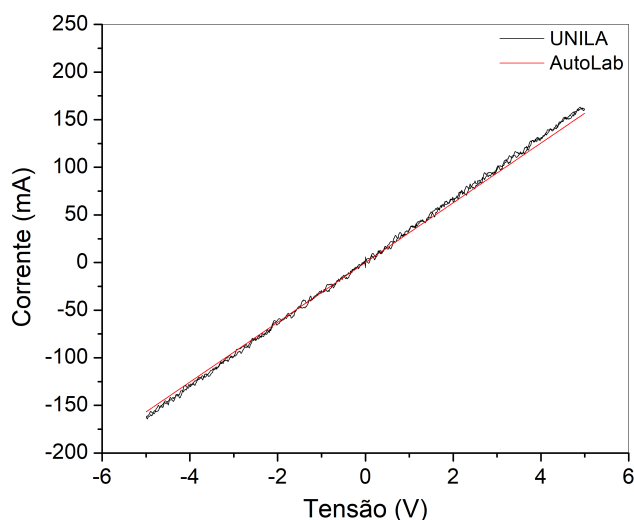


Figura 57: Comparativo de validação (Cenário 1). A linha preta (UNILA corrigida) sobrepõe-se à linha vermelha (AutoLab), indicando alta correlação

Fonte: O autor

5.6 VALIDAÇÃO POTENCIOSTATO: PROTÓTIPO (ESCALA DE $\pm 10 \text{ mA}$)

O cenário de teste (Figura 58), para o teste na menor escala de corrente, consiste na aplicação de uma diferença de potencial de $\pm 4 \text{ V}$, com velocidade de varredura de $0,1 \text{ V/s}$ e passo de $0,5 \text{ mV}$.

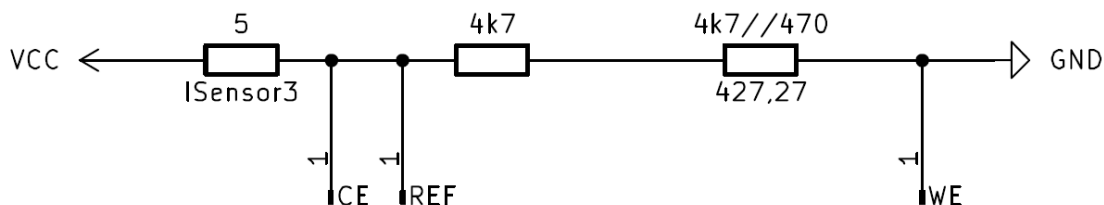


Figura 58: Esquemático da *Dummy Cell* utilizada no cenário de validação

Fonte: O autor

O perfil obtido na voltametria cíclica revela um comportamento estritamente ôhmico, evidenciado pela linearidade da resposta corrente-potencial e pela intersecção precisa na origem do plano cartesiano. A inexistência de histerese ou distorções na reta de varredura comprova a estabilidade do sistema de controle de potencial e a eficácia da malha de realimentação do potenciostato, que se manteve livre de oscilações e mesmo nos limites operacionais de tensão. Paralelamente, os resultados validam o estágio de sensoriamento de corrente, demonstrando que o circuito de transimpedância opera dentro de sua zona de linearidade, sem ocorrência de saturação ou discrepâncias estatísticas que pudessem comprometer a integridade dos dados coletados.

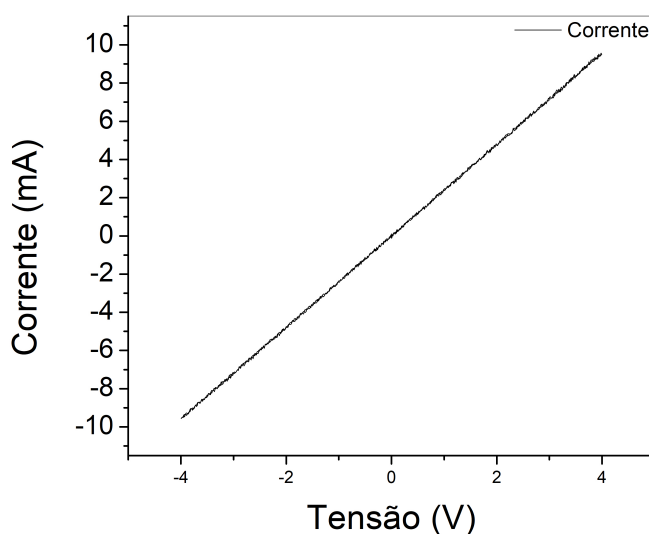


Figura 59: Voltametria verificando o comportamento da escala de 10mA

Fonte: O autor

5.7 ESTABILIDADE DO ELETRODO DE REFERÊNCIA

A estabilidade temporal do Eletrodo de Referência de Hidrogênio (ERH) desenvolvido foi avaliada por meio de um ensaio de cronoamperometria de longa duração. A configuração experimental consistiu em uma célula eletroquímica composta por um eletrodo de trabalho de platina, um contra eletrodo de aço inoxidável e o ERH como referência, todos imersos em solução de H_2SO_4 ($0,1 \text{ mol L}^{-1}$), conforme ilustrado na Figura 60.

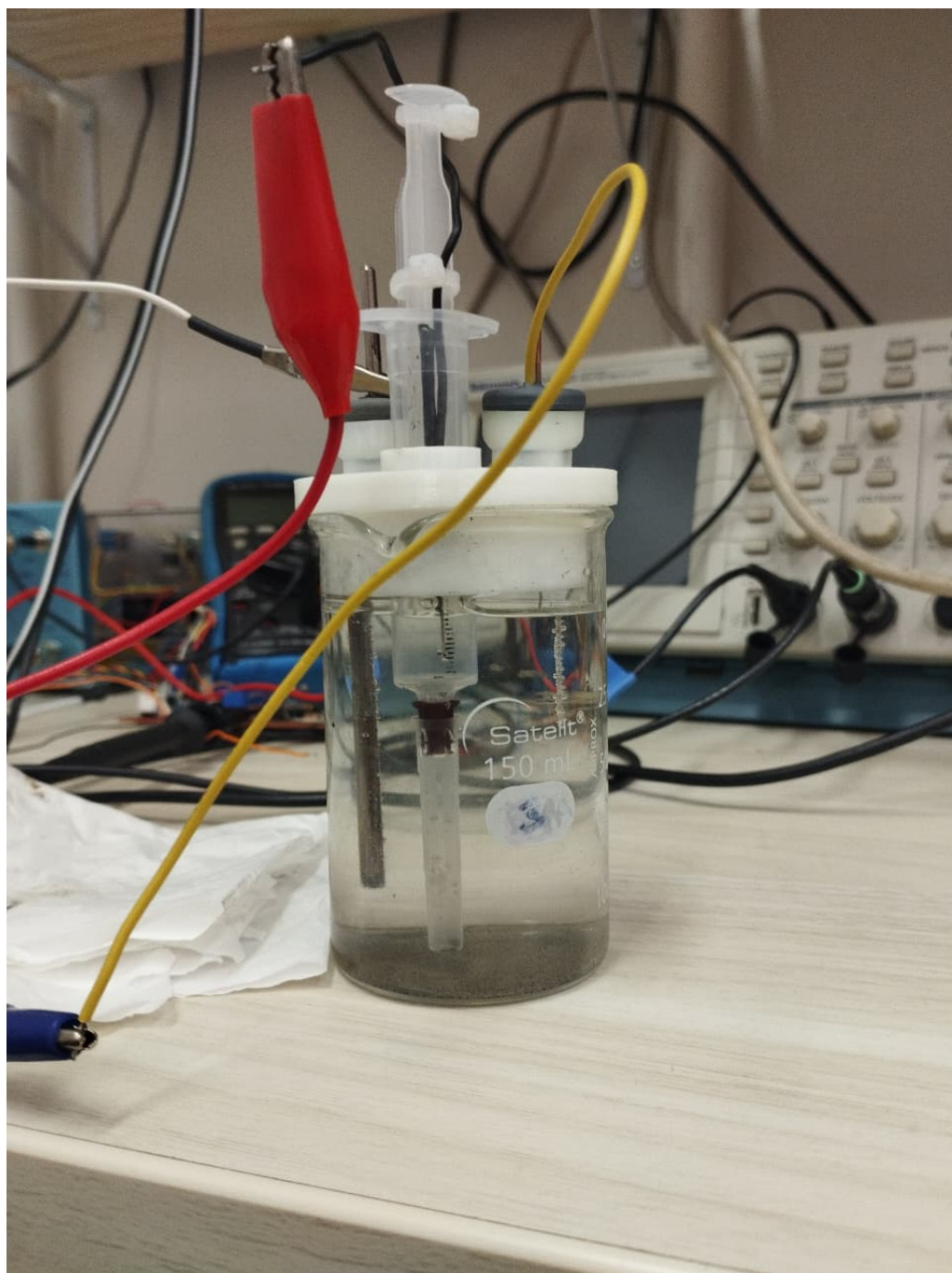


Figura 60: Montagem da célula eletroquímica para o teste de estabilidade

Fonte: O autor

O comportamento dinâmico do sistema sob potencial fixo de -3 V é apresentado na Figura 61. Observa-se, nos instantes iniciais (até aproximadamente 500 s), um incremento gradual na corrente antes da sua estabilização. Este fenômeno é atribuído ao período de ativação da superfície catalítica do eletrodo e ao rearranjo da dupla camada elétrica na interface eletrodo/eletrólito.

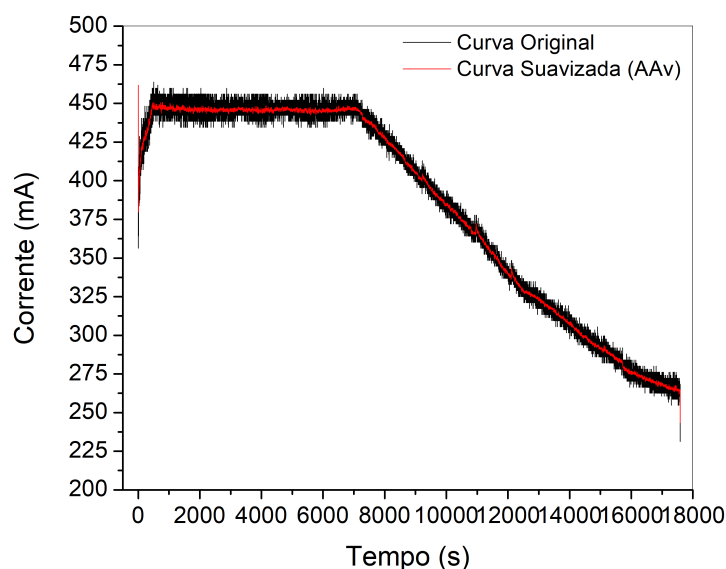


Figura 61: Teste de longa duração do Eletrodo de Referência sob potencial fixo de -3 V. Estabilidade mantida por aproximadamente 2 horas (7000 s)

Fonte: O autor

Após o período de indução, o sistema exibiu um patamar de corrente estável em torno de 450 mA por mais de 7.000 segundos. A partir deste ponto, iniciou-se um decaimento contínuo e linear da corrente até a interrupção do ensaio em 18.000 segundos. É importante ressaltar que a bolha de hidrogênio, essencial para o funcionamento do ERH, manteve-se íntegra e presente durante toda a medição.

A inspeção visual pós-teste (Figura 62) revelou que a queda de desempenho não está associada à falha do eletrodo de referência ou à difusão da bolha de gás. O fator limitante foi a degradação severa do contra eletrodo de aço inoxidável, que sofreu oxidação intensa e consumo de massa pelo processo eletrolítico, alterando a composição química do eletrólito e a geometria da célula.



Figura 62: Deterioração do contra eletrodo após 5 horas de teste

Fonte: O autor

Devido à instabilidade gerada pela oxidação do contra eletrodo e à consequente contaminação da solução por íons metálicos, a aplicação da Equação 28 para calcular a taxa de deriva de potencial (ΔV_{drift}) do ERH de forma isolada foi prejudicada e inviável.

$$\Delta V_{drift} = \frac{V(t_{final}) - V(t_{inicial})}{\Delta t} \quad (\text{mV/h}) \quad (28)$$

Contudo, os resultados validam a robustez do projeto do eletrodo de referência. Ele manteve sua integridade física e funcionalidade sob corrente elevada (0,45 A) por um período superior a duas horas. Isso demonstra que a estratégia de condicionamento de sinal com alta impedância de entrada (buffer OPA128) é plenamente adequada para a leitura precisa do eletrodo de referência desenvolvido e de eletrodos de referência comerciais.

6 CONSIDERAÇÕES FINAIS

O presente trabalho atingiu seu objetivo principal ao projetar, desenvolver e validar um protótipo de potenciostato modular de baixo custo, capaz de realizar ensaios de voltametria cíclica. A integração entre a plataforma Arduino, a linguagem Python e componentes eletrônicos de precisão, como o DAC714 e o buffer OPA128, resultou em um instrumento robusto.

Os ensaios de caracterização elétrica demonstraram uma resposta linear consistente e um tempo de acomodação adequado para as taxas de varredura típicas de experimentos eletroquímicos. A validação comparativa com o potenciostato comercial Autolab PGSTAT302N, utilizando uma *dummy cell*, confirmou a precisão do protótipo na aplicação de potencial e leitura de corrente, validando a topologia eletrônica adotada.

Uma das principais virtudes do equipamento desenvolvido é a sua modularidade. O sensor de corrente, baseado no circuito integrado INA282, pode ser facilmente modificado para entregar novas escalas de medição. Essa adaptação é realizada através da alteração do resistor de *shunt*, respeitando-se os limites de tensão de modo comum do componente, que operam entre -14 V e +80 V. Essa flexibilidade amplia significativamente o escopo de utilização do instrumento: o equipamento pode ser empregado tanto em aulas experimentais, onde o elemento ou evento de interesse acontece tipicamente na escala de miliamperes (mA), quanto em atividades de pesquisa que demandem aplicações de alta corrente.

Em relação à capacidade de potência, a fonte de alimentação, da maneira que está configurada atualmente, permite que o sistema entregue em plena carga de aproximadamente $\pm 6,0$ V e $\pm 2,5$ A. No entanto, a arquitetura do driver de potência foi dimensionada para suportar demandas superiores. Alterando-se o transformador para um modelo de ± 40 V com 5 A, o sistema torna-se capaz de entregar até ± 30 V com ± 5 A na saída, o que expande a janela de trabalho para ensaios de eletrodeposição ou corrosão mais exigentes.

Como perspectivas de trabalhos futuros e aprimoramento contínuo do protótipo, destaca-se o desenvolvimento da versão final deste protótipo didático, que será composto por componentes eletrônicos de fácil acesso no mercado, de custo menor e com a montagem facilitada através de uma placa de circuito impresso (PCB). Também temos que com o conhecimento acumulado, podemos produzir um potenciostato customizado, o que permite expandir as possibilidades de pesquisa da universidade.

Conclui-se que o potenciostato modular desenvolvido não apenas cumpre sua função como ferramenta didática acessível, mas também estabelece uma base sólida para o desenvolvimento de instrumentação científica aberta (Open Hardware) na instituição, fomentando a autonomia tecnológica e o aprendizado prático interdisciplinar em Engenharia Física.

REFERÊNCIAS

ARDUINO DOCUMENTATION. **Arduino Nano — Technical Reference**. Acesso em: 9 nov. 2025. 2024. Disponível em: <https://docs.arduino.cc/hardware/nano/>.

ARDUINO FOUNDATION. **Arduino Documentation — Open Source and Hardware Philosophy**. Acesso em: 9 nov. 2025. 2023. Disponível em: <https://docs.arduino.cc/>.

ATMEL CORPORATION. **ATmega328P — 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash**. Acesso em: 9 nov. 2025. 2015. Disponível em: https://ww1.microchip.com/downloads/en/devicedoc/atmel-7810-8-bit-avr-atmega328p_datasheet.pdf.

BARD, Allen J.; FAULKNER, Larry R. **Electrochemical Methods: Fundamentals and Applications**. 2nd. New York: John Wiley & Sons, 2001.

BUSONI, L.; CARLÀ, M.; LANZI, L. A comparison between potentiostatic circuits with grounded work or auxiliary electrode. **Review of Scientific Instruments**, American Institute of Physics, v. 73, n. 4, p. 1921–1923, abr. 2002. DOI: 10.1063/1.1463715.

CAMBURN, Bradley A. et al. A Systematic Review of Open-Source Hardware Design. **Journal of Open Hardware**, v. 1, n. 1, p. 1–13, 2017. DOI: 10.5334/joh.2.

CORP., National Instruments. **Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing**. Acesso em: 08 nov. 2025. 30 maio 2025. Disponível em: <https://www.ni.com/en/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/acquiring-an-analog-signal-bandwidth-nyquist-sampling-theorem-.html>.

CREATIVE COMMONS. **Atribuição-CompartilhaIgual 3.0 Não Adaptada (CC BY-SA 3.0)**. Acesso em: 9 nov. 2025. 2013. Disponível em: https://creativecommons.org/licenses/by-sa/3.0/deed.pt_BR.

DEVICES, Analog. **OP07: Ultra Low Offset Voltage Operational Amplifier**. 2020. Datasheet. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/OP07.pdf>.

FLOYD, Thomas L. **Sistemas digitais: fundamentos e aplicações**. 9. ed. São Paulo: Pearson Prentice Hall, 2007.

INSTRUMENTS, Texas. **DAC714 16-Bit Monolithic Digital-to-Analog Converter**. 2002. Datasheet. Disponível em: <https://www.ti.com/lit/ds/symlink/dac714.pdf>.

INSTRUMENTS, Texas. **INA282: High-Voltage, High-Accuracy Current-Shunt Monitor**. 2009. Datasheet. Disponível em: <https://www.ti.com/lit/ds/symlink/ina282.pdf>.

INSTRUMENTS, Texas. **OPA128 Ultra-Low Bias Current Operational Amplifier**. 2013. Datasheet. Disponível em: <https://www.ti.com/lit/ds/symlink/opa128.pdf>.

INSTRUMENTS, Texas. **OPA541 High-Voltage, High-Current Operational Amplifier**. 2016. Datasheet. Disponível em: <https://www.ti.com/lit/ds/symlink/opa541.pdf>.

INSTRUMENTS, Texas. **TL431 Programmable Precision Reference**. 2015. Datasheet. Disponível em: <https://www.ti.com/lit/ds/symlink/tl431.pdf>.

KOUTILELLIS, G. D.; ECONOMOU, A.; EFSTATHIOU, C. E. A potentiostat featuring an integrator transimpedance amplifier for the measurement of very low currents—Proof-of-principle application in microfluidic separations and voltammetry. **Review of Scientific Instruments**, AIP Publishing, Athens, Greece, v. 91, n. 12, p. 124104, 2020. DOI: 10.1063/5.0020841.

LAST MINUTE ENGINEERS. **Arduino Nano Pinout, Specifications, Schematic & Datasheet**. Acesso em: 9 nov. 2025. 2024. Disponível em: <https://lastminuteengineers.com/>.

MATTOS, Oscar R.; SILVA, Mariano S. **Um potenciostato/galvanostato rápido de projeto nacional**. Rio de Janeiro, Brasil, 1983.

METROHM. **Compreendendo a voltametria de varredura linear e a voltametria cíclica**. Acesso em: 06, 01. 2026. 2025. Disponível em: https://www.metrohm.com/pt_br/fique-por-dentro/blog/2025/compreendendo-a-voltametria-de-varredura-linear-e-cvs.html.

PYTHON SOFTWARE FOUNDATION. **Python Official Documentation**. Acesso em: 9 nov. 2025. 2024. Disponível em: <https://www.python.org/doc/>.

RED HAT, INC. **O que é Open Hardware?** Acesso em: 9 nov. 2025. 2023. Disponível em: <https://www.redhat.com/pt-br/topics/open-source/what-is-open-hardware>.

SKOOG, Douglas A.; HOLLER, F. James; NIEMAN, Timothy A. **Princípios de Análise Instrumental**. 5. ed. São Paulo: McGraw-Hill, 1992. Tradução da 5ª edição norte-americana.

SKOOG, Douglas A.; WEST, Donald M. et al. **Fundamentos de Química Analítica**. 9. ed. São Paulo: Cengage Learning, 2017.

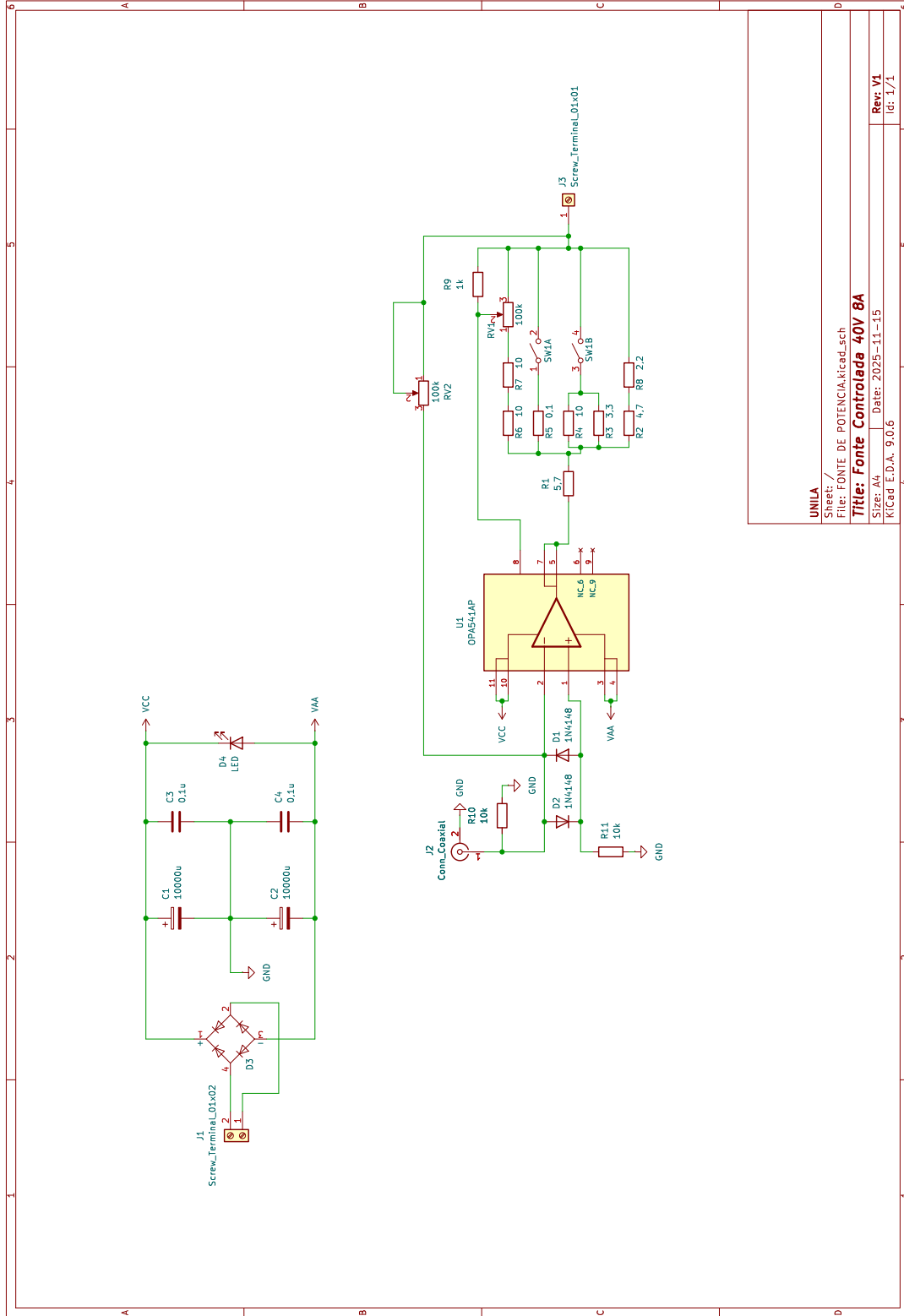
SMITH, Timothy J.; STEVENSON, Keith J. Reference Electrodes. *In*: ZOSKI, Cynthia G. (ed.). **Handbook of Electrochemistry**. Amsterdam: Elsevier, 2007. cap. 4.

SOUZA, Alexandre Kennedy Pinto. **Desenvolvimento de potenciostatos para caracterização de células eletroquímicas**. 2016. f. 81. Diss. (Mestrado) – Universidade Federal do Amazonas, Manaus. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Amazonas.

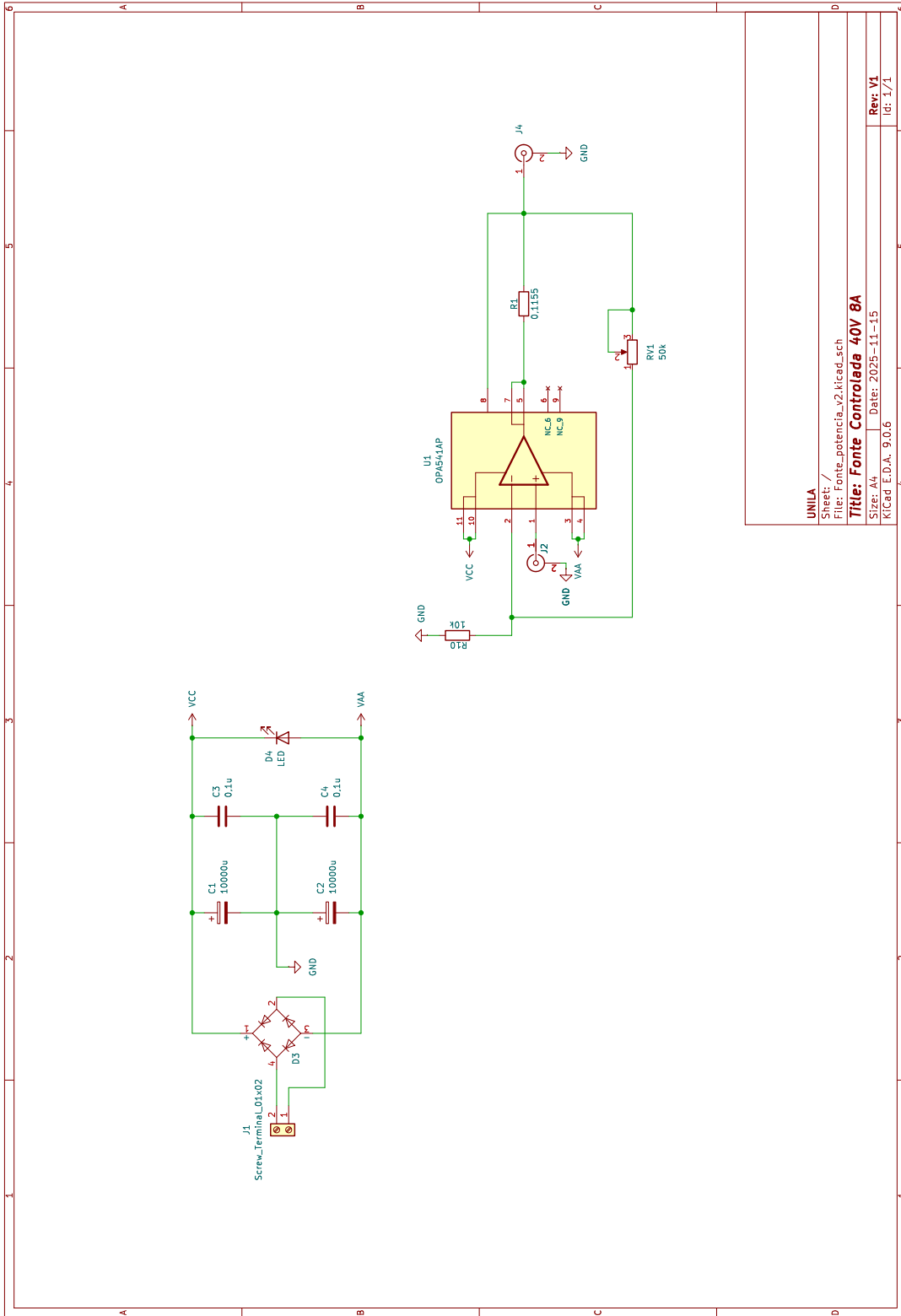
WIRING PROJECT. **Wiring: The Programming Framework Behind Arduino**. Acesso em: 9 nov. 2025. 2003. Disponível em: <http://wiring.org.co/>.

YARNITZKY, C. N. Design and Construction of a Potentiostat for a Chemical Metal-Walles Reactor. **Journal of Electroanalytical Chemistry**, maio 2000.

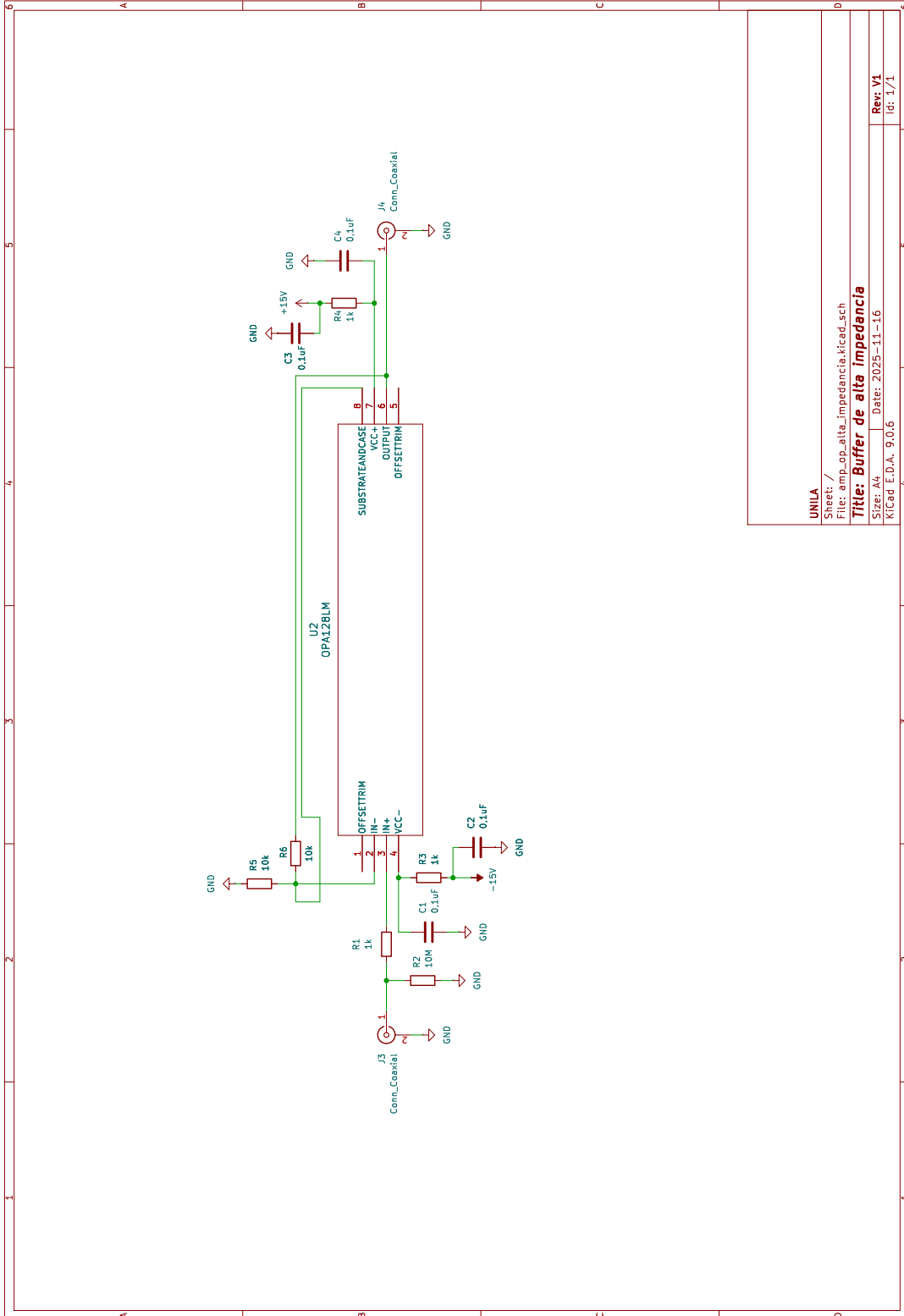
ANEXO A – FONTE DE POTÊNCIA



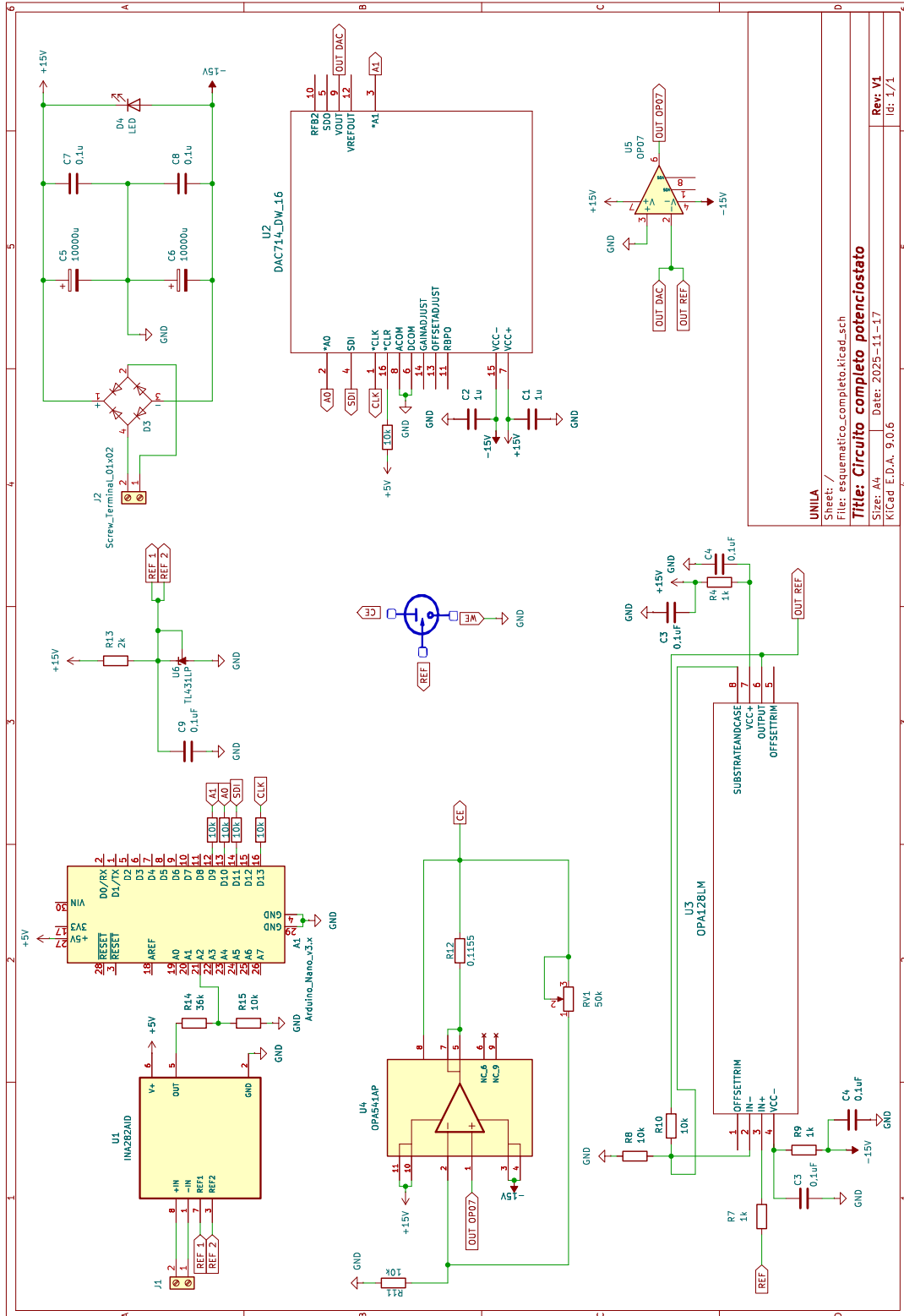
ANEXO B – FONTE DE POTÊNCIA SIMPLIFICADA



ANEXO C – BUFFER DE ALTA IMPEDÂNCIA PARA ELETRODO DE REFERÊNCIA



ANEXO E – CIRCUITO COMPLETO



UNILA
Sheet: /
File: esquematico_completo.kicad.sch
Title: Circuito completo potencioestado
Size: A4 | Date: 2025-11-17
KICad E.D.A. 9.0.6
Rev: V1
Id: 1/1

ANEXO F – Código Python - Interface

```

1  # Python Dependencies Installation
2
3  # Install
4  %pip install pyserial
5  %pip install numpy
6  %pip install PyQt5
7  %pip install pyqtgraph
8
9  # Verify installation
10 import sys
11 import serial
12 import numpy as np
13 from PyQt5.QtWidgets import QApplication
14 import pyqtgraph as pg
15
16 print(" All libraries installed successfully!")
17 print(f"Python version: {sys.version}")
18 print(f"pyserial version: {serial.__version__}")
19 print(f"numpy version: {np.__version__}")
20 print(f"pyqtgraph version: {pg.__version__}")
21
22 import sys
23 import serial
24 import numpy as np
25 from PyQt5.QtWidgets import (
26     QApplication, QMainWindow, QVBoxLayout, QWidget, QLabel, QPushButton, QDialog,
27     QMessageBox, QComboBox, QHBoxLayout
28 )
29 import pyqtgraph as pg
30 import os
31
32 class MainWindow(QMainWindow):
33     def __init__(self):
34         super().__init__()
35         self.setWindowTitle("Voltage vs Current Graph")
36         self.setGeometry(100, 100, 1000, 800)
37
38         # Central widget configuration
39         central_widget = QWidget()
40         self.setCentralWidget(central_widget)
41         layout = QVBoxLayout(central_widget)
42
43         # Status label
44         self.status_label = QLabel("Status: Waiting for connection and start...")
45         layout.addWidget(self.status_label)

```

```

45
46     # Label to show constant voltage applied
47     self.tensao_constante_label = QLabel("Constant Voltage: -- V")
48     self.tensao_constante_label.setStyleSheet("font-weight: bold; font-size: 12pt;
49     color: #0066cc;")
50     layout.addWidget(self.tensao_constante_label)
51     self.tensao_constante_label.hide() # Initially hidden
52
53     # Label to show estimated measurement time
54     self.tempo_estimado_label = QLabel("Estimated Time: -- s | Estimated Data: --")
55     self.tempo_estimado_label.setStyleSheet("font-weight: bold; font-size: 11pt;
56     color: #006600;")
57     layout.addWidget(self.tempo_estimado_label)
58
59     # System configuration info box
60     info_box = QLabel("""
61     <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;
62     border-radius: 5px;">
63     <b>System Configuration:</b><br>
64     • DAC714: Range ±10V, 16-bit resolution (corrected for inversion)<br>
65     • INA282: 0V=-1A, 0.55V=0A, 1.1V=+1A (internal 1.1V reference)<br>
66     • Pin A2: Analog current reading<br>
67     • Scale: Current in mA
68     </div>
69     """)
70     info_box.setWordWrap(True)
71     layout.addWidget(info_box)
72
73     # Graph configuration
74     self.plot_widget = pg.PlotWidget()
75     self.plot_widget.setTitle("DAC Voltage vs Current (INA282)")
76     self.plot_widget.setLabel('left', 'Current (mA)')
77     self.plot_widget.setLabel('bottom', 'DAC Voltage (V)')
78     self.plot_widget.showGrid(x=True, y=True)
79     # Removed fixed limits for auto adjustment
80     # self.plot_widget.setXRange(0, 2)
81     # self.plot_widget.setYRange(-200, 200)
82     self.plot_widget.setMouseEnabled(x=False, y=False) # Disable mouse zoom
83
84     layout.addWidget(self.plot_widget)
85
86     # Operation mode (Ramp or Constant Voltage)
87     self.modos_operacao = "Rampa" # Default
88     modo_layout = QHBoxLayout()
89     modo_label = QLabel("Operation Mode:")
90     self.modos_combo = QComboBox()
91     self.modos_combo.addItem("Rampa")
92     self.modos_combo.addItem("Tensão Constante")

```

```

89     self.modos_combo.currentTextChanged.connect(self.on_modos_changed)
90     modos_layout.addWidget(modos_label)
91     modos_layout.addWidget(self.modos_combo)
92     modos_layout.addStretch()
93     layout.addLayout(modos_layout)
94
95     # Data initialization
96     self.time_data = []
97     self.dac_voltage_data = [] # DAC voltage
98     self.current_data = [] # Current in mA from INA282
99     self.curve = self.plot_widget.plot(pen='b')
100    self.tensao_constante_aplicada = 0.0 # Constant voltage in constant mode
101
102    # Program state: starts stopped (not reading data)
103    self.running = False
104
105    # Serial port configuration
106    self.serial_port = None
107    self.serial_connected = False
108    try:
109        self.serial_port = serial.Serial('COM11', 115200, timeout=1)
110        # Wait for Arduino ready signal
111        self.serial_port.reset_input_buffer()
112        self.serial_port.reset_output_buffer()
113        ready = False
114        for _ in range(200): # ~20s to accommodate restarts/bootloaders
115            line = self.serial_port.readline().decode('utf-8', errors='ignore').
strip()
116            if line == 'ACK:READY':
117                ready = True
118                break
119            if ready:
120                self.status_label.setText("Status: Connected and ready (ACK:READY).")
121            else:
122                self.status_label.setText("Status: Connected. No ACK:READY; Arduino may
be restarting. You can start.")
123                self.serial_connected = True
124            except Exception as e:
125                self.status_label.setText("Demo mode: no serial connection (COM11
unavailable).")
126                print(f"Error opening serial port. Application will continue in demo mode.
Detail: {e}")
127
128    # Control buttons
129    self.control_button = QPushButton("Configure Parameters")
130    self.control_button.clicked.connect(self.configure_parameters)
131    layout.addWidget(self.control_button)

```

```

132
133     self.start_button = QPushButton("Start")
134     self.start_button.clicked.connect(self.start_acquisition)
135     layout.addWidget(self.start_button)
136
137     self.stop_button = QPushButton("Stop")
138     self.stop_button.clicked.connect(self.stop_acquisition)
139     self.stop_button.setEnabled(False)
140     layout.addWidget(self.stop_button)
141
142     self.clear_button = QPushButton("Clear Data")
143     self.clear_button.clicked.connect(self.clear_data)
144     layout.addWidget(self.clear_button)
145
146     self.save_button = QPushButton("Save Data")
147     self.save_button.clicked.connect(self.save_data)
148     layout.addWidget(self.save_button)
149
150     # Timer for graph update
151     self.timer = pg.QtCore.QTimer()
152     self.timer.timeout.connect(self.update_plot)
153     # Don't start timer yet; will be activated with start button
154
155     def on_modos_changed(self, modo):
156         """Updates interface when operation mode changes"""
157         self.modo_operacao = modo
158         if modo == "Tensão Constante":
159             self.plot_widget.setTitle("Time vs Current (Constant Voltage)")
160             self.plot_widget.setLabel('bottom', 'Time (s)')
161             self.tensao_constante_label.show()
162         else: # Rampa
163             self.plot_widget.setTitle("DAC Voltage vs Current (INA282)")
164             self.plot_widget.setLabel('bottom', 'DAC Voltage (V)')
165             self.tensao_constante_label.hide()
166         # Clear data when changing mode
167         if not self.running:
168             self.clear_data()
169
170     def configure_parameters(self):
171         if self.running:
172             QMessageBox.warning(self, "Warning", "Stop acquisition before changing
173 parameters.")
174             return
175
176         if self.modo_operacao == "Tensão Constante":
177             # Constant voltage mode: asks only for voltage and maintains unlimited
178             # execution with fixed TX

```

```

177         tensao_const, ok = QInputDialog.getDouble(
178             self, "Constant Voltage (V)", "Enter Constant Voltage (V):", 0.0, -10.0,
10.0, 6)
179         if not ok:
180             return
181
182         # Store constant voltage and mark unlimited duration
183         self.tensao_constante_aplicada = tensao_const
184         self.duracao_constante = None
185         self.tensao_constante_label.setText(f"Constant Voltage: {tensao_const:.6f} V
")
186
187         # Set fixed TX_INTERVAL at 500 ms and unlimited duration (0 signals no limit
)
188         tx_interval_ms = 500
189         duracao = 0 # Interpreted by firmware as continuous execution
190
191         self.tempo_estimado_label.setText(
192             "Estimated Time: Unlimited | TX Interval: 500 ms"
193         )
194
195         # Send command for constant mode with voltage and TX_INTERVAL
196         # Format: "MODE:CONSTANT,tensao,duracao,tx_interval_ms"
197         params = f"MODE:CONSTANT,{tensao_const},{duracao},{tx_interval_ms}\n"
198         if self.serial_connected:
199             try:
200                 self.serial_port.reset_input_buffer()
201                 self.serial_port.reset_output_buffer()
202                 self.serial_port.write(params.encode())
203                 self.serial_port.flush()
204                 self.status_label.setText(
205                     f"Status: Constant Voltage mode configured. Voltage: {
tensao_const:.6f}V, "
206                     "Unlimited time, TX Interval 500ms"
207                 )
208             except Exception as e:
209                 self.status_label.setText(f"Error sending parameters: {str(e)}")
210             else:
211                 self.status_label.setText(
212                     "Status: Parameters stored locally (no serial connection)."
213                 )
214         else:
215             # Ramp mode: asks for all ramp parameters
216             # Collect user parameters with 6 decimal places for maximum precision
217             # DAC714 minimum step: 20V/65536 0.000305V (range ±10V - bipolar mode)
218             tens_min, ok1 = QInputDialog.getDouble(

```

```

219         self, "Minimum Voltage (V)", "Enter Minimum Voltage (V):", -1.2, -10.0,
10.0, 6)
220     if not ok1:
221         return
222     tens_max, ok2 = QInputDialog.getDouble(
223         self, "Maximum Voltage (V)", "Enter Maximum Voltage (V):", 3.0, -10.0,
10.0, 6)
224     if not ok2:
225         return
226     velocidade, ok3 = QInputDialog.getDouble(
227         self, "Velocity (V/s)", "Enter Ramp Velocity (V/s):", 0.2, 0.0001, 20.0,
6)
228     if not ok3:
229         return
230     passos_seg, ok4 = QInputDialog.getInt(
231         self, "Steps per second", "Enter Number of Steps:", 50, 1, 2000)
232     if not ok4:
233         return
234     qtd_ciclos, ok5 = QInputDialog.getInt(
235         self, "Number of Cycles", "Enter Number of Cycles:", 3, 1, 1000)
236     if not ok5:
237         return
238
239     if tens_min >= tens_max:
240         QMessageBox.critical(self, "Error", "Minimum voltage must be less than
maximum.")
241         return
242
243     # Ask for desired number of points
244     num_pontos, ok6 = QInputDialog.getInt(
245         self, "Number of Points", "Enter Desired Number of Points in Graph:",
800, 100, 100000, 1)
246     if not ok6:
247         return
248
249     # Calculate voltage step and validate DAC resolution
250     # DAC714 minimum resolution: 20V/65536 0.000305V (range ±10V - bipolar
mode)
251     RESOLUCAO_DAC = 0.000305
252     tensao_por_passo = velocidade / passos_seg
253
254     # If step is smaller than resolution, adjust passos_seg
255     if tensao_por_passo < RESOLUCAO_DAC:
256         passos_seg_ajustado = int(velocidade / RESOLUCAO_DAC) + 1
257         QMessageBox.warning(
258             self, "Automatic Adjustment",

```

```

259         f"Calculated step ({tensao_por_passo:.6f}V) is smaller than DAC
resolution ({RESOLUCAO_DAC}V).\n"
260         f"Steps per second adjusted from {passos_seg} to {
passos_seg_ajustado}."
261     )
262     passos_seg = passos_seg_ajustado
263     tensao_por_passo = velocidade / passos_seg
264
265     # Store resolution for later use
266     self.resolucao_dac = RESOLUCAO_DAC
267
268     # Store parameters and derived calculations
269     self.tens_min = tens_min
270     self.tens_max = tens_max
271     self.velocidade = velocidade
272     self.passos_seg = passos_seg
273     self.qtd_ciclos = qtd_ciclos
274     self.duracao_passo = 1.0 / passos_seg
275     self.tensao_por_passo = tensao_por_passo
276
277     # Calculate total measurement time (up and down per cycle)
278     amplitude = tens_max - tens_min
279     tempo_por_ciclo = 2 * (amplitude / velocidade) # Up and down
280     tempo_total = qtd_ciclos * tempo_por_ciclo
281
282     # Calculate TX_INTERVAL_MS based on desired number of points
283     tx_interval_ms = max(10, int((tempo_total * 1000) / num_pontos)) # Minimum
10ms
284
285     # Calculate real number of data points with calculated interval
286     num_dados_estimado = int((tempo_total * 1000) / tx_interval_ms)
287
288     # Update estimated time label
289     self.tempo_estimado_label.setText(
290         f"Estimated Time: {tempo_total:.2f} s | Estimated Data: {
num_dados_estimado} | TX Interval: {tx_interval_ms} ms"
291     )
292
293     # Send parameters to Arduino (ramp mode)
294     # Format: "MODE:RAMP,tens_min,tens_max,velocidade,passos_seg,qtd_ciclos,
tx_interval_ms"
295     params = f"MODE:RAMP,{tens_min},{tens_max},{velocidade},{passos_seg},{
qtd_ciclos},{tx_interval_ms}\n"
296     if self.serial_connected:
297         try:
298             self.serial_port.reset_input_buffer()
299             self.serial_port.reset_output_buffer()

```

```

300         self.serial_port.write(params.encode())
301         self.serial_port.flush()
302         self.status_label.setText(
303             f>Status: Ramp parameters sent. Δt step = {self.duracao_passo:.6
f}s, "
304             f"ΔV step = {self.tensao_por_passo:.6f}V ({self.resolucao_dac}V
). Total time: {tempo_total:.2f}s, Data: {num_dados_estimado}"
305         )
306     except Exception as e:
307         self.status_label.setText(f"Error sending parameters: {str(e)}")
308     else:
309         self.status_label.setText(
310             "Status: Ramp parameters stored (no serial connection)."
311         )
312
313     def start_acquisition(self):
314         if self.running:
315             return
316         if not self.serial_connected:
317             QMessageBox.information(
318                 self,
319                 "No Connection",
320                 "No connection with Arduino. Connect device to start acquisition."
321             )
322             self.status_label.setText("Status: No serial connection. Acquisition not
started.")
323             return
324         # Clear data before starting
325         self.clear_data()
326         # Store initial time for both modes (relative time starts at 0)
327         import time
328         self.tempo_inicial = time.time()
329         self.tempo_inicial_arduino = None # Will be set when receiving first data
330         self.running = True
331         self.timer.start(50)
332         self.start_button.setEnabled(False)
333         self.stop_button.setEnabled(True)
334         self.control_button.setEnabled(False)
335         self.modos_combo.setEnabled(False) # Disable mode change during execution
336         try:
337             self.serial_port.reset_input_buffer()
338             self.serial_port.reset_output_buffer()
339             self.serial_port.write(b"START\n")
340             self.serial_port.flush()
341             # Wait for ACK
342             ack_ok = False
343             for _ in range(200): # ~20s

```

```

344         line = self.serial_port.readline().decode('utf-8', errors='ignore').
strip()
345         if line == 'ACK:START' or line == 'ACK:RUNNING':
346             ack_ok = True
347             break
348         if ack_ok:
349             modo_texto = "Tensão Constante" if self.modo_operacao == "Tensão
Constante" else "Rampa"
350             self.status_label.setText(f"Status: Acquisition started ({modo_texto}) -
ACK received.")
351         else:
352             self.status_label.setText("Status: Acquisition started (no ACK; check
connection).")
353     except Exception as e:
354         self.status_label.setText(f"Error sending START command: {str(e)}")
355
356     def stop_acquisition(self):
357         if not self.running:
358             return
359         self.running = False
360         self.timer.stop()
361         self.start_button.setEnabled(True)
362         self.stop_button.setEnabled(False)
363         self.control_button.setEnabled(True)
364         self.modo_combo.setEnabled(True) # Re-enable mode change
365         if self.serial_connected:
366             try:
367                 self.serial_port.write(b"RESET\n")
368                 self.serial_port.flush()
369                 # Optionally wait for ACK:RESET
370                 for _ in range(20):
371                     line = self.serial_port.readline().decode('utf-8', errors='ignore').
strip()
372                     if line == 'ACK:RESET':
373                         break
374             except Exception as e:
375                 self.status_label.setText(f"Error sending RESET command: {str(e)}")
376                 self.status_label.setText("Status: Acquisition stopped.")
377
378     def clear_data(self):
379         if self.running:
380             QMessageBox.warning(self, "Warning", "Stop acquisition before clearing data.
")
381             return
382         self.current_data.clear()
383         self.time_data.clear()
384         self.dac_voltage_data.clear()

```

```

385     self.curve.clear()
386     # Reset relative time when clearing data
387     if hasattr(self, 'tempo_inicial'):
388         del self.tempo_inicial
389     if hasattr(self, 'tempo_inicial_arduino'):
390         self.tempo_inicial_arduino = None
391     self.status_label.setText("Status: Data cleared.")
392
393     def save_data(self):
394         if len(self.current_data) == 0:
395             QMessageBox.information(self, "Info", "No data to save.")
396             return
397
398         downloads_path = os.path.join(os.path.expanduser("~"), "Downloads")
399
400         # Create filename with date/time and system parameters
401         from datetime import datetime
402         now = datetime.now()
403         timestamp = now.strftime("%Y%m%d_%H%M%S")
404
405         # Try to get current parameters (if available)
406         try:
407             # Use stored parameters when configured
408             tens_min = getattr(self, 'tens_min', -1.2)
409             tens_max = getattr(self, 'tens_max', 3.0)
410             velocidade = getattr(self, 'velocidade', 0.2)
411             passos_seg = getattr(self, 'passos_seg', 50)
412             qtd_ciclos = getattr(self, 'qtd_ciclos', 3)
413
414             filename = (
415                 f"dados_ina282_{timestamp}_min{tens_min}_max{tens_max}"
416                 f"_vel{velocidade}_passos{passos_seg}_ciclos{qtd_ciclos}.csv"
417             )
418         except:
419             filename = f"dados_ina282_{timestamp}.csv"
420
421         file_path = os.path.join(downloads_path, filename)
422
423         try:
424             with open(file_path, 'w') as f:
425                 f.write("Tempo (s),Tensao_DAC (V),Corrente (mA)\n")
426                 for i in range(len(self.current_data)):
427                     t = self.time_data[i] if i < len(self.time_data) else i * 0.5
428                     dac_voltage = self.dac_voltage_data[i] if i < len(self.
dac_voltage_data) else 0.0
429                     f.write(f"{t:.3f},{dac_voltage:.6f},{self.current_data[i]:.6f}\n")
430

```

```

431         self.status_label.setText(f"Status: Data saved to {file_path}")
432     except Exception as e:
433         self.status_label.setText(f"Error saving data: {str(e)}")
434
435     def update_plot(self):
436         if not self.running:
437             return
438
439         if not self.serial_connected or not self.serial_port:
440             return
441
442         if self.serial_port.in_waiting > 0:
443             try:
444                 line = self.serial_port.readline().decode('utf-8', errors='ignore').
strip()
445
446                 # Check if received ACK:DONE (constant mode finished)
447                 if line == 'ACK:DONE':
448                     # Automatically stop acquisition when time ends in constant mode
449                     if self.modoo_peracao == "Tensão Constante":
450                         self.stop_acquisition()
451                         self.status_label.setText("Status: Measurement completed
automatically. Voltage zeroed.")
452                         return
453
454                     # Ignore other status/ACK messages
455                     if line.startswith('ACK:') or line.startswith('ERR:') or len(line) == 0:
456                         return
457
458                     if ',' in line:
459                         parts = line.split(',')
460                         if len(parts) != 4: # Expected: time, tensao_dac, corrente_mA,
tensao_dac_duplicated
461                             return
462
463                         time_str, dac_voltage_str, current_str, _ = parts # Ignore 4th
duplicated column
464
465                         try:
466                             time = float(time_str)
467                             dac_voltage = float(dac_voltage_str)
468                             current = float(current_str) # Already in mA from Arduino
469
470                         except ValueError:
471                             return
472
473                     # Use relative time since start for both modes (starts at 0s)
474                     # On first data, store Arduino time as reference
475                     if self.tempo_inicial_arduino is None:
476                         self.tempo_inicial_arduino = time
477
478                     # Calculate relative time since first received data
479                     tempo_relativo = time - self.tempo_inicial_arduino

```

```

474         self.time_data.append(tempo_relativo)
475
476         self.dac_voltage_data.append(dac_voltage)
477         self.current_data.append(current)
478
479         # Update graph depending on mode
480         # Apply transformations: X axis multiplied by -1 only in ramp mode
481         if self.modo_operacao == "Tensão Constante":
482             # Constant mode: time on X axis (no inversion), current on Y
axis without adjustment
483             x_data = self.time_data
484             y_data = self.current_data
485             self.curve.setData(x_data, y_data)
486             tempo_atual = self.time_data[-1] if self.time_data else 0.0
487             self.status_label.setText(
488                 f"Status: Time: {tempo_atual:.3f}s, Current: {current:.6f}mA
, Voltage: {dac_voltage:.6f}V"
489             )
490         else:
491             # Ramp mode: voltage on X axis (inverted), current on Y axis
without adjustment
492             # Transform: X = -X
493             x_data = [-v for v in self.dac_voltage_data]
494             y_data = self.current_data
495             self.curve.setData(x_data, y_data)
496             self.status_label.setText(
497                 f"Status: Data received - DAC Voltage: {dac_voltage:.6f}V,
Current: {current:.6f}mA"
498             )
499
500             # Auto adjust axes to include all received data
501             self.plot_widget.enableAutoRange(axis=pg.ViewBox.XAxis, enable=True)
502             self.plot_widget.enableAutoRange(axis=pg.ViewBox.YAxis, enable=True)
503         except Exception as e:
504             self.status_label.setText(f"Error reading serial: {str(e)}")
505
506     def closeEvent(self, event):
507         self.timer.stop()
508         if self.serial_port and self.serial_port.is_open:
509             self.serial_port.close()
510         event.accept()
511
512 if __name__ == '__main__':
513     app = QApplication(sys.argv)
514     window = MainWindow()
515     window.show()
516     sys.exit(app.exec_())

```

ANEXO G – Código Arduino

```

1 #include <SPI.h>
2 #include <math.h>
3
4 // DAC714 pins
5 const int clkPin = 13;
6 const int a0Pin = 10;
7 const int a1Pin = 9;
8 const int clrPin = 8;
9 // INA282 sensor pin
10 const int INA282_PIN = A2;
11 // INA282 calibration offset (in mA)
12 // Adjust this value to compensate offset when should be 0A
13 const float INA282_OFFSET_MA = -20.0;
14 // Operation mode
15 enum ModoOperacao {
16     MODO_RAMPA,
17     MODO_CONSTANTE
18 };
19 ModoOperacao modoAtual = MODO_RAMPA;
20 // Configurable ramp parameters
21 float TENS_MIN = -1.2;
22 float TENS_MAX = 3.0;
23 float VELOCIDADE = 0.2;           // V/s
24 int PASSOS_POR_SEG = 50;         // steps per second
25 int QTD_CICLOS = 3;
26 // Constant mode parameters
27 float TENSAO_CONSTANTE = 0.0;
28 float DURACAO_CONSTANTE = 60.0; // Duration in seconds
29 unsigned long tempoInicioConstante = 0;
30 // Derived parameters
31 float VOLT_POR_PASSO = 0.0;
32 unsigned long PASSO_INTERVAL_MS = 20;
33 // Ramp state
34 float tensao = 0.0;
35 int direcao = 1;                 // +1 rising, -1 falling
36 int cicloAtual = 0;
37 bool running = false;
38 bool retornandoZero = false;
39 // Transmission control (configurable)
40 unsigned long TX_INTERVAL_MS = 500;
41 unsigned long _lastTxMs = 0;
42
43 void setup() {
44     pinMode(clkPin, OUTPUT);
45     pinMode(a0Pin, OUTPUT);

```

```

46 pinMode(a1Pin, OUTPUT);
47 pinMode(clrPin, OUTPUT);
48
49 digitalWrite(clkPin, LOW);
50 digitalWrite(a0Pin, HIGH);
51 digitalWrite(a1Pin, HIGH);
52 digitalWrite(clrPin, HIGH);
53
54 Serial.begin(115200);
55 SPI.begin();
56 SPI.setDataMode(SPI_MODE1);
57 SPI.setBitOrder(MSBFIRST);
58
59 // Analog pin configuration for INA282
60 pinMode(INA282_PIN, INPUT);
61
62 // Configure internal 1.1V reference for INA282 reading (Arduino Nano)
63 // IMPORTANT: Must be called BEFORE any analogRead()
64 analogReference(INTERNAL);
65
66 // Ensure DAC714 reset
67 digitalWrite(clrPin, LOW);
68 delayMicroseconds(5);
69 digitalWrite(clrPin, HIGH);
70
71 writeDAC(voltageToDAC(0.0)); // Initialize at 0V
72 delay(1000);
73
74 atualizarDerivados();
75 Serial.println("ACK:READY");
76 }
77
78 void loop() {
79   processarSerial();
80
81   if (running) {
82     if (modoAtual == MODO_CONSTANTE) {
83       // Constant mode: maintains fixed voltage. If duration > 0, ends automatically.
84       unsigned long tempoAtual = millis();
85       float tempoDecorrido = (tempoAtual - tempoInicioConstante) / 1000.0;
86
87       bool temLimite = DURACAO_CONSTANTE > 0.0f;
88       if (temLimite && tempoDecorrido >= DURACAO_CONSTANTE) {
89         // Duration reached: zero voltage immediately and stop
90         tensao = 0.0;
91         writeDAC(voltageToDAC(0.0));
92         running = false;

```

```

93     retornandoZero = false; // No need to return, already at zero
94     Serial.println("ACK:DONE");
95     } else {
96         // Maintain constant voltage
97         tensao = TENSAO_CONSTANTE;
98         writeDAC(voltageToDAC(tensao));
99         delay(10); // Small delay to avoid overload
100    }
101    } else {
102        // Ramp mode: execute ramp step
103        executarPassoRampa();
104        delay(PASSO_INTERVAL_MS);
105    }
106 }
107 else if (retornandoZero) {
108     retornarParaZero();
109     delay(PASSO_INTERVAL_MS);
110 }
111 else {
112     if (fabsf(tensao) > 0.0001f) {
113         tensao = 0.0f;
114         writeDAC(voltageToDAC(0.0f));
115     }
116     delay(PASSO_INTERVAL_MS);
117 }
118
119 readSensors();
120 }
121
122 void processarSerial() {
123     // Receive data and commands from serial
124     if (Serial.available()) {
125         String input = Serial.readStringUntil('\n');
126         input.trim();
127
128         if (input.equalsIgnoreCase("START")) {
129             iniciarOperacao();
130         }
131         else if (input.equalsIgnoreCase("RESET")) {
132             Serial.println("ACK:RESET");
133             running = false;
134             retornandoZero = true;
135         }
136         else if (input.startsWith("MODE:")) {
137             parseModeCommand(input);
138         }
139         else if (input.indexOf(',') > 0) {

```

```
140     parseParameters(input);
141     }
142 }
143 }
144
145 void iniciarOperacao() {
146     if (running) {
147         return;
148     }
149
150     running = true;
151     retornandoZero = false;
152
153     if (modoAtual == MODO_CONSTANTE) {
154         // Constant mode: apply constant voltage directly and store initial time
155         tensao = TENSAO_CONSTANTE;
156         tempoInicioConstante = millis();
157         writeDAC(voltageToDAC(tensao));
158     } else {
159         // Ramp mode: start from zero
160         direcao = 1;
161         cicloAtual = 0;
162         tensao = 0.0;
163         writeDAC(voltageToDAC(tensao));
164     }
165
166     Serial.println("ACK:START");
167     Serial.println("ACK:RUNNING");
168 }
169
170 void executarPassoRampa() {
171     tensao += direcao * VOLT_POR_PASSO;
172
173     if (direcao > 0 && tensao >= TENS_MAX) {
174         tensao = TENS_MAX;
175         direcao = -1;
176     } else if (direcao < 0 && tensao <= TENS_MIN) {
177         tensao = TENS_MIN;
178         direcao = 1;
179         cicloAtual++;
180         if (cicloAtual >= QTD_CICLOS) {
181             running = false;
182             retornandoZero = true;
183             Serial.println("ACK:DONE");
184         }
185     }
186 }
```

```

187 writeDAC(voltageToDAC(tensao));
188 }
189
190 void retornarParaZero() {
191     if (fabsf(tensao) <= fabsf(VOLT_POR_PASSO)) {
192         tensao = 0.0f;
193         retornandoZero = false;
194         writeDAC(voltageToDAC(0.0f));
195         return;
196     }
197
198     if (tensao > 0.0f) {
199         tensao -= VOLT_POR_PASSO;
200     } else {
201         tensao += VOLT_POR_PASSO;
202     }
203
204     writeDAC(voltageToDAC(tensao));
205 }
206
207 void parseModeCommand(String input) {
208     // Expected format: "MODE:CONSTANT,tensao,duracao,tx_interval_ms" or "MODE:RAMP,
209     // tens_min,tens_max,velocidade,passos_seg,qtd_ciclos,tx_interval_ms"
210     int colonIndex = input.indexOf(':');
211     int commaIndex = input.indexOf(',');
212
213     if (colonIndex > 0 && commaIndex > colonIndex) {
214         String modeStr = input.substring(colonIndex + 1, commaIndex);
215
216         if (modeStr.equalsIgnoreCase("CONSTANT")) {
217             // Constant mode: "MODE:CONSTANT,tensao,duracao,tx_interval_ms"
218             int commaIndex2 = input.indexOf(',', commaIndex + 1);
219             int commaIndex3 = input.indexOf(',', commaIndex2 + 1);
220
221             if (commaIndex2 > 0 && commaIndex3 > 0) {
222                 float tensaoTemp = input.substring(commaIndex + 1, commaIndex2).toFloat();
223                 float duracaoTemp = input.substring(commaIndex2 + 1, commaIndex3).toFloat();
224                 unsigned long txIntervalTemp = input.substring(commaIndex3 + 1).toInt();
225
226                 if (tensaoTemp >= -10.0 && tensaoTemp <= 10.0 && duracaoTemp >= 0 &&
227                 txIntervalTemp >= 10) {
228                     modoAtual = MODO_CONSTANTE;
229                     TENSAO_CONSTANTE = tensaoTemp;
230                     DURACAO_CONSTANTE = duracaoTemp;
231                     TX_INTERVAL_MS = txIntervalTemp;
232                     Serial.print("CONSTANT mode configured. Voltage: ");
233                     Serial.print(TENSAO_CONSTANTE, 6);

```

```

232     Serial.print(", Duration: ");
233     if (DURACAO_CONSTANTE > 0.0f) {
234         Serial.print(DURACAO_CONSTANTE, 2);
235         Serial.print("s");
236     } else {
237         Serial.print("unlimited");
238     }
239     Serial.print(", TX Interval: ");
240     Serial.print(TX_INTERVAL_MS);
241     Serial.println("ms");
242 } else {
243     Serial.println("Error: invalid constant mode parameters.");
244 }
245 } else {
246     Serial.println("Error: invalid constant mode format.");
247 }
248 }
249 else if (modeStr.equalsIgnoreCase("RAMP")) {
250     // Ramp mode: "MODE:RAMP,tens_min,tens_max,velocidade,passos_seg,qtd_ciclos,
tx_interval_ms"
251     modoAtual = MODO_RAMPA;
252     String paramsStr = input.substring(commaIndex + 1);
253     parseParameters(paramsStr);
254 }
255 else {
256     Serial.println("Error: unknown mode. Use CONSTANT or RAMP.");
257 }
258 } else {
259     Serial.println("Error: invalid MODE command format.");
260 }
261 }
262
263 void parseParameters(String input) {
264     // Split input string into parts (format: tens_min,tens_max,velocidade,passos_seg,
qtd_ciclos,tx_interval_ms)
265     int commaIndex1 = input.indexOf(',');
266     int commaIndex2 = input.indexOf(',', commaIndex1 + 1);
267     int commaIndex3 = input.indexOf(',', commaIndex2 + 1);
268     int commaIndex4 = input.indexOf(',', commaIndex3 + 1);
269     int commaIndex5 = input.indexOf(',', commaIndex4 + 1);
270
271     if (commaIndex1 > 0 && commaIndex2 > 0 && commaIndex3 > 0 && commaIndex4 > 0 &&
commaIndex5 > 0) {
272         float tensMinTemp = input.substring(0, commaIndex1).toFloat();
273         float tensMaxTemp = input.substring(commaIndex1 + 1, commaIndex2).toFloat();
274         float velocidadeTemp = input.substring(commaIndex2 + 1, commaIndex3).toFloat();
275         int passosTemp = input.substring(commaIndex3 + 1, commaIndex4).toInt();

```

```

276     int ciclosTemp = input.substring(commaIndex4 + 1, commaIndex5).toInt();
277     unsigned long txIntervalTemp = input.substring(commaIndex5 + 1).toInt();
278
279     if (tensMinTemp < tensMaxTemp && velocidadeTemp > 0 && passosTemp > 0 && ciclosTemp
    > 0 && txIntervalTemp >= 10) {
280         TENS_MIN = tensMinTemp;
281         TENS_MAX = tensMaxTemp;
282         VELOCIDADE = velocidadeTemp;
283         PASSOS_POR_SEG = passosTemp;
284         QTD_CICLOS = ciclosTemp;
285         TX_INTERVAL_MS = txIntervalTemp;
286         atualizarDerivados();
287
288         Serial.print("RAMP parameters updated: ");
289         Serial.print("T_MIN: "); Serial.print(TENS_MIN);
290         Serial.print(", T_MAX: "); Serial.print(TENS_MAX);
291         Serial.print(", VEL: "); Serial.print(VELOCIDADE);
292         Serial.print(", PASSOS: "); Serial.print(PASSOS_POR_SEG);
293         Serial.print(", CICLOS: "); Serial.print(QTD_CICLOS);
294         Serial.print(", TX Interval: "); Serial.print(TX_INTERVAL_MS);
295         Serial.println("ms");
296     } else {
297         Serial.println("Error: invalid parameters.");
298     }
299 } else {
300     Serial.println("Error: invalid parameter format.");
301 }
302 }
303
304 void atualizarDerivados() {
305     VOLT_POR_PASSO = VELOCIDADE / PASSOS_POR_SEG;
306     if (VOLT_POR_PASSO <= 0.0) {
307         VOLT_POR_PASSO = 0.0001;
308     }
309
310     float intervaloMs = 1000.0 / PASSOS_POR_SEG;
311     if (intervaloMs < 1.0) {
312         intervaloMs = 1.0;
313     }
314     PASSO_INTERVAL_MS = (unsigned long)(intervaloMs + 0.5);
315 }
316
317 void writeDAC(uint16_t value) {
318     digitalWrite(aOPin, LOW);
319     SPI.transfer16(value);
320     digitalWrite(aOPin, HIGH);
321     digitalWrite(a1Pin, LOW);

```

```
322 delayMicroseconds(1);
323 digitalWrite(a1Pin, HIGH);
324 }
325
326 // Convert voltage (-10V to +10V) to two's complement (16 bits)
327 uint16_t voltageToDAC(float voltage) {
328     int16_t dac_val = (int16_t)((voltage / 10.0) * 32768.0); // ±10V → ±0x8000
329     return (uint16_t)dac_val;
330 }
331
332 // Convert two's complement to voltage
333 float dacToVoltage(uint16_t dacValue) {
334     return ((int16_t)dacValue) * 10.0 / 32768.0;
335 }
336
337 void readSensors() {
338     unsigned long now = millis();
339     if (now - _lastTxMs < TX_INTERVAL_MS) {
340         return;
341     }
342     _lastTxMs = now;
343
344     float time_s = now / 1000.0;
345
346     int adcValue = analogRead(INA282_PIN);
347     // Uses internal 1.1V reference: 1023 = 1.1V
348     // ADC conversion: read value (0-1023) to voltage (0-1.1V)
349     float voltageINA282 = adcValue * (1.1 / 1023.0);
350
351     // Voltage to current conversion:
352     // 1A → 1.1V, 0A → 0.55V, -1A → 0V
353     // Linear formula: current (mA) = (voltage - 0.55) * (1000.0 / 0.55)
354     // Apply calibration offset to compensate deviation when should be 0A
355     float current_mA = (voltageINA282 - 0.55) * (1000.0 / 0.55) + INA282_OFFSET_MA;
356
357     Serial.print(time_s, 3);
358     Serial.print(",");
359     Serial.print(tensao, 6);
360     Serial.print(",");
361     Serial.print(current_mA, 6);
362     Serial.print(",");
363     Serial.println(tensao, 6);
364 }
```